



# RTEMS POSIX 1003.1 Compliance Guide

*Release 6.2-rc2 (6th October 2025)*

© 1988-2025 RTEMS Project and contributors



# CONTENTS

|          |                               |          |
|----------|-------------------------------|----------|
| <b>1</b> | <b>Preface</b>                | <b>3</b> |
| <b>2</b> | <b>Standards</b>              | <b>5</b> |
| <b>3</b> | <b>RTEMS Complete Profile</b> | <b>7</b> |
| 3.1      | Summary . . . . .             | 8        |
| 3.2      | <aio.h> . . . . .             | 9        |
| 3.3      | <arpa/inet.h> . . . . .       | 10       |
| 3.4      | <assert.h> . . . . .          | 11       |
| 3.5      | <complex.h> . . . . .         | 12       |
| 3.6      | <ctype.h> . . . . .           | 14       |
| 3.7      | <devctl.h> . . . . .          | 15       |
| 3.8      | <dirent.h> . . . . .          | 16       |
| 3.9      | <dlfcn.h> . . . . .           | 17       |
| 3.10     | <errno.h> . . . . .           | 18       |
| 3.11     | <fcntl.h> . . . . .           | 19       |
| 3.12     | <fenv.h> . . . . .            | 20       |
| 3.13     | <fmtmsg.h> . . . . .          | 21       |
| 3.14     | <fnmatch.h> . . . . .         | 22       |
| 3.15     | <ftw.h> . . . . .             | 23       |
| 3.16     | <glob.h> . . . . .            | 24       |
| 3.17     | <grp.h> . . . . .             | 25       |
| 3.18     | <iconv.h> . . . . .           | 26       |
| 3.19     | <inttypes.h> . . . . .        | 27       |
| 3.20     | <langinfo.h> . . . . .        | 28       |
| 3.21     | <libgen.h> . . . . .          | 29       |
| 3.22     | <libintl.h> . . . . .         | 30       |
| 3.23     | <locale.h> . . . . .          | 31       |
| 3.24     | <math.h> . . . . .            | 32       |
| 3.25     | <monetary.h> . . . . .        | 38       |
| 3.26     | <mqueue.h> . . . . .          | 39       |
| 3.27     | <ndbm.h> . . . . .            | 40       |
| 3.28     | <net/if.h> . . . . .          | 41       |
| 3.29     | <netdb.h> . . . . .           | 42       |
| 3.30     | <nl_types.h> . . . . .        | 43       |
| 3.31     | <poll.h> . . . . .            | 44       |
| 3.32     | <pthread.h> . . . . .         | 45       |
| 3.33     | <pwd.h> . . . . .             | 49       |

|          |                             |            |
|----------|-----------------------------|------------|
| 3.34     | <regex.h>                   | 50         |
| 3.35     | <sched.h>                   | 51         |
| 3.36     | <search.h>                  | 52         |
| 3.37     | <semaphore.h>               | 53         |
| 3.38     | <setjmp.h>                  | 54         |
| 3.39     | <signal.h>                  | 55         |
| 3.40     | <spawn.h>                   | 56         |
| 3.41     | <stdarg.h>                  | 57         |
| 3.42     | <stdatomic.h>               | 58         |
| 3.43     | <stddef.h>                  | 59         |
| 3.44     | <stdint.h>                  | 60         |
| 3.45     | <stdio.h>                   | 61         |
| 3.46     | <stdlib.h>                  | 64         |
| 3.47     | <string.h>                  | 67         |
| 3.48     | <strings.h>                 | 69         |
| 3.49     | <stropts.h>                 | 70         |
| 3.50     | <sys/ipc.h>                 | 71         |
| 3.51     | <sys/mman.h>                | 72         |
| 3.52     | <sys/msg.h>                 | 73         |
| 3.53     | <sys/resource.h>            | 74         |
| 3.54     | <sys/select.h>              | 75         |
| 3.55     | <sys/sem.h>                 | 76         |
| 3.56     | <sys/shm.h>                 | 77         |
| 3.57     | <sys/socket.h>              | 78         |
| 3.58     | <sys/stat.h>                | 79         |
| 3.59     | <sys/statvfs.h>             | 80         |
| 3.60     | <sys/time.h>                | 81         |
| 3.61     | <sys/uio.h>                 | 82         |
| 3.62     | <sys/utsname.h>             | 83         |
| 3.63     | <sys/wait.h>                | 84         |
| 3.64     | <syslog.h>                  | 85         |
| 3.65     | <termios.h>                 | 86         |
| 3.66     | <threads.h>                 | 87         |
| 3.67     | <time.h>                    | 88         |
| 3.68     | <trace.h>                   | 90         |
| 3.69     | <ulimit.h>                  | 92         |
| 3.70     | <unistd.h>                  | 93         |
| 3.71     | <utime.h>                   | 96         |
| 3.72     | <utmpx.h>                   | 97         |
| 3.73     | <wchar.h>                   | 98         |
| 3.74     | <wctype.h>                  | 101        |
| 3.75     | <wordexp.h>                 | 103        |
| <b>4</b> | <b>POSIX-2024 (Issue 8)</b> | <b>105</b> |
| 4.1      | Summary                     | 106        |
| 4.2      | <aio.h>                     | 107        |
| 4.3      | <arpa/inet.h>               | 108        |
| 4.4      | <assert.h>                  | 109        |
| 4.5      | <complex.h>                 | 110        |
| 4.6      | <ctype.h>                   | 112        |
| 4.7      | <dirent.h>                  | 113        |

|      |                  |     |
|------|------------------|-----|
| 4.8  | <dlfcn.h>        | 114 |
| 4.9  | <errno.h>        | 115 |
| 4.10 | <fcntl.h>        | 116 |
| 4.11 | <fenv.h>         | 117 |
| 4.12 | <fmtmsg.h>       | 118 |
| 4.13 | <fnmatch.h>      | 119 |
| 4.14 | <ftw.h>          | 120 |
| 4.15 | <glob.h>         | 121 |
| 4.16 | <grp.h>          | 122 |
| 4.17 | <iconv.h>        | 123 |
| 4.18 | <inttypes.h>     | 124 |
| 4.19 | <langinfo.h>     | 125 |
| 4.20 | <libgen.h>       | 126 |
| 4.21 | <libintl.h>      | 127 |
| 4.22 | <locale.h>       | 128 |
| 4.23 | <math.h>         | 129 |
| 4.24 | <monetary.h>     | 135 |
| 4.25 | <mqueue.h>       | 136 |
| 4.26 | <ndbm.h>         | 137 |
| 4.27 | <net/if.h>       | 138 |
| 4.28 | <netdb.h>        | 139 |
| 4.29 | <nl_types.h>     | 140 |
| 4.30 | <poll.h>         | 141 |
| 4.31 | <pthread.h>      | 142 |
| 4.32 | <pwd.h>          | 145 |
| 4.33 | <regex.h>        | 146 |
| 4.34 | <sched.h>        | 147 |
| 4.35 | <search.h>       | 148 |
| 4.36 | <semaphore.h>    | 149 |
| 4.37 | <setjmp.h>       | 150 |
| 4.38 | <signal.h>       | 151 |
| 4.39 | <spawn.h>        | 152 |
| 4.40 | <stdarg.h>       | 153 |
| 4.41 | <stdatomic.h>    | 154 |
| 4.42 | <stddef.h>       | 155 |
| 4.43 | <stdint.h>       | 156 |
| 4.44 | <stdio.h>        | 157 |
| 4.45 | <stdlib.h>       | 160 |
| 4.46 | <string.h>       | 163 |
| 4.47 | <strings.h>      | 165 |
| 4.48 | <sys/ipc.h>      | 166 |
| 4.49 | <sys/mman.h>     | 167 |
| 4.50 | <sys/msg.h>      | 168 |
| 4.51 | <sys/resource.h> | 169 |
| 4.52 | <sys/select.h>   | 170 |
| 4.53 | <sys/sem.h>      | 171 |
| 4.54 | <sys/shm.h>      | 172 |
| 4.55 | <sys/socket.h>   | 173 |
| 4.56 | <sys/stat.h>     | 174 |
| 4.57 | <sys/statvfs.h>  | 175 |
| 4.58 | <sys/time.h>     | 176 |

|          |                             |            |
|----------|-----------------------------|------------|
| 4.59     | <sys/uio.h>                 | 177        |
| 4.60     | <sys/utsname.h>             | 178        |
| 4.61     | <sys/wait.h>                | 179        |
| 4.62     | <syslog.h>                  | 180        |
| 4.63     | <termios.h>                 | 181        |
| 4.64     | <threads.h>                 | 182        |
| 4.65     | <time.h>                    | 183        |
| 4.66     | <unistd.h>                  | 185        |
| 4.67     | <utmpx.h>                   | 188        |
| 4.68     | <wchar.h>                   | 189        |
| 4.69     | <wctype.h>                  | 192        |
| 4.70     | <wordexp.h>                 | 194        |
| <b>5</b> | <b>POSIX-2017 (Issue 7)</b> | <b>195</b> |
| 5.1      | Summary                     | 196        |
| 5.2      | <aio.h>                     | 197        |
| 5.3      | <arpa/inet.h>               | 198        |
| 5.4      | <assert.h>                  | 199        |
| 5.5      | <complex.h>                 | 200        |
| 5.6      | <ctype.h>                   | 202        |
| 5.7      | <dirent.h>                  | 203        |
| 5.8      | <dlfcn.h>                   | 204        |
| 5.9      | <errno.h>                   | 205        |
| 5.10     | <fcntl.h>                   | 206        |
| 5.11     | <fenv.h>                    | 207        |
| 5.12     | <fmtmsg.h>                  | 208        |
| 5.13     | <fnmatch.h>                 | 209        |
| 5.14     | <ftw.h>                     | 210        |
| 5.15     | <glob.h>                    | 211        |
| 5.16     | <grp.h>                     | 212        |
| 5.17     | <iconv.h>                   | 213        |
| 5.18     | <inttypes.h>                | 214        |
| 5.19     | <langinfo.h>                | 215        |
| 5.20     | <libgen.h>                  | 216        |
| 5.21     | <locale.h>                  | 217        |
| 5.22     | <math.h>                    | 218        |
| 5.23     | <monetary.h>                | 224        |
| 5.24     | <mqueue.h>                  | 225        |
| 5.25     | <ndbm.h>                    | 226        |
| 5.26     | <net/if.h>                  | 227        |
| 5.27     | <netdb.h>                   | 228        |
| 5.28     | <nl_types.h>                | 229        |
| 5.29     | <poll.h>                    | 230        |
| 5.30     | <pthread.h>                 | 231        |
| 5.31     | <pwd.h>                     | 235        |
| 5.32     | <regex.h>                   | 236        |
| 5.33     | <sched.h>                   | 237        |
| 5.34     | <search.h>                  | 238        |
| 5.35     | <semaphore.h>               | 239        |
| 5.36     | <setjmp.h>                  | 240        |
| 5.37     | <signal.h>                  | 241        |

|          |                                 |            |
|----------|---------------------------------|------------|
| 5.38     | <spawn.h>                       | 242        |
| 5.39     | <stdarg.h>                      | 243        |
| 5.40     | <stddef.h>                      | 244        |
| 5.41     | <stdint.h>                      | 245        |
| 5.42     | <stdio.h>                       | 246        |
| 5.43     | <stdlib.h>                      | 249        |
| 5.44     | <string.h>                      | 252        |
| 5.45     | <strings.h>                     | 254        |
| 5.46     | <stropts.h>                     | 255        |
| 5.47     | <sys/ipc.h>                     | 256        |
| 5.48     | <sys/mman.h>                    | 257        |
| 5.49     | <sys/msg.h>                     | 258        |
| 5.50     | <sys/resource.h>                | 259        |
| 5.51     | <sys/select.h>                  | 260        |
| 5.52     | <sys/sem.h>                     | 261        |
| 5.53     | <sys/shm.h>                     | 262        |
| 5.54     | <sys/socket.h>                  | 263        |
| 5.55     | <sys/stat.h>                    | 264        |
| 5.56     | <sys/statvfs.h>                 | 265        |
| 5.57     | <sys/time.h>                    | 266        |
| 5.58     | <sys/uio.h>                     | 267        |
| 5.59     | <sys/utsname.h>                 | 268        |
| 5.60     | <sys/wait.h>                    | 269        |
| 5.61     | <syslog.h>                      | 270        |
| 5.62     | <termios.h>                     | 271        |
| 5.63     | <time.h>                        | 272        |
| 5.64     | <trace.h>                       | 274        |
| 5.65     | <ulimit.h>                      | 276        |
| 5.66     | <unistd.h>                      | 277        |
| 5.67     | <utime.h>                       | 280        |
| 5.68     | <utmpx.h>                       | 281        |
| 5.69     | <wchar.h>                       | 282        |
| 5.70     | <wctype.h>                      | 285        |
| 5.71     | <wordexp.h>                     | 287        |
| <b>6</b> | <b>POSIX-2008 (Issue 6 TC2)</b> | <b>289</b> |
| 6.1      | Summary                         | 290        |
| 6.2      | <aio.h>                         | 291        |
| 6.3      | <arpa/inet.h>                   | 292        |
| 6.4      | <assert.h>                      | 293        |
| 6.5      | <complex.h>                     | 294        |
| 6.6      | <ctype.h>                       | 296        |
| 6.7      | <dirent.h>                      | 297        |
| 6.8      | <dlfcn.h>                       | 298        |
| 6.9      | <errno.h>                       | 299        |
| 6.10     | <fcntl.h>                       | 300        |
| 6.11     | <fenv.h>                        | 301        |
| 6.12     | <fmtmsg.h>                      | 302        |
| 6.13     | <fnmatch.h>                     | 303        |
| 6.14     | <ftw.h>                         | 304        |
| 6.15     | <glob.h>                        | 305        |

|      |                  |     |
|------|------------------|-----|
| 6.16 | <grp.h>          | 306 |
| 6.17 | <iconv.h>        | 307 |
| 6.18 | <inttypes.h>     | 308 |
| 6.19 | <langinfo.h>     | 309 |
| 6.20 | <libgen.h>       | 310 |
| 6.21 | <locale.h>       | 311 |
| 6.22 | <math.h>         | 312 |
| 6.23 | <monetary.h>     | 318 |
| 6.24 | <mqueue.h>       | 319 |
| 6.25 | <ndbm.h>         | 320 |
| 6.26 | <net/if.h>       | 321 |
| 6.27 | <netdb.h>        | 322 |
| 6.28 | <nl_types.h>     | 323 |
| 6.29 | <poll.h>         | 324 |
| 6.30 | <pthread.h>      | 325 |
| 6.31 | <pwd.h>          | 329 |
| 6.32 | <regex.h>        | 330 |
| 6.33 | <sched.h>        | 331 |
| 6.34 | <search.h>       | 332 |
| 6.35 | <semaphore.h>    | 333 |
| 6.36 | <setjmp.h>       | 334 |
| 6.37 | <signal.h>       | 335 |
| 6.38 | <spawn.h>        | 336 |
| 6.39 | <stdarg.h>       | 337 |
| 6.40 | <stddef.h>       | 338 |
| 6.41 | <stdint.h>       | 339 |
| 6.42 | <stdio.h>        | 340 |
| 6.43 | <stdlib.h>       | 343 |
| 6.44 | <string.h>       | 346 |
| 6.45 | <strings.h>      | 348 |
| 6.46 | <stropts.h>      | 349 |
| 6.47 | <sys/ipc.h>      | 350 |
| 6.48 | <sys/mman.h>     | 351 |
| 6.49 | <sys/msg.h>      | 352 |
| 6.50 | <sys/resource.h> | 353 |
| 6.51 | <sys/select.h>   | 354 |
| 6.52 | <sys/sem.h>      | 355 |
| 6.53 | <sys/shm.h>      | 356 |
| 6.54 | <sys/socket.h>   | 357 |
| 6.55 | <sys/stat.h>     | 358 |
| 6.56 | <sys/statvfs.h>  | 359 |
| 6.57 | <sys/time.h>     | 360 |
| 6.58 | <sys/uio.h>      | 361 |
| 6.59 | <sys/utsname.h>  | 362 |
| 6.60 | <sys/wait.h>     | 363 |
| 6.61 | <syslog.h>       | 364 |
| 6.62 | <termios.h>      | 365 |
| 6.63 | <time.h>         | 366 |
| 6.64 | <trace.h>        | 368 |
| 6.65 | <ulimit.h>       | 370 |
| 6.66 | <unistd.h>       | 371 |



|          |                             |            |
|----------|-----------------------------|------------|
| 6.67     | <utime.h>                   | 374        |
| 6.68     | <utmpx.h>                   | 375        |
| 6.69     | <wchar.h>                   | 376        |
| 6.70     | <wctype.h>                  | 379        |
| 6.71     | <wordexp.h>                 | 381        |
| <b>7</b> | <b>POSIX-2003 (Issue 6)</b> | <b>383</b> |
| 7.1      | Summary                     | 384        |
| 7.2      | <aio.h>                     | 385        |
| 7.3      | <arpa/inet.h>               | 386        |
| 7.4      | <assert.h>                  | 387        |
| 7.5      | <complex.h>                 | 388        |
| 7.6      | <ctype.h>                   | 390        |
| 7.7      | <dirent.h>                  | 391        |
| 7.8      | <dlfcn.h>                   | 392        |
| 7.9      | <errno.h>                   | 393        |
| 7.10     | <fcntl.h>                   | 394        |
| 7.11     | <fenv.h>                    | 395        |
| 7.12     | <fmtmsg.h>                  | 396        |
| 7.13     | <fnmatch.h>                 | 397        |
| 7.14     | <ftw.h>                     | 398        |
| 7.15     | <glob.h>                    | 399        |
| 7.16     | <grp.h>                     | 400        |
| 7.17     | <iconv.h>                   | 401        |
| 7.18     | <inttypes.h>                | 402        |
| 7.19     | <langinfo.h>                | 403        |
| 7.20     | <libgen.h>                  | 404        |
| 7.21     | <locale.h>                  | 405        |
| 7.22     | <math.h>                    | 406        |
| 7.23     | <monetary.h>                | 412        |
| 7.24     | <mqueue.h>                  | 413        |
| 7.25     | <ndbm.h>                    | 414        |
| 7.26     | <net/if.h>                  | 415        |
| 7.27     | <netdb.h>                   | 416        |
| 7.28     | <nl_types.h>                | 417        |
| 7.29     | <poll.h>                    | 418        |
| 7.30     | <pthread.h>                 | 419        |
| 7.31     | <pwd.h>                     | 422        |
| 7.32     | <regex.h>                   | 423        |
| 7.33     | <sched.h>                   | 424        |
| 7.34     | <search.h>                  | 425        |
| 7.35     | <semaphore.h>               | 426        |
| 7.36     | <setjmp.h>                  | 427        |
| 7.37     | <signal.h>                  | 428        |
| 7.38     | <spawn.h>                   | 429        |
| 7.39     | <stdarg.h>                  | 430        |
| 7.40     | <stddef.h>                  | 431        |
| 7.41     | <stdint.h>                  | 432        |
| 7.42     | <stdio.h>                   | 433        |
| 7.43     | <stdlib.h>                  | 435        |
| 7.44     | <string.h>                  | 438        |

|          |   |            |
|----------|---|------------|
| 7.45     | <strings.h>                               | 439        |
| 7.46     | <stropts.h>                               | 440        |
| 7.47     | <sys/ipc.h>                               | 441        |
| 7.48     | <sys/mman.h>                              | 442        |
| 7.49     | <sys/msg.h>                               | 443        |
| 7.50     | <sys/resource.h>                          | 444        |
| 7.51     | <sys/select.h>                            | 445        |
| 7.52     | <sys/sem.h>                               | 446        |
| 7.53     | <sys/shm.h>                               | 447        |
| 7.54     | <sys/socket.h>                            | 448        |
| 7.55     | <sys/stat.h>                              | 449        |
| 7.56     | <sys/statvfs.h>                           | 450        |
| 7.57     | <sys/time.h>                              | 451        |
| 7.58     | <sys/uio.h>                               | 452        |
| 7.59     | <sys/utsname.h>                           | 453        |
| 7.60     | <sys/wait.h>                              | 454        |
| 7.61     | <syslog.h>                                | 455        |
| 7.62     | <termios.h>                               | 456        |
| 7.63     | <time.h>                                  | 457        |
| 7.64     | <trace.h>                                 | 459        |
| 7.65     | <ulimit.h>                                | 461        |
| 7.66     | <unistd.h>                                | 462        |
| 7.67     | <utime.h>                                 | 465        |
| 7.68     | <utmpx.h>                                 | 466        |
| 7.69     | <wchar.h>                                 | 467        |
| 7.70     | <wctype.h>                                | 469        |
| 7.71     | <wordexp.h>                               | 470        |
| <b>8</b> | <b>POSIX PSE51 - Minimal</b>              | <b>471</b> |
| 8.1      | Summary                                   | 472        |
| 8.2      | <ctype.h>                                 | 473        |
| 8.3      | <errno.h>                                 | 474        |
| 8.4      | <fcntl.h>                                 | 475        |
| 8.5      | <fenv.h>                                  | 476        |
| 8.6      | <inttypes.h>                              | 477        |
| 8.7      | <locale.h>                                | 478        |
| 8.8      | <pthread.h>                               | 479        |
| 8.9      | <sched.h>                                 | 482        |
| 8.10     | <semaphore.h>                             | 483        |
| 8.11     | <setjmp.h>                                | 484        |
| 8.12     | <signal.h>                                | 485        |
| 8.13     | <stdarg.h>                                | 486        |
| 8.14     | <stddef.h>                                | 487        |
| 8.15     | <stdio.h>                                 | 488        |
| 8.16     | <stdlib.h>                                | 490        |
| 8.17     | <string.h>                                | 491        |
| 8.18     | <sys/mman.h>                              | 492        |
| 8.19     | <sys/utsname.h>                           | 493        |
| 8.20     | <time.h>                                  | 494        |
| 8.21     | <unistd.h>                                | 495        |
| <b>9</b> | <b>POSIX PSE52 - Real-Time Controller</b> | <b>497</b> |

|           |                                |            |
|-----------|--------------------------------|------------|
| 9.1       | Summary                        | 498        |
| 9.2       | <complex.h>                    | 499        |
| 9.3       | <ctype.h>                      | 501        |
| 9.4       | <dirent.h>                     | 502        |
| 9.5       | <errno.h>                      | 503        |
| 9.6       | <fcntl.h>                      | 504        |
| 9.7       | <fenv.h>                       | 505        |
| 9.8       | <inttypes.h>                   | 506        |
| 9.9       | <locale.h>                     | 507        |
| 9.10      | <math.h>                       | 508        |
| 9.11      | <mqueue.h>                     | 514        |
| 9.12      | <pthread.h>                    | 515        |
| 9.13      | <sched.h>                      | 518        |
| 9.14      | <semaphore.h>                  | 519        |
| 9.15      | <setjmp.h>                     | 520        |
| 9.16      | <signal.h>                     | 521        |
| 9.17      | <stdarg.h>                     | 522        |
| 9.18      | <stddef.h>                     | 523        |
| 9.19      | <stdio.h>                      | 524        |
| 9.20      | <stdlib.h>                     | 526        |
| 9.21      | <string.h>                     | 527        |
| 9.22      | <sys/mman.h>                   | 528        |
| 9.23      | <sys/stat.h>                   | 529        |
| 9.24      | <sys/utsname.h>                | 530        |
| 9.25      | <time.h>                       | 531        |
| 9.26      | <trace.h>                      | 532        |
| 9.27      | <unistd.h>                     | 534        |
| 9.28      | <utime.h>                      | 535        |
| <b>10</b> | <b>POSIX PSE53 - Dedicated</b> | <b>537</b> |
| 10.1      | Summary                        | 538        |
| 10.2      | <aio.h>                        | 539        |
| 10.3      | <arpa/inet.h>                  | 540        |
| 10.4      | <assert.h>                     | 541        |
| 10.5      | <complex.h>                    | 542        |
| 10.6      | <ctype.h>                      | 544        |
| 10.7      | <dirent.h>                     | 545        |
| 10.8      | <errno.h>                      | 546        |
| 10.9      | <fcntl.h>                      | 547        |
| 10.10     | <fenv.h>                       | 548        |
| 10.11     | <inttypes.h>                   | 549        |
| 10.12     | <locale.h>                     | 550        |
| 10.13     | <math.h>                       | 551        |
| 10.14     | <mqueue.h>                     | 557        |
| 10.15     | <net/if.h>                     | 558        |
| 10.16     | <netdb.h>                      | 559        |
| 10.17     | <pthread.h>                    | 560        |
| 10.18     | <sched.h>                      | 563        |
| 10.19     | <semaphore.h>                  | 564        |
| 10.20     | <setjmp.h>                     | 565        |
| 10.21     | <signal.h>                     | 566        |

|           |                                   |            |
|-----------|-----------------------------------|------------|
| 10.22     | <spawn.h>                         | 567        |
| 10.23     | <stdarg.h>                        | 568        |
| 10.24     | <stddef.h>                        | 569        |
| 10.25     | <stdio.h>                         | 570        |
| 10.26     | <stdlib.h>                        | 572        |
| 10.27     | <string.h>                        | 574        |
| 10.28     | <sys/mman.h>                      | 575        |
| 10.29     | <sys/select.h>                    | 576        |
| 10.30     | <sys/socket.h>                    | 577        |
| 10.31     | <sys/stat.h>                      | 578        |
| 10.32     | <sys/time.h>                      | 579        |
| 10.33     | <sys/utsname.h>                   | 580        |
| 10.34     | <sys/wait.h>                      | 581        |
| 10.35     | <time.h>                          | 582        |
| 10.36     | <trace.h>                         | 583        |
| 10.37     | <unistd.h>                        | 585        |
| 10.38     | <utime.h>                         | 587        |
| <b>11</b> | <b>POSIX PSE54 - Multipurpose</b> | <b>589</b> |
| 11.1      | Summary                           | 590        |
| 11.2      | <aio.h>                           | 591        |
| 11.3      | <arpa/inet.h>                     | 592        |
| 11.4      | <assert.h>                        | 593        |
| 11.5      | <complex.h>                       | 594        |
| 11.6      | <ctype.h>                         | 596        |
| 11.7      | <dirent.h>                        | 597        |
| 11.8      | <dlfcn.h>                         | 598        |
| 11.9      | <errno.h>                         | 599        |
| 11.10     | <fcntl.h>                         | 600        |
| 11.11     | <fenv.h>                          | 601        |
| 11.12     | <fnmatch.h>                       | 602        |
| 11.13     | <glob.h>                          | 603        |
| 11.14     | <grp.h>                           | 604        |
| 11.15     | <inttypes.h>                      | 605        |
| 11.16     | <locale.h>                        | 606        |
| 11.17     | <math.h>                          | 607        |
| 11.18     | <mqueue.h>                        | 613        |
| 11.19     | <net/if.h>                        | 614        |
| 11.20     | <netdb.h>                         | 615        |
| 11.21     | <pthread.h>                       | 616        |
| 11.22     | <pwd.h>                           | 619        |
| 11.23     | <regex.h>                         | 620        |
| 11.24     | <sched.h>                         | 621        |
| 11.25     | <semaphore.h>                     | 622        |
| 11.26     | <setjmp.h>                        | 623        |
| 11.27     | <signal.h>                        | 624        |
| 11.28     | <spawn.h>                         | 625        |
| 11.29     | <stdarg.h>                        | 626        |
| 11.30     | <stddef.h>                        | 627        |
| 11.31     | <stdio.h>                         | 628        |
| 11.32     | <stdlib.h>                        | 630        |

|           |                             |            |
|-----------|-----------------------------|------------|
| 11.33     | <string.h>                  | 632        |
| 11.34     | <sys/mman.h>                | 633        |
| 11.35     | <sys/select.h>              | 634        |
| 11.36     | <sys/socket.h>              | 635        |
| 11.37     | <sys/stat.h>                | 636        |
| 11.38     | <sys/time.h>                | 637        |
| 11.39     | <sys/utsname.h>             | 638        |
| 11.40     | <sys/wait.h>                | 639        |
| 11.41     | <syslog.h>                  | 640        |
| 11.42     | <termios.h>                 | 641        |
| 11.43     | <time.h>                    | 642        |
| 11.44     | <trace.h>                   | 643        |
| 11.45     | <unistd.h>                  | 645        |
| 11.46     | <utime.h>                   | 647        |
| 11.47     | <wchar.h>                   | 648        |
| 11.48     | <wctype.h>                  | 650        |
| 11.49     | <wordexp.h>                 | 651        |
| <b>12</b> | <b>C99 Standard Library</b> | <b>653</b> |
| 12.1      | Summary                     | 654        |
| 12.2      | <assert.h>                  | 655        |
| 12.3      | <complex.h>                 | 656        |
| 12.4      | <ctype.h>                   | 658        |
| 12.5      | <errno.h>                   | 659        |
| 12.6      | <fenv.h>                    | 660        |
| 12.7      | <inttypes.h>                | 661        |
| 12.8      | <locale.h>                  | 662        |
| 12.9      | <math.h>                    | 663        |
| 12.10     | <setjmp.h>                  | 669        |
| 12.11     | <signal.h>                  | 670        |
| 12.12     | <stdarg.h>                  | 671        |
| 12.13     | <stddef.h>                  | 672        |
| 12.14     | <stdint.h>                  | 673        |
| 12.15     | <stdio.h>                   | 674        |
| 12.16     | <stdlib.h>                  | 676        |
| 12.17     | <string.h>                  | 678        |
| 12.18     | <time.h>                    | 679        |
| 12.19     | <wchar.h>                   | 680        |
| 12.20     | <wctype.h>                  | 682        |
| <b>13</b> | <b>C11 Standard Library</b> | <b>683</b> |
| 13.1      | Summary                     | 684        |
| 13.2      | <assert.h>                  | 685        |
| 13.3      | <complex.h>                 | 686        |
| 13.4      | <ctype.h>                   | 688        |
| 13.5      | <errno.h>                   | 689        |
| 13.6      | <fenv.h>                    | 690        |
| 13.7      | <inttypes.h>                | 691        |
| 13.8      | <locale.h>                  | 692        |
| 13.9      | <math.h>                    | 693        |
| 13.10     | <setjmp.h>                  | 699        |
| 13.11     | <signal.h>                  | 700        |

|           |  |            |
|-----------|--|------------|
| 13.12     | <stdarg.h>   | 701        |
| 13.13     | <stdatomic.h>  | 702        |
| 13.14     | <stddef.h>   | 703        |
| 13.15     | <stdint.h>   | 704        |
| 13.16     | <stdio.h>  | 705        |
| 13.17     | <stdlib.h>   | 707        |
| 13.18     | <string.h>   | 709        |
| 13.19     | <threads.h>  | 710        |
| 13.20     | <time.h>   | 711        |
| 13.21     | <wchar.h>  | 712        |
| 13.22     | <wctype.h>   | 714        |
| <b>14</b> | <b>C17 Standard Library</b>                          | <b>715</b> |
| 14.1      | Summary  | 716        |
| 14.2      | <assert.h>   | 717        |
| 14.3      | <complex.h>  | 718        |
| 14.4      | <ctype.h>  | 720        |
| 14.5      | <errno.h>  | 721        |
| 14.6      | <fenv.h>   | 722        |
| 14.7      | <inttypes.h>   | 723        |
| 14.8      | <locale.h>   | 724        |
| 14.9      | <math.h>   | 725        |
| 14.10     | <setjmp.h>   | 731        |
| 14.11     | <signal.h>   | 732        |
| 14.12     | <stdarg.h>   | 733        |
| 14.13     | <stdatomic.h>  | 734        |
| 14.14     | <stddef.h>   | 735        |
| 14.15     | <stdio.h>  | 736        |
| 14.16     | <stdlib.h>   | 738        |
| 14.17     | <string.h>   | 740        |
| 14.18     | <threads.h>  | 741        |
| 14.19     | <time.h>   | 742        |
| 14.20     | <wchar.h>  | 743        |
| 14.21     | <wctype.h>   | 745        |
| <b>15</b> | <b>FACE Technical Standard, Edition 2.1 Security</b> | <b>747</b> |
| 15.1      | Summary  | 748        |
| 15.2      | <arpa/inet.h>  | 749        |
| 15.3      | <ctype.h>  | 750        |
| 15.4      | <devctl.h>   | 751        |
| 15.5      | <errno.h>  | 752        |
| 15.6      | <math.h>   | 753        |
| 15.7      | <netdb.h>  | 754        |
| 15.8      | <pthread.h>  | 755        |
| 15.9      | <sched.h>  | 757        |
| 15.10     | <semaphore.h>  | 758        |
| 15.11     | <signal.h>   | 759        |
| 15.12     | <stdlib.h>   | 760        |
| 15.13     | <string.h>   | 761        |
| 15.14     | <sys/mman.h>   | 762        |
| 15.15     | <sys/socket.h>                                       | 763        |
| 15.16     | <sys/stat.h>   | 764        |

|           |   |            |
|-----------|---|------------|
| 15.17     | <time.h>  | 765        |
| 15.18     | <unistd.h>  | 766        |
| <b>16</b> | <b>FACE Technical Standard, Edition 2.1 Safety Base</b>     | <b>767</b> |
| 16.1      | Summary   | 768        |
| 16.2      | <arpa/inet.h>   | 769        |
| 16.3      | <ctype.h>   | 770        |
| 16.4      | <devctl.h>  | 771        |
| 16.5      | <dirent.h>  | 772        |
| 16.6      | <errno.h>   | 773        |
| 16.7      | <fcntl.h>   | 774        |
| 16.8      | <math.h>  | 775        |
| 16.9      | <mqueue.h>  | 776        |
| 16.10     | <netdb.h>   | 777        |
| 16.11     | <pthread.h>   | 778        |
| 16.12     | <sched.h>   | 780        |
| 16.13     | <semaphore.h>   | 781        |
| 16.14     | <signal.h>  | 782        |
| 16.15     | <stdio.h>   | 783        |
| 16.16     | <stdlib.h>  | 784        |
| 16.17     | <string.h>  | 785        |
| 16.18     | <sys/mman.h>  | 786        |
| 16.19     | <sys/select.h>  | 787        |
| 16.20     | <sys/socket.h>  | 788        |
| 16.21     | <sys/stat.h>  | 789        |
| 16.22     | <time.h>  | 790        |
| 16.23     | <unistd.h>  | 791        |
| <b>17</b> | <b>FACE Technical Standard, Edition 2.1 Safety Extended</b> | <b>793</b> |
| 17.1      | Summary   | 794        |
| 17.2      | <arpa/inet.h>   | 795        |
| 17.3      | <ctype.h>   | 796        |
| 17.4      | <devctl.h>  | 797        |
| 17.5      | <dirent.h>  | 798        |
| 17.6      | <errno.h>   | 799        |
| 17.7      | <fcntl.h>   | 800        |
| 17.8      | <math.h>  | 801        |
| 17.9      | <mqueue.h>  | 802        |
| 17.10     | <netdb.h>   | 803        |
| 17.11     | <pthread.h>   | 804        |
| 17.12     | <sched.h>   | 806        |
| 17.13     | <semaphore.h>   | 807        |
| 17.14     | <setjmp.h>  | 808        |
| 17.15     | <signal.h>  | 809        |
| 17.16     | <spawn.h>   | 810        |
| 17.17     | <stdarg.h>  | 811        |
| 17.18     | <stdio.h>   | 812        |
| 17.19     | <stdlib.h>  | 813        |
| 17.20     | <string.h>  | 814        |
| 17.21     | <sys/mman.h>  | 815        |
| 17.22     | <sys/select.h>  | 816        |
| 17.23     | <sys/socket.h>  | 817        |

|           |   |            |
|-----------|---|------------|
| 17.24     | <sys/stat.h>  | 818        |
| 17.25     | <sys/time.h>  | 819        |
| 17.26     | <sys/utsname.h>   | 820        |
| 17.27     | <sys/wait.h>  | 821        |
| 17.28     | <time.h>  | 822        |
| 17.29     | <unistd.h>  | 823        |
| <b>18</b> | <b>FACE Technical Standard, Edition 2.1 General Purpose</b> | <b>825</b> |
| 18.1      | Summary   | 826        |
| 18.2      | <aio.h>   | 827        |
| 18.3      | <arpa/inet.h>   | 828        |
| 18.4      | <assert.h>  | 829        |
| 18.5      | <complex.h>   | 830        |
| 18.6      | <ctype.h>   | 832        |
| 18.7      | <devctl.h>  | 833        |
| 18.8      | <dirent.h>  | 834        |
| 18.9      | <errno.h>   | 835        |
| 18.10     | <fcntl.h>   | 836        |
| 18.11     | <fenv.h>  | 837        |
| 18.12     | <inttypes.h>  | 838        |
| 18.13     | <locale.h>  | 839        |
| 18.14     | <math.h>  | 840        |
| 18.15     | <mqueue.h>  | 846        |
| 18.16     | <net/if.h>  | 847        |
| 18.17     | <netdb.h>   | 848        |
| 18.18     | <pthread.h>   | 849        |
| 18.19     | <sched.h>   | 852        |
| 18.20     | <semaphore.h>   | 853        |
| 18.21     | <setjmp.h>  | 854        |
| 18.22     | <signal.h>  | 855        |
| 18.23     | <spawn.h>   | 856        |
| 18.24     | <stdarg.h>  | 857        |
| 18.25     | <stdio.h>   | 858        |
| 18.26     | <stdlib.h>  | 860        |
| 18.27     | <string.h>  | 862        |
| 18.28     | <sys/mman.h>  | 863        |
| 18.29     | <sys/select.h>  | 864        |
| 18.30     | <sys/socket.h>  | 865        |
| 18.31     | <sys/stat.h>  | 866        |
| 18.32     | <sys/time.h>  | 867        |
| 18.33     | <sys/utsname.h>   | 868        |
| 18.34     | <sys/wait.h>  | 869        |
| 18.35     | <time.h>  | 870        |
| 18.36     | <unistd.h>  | 871        |
| 18.37     | <wchar.h>   | 873        |
| 18.38     | <wctype.h>  | 875        |
| <b>19</b> | <b>FACE Technical Standard, Edition 3.0 Security</b>        | <b>877</b> |
| 19.1      | Summary   | 878        |
| 19.2      | <arpa/inet.h>   | 879        |
| 19.3      | <ctype.h>   | 880        |
| 19.4      | <devctl.h>  | 881        |



|           |   |            |
|-----------|---|------------|
| 19.5      | <errno.h>   | 882        |
| 19.6      | <math.h>  | 883        |
| 19.7      | <netdb.h>   | 884        |
| 19.8      | <pthread.h>   | 885        |
| 19.9      | <sched.h>   | 887        |
| 19.10     | <semaphore.h>   | 888        |
| 19.11     | <signal.h>  | 889        |
| 19.12     | <stddef.h>  | 890        |
| 19.13     | <stdint.h>  | 891        |
| 19.14     | <stdlib.h>  | 892        |
| 19.15     | <string.h>  | 893        |
| 19.16     | <sys/mman.h>  | 894        |
| 19.17     | <sys/socket.h>  | 895        |
| 19.18     | <sys/stat.h>  | 896        |
| 19.19     | <time.h>  | 897        |
| 19.20     | <unistd.h>  | 898        |
| <b>20</b> | <b>FACE Technical Standard, Edition 3.0 Safety Base</b>     | <b>899</b> |
| 20.1      | Summary   | 900        |
| 20.2      | <arpa/inet.h>   | 901        |
| 20.3      | <ctype.h>   | 902        |
| 20.4      | <devctl.h>  | 903        |
| 20.5      | <dirent.h>  | 904        |
| 20.6      | <errno.h>   | 905        |
| 20.7      | <fcntl.h>   | 906        |
| 20.8      | <math.h>  | 907        |
| 20.9      | <mqueue.h>  | 908        |
| 20.10     | <netdb.h>   | 909        |
| 20.11     | <pthread.h>   | 910        |
| 20.12     | <sched.h>   | 912        |
| 20.13     | <semaphore.h>   | 913        |
| 20.14     | <signal.h>  | 914        |
| 20.15     | <stddef.h>  | 915        |
| 20.16     | <stdint.h>  | 916        |
| 20.17     | <stdio.h>   | 917        |
| 20.18     | <stdlib.h>  | 918        |
| 20.19     | <string.h>  | 919        |
| 20.20     | <sys/mman.h>  | 920        |
| 20.21     | <sys/select.h>  | 921        |
| 20.22     | <sys/socket.h>  | 922        |
| 20.23     | <sys/stat.h>  | 923        |
| 20.24     | <time.h>  | 924        |
| 20.25     | <unistd.h>  | 925        |
| <b>21</b> | <b>FACE Technical Standard, Edition 3.0 Safety Extended</b> | <b>927</b> |
| 21.1      | Summary   | 928        |
| 21.2      | <arpa/inet.h>   | 929        |
| 21.3      | <ctype.h>   | 930        |
| 21.4      | <devctl.h>  | 931        |
| 21.5      | <dirent.h>  | 932        |
| 21.6      | <errno.h>   | 933        |
| 21.7      | <fcntl.h>   | 934        |

|           |   |            |
|-----------|---|------------|
| 21.8      | <math.h>  | 935        |
| 21.9      | <mqueue.h>  | 936        |
| 21.10     | <netdb.h>   | 937        |
| 21.11     | <pthread.h>   | 938        |
| 21.12     | <sched.h>   | 940        |
| 21.13     | <semaphore.h>   | 941        |
| 21.14     | <setjmp.h>  | 942        |
| 21.15     | <signal.h>  | 943        |
| 21.16     | <stdarg.h>  | 944        |
| 21.17     | <stddef.h>  | 945        |
| 21.18     | <stdint.h>  | 946        |
| 21.19     | <stdio.h>   | 947        |
| 21.20     | <stdlib.h>  | 948        |
| 21.21     | <string.h>  | 949        |
| 21.22     | <sys/mman.h>  | 950        |
| 21.23     | <sys/select.h>  | 951        |
| 21.24     | <sys/socket.h>  | 952        |
| 21.25     | <sys/stat.h>  | 953        |
| 21.26     | <sys/utsname.h>   | 954        |
| 21.27     | <time.h>  | 955        |
| 21.28     | <unistd.h>  | 956        |
| <b>22</b> | <b>FACE Technical Standard, Edition 3.0 General Purpose</b> | <b>957</b> |
| 22.1      | Summary   | 958        |
| 22.2      | <aio.h>   | 959        |
| 22.3      | <arpa/inet.h>   | 960        |
| 22.4      | <complex.h>   | 961        |
| 22.5      | <ctype.h>   | 963        |
| 22.6      | <devctl.h>  | 964        |
| 22.7      | <dirent.h>  | 965        |
| 22.8      | <errno.h>   | 966        |
| 22.9      | <fcntl.h>   | 967        |
| 22.10     | <fenv.h>  | 968        |
| 22.11     | <inttypes.h>  | 969        |
| 22.12     | <locale.h>  | 970        |
| 22.13     | <math.h>  | 971        |
| 22.14     | <mqueue.h>  | 977        |
| 22.15     | <net/if.h>  | 978        |
| 22.16     | <netdb.h>   | 979        |
| 22.17     | <pthread.h>   | 980        |
| 22.18     | <sched.h>   | 983        |
| 22.19     | <semaphore.h>   | 984        |
| 22.20     | <setjmp.h>  | 985        |
| 22.21     | <signal.h>  | 986        |
| 22.22     | <stdarg.h>  | 987        |
| 22.23     | <stddef.h>  | 988        |
| 22.24     | <stdint.h>  | 989        |
| 22.25     | <stdio.h>   | 990        |
| 22.26     | <stdlib.h>  | 992        |
| 22.27     | <string.h>  | 994        |
| 22.28     | <sys/mman.h>  | 995        |

|           |   |             |
|-----------|---|-------------|
| 22.29     | <sys/select.h>  | 996         |
| 22.30     | <sys/socket.h>  | 997         |
| 22.31     | <sys/stat.h>  | 998         |
| 22.32     | <sys/utsname.h>   | 999         |
| 22.33     | <time.h>  | 1000        |
| 22.34     | <unistd.h>  | 1001        |
| 22.35     | <wchar.h>   | 1003        |
| 22.36     | <wctype.h>  | 1004        |
| <b>23</b> | <b>FACE Technical Standard, Edition 3.1 Security</b>    | <b>1005</b> |
| 23.1      | Summary   | 1006        |
| 23.2      | <arpa/inet.h>   | 1007        |
| 23.3      | <ctype.h>   | 1008        |
| 23.4      | <devctl.h>  | 1009        |
| 23.5      | <errno.h>   | 1010        |
| 23.6      | <math.h>  | 1011        |
| 23.7      | <netdb.h>   | 1012        |
| 23.8      | <pthread.h>   | 1013        |
| 23.9      | <sched.h>   | 1015        |
| 23.10     | <semaphore.h>   | 1016        |
| 23.11     | <signal.h>  | 1017        |
| 23.12     | <stddef.h>  | 1018        |
| 23.13     | <stdint.h>  | 1019        |
| 23.14     | <stdlib.h>  | 1020        |
| 23.15     | <string.h>  | 1021        |
| 23.16     | <sys/mman.h>  | 1022        |
| 23.17     | <sys/socket.h>  | 1023        |
| 23.18     | <sys/stat.h>  | 1024        |
| 23.19     | <time.h>  | 1025        |
| 23.20     | <unistd.h>  | 1026        |
| <b>24</b> | <b>FACE Technical Standard, Edition 3.1 Safety Base</b> | <b>1027</b> |
| 24.1      | Summary   | 1028        |
| 24.2      | <arpa/inet.h>   | 1029        |
| 24.3      | <ctype.h>   | 1030        |
| 24.4      | <devctl.h>  | 1031        |
| 24.5      | <dirent.h>  | 1032        |
| 24.6      | <errno.h>   | 1033        |
| 24.7      | <fcntl.h>   | 1034        |
| 24.8      | <math.h>  | 1035        |
| 24.9      | <mqueue.h>  | 1036        |
| 24.10     | <netdb.h>   | 1037        |
| 24.11     | <pthread.h>   | 1038        |
| 24.12     | <sched.h>   | 1040        |
| 24.13     | <semaphore.h>   | 1041        |
| 24.14     | <signal.h>  | 1042        |
| 24.15     | <stddef.h>  | 1043        |
| 24.16     | <stdint.h>  | 1044        |
| 24.17     | <stdio.h>   | 1045        |
| 24.18     | <stdlib.h>  | 1046        |
| 24.19     | <string.h>  | 1047        |
| 24.20     | <sys/mman.h>  | 1048        |

|           |   |             |
|-----------|---|-------------|
| 24.21     | <sys/select.h>  | 1049        |
| 24.22     | <sys/socket.h>  | 1050        |
| 24.23     | <sys/stat.h>  | 1051        |
| 24.24     | <time.h>  | 1052        |
| 24.25     | <unistd.h>  | 1053        |
| <b>25</b> | <b>FACE Technical Standard, Edition 3.1 Safety Extended</b> | <b>1055</b> |
| 25.1      | Summary   | 1056        |
| 25.2      | <arpa/inet.h>   | 1057        |
| 25.3      | <ctype.h>   | 1058        |
| 25.4      | <devctl.h>  | 1059        |
| 25.5      | <dirent.h>  | 1060        |
| 25.6      | <errno.h>   | 1061        |
| 25.7      | <fcntl.h>   | 1062        |
| 25.8      | <math.h>  | 1063        |
| 25.9      | <mqueue.h>  | 1064        |
| 25.10     | <netdb.h>   | 1065        |
| 25.11     | <pthread.h>   | 1066        |
| 25.12     | <sched.h>   | 1068        |
| 25.13     | <semaphore.h>   | 1069        |
| 25.14     | <setjmp.h>  | 1070        |
| 25.15     | <signal.h>  | 1071        |
| 25.16     | <stdarg.h>  | 1072        |
| 25.17     | <stddef.h>  | 1073        |
| 25.18     | <stdint.h>  | 1074        |
| 25.19     | <stdio.h>   | 1075        |
| 25.20     | <stdlib.h>  | 1076        |
| 25.21     | <string.h>  | 1077        |
| 25.22     | <sys/mman.h>  | 1078        |
| 25.23     | <sys/select.h>  | 1079        |
| 25.24     | <sys/socket.h>  | 1080        |
| 25.25     | <sys/stat.h>  | 1081        |
| 25.26     | <sys/utsname.h>   | 1082        |
| 25.27     | <time.h>  | 1083        |
| 25.28     | <unistd.h>  | 1084        |
| <b>26</b> | <b>FACE Technical Standard, Edition 3.1 General Purpose</b> | <b>1085</b> |
| 26.1      | Summary   | 1086        |
| 26.2      | <aio.h>   | 1087        |
| 26.3      | <arpa/inet.h>   | 1088        |
| 26.4      | <complex.h>   | 1089        |
| 26.5      | <ctype.h>   | 1091        |
| 26.6      | <devctl.h>  | 1092        |
| 26.7      | <dirent.h>  | 1093        |
| 26.8      | <errno.h>   | 1094        |
| 26.9      | <fcntl.h>   | 1095        |
| 26.10     | <fenv.h>  | 1096        |
| 26.11     | <inttypes.h>  | 1097        |
| 26.12     | <locale.h>  | 1098        |
| 26.13     | <math.h>  | 1099        |
| 26.14     | <mqueue.h>  | 1105        |
| 26.15     | <net/if.h>  | 1106        |

|           |   |             |
|-----------|---|-------------|
| 26.16     | <netdb.h>   | 1107        |
| 26.17     | <pthread.h>   | 1108        |
| 26.18     | <sched.h>   | 1111        |
| 26.19     | <semaphore.h>   | 1112        |
| 26.20     | <setjmp.h>  | 1113        |
| 26.21     | <signal.h>  | 1114        |
| 26.22     | <stdarg.h>  | 1115        |
| 26.23     | <stddef.h>  | 1116        |
| 26.24     | <stdint.h>  | 1117        |
| 26.25     | <stdio.h>   | 1118        |
| 26.26     | <stdlib.h>  | 1120        |
| 26.27     | <string.h>  | 1121        |
| 26.28     | <sys/mman.h>  | 1122        |
| 26.29     | <sys/select.h>  | 1123        |
| 26.30     | <sys/socket.h>  | 1124        |
| 26.31     | <sys/stat.h>  | 1125        |
| 26.32     | <sys/utsname.h>   | 1126        |
| 26.33     | <time.h>  | 1127        |
| 26.34     | <unistd.h>  | 1128        |
| <b>27</b> | <b>FACE Technical Standard, Edition 3.2 Security</b>    | <b>1131</b> |
| 27.1      | Summary   | 1132        |
| 27.2      | <arpa/inet.h>   | 1133        |
| 27.3      | <ctype.h>   | 1134        |
| 27.4      | <devctl.h>  | 1135        |
| 27.5      | <errno.h>   | 1136        |
| 27.6      | <math.h>  | 1137        |
| 27.7      | <netdb.h>   | 1138        |
| 27.8      | <pthread.h>   | 1139        |
| 27.9      | <sched.h>   | 1141        |
| 27.10     | <semaphore.h>   | 1142        |
| 27.11     | <signal.h>  | 1143        |
| 27.12     | <stddef.h>  | 1144        |
| 27.13     | <stdint.h>  | 1145        |
| 27.14     | <stdlib.h>  | 1146        |
| 27.15     | <string.h>  | 1147        |
| 27.16     | <sys/mman.h>  | 1148        |
| 27.17     | <sys/socket.h>  | 1149        |
| 27.18     | <sys/stat.h>  | 1150        |
| 27.19     | <time.h>  | 1151        |
| 27.20     | <unistd.h>  | 1152        |
| <b>28</b> | <b>FACE Technical Standard, Edition 3.2 Safety Base</b> | <b>1153</b> |
| 28.1      | Summary   | 1154        |
| 28.2      | <arpa/inet.h>   | 1155        |
| 28.3      | <ctype.h>   | 1156        |
| 28.4      | <devctl.h>  | 1157        |
| 28.5      | <dirent.h>  | 1158        |
| 28.6      | <errno.h>   | 1159        |
| 28.7      | <fcntl.h>   | 1160        |
| 28.8      | <math.h>  | 1161        |
| 28.9      | <mqueue.h>  | 1162        |

|           |   |             |
|-----------|---|-------------|
| 28.10     | <netdb.h>   | 1163        |
| 28.11     | <pthread.h>   | 1164        |
| 28.12     | <sched.h>   | 1166        |
| 28.13     | <semaphore.h>   | 1167        |
| 28.14     | <signal.h>  | 1168        |
| 28.15     | <stddef.h>  | 1169        |
| 28.16     | <stdint.h>  | 1170        |
| 28.17     | <stdio.h>   | 1171        |
| 28.18     | <stdlib.h>  | 1172        |
| 28.19     | <string.h>  | 1173        |
| 28.20     | <sys/mman.h>  | 1174        |
| 28.21     | <sys/select.h>  | 1175        |
| 28.22     | <sys/socket.h>  | 1176        |
| 28.23     | <sys/stat.h>  | 1177        |
| 28.24     | <time.h>  | 1178        |
| 28.25     | <unistd.h>  | 1179        |
| <b>29</b> | <b>FACE Technical Standard, Edition 3.2 Safety Extended</b> | <b>1181</b> |
| 29.1      | Summary   | 1182        |
| 29.2      | <arpa/inet.h>   | 1183        |
| 29.3      | <ctype.h>   | 1184        |
| 29.4      | <devctl.h>  | 1185        |
| 29.5      | <dirent.h>  | 1186        |
| 29.6      | <errno.h>   | 1187        |
| 29.7      | <fcntl.h>   | 1188        |
| 29.8      | <math.h>  | 1189        |
| 29.9      | <mqueue.h>  | 1190        |
| 29.10     | <netdb.h>   | 1191        |
| 29.11     | <pthread.h>   | 1192        |
| 29.12     | <sched.h>   | 1194        |
| 29.13     | <semaphore.h>   | 1195        |
| 29.14     | <setjmp.h>  | 1196        |
| 29.15     | <signal.h>  | 1197        |
| 29.16     | <stdarg.h>  | 1198        |
| 29.17     | <stddef.h>  | 1199        |
| 29.18     | <stdint.h>  | 1200        |
| 29.19     | <stdio.h>   | 1201        |
| 29.20     | <stdlib.h>  | 1202        |
| 29.21     | <string.h>  | 1203        |
| 29.22     | <sys/mman.h>  | 1204        |
| 29.23     | <sys/select.h>  | 1205        |
| 29.24     | <sys/socket.h>  | 1206        |
| 29.25     | <sys/stat.h>  | 1207        |
| 29.26     | <sys/utsname.h>   | 1208        |
| 29.27     | <time.h>  | 1209        |
| 29.28     | <unistd.h>  | 1210        |
| <b>30</b> | <b>FACE Technical Standard, Edition 3.2 General Purpose</b> | <b>1211</b> |
| 30.1      | Summary   | 1212        |
| 30.2      | <aio.h>   | 1213        |
| 30.3      | <arpa/inet.h>   | 1214        |
| 30.4      | <complex.h>   | 1215        |

|           |   |             |
|-----------|---|-------------|
| 30.5      | <ctype.h>   | 1217        |
| 30.6      | <devctl.h>  | 1218        |
| 30.7      | <dirent.h>  | 1219        |
| 30.8      | <errno.h>   | 1220        |
| 30.9      | <fcntl.h>   | 1221        |
| 30.10     | <fenv.h>  | 1222        |
| 30.11     | <inttypes.h>  | 1223        |
| 30.12     | <locale.h>  | 1224        |
| 30.13     | <math.h>  | 1225        |
| 30.14     | <mqueue.h>  | 1231        |
| 30.15     | <net/if.h>  | 1232        |
| 30.16     | <netdb.h>   | 1233        |
| 30.17     | <pthread.h>   | 1234        |
| 30.18     | <sched.h>   | 1237        |
| 30.19     | <semaphore.h>   | 1238        |
| 30.20     | <setjmp.h>  | 1239        |
| 30.21     | <signal.h>  | 1240        |
| 30.22     | <stdarg.h>  | 1241        |
| 30.23     | <stddef.h>  | 1242        |
| 30.24     | <stdint.h>  | 1243        |
| 30.25     | <stdio.h>   | 1244        |
| 30.26     | <stdlib.h>  | 1246        |
| 30.27     | <string.h>  | 1247        |
| 30.28     | <sys/mman.h>  | 1248        |
| 30.29     | <sys/select.h>  | 1249        |
| 30.30     | <sys/socket.h>  | 1250        |
| 30.31     | <sys/stat.h>  | 1251        |
| 30.32     | <sys/utsname.h>   | 1252        |
| 30.33     | <time.h>  | 1253        |
| 30.34     | <unistd.h>  | 1254        |
| <b>31</b> | <b>Software Communications Architecture 2.2.2 AEP</b>   | <b>1257</b> |
| 31.1      | Summary   | 1258        |
| 31.2      | <ctype.h>   | 1259        |
| 31.3      | <dirent.h>  | 1260        |
| 31.4      | <fcntl.h>   | 1261        |
| 31.5      | <locale.h>  | 1262        |
| 31.6      | <math.h>  | 1263        |
| 31.7      | <pthread.h>   | 1264        |
| 31.8      | <semaphore.h>   | 1267        |
| 31.9      | <setjmp.h>  | 1268        |
| 31.10     | <signal.h>  | 1269        |
| 31.11     | <stdio.h>   | 1270        |
| 31.12     | <stdlib.h>  | 1272        |
| 31.13     | <string.h>  | 1273        |
| 31.14     | <sys/stat.h>  | 1274        |
| 31.15     | <time.h>  | 1275        |
| 31.16     | <unistd.h>  | 1276        |
| 31.17     | <utime.h>   | 1277        |
| <b>32</b> | <b>Software Communications Architecture 4.1 Ultra Lightweight Application Environment Profile</b> | <b>1279</b> |

|           |   |             |
|-----------|---|-------------|
| 32.1      | Summary . . . . .   | 1280        |
| 32.2      | <math.h> . . . . .  | 1281        |
| 32.3      | <mqueue.h> . . . . .  | 1282        |
| 32.4      | <pthread.h> . . . . .   | 1283        |
| 32.5      | <semaphore.h> . . . . .   | 1284        |
| 32.6      | <time.h> . . . . .  | 1285        |
| <b>33</b> | <b>Software Communications Architecture 4.1 Lightweight Application Environment Profile</b> | <b>1287</b> |
| 33.1      | Summary . . . . .   | 1288        |
| 33.2      | <ctype.h> . . . . .   | 1289        |
| 33.3      | <fcntl.h> . . . . .   | 1290        |
| 33.4      | <math.h> . . . . .  | 1291        |
| 33.5      | <mqueue.h> . . . . .  | 1292        |
| 33.6      | <pthread.h> . . . . .   | 1293        |
| 33.7      | <semaphore.h> . . . . .   | 1294        |
| 33.8      | <stdio.h> . . . . .   | 1295        |
| 33.9      | <stdlib.h> . . . . .  | 1296        |
| 33.10     | <string.h> . . . . .  | 1297        |
| 33.11     | <time.h> . . . . .  | 1298        |
| 33.12     | <unistd.h> . . . . .  | 1299        |
| <b>34</b> | <b>Software Communications Architecture 4.1 [Full] Appliation Environment Profile</b>       | <b>1301</b> |
| 34.1      | Summary . . . . .   | 1302        |
| 34.2      | <arpa/inet.h> . . . . .   | 1303        |
| 34.3      | <ctype.h> . . . . .   | 1304        |
| 34.4      | <dirent.h> . . . . .  | 1305        |
| 34.5      | <errno.h> . . . . .   | 1306        |
| 34.6      | <fcntl.h> . . . . .   | 1307        |
| 34.7      | <math.h> . . . . .  | 1308        |
| 34.8      | <mqueue.h> . . . . .  | 1309        |
| 34.9      | <pthread.h> . . . . .   | 1310        |
| 34.10     | <semaphore.h> . . . . .   | 1312        |
| 34.11     | <signal.h> . . . . .  | 1313        |
| 34.12     | <stdarg.h> . . . . .  | 1314        |
| 34.13     | <stdio.h> . . . . .   | 1315        |
| 34.14     | <stdlib.h> . . . . .  | 1317        |
| 34.15     | <string.h> . . . . .  | 1318        |
| 34.16     | <sys/select.h> . . . . .  | 1319        |
| 34.17     | <sys/socket.h> . . . . .  | 1320        |
| 34.18     | <sys/stat.h> . . . . .  | 1321        |
| 34.19     | <time.h> . . . . .  | 1322        |
| 34.20     | <unistd.h> . . . . .  | 1323        |
| <b>35</b> | <b>Glossary</b>   | <b>1325</b> |
|           | <b>Index</b>  | <b>1327</b> |



## Copyrights and License

© 2017 Chris Johns

© 1988, 2018 On-Line Applications Research Corporation (OAR)

This document is available under the [Creative Commons Attribution-ShareAlike 4.0 International Public License](#).

The authors have used their best efforts in preparing this material. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. No warranty of any kind, expressed or implied, with regard to the software or the material contained in this document is provided. No liability arising out of the application or use of any product described in this document is assumed. The authors reserve the right to revise this material and to make changes from time to time in the content hereof without obligation to notify anyone of such revision or changes.

The RTEMS Project is hosted at <https://www.rtems.org>. Any inquiries concerning RTEMS, its related support components, or its documentation should be directed to the RTEMS Project community.

## RTEMS Online Resources

|                  |   |
|------------------|---|
| Home             | <a href="https://www.rtems.org">https://www.rtems.org</a>       |
| Documentation    | <a href="https://docs.rtems.org">https://docs.rtems.org</a>     |
| Mailing Lists    | <a href="https://lists.rtems.org">https://lists.rtems.org</a>   |
| Bug Reporting    | <a href="https://gitlab.rtems.org">https://gitlab.rtems.org</a> |
| Git Repositories | <a href="https://gitlab.rtems.org">https://gitlab.rtems.org</a> |
| Developers       | <a href="https://gitlab.rtems.org">https://gitlab.rtems.org</a> |



## PREFACE

RTEMS supports a variety of POSIX and BSD features including some POSIX methods that are now deemed obsolete and some methods for compatibility with GNU/Linux and FreeBSD. There are multiple POSIX standard versions as well as multiple efforts to tailor (e.g. profile) POSIX for embedded environments. They range in size from less than 200 required capabilities to the full POSIX standard which has over 1200 required capabilities. This document reports on the alignment of RTEMS with various standard versions and defined profiles.

RTEMS supports a number of POSIX process, user, and group oriented routines in what is referred to as a “SUSP” (Single-User, Single Process) manner. RTEMS supports a single process, multithreaded POSIX environment. In a pure world, there would be no reason to even include routines like `getpid()` when there can only be one process. But providing routines like `getpid()` and making them work in a sensible fashion for an embedded environment while not returning `ENOSYS` (for not implemented) makes it significantly easier to port code from a UNIX environment without modifying it.

In general, adding missing methods is always an open project for a volunteer. If considering addressing missing methods, please discuss this on mailing list. Some are properly implemented in the Newlib C Standard Library used by RTEMS. Others may require target architecture specific implementations. Still others may be impossible to implement without multiple processes or can only be implemented in a restricted fashion.

Missing methods required by the C99 standard or FACE Technical Standard Edition 3.0 General Purpose Profile are good candidates to add. Proposals to add missing methods from the C11 standard should be reviewed by RTEMS core developers to ensure the effort is well spent. There are rumors that some optional methods that are not being widely implemented will be removed in a future versino of the C Programming Language standard.

The next chapter in this document describes each of the standards with which the RTEMS alignment is tracked. Each subsequent chapter in this document presents the alignment of RTEMS with a specific standard version or defined profile. Each section with a chapter details the alignment of a specific header file relative to the chapter’s standard or profile. The implementation status of the items required by the standard are listed.



# STANDARDS

This chapter describes each of the standards which RTEMS tracks API alignment with. As a general rule, these standards are related to the POSIX or C programming language standards. Many are the result of domain specific efforts to define subsets or profiles or the full POSIX standard which are suitable for a specific domain. Each API set is considered a “profile” against which the full capability set of RTEMS is evaluated.

The RTEMS Complete Profile is the complete set of POSIX, BSD, and C programming language functions supported by RTEMS. This profile is independent of any standard and represents a union of multiple standards. For example, RTEMS supports BSD derived functions that are not in POSIX.

The IEEE Standard 1003.1 is the POSIX standard which is maintained by The Open Group. Specifically, IEEE Standard 1003.1-2003 is the 2003 edition of the POSIX standard which is referred to by The Open Group as Issue 6. IEEE Standard 1003.1-2008 is the 2008 Edition of the standard with two Technical Corrigenda applied. It does not have an issue number associated with it. IEEE Standard 1003.1-2017 is also known as Issue 7. Each edition of the POSIX standard tends to add some functions, deprecate some functions, and obsolete (e.g. remove) other functions.

API differences between Issue 5 and Issue 6 are documented at [https://pubs.opengroup.org/onlinepubs/009695399/xrat/xsh\\_chap01.html](https://pubs.opengroup.org/onlinepubs/009695399/xrat/xsh_chap01.html). There is not a summary for the changes between Issue 6 as published and what was released as 1003.1-2008. However, there is a summary of API changes from Issue 6 to Issue 7 (POSIX 1003.1-2017) at [https://pubs.opengroup.org/onlinepubs/9699919799/xrat/V4\\_xsh\\_chap01.html](https://pubs.opengroup.org/onlinepubs/9699919799/xrat/V4_xsh_chap01.html).

Issue 8 was published in June 2024. The HTML version is not yet available on the web. It did include a list of newly added functions and deprecated functions.

PSE51 through PSE54 are Open Group defined profiles of the 2003 edition of the POSIX standard. These profiles are:

- Profile 54 - Multipurpose
  - 1003.1-2003 Base Multi-process, Threads and File System
- Profile 53 - Dedicated
  - Multi-process, Threads and File System
- Profile 52 - Controller
  - Single Process, Threads, and File System
- Profile 51 - Minimal

– Single Process, Threads, with No File System

The C99 Programming Language standard defines the Standard C Library. This library is largely included by reference in the POSIX standard.

The C11 Programming Language standard defines also defines an updated version of the Standard C Library. It deletes a few functions from the C99 version but adds many functions. A large portion of these functions are optional and not commonly implemented.

Similarly, the C17 Programming Language standard defines an updated version of the Standard C Library.

The Open Group FACE Consortium (<https://www.opengroup.org/face>) has defined four POSIX profiles targetting the avionics application domain. The FACE Technical Standard has been through multiple revisions and the POSIX API profiles are identical in Editions 1.0, 2.0, 2.1, and 2.1.1. In these editions, the profiles and the approximate number of APIs in each are as follows:

- Security - ~165 APIs, single process, no FILE \*
- Safety Basic - ~250 APIs, single process, some FILE \*
- Safety Extended - ~335 APIs, multi-process, more FILE \*
- General Purpose - ~825 APIs, multi-process, much more

FACE Technical Standard, Edition 3.0 adds the requirement for an operating system to support `clock_nanosleep()` in all profiles and defines one additional subcommand for the `posix_devctl()` functions.

FACE Technical Standard, Edition 3.1 has a number of minor changes to the profiles. Most of these were to improve alignment with the Software Communications Architecture (SCA) profiles. Additionally, some inconsistencies in the profiles were noticed and addressed while doing the alignment review.

FACE Technical Standard, Edition 3.2 has a few minor changes to the profiles.

RTEMS provides all of the functions required by the FACE Safety BASE profile and all of the functions in the Safety Extended profile which do not require multiple processes. Similarly, RTEMS provides nearly all of the functions in the General Purpose profile which do not require multiple processes. Support for the functions defined in `fenv.h` is processor architecture dependent.

The Software Communications Architecture (SCA) specification targets the requirements for software-defined radios. This specification was originally developed in support of the Joint Tactical Radio System (JTRS) program in conjunction with the Object Management Group (OMG). This standard is now maintained by the Wireless Innovation Forum with support from the U.S. Navy Joint Tactical Network Center (JTNC). Some URLs of interest:

- SCA at Wireless Innovation Forum - <http://www.wirelessinnovation.org/sca-based-standards-library>
- JTRS - [https://en.wikipedia.org/wiki/Joint\\_Tactical\\_Radio\\_System](https://en.wikipedia.org/wiki/Joint_Tactical_Radio_System)
- JTNC - <http://www.public.navy.mil/jtnc/Pages/home.aspx>

The SCA standard is hosted at the Wireless Innovation Forum with JTNC hosting supplemental information.

RTEMS includes all functions required by the SCA POSIX profiles.

# RTEMS COMPLETE PROFILE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 3.1 Summary

The follow table summarizes RTEMS supported methods for all tracked standards:

|               |      |
|---------------|------|
| Supported     | 1063 |
| ENOSYS        | 18   |
| Not supported | 192  |



## 3.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

### 3.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

### 3.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

### 3.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 3.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalnum_l()`
- `isalpha()`
- `isalpha_l()`
- `isascii()`
- `isblank()`
- `isblank_l()`
- `iscntrl()`
- `iscntrl_l()`
- `isdigit()`
- `isdigit_l()`
- `isgraph()`
- `isgraph_l()`
- `islower()`
- `islower_l()`
- `isprint()`
- `isprint_l()`
- `ispunct()`
- `ispunct_l()`
- `isspace()`
- `isspace_l()`
- `isupper()`
- `isupper_l()`
- `isxdigit()`
- `isxdigit_l()`
- `toascii()`
- `tolower()`
- `tolower_l()`
- `toupper()`
- `toupper_l()`

### 3.7 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

### 3.8 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `alphasort()`
- `closedir()`
- `fdopendir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in <dirent.h> are not supported:

- `dirfd()`



### 3.9 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dLError()`
- `dlopen()`
- `dlsym()`

### 3.10 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

### 3.11 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix\_fadvise()
- posix\_fallocate()
- posix\_openpt()

### 3.12 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

### 3.13 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- `fmtmsg()`

### 3.14 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- fnmatch()

### 3.15 <ftw.h>

The following methods and variables in <ftw.h> are supported:

- `ftw()`
- `nftw()`

### 3.16 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()



### 3.17 <grp.h>

The following methods and variables in <grp.h> are supported:

- `endgrent()`
- `getgrent()`
- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`
- `setgrent()`

### 3.18 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv\_close()
- iconv\_open()

### 3.19 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 3.20 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl\_langinfo()
- nl\_langinfo\_l()

## 3.21 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- `basename()`
- `dirname()`

## 3.22 <libintl.h>

The following methods and variables in <libintl.h> are supported:

- `textdomain()`

### 3.23 <locale.h>

The following methods and variables in <locale.h> are supported:

- duplocale()
- freelocale()
- localeconv()
- newlocale()
- setlocale()
- uselocale()

## 3.24 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asin1()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`



- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalb()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`

- `sqrt()`
- `sqrtf()`
- `sqrtrl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `j0()`
- `j1()`
- `jn()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

### 3.25 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- `strfmon()`
- `strfmon_l()`

## 3.26 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

### 3.27 <ndbm.h>

The following methods and variables in <ndbm.h> are supported:

- dbm\_clearerr()
- dbm\_close()
- dbm\_delete()
- dbm\_error()
- dbm\_fetch()
- dbm\_firstkey()
- dbm\_nextkey()
- dbm\_open()
- dbm\_store()



### 3.28 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 3.29 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostbyaddr()`
- `gethostbyname()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `h_errno`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

### 3.30 <nl\_types.h>

The following methods and variables in <nl\_types.h> are not supported:

- catclose()
- catgets()
- catopen()

### 3.31 <poll.h>

The following methods and variables in <poll.h> are supported:

- poll()

### 3.32 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_getpshared()
- pthread\_barrierattr\_init()
- pthread\_barrierattr\_setpshared()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()

- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`

- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`
- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`



### 3.33 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

### 3.34 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

### 3.35 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

### 3.36 <search.h>

The following methods and variables in <search.h> are supported:

- `hcreate()`
- `hdestroy()`
- `hsearch()`
- `tdelete()`
- `tfind()`
- `tsearch()`
- `twalk()`

The following methods and variables in <search.h> are not supported:

- `insque()`
- `lfind()`
- `lsearch()`
- `remque()`

### 3.37 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

### 3.38 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

The following methods and variables in <setjmp.h> are not supported:

- `_longjmp()`
- `_setjmp()`

### 3.39 <signal.h>

The following methods and variables in <signal.h> are supported:

- `bsd_signal()`
- `kill()`
- `psignal()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sig2str()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

The following methods and variables in <signal.h> are not supported:

- `killpg()`
- `psiginfo()`
- `sigaltstack()`
- `sighold()`
- `sigignore()`
- `siginterrupt()`
- `sigpause()`
- `sigrelse()`
- `sigset()`

### 3.40 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnnp()



### 3.41 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va\_arg()
- va\_copy()
- va\_end()
- va\_start()

## 3.42 <stdatomic.h>

The following methods and variables in <stdatomic.h> are supported:

- `ATOMIC_VAR_INIT()`
- `atomic_compare_exchange_strong()`
- `atomic_compare_exchange_strong_explicit()`
- `atomic_compare_exchange_weak()`
- `atomic_compare_exchange_weak_explicit()`
- `atomic_exchange()`
- `atomic_exchange_explicit()`
- `atomic_fetch_key()`
- `atomic_fetch_key_explicit()`
- `atomic_flag_clear()`
- `atomic_flag_clear_explicit()`
- `atomic_flag_test_and_set()`
- `atomic_flag_test_and_set_explicit()`
- `atomic_init()`
- `atomic_is_lock_free()`
- `atomic_load()`
- `atomic_load_explicit()`
- `atomic_signal_fence()`
- `atomic_store()`
- `atomic_store_explicit()`
- `atomic_thread_fence()`

### 3.43 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 3.44 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

### 3.45 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()

- `getchar_unlocked()`
- `gets()`
- `open_memstream()`
- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tempnam()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vdprintf()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `getdelim()`
- `getline()`
- `pclose()`
- `popen()`
- `renameat()`

## 3.46 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `aligned_alloc()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `ecvt()`
- `erand48()`
- `exit()`
- `fcvt()`
- `free()`
- `gcvt()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`



- `mbstowcs()`
- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`
- `mktemp()`
- `mktime()`
- `mrnd48()`
- `nrnd48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `qsort_r()`
- `quick_exit()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `reallocarray()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`
- `setkey()`
- `setstate()`
- `unlockpt()`

### 3.47 <string.h>

The following methods and variables in <string.h> are supported:

- memccpy()
- memchr()
- memcmp()
- memcpy()
- memmem()
- memmove()
- memset()
- stpcpy()
- stpncpy()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcoll\_l()
- strcpy()
- strcspn()
- strdup()
- strerror()
- strerror\_l()
- strerror\_r()
- strlcat()
- strlcpy()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strndup()
- strnlen()
- strpbrk()
- strrchr()
- strsignal()
- strspn()

- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`
- `strxfrm_l()`

### 3.48 <strings.h>

The following methods and variables in <strings.h> are supported:

- `bcmp()`
- `bcopy()`
- `bzero()`
- `ffs()`
- `ftime()`
- `index()`
- `rindex()`
- `strcasecmp()`
- `strcasecmp_l()`
- `strncasecmp()`
- `strncasecmp_l()`

### 3.49 <stropts.h>

The following methods and variables in <stropts.h> are supported:

- `ioctl()`

The following methods and variables in <stropts.h> are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`

### 3.50 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- ftok()

### 3.51 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in <sys/mman.h> are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`



### 3.52 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()

### 3.53 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

### 3.54 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

### 3.55 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- `semctl()`
- `semget()`
- `semop()`

### 3.56 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

### 3.57 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

### 3.58 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()

### 3.59 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()



## 3.60 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- gettimeofday()
- times()
- utimes()

The following methods in <sys/time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- getitimer()
- setitimer()

### 3.61 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

## 3.62 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

### 3.63 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

## 3.64 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- closelog()
- openlog()
- setlogmask()
- syslog()

### 3.65 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()

## 3.66 <threads.h>

The following methods and variables in <threads.h> are supported:

- `call_once()`
- `cnd_broadcast()`
- `cnd_destroy()`
- `cnd_init()`
- `cnd_signal()`
- `cnd_timedwait()`
- `cnd_wait()`
- `mtx_destroy()`
- `mtx_init()`
- `mtx_lock()`
- `mtx_timedlock()`
- `mtx_trylock()`
- `mtx_unlock()`
- `thrd_create()`
- `thrd_current()`
- `thrd_detach()`
- `thrd_equal()`
- `thrd_exit()`
- `thrd_join()`
- `thrd_sleep()`
- `thrd_yield()`
- `tss_create()`
- `tss_delete()`
- `tss_get()`
- `tss_set()`

## 3.67 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- strftime\_l()
- strptime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight



- `getdate()`
- `getdate_err`

### 3.68 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

### 3.69 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- `ulimit()`

## 3.70 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `ualarm()`
- `unlink()`
- `usleep()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`

- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

### 3.71 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`



## 3.72 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

### 3.73 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open\_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrctomb()
- wcscasecmp()
- wcscasecmp\_l()

- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wcsdup()`
- `wcsftime()`
- `wslcat()`
- `wslcpy()`
- `wslen()`
- `wcsncasecmp()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wctod()`
- `wctof()`
- `wctok()`
- `wctol()`
- `wctold()`
- `wctoll()`
- `wctoul()`
- `wctoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`

- wctob()
- wcwidth()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

The following methods and variables in `<wchar.h>` are not supported:

- wcsncasemcp\_l()

### 3.74 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalnum_l()`
- `iswalpha()`
- `iswalpha_l()`
- `iswblank()`
- `iswblank_l()`
- `iswcntrl()`
- `iswcntrl_l()`
- `iswctype()`
- `iswctype_l()`
- `iswdigit()`
- `iswdigit_l()`
- `iswgraph()`
- `iswgraph_l()`
- `iswlower()`
- `iswlower_l()`
- `iswprint()`
- `iswprint_l()`
- `iswpunct()`
- `iswpunct_l()`
- `iswspace()`
- `iswspace_l()`
- `iswupper()`
- `iswupper_l()`
- `iswxdigit()`
- `iswxdigit_l()`
- `towctrans()`
- `towctrans_l()`
- `towlower()`
- `towlower_l()`
- `towupper()`
- `towupper_l()`

- `wctrans()`
- `wctrans_l()`
- `wctype()`
- `wctype_l()`

### 3.75 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()





## POSIX-2024 (ISSUE 8)

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 4.1 Summary

The follow table summarizes alignment with the POSIX-2024 (Issue 8) standard:

|               |      |
|---------------|------|
| Supported     | 1030 |
| ENOSYS        | 16   |
| Not supported | 125  |

## 4.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 4.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 4.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 4.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 4.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalnum_l()`
- `isalpha()`
- `isalpha_l()`
- `isblank()`
- `isblank_l()`
- `iscntrl()`
- `iscntrl_l()`
- `isdigit()`
- `isdigit_l()`
- `isgraph()`
- `isgraph_l()`
- `islower()`
- `islower_l()`
- `isprint()`
- `isprint_l()`
- `ispunct()`
- `ispunct_l()`
- `isspace()`
- `isspace_l()`
- `isupper()`
- `isupper_l()`
- `isxdigit()`
- `isxdigit_l()`
- `tolower()`
- `tolower_l()`
- `toupper()`
- `toupper_l()`



## 4.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `alphasort()`
- `closedir()`
- `fdopendir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in <dirent.h> are not supported:

- `dirfd()`

## 4.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dLError()`
- `dlopen()`
- `dlsym()`

## 4.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 4.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix\_fadvise()
- posix\_fallocate()
- posix\_openpt()

## 4.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 4.12 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- `fmtmsg()`

## 4.13 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- `fnmatch()`

## 4.14 <ftw.h>

The following methods and variables in <ftw.h> are supported:

- `nftw()`



## 4.15 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()

## 4.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- `endgrent()`
- `getgrent()`
- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`
- `setgrent()`

## 4.17 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv\_close()
- iconv\_open()

## 4.18 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 4.19 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl\_langinfo()
- nl\_langinfo\_l()

## 4.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- `basename()`
- `dirname()`

## 4.21 <libintl.h>

The following methods and variables in <libintl.h> are supported:

- `textdomain()`

## 4.22 <locale.h>

The following methods and variables in <locale.h> are supported:

- duplocale()
- freelocale()
- localeconv()
- newlocale()
- setlocale()
- uselocale()



## 4.23 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceil1()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtrf()`
- `sqrtrl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `j0()`
- `j1()`
- `jn()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

## 4.24 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- `strfmon()`
- `strfmon_l()`

## 4.25 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()



## 4.26 <ndbm.h>

The following methods and variables in <ndbm.h> are supported:

- dbm\_clearerr()
- dbm\_close()
- dbm\_delete()
- dbm\_error()
- dbm\_fetch()
- dbm\_firstkey()
- dbm\_nextkey()
- dbm\_open()
- dbm\_store()

## 4.27 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 4.28 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 4.29 <nl\_types.h>

The following methods and variables in <nl\_types.h> are not supported:

- `catclose()`
- `catgets()`
- `catopen()`

## 4.30 <poll.h>

The following methods and variables in <poll.h> are supported:

- poll()

## 4.31 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_getpshared()
- pthread\_barrierattr\_init()
- pthread\_barrierattr\_setpshared()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()

- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`

- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`
- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`



## 4.32 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

### 4.33 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

## 4.34 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 4.35 <search.h>

The following methods and variables in <search.h> are supported:

- `hcreate()`
- `hdestroy()`
- `hsearch()`
- `tdelete()`
- `tfind()`
- `tsearch()`
- `twalk()`

The following methods and variables in <search.h> are not supported:

- `insque()`
- `lfind()`
- `lsearch()`
- `remque()`

## 4.36 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 4.37 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

## 4.38 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- psignal()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sig2str()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- psiginfo()
- sigaltstack()

## 4.39 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnnp()



## 4.40 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 4.41 <stdatomic.h>

The following methods and variables in <stdatomic.h> are supported:

- `ATOMIC_VAR_INIT()`
- `atomic_compare_exchange_strong()`
- `atomic_compare_exchange_strong_explicit()`
- `atomic_compare_exchange_weak()`
- `atomic_compare_exchange_weak_explicit()`
- `atomic_exchange()`
- `atomic_exchange_explicit()`
- `atomic_fetch_key()`
- `atomic_fetch_key_explicit()`
- `atomic_flag_clear()`
- `atomic_flag_clear_explicit()`
- `atomic_flag_test_and_set()`
- `atomic_flag_test_and_set_explicit()`
- `atomic_init()`
- `atomic_is_lock_free()`
- `atomic_load()`
- `atomic_load_explicit()`
- `atomic_signal_fence()`
- `atomic_store()`
- `atomic_store_explicit()`
- `atomic_thread_fence()`

## 4.42 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 4.43 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 4.44 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanff()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()

- `getchar_unlocked()`
- `open_memstream()`
- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vdprintf()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `getdelim()`

- `getline()`
- `pclose()`
- `popen()`
- `renameat()`

## 4.45 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `aligned_alloc()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `erand48()`
- `exit()`
- `free()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mkdtemp()`



- `mkstemp()`
- `mktime()`
- `rand48()`
- `rand48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `qsort_r()`
- `quick_exit()`
- `rand()`
- `random()`
- `realloc()`
- `reallocarray()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`

- ptsname()
- setkey()
- setstate()
- unlockpt()

## 4.46 <string.h>

The following methods and variables in <string.h> are supported:

- memccpy()
- memchr()
- memcmp()
- memcpy()
- memmem()
- memmove()
- memset()
- stpcpy()
- stpncpy()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcoll\_l()
- strcpy()
- strcspn()
- strdup()
- strerror()
- strerror\_l()
- strerror\_r()
- strlcat()
- strlcpy()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strndup()
- strnlen()
- strpbrk()
- strrchr()
- strsignal()
- strspn()

- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`
- `strxfrm_l()`

## 4.47 <strings.h>

The following methods and variables in <strings.h> are supported:

- ffs()
- strcasecmp()
- strcasecmp\_l()
- strncasecmp()
- strncasecmp\_l()

## 4.48 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- ftok()

## 4.49 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in <sys/mman.h> are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`

## 4.50 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()



## 4.51 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

## 4.52 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 4.53 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- `semctl()`
- `semget()`
- `semop()`

## 4.54 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

## 4.55 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

## 4.56 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()

## 4.57 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()

## 4.58 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- `times()`
- `utimes()`



## 4.59 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

## 4.60 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 4.61 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

## 4.62 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

## 4.63 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()

## 4.64 <threads.h>

The following methods and variables in <threads.h> are supported:

- `call_once()`
- `cnd_broadcast()`
- `cnd_destroy()`
- `cnd_init()`
- `cnd_signal()`
- `cnd_timedwait()`
- `cnd_wait()`
- `mtx_destroy()`
- `mtx_init()`
- `mtx_lock()`
- `mtx_timedlock()`
- `mtx_trylock()`
- `mtx_unlock()`
- `thrd_create()`
- `thrd_current()`
- `thrd_detach()`
- `thrd_equal()`
- `thrd_exit()`
- `thrd_join()`
- `thrd_sleep()`
- `thrd_yield()`
- `tss_create()`
- `tss_delete()`
- `tss_get()`
- `tss_set()`

## 4.65 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- strftime\_l()
- strptime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight

- `getdate()`
- `getdate_err`



## 4.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execlp()`

- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

## 4.67 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- `endutxent()`
- `getutxent()`
- `getutxid()`
- `getutxline()`
- `pututxline()`
- `setutxent()`

## 4.68 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open\_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrntomb()
- wcscasecmp()
- wcscasecmp\_l()

- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wcsdup()`
- `wcsftime()`
- `wcslcat()`
- `wcslcpy()`
- `wcslen()`
- `wcsncasecmp()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`

- `wctob()`
- `wcwidth()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

The following methods and variables in `<wchar.h>` are not supported:

- `wcsncasemcp_l()`

## 4.69 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalnum_l()`
- `iswalpha()`
- `iswalpha_l()`
- `iswblank()`
- `iswblank_l()`
- `iswcntrl()`
- `iswcntrl_l()`
- `iswctype()`
- `iswctype_l()`
- `iswdigit()`
- `iswdigit_l()`
- `iswgraph()`
- `iswgraph_l()`
- `iswlower()`
- `iswlower_l()`
- `iswprint()`
- `iswprint_l()`
- `iswpunct()`
- `iswpunct_l()`
- `iswspace()`
- `iswspace_l()`
- `iswupper()`
- `iswupper_l()`
- `iswxdigit()`
- `iswxdigit_l()`
- `towctrans()`
- `towctrans_l()`
- `towlower()`
- `towlower_l()`
- `towupper()`
- `towupper_l()`



- `wctrans()`
- `wctrans_l()`
- `wctype()`
- `wctype_l()`

## 4.70 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()

## POSIX-2017 (ISSUE 7)

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 5.1 Summary

The follow table summarizes alignment with the POSIX-2017 (Issue 7) standard:

|               |     |
|---------------|-----|
| Supported     | 986 |
| ENOSYS        | 18  |
| Not supported | 192 |

## 5.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

### 5.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 5.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 5.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`



- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 5.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalnum_l()`
- `isalpha()`
- `isalpha_l()`
- `isascii()`
- `isblank()`
- `isblank_l()`
- `iscntrl()`
- `iscntrl_l()`
- `isdigit()`
- `isdigit_l()`
- `isgraph()`
- `isgraph_l()`
- `islower()`
- `islower_l()`
- `isprint()`
- `isprint_l()`
- `ispunct()`
- `ispunct_l()`
- `isspace()`
- `isspace_l()`
- `isupper()`
- `isupper_l()`
- `isxdigit()`
- `isxdigit_l()`
- `toascii()`
- `tolower()`
- `tolower_l()`
- `toupper()`
- `toupper_l()`

## 5.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `alphasort()`
- `closedir()`
- `fdopendir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in <dirent.h> are not supported:

- `dirfd()`

## 5.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dLError()`
- `dlopen()`
- `dlsym()`

## 5.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 5.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix\_fadvise()
- posix\_fallocate()
- posix\_openpt()

## 5.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 5.12 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- `fmtmsg()`



## 5.13 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- fnmatch()

## 5.14 <ftw.h>

The following methods and variables in <ftw.h> are supported:

- `ftw()`
- `nftw()`

## 5.15 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()

## 5.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- `endgrent()`
- `getgrent()`
- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`
- `setgrent()`

## 5.17 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv\_close()
- iconv\_open()

## 5.18 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 5.19 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl\_langinfo()
- nl\_langinfo\_l()

## 5.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- `basename()`
- `dirname()`



## 5.21 <locale.h>

The following methods and variables in <locale.h> are supported:

- duplocale()
- freelocale()
- localeconv()
- newlocale()
- setlocale()
- uselocale()

## 5.22 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceil1()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `j0()`
- `j1()`
- `jn()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

## 5.23 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- `strfmon()`
- `strfmon_l()`



## 5.24 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 5.25 <ndbm.h>

The following methods and variables in <ndbm.h> are supported:

- dbm\_clearerr()
- dbm\_close()
- dbm\_delete()
- dbm\_error()
- dbm\_fetch()
- dbm\_firstkey()
- dbm\_nextkey()
- dbm\_open()
- dbm\_store()

## 5.26 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 5.27 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 5.28 <nl\_types.h>

The following methods and variables in <nl\_types.h> are not supported:

- catclose()
- catgets()
- catopen()

## 5.29 <poll.h>

The following methods and variables in <poll.h> are supported:

- poll()

## 5.30 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_getpshared()
- pthread\_barrierattr\_init()
- pthread\_barrierattr\_setpshared()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()

- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`



- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`

- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`

## 5.31 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

## 5.32 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

### 5.33 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 5.34 <search.h>

The following methods and variables in <search.h> are supported:

- `hcreate()`
- `hdestroy()`
- `hsearch()`
- `tdelete()`
- `tfind()`
- `tsearch()`
- `twalk()`

The following methods and variables in <search.h> are not supported:

- `insque()`
- `lfind()`
- `lsearch()`
- `remque()`

## 5.35 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 5.36 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

The following methods and variables in <setjmp.h> are not supported:

- `_longjmp()`
- `_setjmp()`



## 5.37 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- psignal()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- psiginfo()
- sigaltstack()
- sighold()
- sigignore()
- siginterrupt()
- sigpause()
- sigrelse()
- sigset()

## 5.38 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnnp()

## 5.39 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 5.40 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 5.41 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 5.42 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanff()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()

- `getchar_unlocked()`
- `gets()`
- `open_memstream()`
- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tempnam()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vdprintf()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `getdelim()`
- `getline()`
- `pclose()`
- `popen()`
- `renameat()`



## 5.43 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `erand48()`
- `exit()`
- `free()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`

- `mktime()`
- `mrnd48()`
- `nrnd48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`
- `setkey()`
- `setstate()`

- `unlockpt()`

## 5.44 <string.h>

The following methods and variables in <string.h> are supported:

- memccpy()
- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- stpcpy()
- stpncpy()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcoll\_l()
- strcpy()
- strcspn()
- strdup()
- strerror()
- strerror\_l()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strndup()
- strnlen()
- strpbrk()
- strrchr()
- strsignal()
- strspn()
- strstr()
- strtok()
- strtok\_r()

- `strxfrm()`
- `strxfrm_l()`

## 5.45 <strings.h>

The following methods and variables in <strings.h> are supported:

- ffs()
- strcasecmp()
- strcasecmp\_l()
- strncasecmp()
- strncasecmp\_l()

## 5.46 <stropts.h>

The following methods and variables in <stropts.h> are supported:

- `ioctl()`

The following methods and variables in <stropts.h> are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`

## 5.47 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- ftok()



## 5.48 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in <sys/mman.h> are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`

## 5.49 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()

## 5.50 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

## 5.51 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 5.52 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- `semctl()`
- `semget()`
- `semop()`

## 5.53 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

## 5.54 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

## 5.55 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()



## 5.56 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()

## 5.57 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- gettimeofday()
- times()
- utimes()

The following methods in <sys/time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- getitimer()
- setitimer()

## 5.58 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

## 5.59 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 5.60 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

## 5.61 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

## 5.62 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()

## 5.63 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- strftime\_l()
- strptime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight



- `getdate()`
- `getdate_err`

## 5.64 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

## 5.65 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- `ulimit()`

## 5.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execlp()`

- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

## 5.67 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`



## 5.68 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

## 5.69 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open\_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrntomb()
- wcscasecmp()
- wcscasecmp\_l()

- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wcsdup()`
- `wcsftime()`
- `wcslen()`
- `wcsncasecmp()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`
- `wctob()`
- `wcwidth()`

- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

The following methods and variables in `<wchar.h>` are not supported:

- `wcsncasemcp_l()`

## 5.70 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalnum_l()`
- `iswalpha()`
- `iswalpha_l()`
- `iswblank()`
- `iswblank_l()`
- `iswcntrl()`
- `iswcntrl_l()`
- `iswctype()`
- `iswctype_l()`
- `iswdigit()`
- `iswdigit_l()`
- `iswgraph()`
- `iswgraph_l()`
- `iswlower()`
- `iswlower_l()`
- `iswprint()`
- `iswprint_l()`
- `iswpunct()`
- `iswpunct_l()`
- `iswspace()`
- `iswspace_l()`
- `iswupper()`
- `iswupper_l()`
- `iswxdigit()`
- `iswxdigit_l()`
- `towctrans()`
- `towctrans_l()`
- `towlower()`
- `towlower_l()`
- `towupper()`
- `towupper_l()`

- `wctrans()`
- `wctrans_l()`
- `wctype()`
- `wctype_l()`

## 5.71 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()





## POSIX-2008 (ISSUE 6 TC2)

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 6.1 Summary

The follow table summarizes alignment with the POSIX-2008 (Issue 6 TC2) standard:

|               |     |
|---------------|-----|
| Supported     | 986 |
| ENOSYS        | 18  |
| Not supported | 192 |

## 6.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 6.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 6.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 6.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 6.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalnum_l()`
- `isalpha()`
- `isalpha_l()`
- `isascii()`
- `isblank()`
- `isblank_l()`
- `iscntrl()`
- `iscntrl_l()`
- `isdigit()`
- `isdigit_l()`
- `isgraph()`
- `isgraph_l()`
- `islower()`
- `islower_l()`
- `isprint()`
- `isprint_l()`
- `ispunct()`
- `ispunct_l()`
- `isspace()`
- `isspace_l()`
- `isupper()`
- `isupper_l()`
- `isxdigit()`
- `isxdigit_l()`
- `toascii()`
- `tolower()`
- `tolower_l()`
- `toupper()`
- `toupper_l()`



## 6.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `alphasort()`
- `closedir()`
- `fdopendir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `scandir()`
- `seekdir()`
- `telldir()`

The following methods and variables in <dirent.h> are not supported:

- `dirfd()`

## 6.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dLError()`
- `dlopen()`
- `dlsym()`

## 6.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 6.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- openat()
- posix\_fadvise()
- posix\_fallocate()
- posix\_openpt()

## 6.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 6.12 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- `fmtmsg()`

## 6.13 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- fnmatch()

## 6.14 <ftw.h>

The following methods and variables in <ftw.h> are supported:

- `ftw()`
- `nftw()`



## 6.15 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()

## 6.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- `endgrent()`
- `getgrent()`
- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`
- `setgrent()`

## 6.17 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv\_close()
- iconv\_open()

## 6.18 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 6.19 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl\_langinfo()
- nl\_langinfo\_l()

## 6.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- `basename()`
- `dirname()`

## 6.21 <locale.h>

The following methods and variables in <locale.h> are supported:

- duplocale()
- freelocale()
- localeconv()
- newlocale()
- setlocale()
- uselocale()

## 6.22 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`



- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtrf()`
- `sqrtrl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `j0()`
- `j1()`
- `jn()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

## 6.23 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- `strfmon()`
- `strfmon_l()`

## 6.24 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 6.25 <ndbm.h>

The following methods and variables in <ndbm.h> are supported:

- dbm\_clearerr()
- dbm\_close()
- dbm\_delete()
- dbm\_error()
- dbm\_fetch()
- dbm\_firstkey()
- dbm\_nextkey()
- dbm\_open()
- dbm\_store()



## 6.26 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 6.27 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 6.28 <nl\_types.h>

The following methods and variables in <nl\_types.h> are not supported:

- catclose()
- catgets()
- catopen()

## 6.29 <poll.h>

The following methods and variables in <poll.h> are supported:

- poll()

## 6.30 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_getpshared()
- pthread\_barrierattr\_init()
- pthread\_barrierattr\_setpshared()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()

- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`

- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

The following methods and variables in `<pthread.h>` are not supported:

- `pthread_mutex_consistent()`

- `pthread_mutexattr_getrobust()`
- `pthread_mutexattr_setrobust()`



## 6.31 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

## 6.32 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

## 6.33 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 6.34 <search.h>

The following methods and variables in <search.h> are supported:

- `hcreate()`
- `hdestroy()`
- `hsearch()`
- `tdelete()`
- `tfind()`
- `tsearch()`
- `twalk()`

The following methods and variables in <search.h> are not supported:

- `insque()`
- `lfind()`
- `lsearch()`
- `remque()`

## 6.35 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 6.36 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

The following methods and variables in <setjmp.h> are not supported:

- \_longjmp()
- \_setjmp()

## 6.37 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- psignal()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

The following methods and variables in <signal.h> are not supported:

- killpg()
- psiginfo()
- sigaltstack()
- sighold()
- sigignore()
- siginterrupt()
- sigpause()
- sigrelse()
- sigset()

## 6.38 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnnp()



## 6.39 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 6.40 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 6.41 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 6.42 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanff()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()

- `getchar_unlocked()`
- `gets()`
- `open_memstream()`
- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tempnam()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vdprintf()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `getdelim()`
- `getline()`
- `pclose()`
- `popen()`
- `renameat()`

## 6.43 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `erand48()`
- `exit()`
- `free()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`

- `mktime()`
- `mrnd48()`
- `nrnd48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`
- `ptsname()`
- `setkey()`
- `setstate()`



- `unlockpt()`

## 6.44 <string.h>

The following methods and variables in <string.h> are supported:

- memccpy()
- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- stpcpy()
- stpncpy()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcoll\_l()
- strcpy()
- strcspn()
- strdup()
- strerror()
- strerror\_l()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strndup()
- strnlen()
- strpbrk()
- strrchr()
- strsignal()
- strspn()
- strstr()
- strtok()
- strtok\_r()

- `strxfrm()`
- `strxfrm_l()`

## 6.45 <strings.h>

The following methods and variables in <strings.h> are supported:

- ffs()
- strcasecmp()
- strcasecmp\_l()
- strncasecmp()
- strncasecmp\_l()

## 6.46 <stropts.h>

The following methods and variables in <stropts.h> are supported:

- `ioctl()`

The following methods and variables in <stropts.h> are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`

## 6.47 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- ftok()

## 6.48 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in <sys/mman.h> are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`

## 6.49 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()



## 6.50 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

## 6.51 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 6.52 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- `semctl()`
- `semget()`
- `semop()`

## 6.53 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

## 6.54 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

## 6.55 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- mknod()
- stat()
- umask()

The following methods and variables in <sys/stat.h> are not supported:

- fchmodat()
- fstatat()
- futimens()
- mkdirat()
- mkfifoat()
- mknodat()
- utimensat()

## 6.56 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()

## 6.57 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- gettimeofday()
- times()
- utimes()

The following methods in <sys/time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- getitimer()
- setitimer()



## 6.58 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

## 6.59 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 6.60 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

## 6.61 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`

## 6.62 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()

## 6.63 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- strftime\_l()
- strptime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight

- `getdate()`
- `getdate_err`

## 6.64 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`



- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

## 6.65 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- `ulimit()`

## 6.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execlp()`

- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `faccessat()`
- `fchownat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

## 6.67 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`

## 6.68 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

## 6.69 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsnrtowcs()
- mbsrtowcs()
- open\_wmemstream()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcpcpy()
- wcpncpy()
- wcrntomb()
- wcscasecmp()
- wcscasecmp\_l()



- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wcsdup()`
- `wcsftime()`
- `wcslen()`
- `wcsncasecmp()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`
- `wctob()`
- `wcwidth()`

- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

The following methods and variables in `<wchar.h>` are not supported:

- `wcsncasemcp_l()`

## 6.70 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalnum_l()`
- `iswalpha()`
- `iswalpha_l()`
- `iswblank()`
- `iswblank_l()`
- `iswcntrl()`
- `iswcntrl_l()`
- `iswctype()`
- `iswctype_l()`
- `iswdigit()`
- `iswdigit_l()`
- `iswgraph()`
- `iswgraph_l()`
- `iswlower()`
- `iswlower_l()`
- `iswprint()`
- `iswprint_l()`
- `iswpunct()`
- `iswpunct_l()`
- `iswspace()`
- `iswspace_l()`
- `iswupper()`
- `iswupper_l()`
- `iswxdigit()`
- `iswxdigit_l()`
- `towctrans()`
- `towctrans_l()`
- `towlower()`
- `towlower_l()`
- `towupper()`
- `towupper_l()`

- `wctrans()`
- `wctrans_l()`
- `wctype()`
- `wctype_l()`

## 6.71 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()



## POSIX-2003 (ISSUE 6)

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 7.1 Summary

The follow table summarizes alignment with the POSIX-2003 (Issue 6) standard:

|               |     |
|---------------|-----|
| Supported     | 938 |
| ENOSYS        | 18  |
| Not supported | 166 |



## 7.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 7.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 7.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 7.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 7.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalpha()`
- `isascii()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `islower_l()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `toascii()`
- `tolower()`
- `toupper()`

## 7.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `seekdir()`

## 7.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dLError()`
- `dlopen()`
- `dlsym()`



## 7.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 7.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- posix\_fadvise()
- posix\_fallocate()
- posix\_openpt()

## 7.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 7.12 <fmtmsg.h>

The following methods and variables in <fmtmsg.h> are not supported:

- `fmtmsg()`

## 7.13 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- `fnmatch()`

## 7.14 <ftw.h>

The following methods and variables in <ftw.h> are supported:

- `ftw()`
- `nftw()`

## 7.15 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()

## 7.16 <grp.h>

The following methods and variables in <grp.h> are supported:

- `endgrent()`
- `getgrent()`
- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`
- `setgrent()`



## 7.17 <iconv.h>

The following methods and variables in <iconv.h> are supported:

- iconv()
- iconv\_close()
- iconv\_open()

## 7.18 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 7.19 <langinfo.h>

The following methods and variables in <langinfo.h> are supported:

- nl\_langinfo()

## 7.20 <libgen.h>

The following methods and variables in <libgen.h> are supported:

- `basename()`
- `dirname()`

## 7.21 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 7.22 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`



- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalb()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`

- `sqrt()`
- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tan1()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `j0()`
- `j1()`
- `jn()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

## 7.23 <monetary.h>

The following methods and variables in <monetary.h> are not supported:

- `strfmon()`

## 7.24 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 7.25 <ndbm.h>

The following methods and variables in <ndbm.h> are supported:

- dbm\_clearerr()
- dbm\_close()
- dbm\_delete()
- dbm\_error()
- dbm\_fetch()
- dbm\_firstkey()
- dbm\_nextkey()
- dbm\_open()
- dbm\_store()

## 7.26 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 7.27 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostbyaddr()`
- `gethostbyname()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `h_errno`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`



## 7.28 <nl\_types.h>

The following methods and variables in <nl\_types.h> are not supported:

- `catclose()`
- `catgets()`
- `catopen()`

## 7.29 <poll.h>

The following methods and variables in <poll.h> are supported:

- poll()

## 7.30 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_getpshared()
- pthread\_barrierattr\_init()
- pthread\_barrierattr\_setpshared()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()

- `pthread_cond_init()`
- `pthread_cond_signal()`
- `pthread_cond_timedwait()`
- `pthread_cond_wait()`
- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`

- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_getpshared()`
- `pthread_rwlockattr_init()`
- `pthread_rwlockattr_setpshared()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_spin_destroy()`
- `pthread_spin_init()`
- `pthread_spin_lock()`
- `pthread_spin_trylock()`
- `pthread_spin_unlock()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 7.31 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `endpwent()`
- `getpwent()`
- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`
- `setpwent()`

## 7.32 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

## 7.33 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()



## 7.34 <search.h>

The following methods and variables in <search.h> are supported:

- `hcreate()`
- `hdestroy()`
- `hsearch()`
- `tdelete()`
- `tfind()`
- `tsearch()`
- `twalk()`

The following methods and variables in <search.h> are not supported:

- `insque()`
- `lfind()`
- `lsearch()`
- `remque()`

## 7.35 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 7.36 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

The following methods and variables in <setjmp.h> are not supported:

- \_longjmp()
- \_setjmp()

## 7.37 <signal.h>

The following methods and variables in <signal.h> are supported:

- `bsd_signal()`
- `kill()`
- `pthread_kill()`
- `pthread_sigmask()`
- `raise()`
- `sigaction()`
- `sigaddset()`
- `sigdelset()`
- `sigemptyset()`
- `sigfillset()`
- `sigismember()`
- `signal()`
- `sigpending()`
- `sigprocmask()`
- `sigqueue()`
- `sigsuspend()`
- `sigtimedwait()`
- `sigwait()`
- `sigwaitinfo()`

The following methods and variables in <signal.h> are not supported:

- `killpg()`
- `sighold()`
- `sigignore()`
- `siginterrupt()`
- `sigpause()`
- `sigrelse()`
- `sigset()`

## 7.38 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- `posix_spawn()`
- `posix_spawn_file_actions_addclose()`
- `posix_spawn_file_actions_adddup2()`
- `posix_spawn_file_actions_addopen()`
- `posix_spawn_file_actions_destroy()`
- `posix_spawn_file_actions_init()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getpgroup()`
- `posix_spawnattr_getschedparam()`
- `posix_spawnattr_getschedpolicy()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setpgroup()`
- `posix_spawnattr_setschedparam()`
- `posix_spawnattr_setschedpolicy()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`
- `posix_spawnp()`

## 7.39 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 7.40 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 7.41 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()



## 7.42 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- `clearerr()`
- `ctermid()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgetpos()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fputc()`
- `fputs()`
- `fread()`
- `freopen()`
- `fscanf()`
- `fseek()`
- `fseeko()`
- `fsetpos()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `getc()`
- `getc_unlocked()`
- `getchar()`
- `getchar_unlocked()`
- `gets()`

- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tempnam()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `pclose()`
- `popen()`

## 7.43 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `drand48()`
- `ecvt()`
- `erand48()`
- `exit()`
- `fcvt()`
- `free()`
- `gcvt()`
- `getenv()`
- `getsubopt()`
- `jrand48()`
- `l64a()`
- `labs()`
- `lcong48()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `lrand48()`
- `malloc()`
- `mblen()`
- `mbstowcs()`

- `mbtowc()`
- `mkstemp()`
- `mktemp()`
- `mktime()`
- `mrnd48()`
- `nrand48()`
- `posix_memalign()`
- `putenv()`
- `qsort()`
- `rand()`
- `rand_r()`
- `random()`
- `realloc()`
- `realpath()`
- `seed48()`
- `setenv()`
- `srand()`
- `srand48()`
- `srandom()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

The following methods and variables in `<stdlib.h>` are not supported:

- `grantpt()`
- `initstate()`

- ptsname()
- setkey()
- setstate()
- unlockpt()

## 7.44 <string.h>

The following methods and variables in <string.h> are supported:

- memccpy()
- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strdup()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 7.45 <strings.h>

The following methods and variables in <strings.h> are supported:

- `bcmp()`
- `bcopy()`
- `bzero()`
- `ffs()`
- `ftime()`
- `index()`
- `rindex()`
- `strcasecmp()`
- `strncasecmp()`

## 7.46 <stropts.h>

The following methods and variables in <stropts.h> are supported:

- `ioctl()`

The following methods and variables in <stropts.h> are not supported:

- `fattach()`
- `fdetach()`
- `getmsg()`
- `getpmsg()`
- `isastream()`
- `putmsg()`
- `putpmsg()`



## 7.47 <sys/ipc.h>

The following methods and variables in <sys/ipc.h> are not supported:

- `ftok()`

## 7.48 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

The following methods and variables in <sys/mman.h> are not supported:

- `posix_mem_offset()`
- `posix_typed_mem_get_info()`
- `posix_typed_mem_open()`

## 7.49 <sys/msg.h>

The following methods and variables in <sys/msg.h> are not supported:

- msgctl()
- msgget()
- msgrcv()
- msgsnd()

## 7.50 <sys/resource.h>

The following methods and variables in <sys/resource.h> are supported:

- getrusage()

The following methods and variables in <sys/resource.h> are not supported:

- getpriority()
- getrlimit()
- setpriority()
- setrlimit()

## 7.51 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 7.52 <sys/sem.h>

The following methods and variables in <sys/sem.h> are not supported:

- semctl()
- semget()
- semop()

## 7.53 <sys/shm.h>

The following methods and variables in <sys/shm.h> are not supported:

- `shmat()`
- `shmctl()`
- `shmdt()`
- `shmget()`

## 7.54 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`



## 7.55 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `chmod()`
- `fchmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `mknod()`
- `stat()`
- `umask()`

## 7.56 <sys/statvfs.h>

The following methods and variables in <sys/statvfs.h> are supported:

- statvfs()

The following methods and variables in <sys/statvfs.h> are not supported:

- fstatvfs()

## 7.57 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- gettimeofday()
- times()
- utimes()

The following methods in <sys/time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- getitimer()
- setitimer()

## 7.58 <sys/uio.h>

The following methods and variables in <sys/uio.h> are supported:

- readv()
- writev()

## 7.59 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 7.60 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

The following methods and variables in <sys/wait.h> are not supported:

- waitid()

## 7.61 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- closelog()
- openlog()
- setlogmask()
- syslog()

## 7.62 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

The following methods and variables in <termios.h> are not supported:

- tcgetsid()



## 7.63 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- strptime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- timezone
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

The following methods and variables in <time.h> are not supported:

- daylight
- getdate()

- `getdate_err`

## 7.64 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

## 7.65 <ulimit.h>

The following methods and variables in <ulimit.h> are not supported:

- `ulimit()`

## 7.66 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchdir()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `lchown()`
- `link()`
- `lseek()`

- `optarg`
- `opterr`
- `optind`
- `optopt`
- `pathconf()`
- `pause()`
- `pipe()`
- `pread()`
- `pwrite()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `swab()`
- `symlink()`
- `sync()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `truncate()`
- `ttyname()`
- `ttyname_r()`
- `ualarm()`
- `unlink()`
- `usleep()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `execl()`

- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`
- `crypt()`
- `encrypt()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `lockf()`
- `nice()`
- `setpgrp()`
- `setregid()`
- `setreuid()`



## 7.67 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`

## 7.68 <utmpx.h>

The following methods and variables in <utmpx.h> are not supported:

- endutxent()
- getutxent()
- getutxid()
- getutxline()
- pututxline()
- setutxent()

## 7.69 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcsnlen()`
- `wcsnrtombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wctob()`
- `wcwidth()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

## 7.70 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

## 7.71 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()

## POSIX PSE51 - MINIMAL

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 8.1 Summary

The follow table summarizes alignment with the POSIX PSE51 - Minimal standard:

|               |     |
|---------------|-----|
| Supported     | 282 |
| ENOSYS        | 2   |
| Not supported | 1   |



## 8.2 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

### 8.3 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 8.4 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- `open()`

## 8.5 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 8.6 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

## 8.7 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 8.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()

- `pthread_condattr_setclock()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`



- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 8.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()

## 8.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 8.11 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

## 8.12 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 8.13 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 8.14 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 8.15 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanff()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- gets()
- perror()
- printf()
- putc()
- putc\_unlocked()
- putchar()
- putchar\_unlocked()
- puts()



- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 8.16 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 8.17 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 8.18 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 8.19 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 8.20 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 8.21 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- close()
- environ
- fdatsync()
- fsync()
- pause()
- read()
- sysconf()
- write()

The following methods and variables in <unistd.h> are not supported:

- confstr()





# POSIX PSE52 - REAL-TIME CONTROLLER

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 9.1 Summary

The follow table summarizes alignment with the POSIX PSE52 - Real-Time Controller standard:

|               |     |
|---------------|-----|
| Supported     | 575 |
| ENOSYS        | 2   |
| Not supported | 52  |

## 9.2 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 9.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 9.4 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 9.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 9.6 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()



## 9.7 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 9.8 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

## 9.9 <locale.h>

The following methods and variables in <locale.h> are supported:

- localeconv()
- setlocale()

## 9.10 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asin1()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfc1()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`



- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 9.11 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 9.12 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()

- `pthread_condattr_setclock()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 9.13 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()

## 9.14 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 9.15 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()



## 9.16 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 9.17 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 9.18 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 9.19 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- gets()
- perror()



## 9.20 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtodf()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 9.21 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

## 9.22 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `msync()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`



## 9.23 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`

## 9.24 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 9.25 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 9.26 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

## 9.27 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- close()
- dup()
- dup2()
- environ
- fdatsync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- link()
- lseek()
- pathconf()
- pause()
- read()
- rmdir()
- sysconf()
- unlink()
- write()

The following methods and variables in <unistd.h> are not supported:

- confstr()

## 9.28 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`





## POSIX PSE53 - DEDICATED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 10.1 Summary

The follow table summarizes alignment with the POSIX PSE53 - Dedicated standard:

|               |     |
|---------------|-----|
| Supported     | 661 |
| ENOSYS        | 15  |
| Not supported | 75  |

## 10.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 10.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 10.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 10.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 10.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()



## 10.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 10.8 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 10.9 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 10.10 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 10.11 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

## 10.12 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 10.13 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asin1()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`



- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 10.14 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 10.15 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 10.16 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 10.17 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_getpshared()



- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 10.18 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 10.19 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 10.20 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

## 10.21 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 10.22 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- `posix_spawn()`
- `posix_spawn_file_actions_addclose()`
- `posix_spawn_file_actions_adddup2()`
- `posix_spawn_file_actions_addopen()`
- `posix_spawn_file_actions_destroy()`
- `posix_spawn_file_actions_init()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getpgroup()`
- `posix_spawnattr_getschedparam()`
- `posix_spawnattr_getschedpolicy()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setpgroup()`
- `posix_spawnattr_setschedparam()`
- `posix_spawnattr_setschedpolicy()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`
- `posix_spawnp()`

## 10.23 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`



## 10.24 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 10.25 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- gets()
- perror()

- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 10.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- \_Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtodf()
- strtol()
- strtold()
- strtoll()
- strtoul()

- strtoull()
- unsetenv()

## 10.27 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 10.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 10.29 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()



## 10.30 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

## 10.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`

## 10.32 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- times()
- utimes()

## 10.33 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 10.34 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()

## 10.35 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

## 10.36 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`



## 10.37 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

The following methods in <unistd.h> are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`

- `exec1p()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`

## 10.38 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`



## POSIX PSE54 - MULTIPURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 11.1 Summary

The follow table summarizes alignment with the POSIX PSE54 - Multipurpose standard:

|               |     |
|---------------|-----|
| Supported     | 805 |
| ENOSYS        | 16  |
| Not supported | 85  |

## 11.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 11.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`



## 11.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 11.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 11.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 11.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 11.8 <dlfcn.h>

The following methods and variables in <dlfcn.h> are supported:

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

## 11.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 11.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

The following methods and variables in <fcntl.h> are not supported:

- posix\_fadvise()
- posix\_fallocate()



## 11.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 11.12 <fnmatch.h>

The following methods and variables in <fnmatch.h> are supported:

- fnmatch()

## 11.13 <glob.h>

The following methods and variables in <glob.h> are supported:

- glob()
- globfree()

## 11.14 <grp.h>

The following methods and variables in <grp.h> are supported:

- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`

## 11.15 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 11.16 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 11.17 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceil1()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`



- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 11.18 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 11.19 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 11.20 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 11.21 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_getpshared()



- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 11.22 <pwd.h>

The following methods and variables in <pwd.h> are supported:

- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`

## 11.23 <regex.h>

The following methods and variables in <regex.h> are supported:

- regcomp()
- regerror()
- regexec()
- regfree()

## 11.24 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 11.25 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 11.26 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

## 11.27 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()



## 11.28 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnp()

## 11.29 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 11.30 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 11.31 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- ctermid()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- gets()

- `perror()`
- `printf()`
- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

The following methods and variables in `<stdio.h>` are not supported:

- `pclose()`
- `popen()`

## 11.32 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mktime()`
- `posix_memalign()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`

- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `unsetenv()`
- `wcstombs()`
- `wctomb()`

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

## 11.33 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()



## 11.34 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munmap()`
- `posix_madvise()`
- `shm_open()`
- `shm_unlink()`

## 11.35 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 11.36 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `recvmsg()`
- `send()`
- `sendmsg()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`
- `socketpair()`

The following methods and variables in <sys/socket.h> are not supported:

- `socketatmark()`

## 11.37 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `chmod()`
- `fchmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

## 11.38 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- `times()`
- `utimes()`

## 11.39 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 11.40 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()

## 11.41 <syslog.h>

The following methods and variables in <syslog.h> are not supported:

- `closelog()`
- `openlog()`
- `setlogmask()`
- `syslog()`



## 11.42 <termios.h>

The following methods and variables in <termios.h> are supported:

- cfgetispeed()
- cfgetospeed()
- cfsetispeed()
- cfsetospeed()
- tcdrain()
- tcflow()
- tcflush()
- tcgetattr()
- tcsendbreak()
- tcsetattr()

## 11.43 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

## 11.44 <trace.h>

The following methods and variables in <trace.h> are not supported:

- `posix_trace_attr_destroy()`
- `posix_trace_attr_getclockres()`
- `posix_trace_attr_getcreatetime()`
- `posix_trace_attr_getgenversion()`
- `posix_trace_attr_getinherited()`
- `posix_trace_attr_getlogfullpolicy()`
- `posix_trace_attr_getlogsize()`
- `posix_trace_attr_getmaxdatasize()`
- `posix_trace_attr_getmaxsystemeventsizesize()`
- `posix_trace_attr_getmaxusereventsizesize()`
- `posix_trace_attr_getname()`
- `posix_trace_attr_getstreamfullpolicy()`
- `posix_trace_attr_getstreamsize()`
- `posix_trace_attr_init()`
- `posix_trace_attr_setinherited()`
- `posix_trace_attr_setlogfullpolicy()`
- `posix_trace_attr_setlogsize()`
- `posix_trace_attr_setmaxdatasize()`
- `posix_trace_attr_setname()`
- `posix_trace_attr_setstreamfullpolicy()`
- `posix_trace_attr_setstreamsize()`
- `posix_trace_clear()`
- `posix_trace_close()`
- `posix_trace_create()`
- `posix_trace_create_withlog()`
- `posix_trace_event()`
- `posix_trace_eventid_equal()`
- `posix_trace_eventid_get_name()`
- `posix_trace_eventid_open()`
- `posix_trace_eventset_add()`
- `posix_trace_eventset_del()`
- `posix_trace_eventset_empty()`

- `posix_trace_eventset_fill()`
- `posix_trace_eventset_ismember()`
- `posix_trace_eventtypelist_getnext_id()`
- `posix_trace_eventtypelist_rewind()`
- `posix_trace_flush()`
- `posix_trace_get_attr()`
- `posix_trace_get_filter()`
- `posix_trace_get_status()`
- `posix_trace_getnext_event()`
- `posix_trace_open()`
- `posix_trace_rewind()`
- `posix_trace_set_filter()`
- `posix_trace_shutdown()`
- `posix_trace_start()`
- `posix_trace_stop()`
- `posix_trace_timedgetnext_event()`
- `posix_trace_trid_eventid_open()`
- `posix_trace_trygetnext_event()`

## 11.45 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getlogin_r()`
- `getopt()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `isatty()`
- `link()`
- `lseek()`
- `optarg`
- `opterr`
- `optind`

- `optopt`
- `pathconf()`
- `pipe()`
- `read()`
- `readlink()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setpgid()`
- `setuid()`
- `sleep()`
- `symlink()`
- `sysconf()`
- `tcgetpgrp()`
- `tcsetpgrp()`
- `ttyname()`
- `ttyname_r()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execlp()`
- `execv()`
- `execve()`
- `execvp()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`

## 11.46 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`

## 11.47 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()



- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

## 11.48 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

## 11.49 <wordexp.h>

The following methods and variables in <wordexp.h> are not supported:

- wordexp()
- wordfree()



## C99 STANDARD LIBRARY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 12.1 Summary

The follow table summarizes alignment with the C99 Standard Library standard:

|               |     |
|---------------|-----|
| Supported     | 488 |
| ENOSYS        | 0   |
| Not supported | 1   |

## 12.2 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 12.3 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`



- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 12.4 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 12.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 12.6 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 12.7 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 12.8 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 12.9 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`



- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 12.10 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

## 12.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- `raise()`
- `signal()`

## 12.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 12.13 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`



## 12.14 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 12.15 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fsetpos()
- ftell()
- fwrite()
- getc()
- getchar()
- gets()
- perror()
- printf()
- putc()
- putchar()
- puts()
- remove()
- rename()
- rewind()
- scanf()

- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 12.16 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mktime()`
- `qsort()`
- `rand()`
- `realloc()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`

- strtoull()
- wcstombs()
- wctomb()

## 12.17 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strxfrm()

## 12.18 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- clock()
- ctime()
- difftime()
- gmtime()
- localtime()
- strftime()
- time()

## 12.19 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()



- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

## 12.20 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

## C11 STANDARD LIBRARY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 13.1 Summary

The follow table summarizes alignment with the C11 Standard Library standard:

|               |     |
|---------------|-----|
| Supported     | 534 |
| ENOSYS        | 0   |
| Not supported | 1   |

## 13.2 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

### 13.3 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 13.4 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 13.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 13.6 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 13.7 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 13.8 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 13.9 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`



- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 13.10 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

## 13.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- raise()
- signal()

## 13.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 13.13 <stdatomic.h>

The following methods and variables in <stdatomic.h> are supported:

- `ATOMIC_VAR_INIT()`
- `atomic_compare_exchange_strong()`
- `atomic_compare_exchange_strong_explicit()`
- `atomic_compare_exchange_weak()`
- `atomic_compare_exchange_weak_explicit()`
- `atomic_exchange()`
- `atomic_exchange_explicit()`
- `atomic_fetch_key()`
- `atomic_fetch_key_explicit()`
- `atomic_flag_clear()`
- `atomic_flag_clear_explicit()`
- `atomic_flag_test_and_set()`
- `atomic_flag_test_and_set_explicit()`
- `atomic_init()`
- `atomic_is_lock_free()`
- `atomic_load()`
- `atomic_load_explicit()`
- `atomic_signal_fence()`
- `atomic_store()`
- `atomic_store_explicit()`
- `atomic_thread_fence()`

## 13.14 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 13.15 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()



## 13.16 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fsetpos()
- ftell()
- fwrite()
- getc()
- getchar()
- perror()
- printf()
- putc()
- putchar()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()

- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 13.17 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `abs()`
- `aligned_alloc()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mktime()`
- `qsort()`
- `quick_exit()`
- `rand()`
- `realloc()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`

- strtoll()
- strtoul()
- strtoull()
- wcstombs()
- wctomb()

## 13.18 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strxfrm()`

## 13.19 <threads.h>

The following methods and variables in <threads.h> are supported:

- `call_once()`
- `cnd_broadcast()`
- `cnd_destroy()`
- `cnd_init()`
- `cnd_signal()`
- `cnd_timedwait()`
- `cnd_wait()`
- `mtx_destroy()`
- `mtx_init()`
- `mtx_lock()`
- `mtx_timedlock()`
- `mtx_trylock()`
- `mtx_unlock()`
- `thrd_create()`
- `thrd_current()`
- `thrd_detach()`
- `thrd_equal()`
- `thrd_exit()`
- `thrd_join()`
- `thrd_sleep()`
- `thrd_yield()`
- `tss_create()`
- `tss_delete()`
- `tss_get()`
- `tss_set()`

## 13.20 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- clock()
- ctime()
- difftime()
- gmtime()
- localtime()
- strftime()
- time()

## 13.21 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()



- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

## 13.22 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

## C17 STANDARD LIBRARY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 14.1 Summary

The follow table summarizes alignment with the C17 Standard Library standard:

|               |     |
|---------------|-----|
| Supported     | 526 |
| ENOSYS        | 1   |
| Not supported | 1   |

## 14.2 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 14.3 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 14.4 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 14.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 14.6 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 14.7 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 14.8 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 14.9 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asin1()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceil1()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabs1()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`



- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 14.10 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

## 14.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- raise()
- signal()

## 14.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 14.13 <stdatomic.h>

The following methods and variables in <stdatomic.h> are supported:

- `ATOMIC_VAR_INIT()`
- `atomic_compare_exchange_strong()`
- `atomic_compare_exchange_strong_explicit()`
- `atomic_compare_exchange_weak()`
- `atomic_compare_exchange_weak_explicit()`
- `atomic_exchange()`
- `atomic_exchange_explicit()`
- `atomic_fetch_key()`
- `atomic_fetch_key_explicit()`
- `atomic_flag_clear()`
- `atomic_flag_clear_explicit()`
- `atomic_flag_test_and_set()`
- `atomic_flag_test_and_set_explicit()`
- `atomic_init()`
- `atomic_is_lock_free()`
- `atomic_load()`
- `atomic_load_explicit()`
- `atomic_signal_fence()`
- `atomic_store()`
- `atomic_store_explicit()`
- `atomic_thread_fence()`

## 14.14 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 14.15 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fsetpos()
- ftell()
- fwrite()
- getc()
- getchar()
- perror()
- printf()
- puts()
- remove()
- rename()
- rewind()
- scanf()
- setbuf()
- setvbuf()
- snprintf()
- sprintf()



- `sscanf()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 14.16 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- \_Exit()
- abort()
- abs()
- aligned\_alloc()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- mktime()
- qsort()
- quick\_exit()
- rand()
- realloc()
- srand()
- strtod()
- strtof()
- strtol()

- strtold()
- strtoll()
- strtoul()
- strtoull()
- wcstombs()
- wctomb()

The following methods in `<stdlib.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `system()`

## 14.17 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strxfrm()

## 14.18 <threads.h>

The following methods and variables in <threads.h> are supported:

- `call_once()`
- `cnd_broadcast()`
- `cnd_destroy()`
- `cnd_init()`
- `cnd_signal()`
- `cnd_timedwait()`
- `cnd_wait()`
- `mtx_destroy()`
- `mtx_init()`
- `mtx_lock()`
- `mtx_timedlock()`
- `mtx_trylock()`
- `mtx_unlock()`
- `thrd_create()`
- `thrd_current()`
- `thrd_detach()`
- `thrd_equal()`
- `thrd_exit()`
- `thrd_join()`
- `thrd_sleep()`
- `thrd_yield()`
- `tss_create()`
- `tss_delete()`
- `tss_get()`
- `tss_set()`

## 14.19 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- clock()
- ctime()
- difftime()
- gmtime()
- localtime()
- strftime()
- time()

## 14.20 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()

- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`



## 14.21 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- iswalnum()
- iswalpha()
- iswblank()
- iswcntrl()
- iswctype()
- iswdigit()
- iswgraph()
- iswlower()
- iswprint()
- iswpunct()
- iswspace()
- iswupper()
- iswxdigit()
- towctrans()
- tolower()
- towupper()
- wctrans()
- wctype()



# FACE TECHNICAL STANDARD, EDITION

## 2.1 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 15.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Security standard:

|               |     |
|---------------|-----|
| Supported     | 162 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 15.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 15.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 15.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 15.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno



## 15.6 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 15.7 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 15.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- `pthread_getcpuclockid()`

## 15.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 15.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 15.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 15.12 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`



## 15.13 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

## 15.14 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mmap()`
- `shm_open()`

## 15.15 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`

## 15.16 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- stat()

## 15.17 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- clock\_settime()
- nanosleep()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 15.18 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- ftruncate()
- pause()

# FACE TECHNICAL STANDARD, EDITION

## 2.1 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 16.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Safety Base standard:

|               |     |
|---------------|-----|
| Supported     | 246 |
| ENOSYS        | 1   |
| Not supported | 0   |



## 16.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 16.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 16.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 16.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 16.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 16.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

## 16.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 16.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()



## 16.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 16.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_equal()
- pthread\_getconcurrency()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_key\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()

- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_getcpuclockid()`

## 16.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 16.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 16.14 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 16.15 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 16.16 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- calloc()
- div()
- labs()
- ldiv()
- malloc()
- mktime()
- rand\_r()
- strtod()
- strtol()
- strtoul()



## 16.17 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 16.18 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `shm_open()`

## 16.19 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 16.20 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 16.21 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 16.22 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 16.23 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `link()`
- `lseek()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`





# FACE TECHNICAL STANDARD, EDITION 2.1 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 17.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Safety Extended standard:

|               |     |
|---------------|-----|
| Supported     | 316 |
| ENOSYS        | 11  |
| Not supported | 9   |

## 17.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 17.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 17.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 17.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 17.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 17.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()



## 17.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 17.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()

## 17.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 17.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()

- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning `-1` and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 17.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 17.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 17.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()



## 17.15 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 17.16 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- `posix_spawn()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`

## 17.17 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va\_arg()
- va\_end()
- va\_start()

## 17.18 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

## 17.19 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `malloc()`
- `mktime()`
- `rand_r()`
- `realloc()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 17.20 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 17.21 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `shm_open()`

## 17.22 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()



## 17.23 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`

## 17.24 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 17.25 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- times()

## 17.26 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 17.27 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- waitpid()

## 17.28 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 17.29 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup2()`
- `environ`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setuid()`
- `sleep()`
- `sysconf()`

- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`



# FACE TECHNICAL STANDARD, EDITION

## 2.1 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 18.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 General Purpose standard:

|               |     |
|---------------|-----|
| Supported     | 773 |
| ENOSYS        | 13  |
| Not supported | 25  |

## 18.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 18.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 18.4 <assert.h>

The following methods and variables in <assert.h> are supported:

- `assert()`

## 18.5 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 18.6 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 18.7 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 18.8 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 18.9 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 18.10 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 18.11 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 18.12 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 18.13 <locale.h>

The following methods and variables in <locale.h> are supported:

- localeconv()
- setlocale()

## 18.14 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`



- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfc1()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 18.15 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 18.16 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 18.17 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`



## 18.18 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_init()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()

- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`

- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 18.19 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

The following methods in <sched.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- sched\_getparam()
- sched\_getscheduler()
- sched\_setparam()
- sched\_setscheduler()

## 18.20 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 18.21 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- `longjmp()`
- `setjmp()`
- `siglongjmp()`
- `sigsetjmp()`

## 18.22 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 18.23 <spawn.h>

The following methods and variables in <spawn.h> are not supported:

- posix\_spawn()
- posix\_spawn\_file\_actions\_addclose()
- posix\_spawn\_file\_actions\_adddup2()
- posix\_spawn\_file\_actions\_addopen()
- posix\_spawn\_file\_actions\_destroy()
- posix\_spawn\_file\_actions\_init()
- posix\_spawnattr\_destroy()
- posix\_spawnattr\_getflags()
- posix\_spawnattr\_getpgroup()
- posix\_spawnattr\_getschedparam()
- posix\_spawnattr\_getschedpolicy()
- posix\_spawnattr\_getsigdefault()
- posix\_spawnattr\_getsigmask()
- posix\_spawnattr\_init()
- posix\_spawnattr\_setflags()
- posix\_spawnattr\_setpgroup()
- posix\_spawnattr\_setschedparam()
- posix\_spawnattr\_setschedpolicy()
- posix\_spawnattr\_setsigdefault()
- posix\_spawnattr\_setsigmask()
- posix\_spawnnp()



## 18.24 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 18.25 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- perror()
- printf()

- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setbuf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 18.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- \_Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbtowc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()

- strtoll()
- strtoul()
- strtoull()
- unsetenv()
- wcstombs()
- wctomb()

## 18.27 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

## 18.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlock()`
- `mlockall()`
- `mmap()`
- `mprotect()`
- `msync()`
- `munlock()`
- `munlockall()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 18.29 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()



## 18.30 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- socketatmark()

## 18.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 18.32 <sys/time.h>

The following methods and variables in <sys/time.h> are supported:

- times()

## 18.33 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 18.34 <sys/wait.h>

The following methods and variables in <sys/wait.h> are supported:

- wait()
- waitpid()

## 18.35 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

## 18.36 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`

- `setegid()`
- `seteuid()`
- `setgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

The following methods in `<unistd.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

The following methods and variables in `<unistd.h>` are not supported:

- `confstr()`



## 18.37 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcsrbrk()
- wcsrchr()
- wcsrtombs()
- wcsspncpy()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcsxfrm()
- wctob()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

## 18.38 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`



# FACE TECHNICAL STANDARD, EDITION 3.0 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 19.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Security standard:

|               |     |
|---------------|-----|
| Supported     | 172 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 19.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 19.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 19.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 19.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 19.6 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 19.7 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 19.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- `pthread_getcpuclockid()`

## 19.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 19.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`



## 19.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 19.12 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 19.13 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 19.14 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 19.15 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

## 19.16 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mmap()`
- `shm_open()`

## 19.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

## 19.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- stat()



## 19.19 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- nanosleep()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 19.20 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- ftruncate()
- pause()

# FACE TECHNICAL STANDARD, EDITION 3.0 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 20.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Safety Base standard:

|               |     |
|---------------|-----|
| Supported     | 254 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 20.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 20.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 20.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 20.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`



## 20.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 20.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

## 20.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 20.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()

## 20.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 20.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_key\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()

- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_getcpuclockid()`

## 20.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()



## 20.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 20.14 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 20.15 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 20.16 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 20.17 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 20.18 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- calloc()
- div()
- labs()
- ldiv()
- malloc()
- mktime()
- rand\_r()
- strtod()
- strtol()
- strtoul()

## 20.19 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 20.20 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `shm_open()`



## 20.21 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 20.22 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

## 20.23 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 20.24 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 20.25 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `link()`
- `lseek()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`



# FACE TECHNICAL STANDARD, EDITION 3.0 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 21.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Safety Extended standard:

|               |     |
|---------------|-----|
| Supported     | 319 |
| ENOSYS        | 2   |
| Not supported | 0   |



## 21.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 21.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 21.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 21.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 21.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 21.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 21.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 21.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()



## 21.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 21.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()

- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 21.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 21.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 21.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()

## 21.15 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 21.16 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`



## 21.17 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 21.18 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 21.19 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

## 21.20 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- malloc()
- mktime()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 21.21 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 21.22 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `mprotect()`
- `shm_open()`

## 21.23 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 21.24 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()



## 21.25 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 21.26 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 21.27 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 21.28 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- chown()
- close()
- dup2()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getuid()
- link()
- lseek()
- pause()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sysconf()
- unlink()
- write()

# FACE TECHNICAL STANDARD, EDITION 3.0 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 22.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 General Purpose standard:

|               |     |
|---------------|-----|
| Supported     | 704 |
| ENOSYS        | 4   |
| Not supported | 3   |

## 22.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 22.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`



## 22.4 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 22.5 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 22.6 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 22.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 22.8 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 22.9 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 22.10 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`



## 22.11 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`
- `wcstoimax()`
- `wcstoumax()`

## 22.12 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`

## 22.13 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtrl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`



## 22.14 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 22.15 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 22.16 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 22.17 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_init()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()

- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`

- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 22.18 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 22.19 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`



## 22.20 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

## 22.21 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 22.22 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va\_arg()
- va\_copy()
- va\_end()
- va\_start()

## 22.23 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 22.24 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 22.25 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- perror()
- printf()

- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 22.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbtowc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()



- `unsetenv()`
- `wcstombs()`
- `wctomb()`

## 22.27 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 22.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mlockall()
- mmap()
- mprotect()
- munmap()
- shm\_open()
- shm\_unlink()

## 22.29 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 22.30 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- socketatmark()

## 22.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 22.32 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 22.33 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()



## 22.34 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin_r()`
- `getuid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setuid()`
- `sysconf()`
- `unlink()`

- `write()`

## 22.35 <wchar.h>

The following methods and variables in <wchar.h> are supported:

- `btowc()`
- `fgetwc()`
- `fgetws()`
- `fputwc()`
- `fputws()`
- `fwide()`
- `fwprintf()`
- `fwscanf()`
- `getwc()`
- `getwchar()`
- `mbrlen()`
- `mbrtowc()`
- `mbsinit()`
- `mbsrtowcs()`

## 22.36 <wctype.h>

The following methods and variables in <wctype.h> are supported:

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`

# FACE TECHNICAL STANDARD, EDITION

## 3.1 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 23.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Security standard:

|               |     |
|---------------|-----|
| Supported     | 174 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 23.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 23.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()



## 23.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 23.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 23.6 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 23.7 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 23.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- `pthread_getcpuclockid()`

## 23.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 23.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`



## 23.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 23.12 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 23.13 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 23.14 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 23.15 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

## 23.16 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `shm_open()`

## 23.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

## 23.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- stat()



## 23.19 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- nanosleep()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 23.20 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- ftruncate()
- gethostname()
- pause()

# FACE TECHNICAL STANDARD, EDITION

## 3.1 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 24.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Safety Base standard:

|               |     |
|---------------|-----|
| Supported     | 254 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 24.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 24.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 24.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 24.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`



## 24.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 24.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

## 24.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 24.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()

## 24.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 24.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_key\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()

- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_getcpuclockid()`

## 24.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()



## 24.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 24.14 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 24.15 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 24.16 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 24.17 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 24.18 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `mktime()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 24.19 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 24.20 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `shm_open()`



## 24.21 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 24.22 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

## 24.23 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 24.24 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 24.25 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `link()`
- `lseek()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`



# FACE TECHNICAL STANDARD, EDITION 3.1 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 25.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Safety Extended standard:

|               |     |
|---------------|-----|
| Supported     | 320 |
| ENOSYS        | 2   |
| Not supported | 0   |



## 25.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 25.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 25.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 25.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 25.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 25.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 25.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 25.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()



## 25.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 25.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()

- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 25.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 25.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 25.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()

## 25.15 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 25.16 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va\_arg()
- va\_copy()
- va\_end()
- va\_start()



## 25.17 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 25.18 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 25.19 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

## 25.20 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- malloc()
- mktime()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 25.21 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 25.22 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `mprotect()`
- `shm_open()`

## 25.23 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 25.24 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getpeername()`
- `getsockname()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `shutdown()`
- `socket()`



## 25.25 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 25.26 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 25.27 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 25.28 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- chown()
- close()
- dup2()
- environ
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getuid()
- link()
- lseek()
- pause()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sysconf()
- unlink()
- write()

# FACE TECHNICAL STANDARD, EDITION

## 3.1 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 26.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 General Purpose standard:

|               |     |
|---------------|-----|
| Supported     | 673 |
| ENOSYS        | 4   |
| Not supported | 5   |

## 26.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

## 26.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`



## 26.4 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`

## 26.5 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 26.6 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 26.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 26.8 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 26.9 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 26.10 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`



## 26.11 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

## 26.12 <locale.h>

The following methods and variables in <locale.h> are supported:

- localeconv()
- setlocale()

## 26.13 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceill()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- cosh()
- coshf()
- coshl()
- cosl()
- erf()
- erfc()
- erfcf()
- erfc1()
- erff()
- erfl()
- exp()
- exp2()
- exp2f()
- exp2l()
- expf()
- expl()
- expm1()
- expm1f()
- expm1l()
- fabs()
- fabsf()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- fmaf()
- fmal()
- fmax()
- fmaxf()
- fmaxl()

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tan1()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`



## 26.14 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 26.15 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()

## 26.16 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 26.17 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_init()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()

- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`

- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 26.18 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 26.19 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`



## 26.20 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

The following methods and variables in <setjmp.h> are not supported:

- \_longjmp()
- \_setjmp()

## 26.21 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 26.22 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 26.23 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 26.24 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 26.25 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- perror()
- printf()

- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 26.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()



## 26.27 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

## 26.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mlockall()`
- `mmap()`
- `mprotect()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 26.29 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 26.30 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- socketatmark()

## 26.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 26.32 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

## 26.33 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

## 26.34 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin\_r()
- getuid()
- link()
- lseek()
- pathconf()
- pause()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sysconf()



- `unlink()`
- `write()`



# FACE TECHNICAL STANDARD, EDITION

## 3.2 SECURITY

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 27.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Security standard:

|               |     |
|---------------|-----|
| Supported     | 172 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 27.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 27.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 27.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 27.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno



## 27.6 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 27.7 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 27.8 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

The following methods in <pthread.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- `pthread_getcpuclockid()`

## 27.9 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 27.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 27.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 27.12 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`



## 27.13 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 27.14 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 27.15 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 27.16 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mmap()`
- `shm_open()`

## 27.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()

## 27.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- stat()

## 27.19 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- nanosleep()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 27.20 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- alarm()
- ftruncate()
- pause()



# FACE TECHNICAL STANDARD, EDITION

## 3.2 SAFETY BASE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 28.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Safety Base standard:

|               |     |
|---------------|-----|
| Supported     | 253 |
| ENOSYS        | 1   |
| Not supported | 0   |

## 28.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 28.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 28.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 28.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 28.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 28.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()



## 28.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 28.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()

## 28.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 28.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_key\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()

- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_getcpuclockid()`

## 28.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_yield()

## 28.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_wait()`

## 28.14 <signal.h>

The following methods and variables in <signal.h> are supported:

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()



## 28.15 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 28.16 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 28.17 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 28.18 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abs()
- atof()
- atoi()
- atol()
- calloc()
- div()
- labs()
- ldiv()
- malloc()
- mktime()
- rand\_r()
- strtod()
- strtol()
- strtoul()

## 28.19 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 28.20 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- mmap()
- shm\_open()

## 28.21 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 28.22 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()



## 28.23 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 28.24 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 28.25 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `alarm()`
- `chdir()`
- `close()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `gethostname()`
- `link()`
- `lseek()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`



# FACE TECHNICAL STANDARD, EDITION 3.2 SAFETY EXTENDED

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 29.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Safety Extended standard:

|               |     |
|---------------|-----|
| Supported     | 319 |
| ENOSYS        | 2   |
| Not supported | 0   |

## 29.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 29.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- `isalnum()`
- `isalpha()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 29.4 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 29.5 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 29.6 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 29.7 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 29.8 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 29.9 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 29.10 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 29.11 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()



- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 29.12 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 29.13 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 29.14 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- siglongjmp()
- sigsetjmp()

## 29.15 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 29.16 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 29.17 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 29.18 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()



## 29.19 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- `clearerr()`
- `fclose()`
- `fdopen()`
- `feof()`
- `ferror()`
- `fflush()`
- `fgetc()`
- `fgets()`
- `fileno()`
- `flockfile()`
- `fopen()`
- `fprintf()`
- `fread()`
- `freopen()`
- `fseek()`
- `fseeko()`
- `ftell()`
- `ftello()`
- `ftrylockfile()`
- `funlockfile()`
- `fwrite()`
- `remove()`
- `rename()`
- `snprintf()`
- `sscanf()`
- `vfprintf()`
- `vsnprintf()`

## 29.20 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- malloc()
- mktime()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 29.21 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

## 29.22 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mmap()`
- `mprotect()`
- `shm_open()`

## 29.23 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

## 29.24 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 29.25 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 29.26 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`



## 29.27 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 29.28 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- chown()
- close()
- dup2()
- environ
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getuid()
- link()
- lseek()
- pause()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sysconf()
- unlink()
- write()

# FACE TECHNICAL STANDARD, EDITION

## 3.2 GENERAL PURPOSE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 30.1 Summary

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 General Purpose standard:

|               |     |
|---------------|-----|
| Supported     | 672 |
| ENOSYS        | 4   |
| Not supported | 5   |

## 30.2 <aio.h>

The following methods and variables in <aio.h> are supported:

- aio\_cancel()
- aio\_error()
- aio\_fsync()
- aio\_read()
- aio\_return()
- aio\_write()
- lio\_listio()

The following methods in <aio.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- aio\_suspend()

### 30.3 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- `htonl()`
- `htons()`
- `inet_addr()`
- `inet_ntoa()`
- `inet_ntop()`
- `inet_pton()`
- `ntohl()`
- `ntohs()`

## 30.4 <complex.h>

The following methods and variables in <complex.h> are supported:

- `cabs()`
- `cabsf()`
- `cabsl()`
- `cacos()`
- `cacosf()`
- `cacosh()`
- `cacoshf()`
- `cacoshl()`
- `cacosl()`
- `carg()`
- `cargf()`
- `cargl()`
- `casin()`
- `casinf()`
- `casinh()`
- `casinhf()`
- `casinhl()`
- `casinl()`
- `catan()`
- `catanf()`
- `catanh()`
- `catanhf()`
- `catanhl()`
- `catanl()`
- `ccos()`
- `ccosf()`
- `ccosh()`
- `ccoshf()`
- `ccoshl()`
- `ccosl()`
- `cexp()`
- `cexpf()`

- `cexpl()`
- `cimag()`
- `cimagf()`
- `cimagl()`
- `clog()`
- `clogf()`
- `clogl()`
- `conj()`
- `conjf()`
- `conjl()`
- `cpow()`
- `cpowf()`
- `cpowl()`
- `cproj()`
- `cprojf()`
- `cprojl()`
- `creal()`
- `crealf()`
- `creall()`
- `csin()`
- `csinf()`
- `csinh()`
- `csinhf()`
- `csinhl()`
- `csinl()`
- `csqrt()`
- `csqrtf()`
- `csqrtl()`
- `ctan()`
- `ctanf()`
- `ctanh()`
- `ctanhf()`
- `ctanhl()`
- `ctanl()`



## 30.5 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 30.6 <devctl.h>

The following methods and variables in <devctl.h> are supported:

- posix\_devctl()

## 30.7 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 30.8 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 30.9 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- fcntl()
- open()

## 30.10 <fenv.h>

The following methods and variables in <fenv.h> are supported:

- `feclearexcept()`
- `fegetenv()`
- `fegetexceptflag()`
- `fegetround()`
- `feholdexcept()`
- `feraiseexcept()`
- `fesetenv()`
- `fesetexceptflag()`
- `fesetround()`
- `fetestexcept()`
- `feupdateenv()`

## 30.11 <inttypes.h>

The following methods and variables in <inttypes.h> are supported:

- `imaxabs()`
- `imaxdiv()`
- `strtoimax()`
- `strtoumax()`

## 30.12 <locale.h>

The following methods and variables in <locale.h> are supported:

- `localeconv()`
- `setlocale()`



## 30.13 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asin1()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `ceil1()`
- `copysign()`
- `copysignf()`
- `copysignl()`
- `cos()`
- `cosf()`

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `erff()`
- `erfl()`
- `exp()`
- `exp2()`
- `exp2f()`
- `exp2l()`
- `expf()`
- `expl()`
- `expm1()`
- `expm1f()`
- `expm1l()`
- `fabs()`
- `fabsf()`
- `fabsl()`
- `fdim()`
- `fdimf()`
- `fdiml()`
- `floor()`
- `floorf()`
- `floorl()`
- `fma()`
- `fmaf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `fpclassify()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isinf()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnan()`
- `isnormal()`
- `isunordered()`
- `ldexp()`
- `ldexpf()`
- `ldexpl()`
- `lgamma()`
- `lgammaf()`
- `lgammal()`
- `llrint()`
- `llrintf()`

- `llrintl()`
- `llround()`
- `llroundf()`
- `llroundl()`
- `log()`
- `log10()`
- `log10f()`
- `log10l()`
- `log1p()`
- `log1pf()`
- `log1pl()`
- `log2()`
- `log2f()`
- `log2l()`
- `logb()`
- `logbf()`
- `logbl()`
- `logf()`
- `logl()`
- `lrint()`
- `lrintf()`
- `lrintl()`
- `lround()`
- `lroundf()`
- `lroundl()`
- `modf()`
- `modff()`
- `modfl()`
- `nan()`
- `nanf()`
- `nanl()`
- `nearbyint()`
- `nearbyintf()`
- `nearbyintl()`

- `nextafter()`
- `nextafterf()`
- `nextafterl()`
- `nexttoward()`
- `nexttowardf()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`
- `remainderf()`
- `remainderl()`
- `remquo()`
- `remquof()`
- `remquol()`
- `rint()`
- `rintf()`
- `rintl()`
- `round()`
- `roundf()`
- `roundl()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`
- `sqrt()`

- `sqrtf()`
- `sqrtl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tan1()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

The following methods and variables in `<math.h>` are not supported:

- `signbit()`

## 30.14 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 30.15 <net/if.h>

The following methods and variables in <net/if.h> are supported:

- if\_freenameindex()
- if\_indextoname()
- if\_nameindex()
- if\_nametoindex()



## 30.16 <netdb.h>

The following methods and variables in <netdb.h> are supported:

- `endhostent()`
- `endnetent()`
- `endprotoent()`
- `endservent()`
- `freeaddrinfo()`
- `gai_strerror()`
- `getaddrinfo()`
- `gethostent()`
- `getnameinfo()`
- `getnetbyaddr()`
- `getnetbyname()`
- `getnetent()`
- `getprotobyname()`
- `getprotobynumber()`
- `getprotoent()`
- `getservbyname()`
- `getservbyport()`
- `getservent()`
- `sethostent()`
- `setnetent()`
- `setprotoent()`
- `setservent()`

## 30.17 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_barrier\_destroy()
- pthread\_barrier\_init()
- pthread\_barrier\_wait()
- pthread\_barrierattr\_destroy()
- pthread\_barrierattr\_init()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()

- `pthread_condattr_destroy()`
- `pthread_condattr_getclock()`
- `pthread_condattr_getpshared()`
- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`

- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

The following methods in `<pthread.h>` are implemented as stubs returning -1 and setting `errno` to `ENOSYS`:

- `pthread_atfork()`
- `pthread_getcpuclockid()`

## 30.18 <sched.h>

The following methods and variables in <sched.h> are supported:

- sched\_get\_priority\_max()
- sched\_get\_priority\_min()
- sched\_rr\_get\_interval()
- sched\_yield()

## 30.19 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 30.20 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()
- siglongjmp()
- sigsetjmp()

The following methods and variables in <setjmp.h> are not supported:

- \_longjmp()
- \_setjmp()

## 30.21 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()



## 30.22 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- `va_arg()`
- `va_copy()`
- `va_end()`
- `va_start()`

## 30.23 <stddef.h>

The following methods and variables in <stddef.h> are supported:

- `offsetof()`

## 30.24 <stdint.h>

The following methods and variables in <stdint.h> are supported:

- INTMAX\_C()
- INTN\_C()
- UINTMAX\_C()
- UINTN\_C()

## 30.25 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- fsetpos()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- getc()
- getc\_unlocked()
- getchar()
- getchar\_unlocked()
- perror()
- printf()

- `putc()`
- `putc_unlocked()`
- `putchar()`
- `putchar_unlocked()`
- `puts()`
- `remove()`
- `rename()`
- `rewind()`
- `scanf()`
- `setvbuf()`
- `snprintf()`
- `sprintf()`
- `sscanf()`
- `stderr`
- `stdin`
- `stdout`
- `tmpfile()`
- `ungetc()`
- `vfprintf()`
- `vfscanf()`
- `vprintf()`
- `vscanf()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`

## 30.26 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mktime()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 30.27 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

## 30.28 <sys/mman.h>

The following methods and variables in <sys/mman.h> are supported:

- `mmap()`
- `mprotect()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`



## 30.29 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- FD\_CLR()
- FD\_ISSET()
- FD\_SET()
- FD\_ZERO()
- select()

The following methods and variables in <sys/select.h> are not supported:

- pselect()

## 30.30 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- listen()
- recv()
- recvfrom()
- recvmsg()
- send()
- sendmsg()
- sendto()
- setsockopt()
- shutdown()
- socket()
- socketpair()

The following methods and variables in <sys/socket.h> are not supported:

- socketatmark()

## 30.31 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- chmod()
- fchmod()
- fstat()
- lstat()
- mkdir()
- mkfifo()
- stat()
- umask()

## 30.32 <sys/utsname.h>

The following methods and variables in <sys/utsname.h> are supported:

- `uname()`

### 30.33 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

The following methods in <time.h> are implemented as stubs returning -1 and setting errno to ENOSYS:

- clock\_getcpuclockid()

## 30.34 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- access()
- alarm()
- chdir()
- chown()
- close()
- dup()
- dup2()
- environ
- fchown()
- fdatasync()
- fpathconf()
- fsync()
- ftruncate()
- getcwd()
- getegid()
- geteuid()
- getgid()
- getgroups()
- gethostname()
- getlogin\_r()
- getuid()
- link()
- lseek()
- pathconf()
- pause()
- read()
- rmdir()
- setegid()
- seteuid()
- setgid()
- setuid()
- sysconf()

- `unlink()`
- `write()`





# SOFTWARE COMMUNICATIONS ARCHITECTURE 2.2.2 AEP

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 31.1 Summary

The follow table summarizes alignment with the Software Communications Architecture 2.2.2 AEP standard:

|               |     |
|---------------|-----|
| Supported     | 243 |
| ENOSYS        | 0   |
| Not supported | 0   |

## 31.2 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

### 31.3 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 31.4 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

## 31.5 <locale.h>

The following methods and variables in <locale.h> are supported:

- `setlocale()`

## 31.6 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `asin()`
- `atan()`
- `atan2()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 31.7 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstackaddr()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstackaddr()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_getpshared()



- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setschedparam()`
- `pthread_setspecific()`
- `pthread_testcancel()`

## 31.8 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`

## 31.9 <setjmp.h>

The following methods and variables in <setjmp.h> are supported:

- longjmp()
- setjmp()

## 31.10 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigsuspend()
- sigwait()

## 31.11 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- getc()
- getchar()
- gets()
- perror()
- printf()
- putc()
- putchar()
- puts()
- remove()
- rename()
- rewind()

- `scanf()`
- `setbuf()`
- `setvbuf()`
- `sprintf()`
- `sscanf()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`

## 31.12 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- malloc()
- qsort()
- rand()
- rand\_r()
- realloc()
- srand()



## 31.13 <string.h>

The following methods and variables in <string.h> are supported:

- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`

## 31.14 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`

## 31.15 <time.h>

The following methods and variables in <time.h> are supported:

- asctime()
- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime()
- ctime\_r()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 31.16 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `chdir()`
- `close()`
- `fpathconf()`
- `getcwd()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`

## 31.17 <utime.h>

The following methods and variables in <utime.h> are supported:

- `utime()`



# SOFTWARE COMMUNICATIONS ARCHITECTURE 4.1 ULTRA LIGHTWEIGHT APPLICATION ENVIRONMENT PROFILE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 32.1 Summary

The follow table summarizes alignment with the Software Communications Architecture 4.1 Ultra Lightweight Application Environment Profile standard:

|               |    |
|---------------|----|
| Supported     | 22 |
| ENOSYS        | 0  |
| Not supported | 0  |



## 32.2 <math.h>

The following methods and variables in <math.h> are supported:

- exp()
- exp2()

### 32.3 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_open()
- mq\_receive()
- mq\_send()

## 32.4 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_settype()
- pthread\_self()

## 32.5 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_init()`
- `sem_post()`
- `sem_wait()`

## 32.6 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- timer\_create()
- timer\_settime()



# SOFTWARE COMMUNICATIONS ARCHITECTURE 4.1 LIGHTWEIGHT APPLICATION ENVIRONMENT PROFILE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 33.1 Summary

The follow table summarizes alignment with the Software Communications Architecture 4.1 Lightweight Application Environment Profile standard:

|               |     |
|---------------|-----|
| Supported     | 110 |
| ENOSYS        | 0   |
| Not supported | 0   |



## 33.2 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

### 33.3 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- open()

## 33.4 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 33.5 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_open()
- mq\_receive()
- mq\_send()

## 33.6 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_wait()
- pthread\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_getpshared()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_settype()
- pthread\_self()

## 33.7 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_getvalue()`
- `sem_init()`
- `sem_post()`
- `sem_wait()`

## 33.8 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- `sscanf()`

## 33.9 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `free()`
- `malloc()`
- `qsort()`
- `rand()`
- `realloc()`
- `srand()`



## 33.10 <string.h>

The following methods and variables in <string.h> are supported:

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strchr()`
- `strcmp()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`

## 33.11 <time.h>

The following methods and variables in <time.h> are supported:

- clock\_getres()
- clock\_gettime()
- gmtime()
- localtime()
- strftime()
- time()
- timer\_create()
- timer\_gettime()
- timer\_settime()

## 33.12 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- close()
- read()
- write()



# SOFTWARE COMMUNICATIONS ARCHITECTURE 4.1 [FULL] APPLIATION ENVIRONMENT PROFILE

This chapter has a subsection per header file to detail the methods provided by RTEMS that are in that header file.

## 34.1 Summary

The follow table summarizes alignment with the Software Communications Architecture 4.1 [Full] Appliation Environment Profile standard:

|               |     |
|---------------|-----|
| Supported     | 255 |
| ENOSYS        | 0   |
| Not supported | 0   |

## 34.2 <arpa/inet.h>

The following methods and variables in <arpa/inet.h> are supported:

- htonl()
- htons()
- ntohl()
- ntohs()

### 34.3 <ctype.h>

The following methods and variables in <ctype.h> are supported:

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()



## 34.4 <dirent.h>

The following methods and variables in <dirent.h> are supported:

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 34.5 <errno.h>

The following methods and variables in <errno.h> are supported:

- errno

## 34.6 <fcntl.h>

The following methods and variables in <fcntl.h> are supported:

- creat()
- open()

## 34.7 <math.h>

The following methods and variables in <math.h> are supported:

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 34.8 <mqqueue.h>

The following methods and variables in <mqqueue.h> are supported:

- mq\_close()
- mq\_getattr()
- mq\_notify()
- mq\_open()
- mq\_receive()
- mq\_send()
- mq\_setattr()
- mq\_unlink()

## 34.9 <pthread.h>

The following methods and variables in <pthread.h> are supported:

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_init()
- pthread\_create()
- pthread\_detach()
- pthread\_equal()
- pthread\_exit()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_join()
- pthread\_key\_create()
- pthread\_key\_delete()
- pthread\_mutex\_destroy()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_getpshared()

- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setschedparam()`
- `pthread_setspecific()`
- `pthread_testcancel()`

## 34.10 <semaphore.h>

The following methods and variables in <semaphore.h> are supported:

- `sem_close()`
- `sem_destroy()`
- `sem_getvalue()`
- `sem_init()`
- `sem_open()`
- `sem_post()`
- `sem_timedwait()`
- `sem_trywait()`
- `sem_unlink()`
- `sem_wait()`



## 34.11 <signal.h>

The following methods and variables in <signal.h> are supported:

- kill()
- pthread\_kill()
- pthread\_sigmask()
- raise()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- signal()
- sigpending()
- sigprocmask()
- sigsuspend()
- sigwait()

## 34.12 <stdarg.h>

The following methods and variables in <stdarg.h> are supported:

- va\_arg()
- va\_copy()
- va\_end()
- va\_start()

### 34.13 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- getc()
- getchar()
- perror()
- printf()
- putc()
- putchar()
- remove()
- rename()
- rewind()
- setbuf()
- setvbuf()

- `snprintf()`
- `sscanf()`
- `ungetc()`
- `vsnprintf()`

## 34.14 <stdlib.h>

The following methods and variables in <stdlib.h> are supported:

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- labs()
- malloc()
- qsort()
- rand()
- rand\_r()
- realloc()
- srand()
- strtod()
- strtol()
- strtoul()

## 34.15 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strchr()
- strcmp()
- strcoll()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 34.16 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- `select()`

## 34.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- accept()
- bind()
- connect()
- getsockopt()
- listen()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- socket()



## 34.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`

## 34.19 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime\_r()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 34.20 <unistd.h>

The following methods and variables in <unistd.h> are supported:

- `access()`
- `chdir()`
- `close()`
- `fpathconf()`
- `getcwd()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`



# GLOSSARY

## **POSIX**

Portable Operating System Interface is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.



# INDEX

## P

POSIX, **1325**