



# RTEMS POSIX 1003.1 Compliance Guide

*Release 7.0-rc1 (24th January 2026)*

© 1988-2026 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	<aio.h> . . . . .	8
3.2	<arpa/inet.h> . . . . .	9
3.3	<assert.h> . . . . .	10
3.4	<complex.h> . . . . .	11
3.5	<ctype.h> . . . . .	13
3.6	<devctl.h> . . . . .	14
3.7	<dirent.h> . . . . .	15
3.8	<dlfcn.h> . . . . .	16
3.9	<errno.h> . . . . .	17
3.10	<fcntl.h> . . . . .	18
3.11	<fenv.h> . . . . .	19
3.12	<fmtmsg.h> . . . . .	20
3.13	<fnmatch.h> . . . . .	21
3.14	<ftw.h> . . . . .	22
3.15	<glob.h> . . . . .	23
3.16	<grp.h> . . . . .	24
3.17	<iconv.h> . . . . .	25
3.18	<inttypes.h> . . . . .	26
3.19	<langinfo.h> . . . . .	27
3.20	<libgen.h> . . . . .	28
3.21	<libintl.h> . . . . .	29
3.22	<locale.h> . . . . .	30
3.23	<math.h> . . . . .	31
3.24	<monetary.h> . . . . .	37
3.25	<mqueue.h> . . . . .	38
3.26	<ndbm.h> . . . . .	39
3.27	<net/if.h> . . . . .	40
3.28	<netdb.h> . . . . .	41
3.29	<nl_types.h> . . . . .	42
3.30	<poll.h> . . . . .	43
3.31	<pthread.h> . . . . .	44
3.32	<pwd.h> . . . . .	48
3.33	<regex.h> . . . . .	49

3.34	<sched.h>	50
3.35	<search.h>	51
3.36	<semaphore.h>	52
3.37	<setjmp.h>	53
3.38	<signal.h>	54
3.39	<spawn.h>	55
3.40	<stdarg.h>	56
3.41	<stdatomic.h>	57
3.42	<stddef.h>	58
3.43	<stdint.h>	59
3.44	<stdio.h>	60
3.45	<stdlib.h>	63
3.46	<string.h>	66
3.47	<strings.h>	68
3.48	<stropts.h>	69
3.49	<sys/ipc.h>	70
3.50	<sys/mman.h>	71
3.51	<sys/msg.h>	72
3.52	<sys/resource.h>	73
3.53	<sys/select.h>	74
3.54	<sys/sem.h>	75
3.55	<sys/shm.h>	76
3.56	<sys/socket.h>	77
3.57	<sys/stat.h>	78
3.58	<sys/statvfs.h>	79
3.59	<sys/time.h>	80
3.60	<sys/uio.h>	81
3.61	<sys/utsname.h>	82
3.62	<sys/wait.h>	83
3.63	<syslog.h>	84
3.64	<termios.h>	85
3.65	<threads.h>	86
3.66	<time.h>	87
3.67	<trace.h>	89
3.68	<ulimit.h>	91
3.69	<unistd.h>	92
3.70	<utime.h>	95
3.71	<utmpx.h>	96
3.72	<wchar.h>	97
3.73	<wctype.h>	100
3.74	<wordexp.h>	102
<b>4</b>	<b>POSIX-2024 (Issue 8)</b>	<b>103</b>
4.1	<aio.h>	104
4.2	<arpa/inet.h>	105
4.3	<assert.h>	106
4.4	<complex.h>	107
4.5	<ctype.h>	109
4.6	<dirent.h>	110
4.7	<dlfcn.h>	111
4.8	<errno.h>	112

4.9	<fcntl.h>	113
4.10	<fenv.h>	114
4.11	<fmtmsg.h>	115
4.12	<fnmatch.h>	116
4.13	<ftw.h>	117
4.14	<glob.h>	118
4.15	<grp.h>	119
4.16	<iconv.h>	120
4.17	<inttypes.h>	121
4.18	<langinfo.h>	122
4.19	<libgen.h>	123
4.20	<libintl.h>	124
4.21	<locale.h>	125
4.22	<math.h>	126
4.23	<monetary.h>	132
4.24	<mqueue.h>	133
4.25	<ndbm.h>	134
4.26	<net/if.h>	135
4.27	<netdb.h>	136
4.28	<nl_types.h>	137
4.29	<poll.h>	138
4.30	<pthread.h>	139
4.31	<pwd.h>	142
4.32	<regex.h>	143
4.33	<sched.h>	144
4.34	<search.h>	145
4.35	<semaphore.h>	146
4.36	<setjmp.h>	147
4.37	<signal.h>	148
4.38	<spawn.h>	149
4.39	<stdarg.h>	150
4.40	<stdatomic.h>	151
4.41	<stddef.h>	152
4.42	<stdint.h>	153
4.43	<stdio.h>	154
4.44	<stdlib.h>	157
4.45	<string.h>	160
4.46	<strings.h>	162
4.47	<sys/ipc.h>	163
4.48	<sys/mman.h>	164
4.49	<sys/msg.h>	165
4.50	<sys/resource.h>	166
4.51	<sys/select.h>	167
4.52	<sys/sem.h>	168
4.53	<sys/shm.h>	169
4.54	<sys/socket.h>	170
4.55	<sys/stat.h>	171
4.56	<sys/statvfs.h>	172
4.57	<sys/time.h>	173
4.58	<sys/uio.h>	174
4.59	<sys/utsname.h>	175

4.60	<sys/wait.h>	176
4.61	<syslog.h>	177
4.62	<termios.h>	178
4.63	<threads.h>	179
4.64	<time.h>	180
4.65	<unistd.h>	182
4.66	<utmpx.h>	185
4.67	<wchar.h>	186
4.68	<wctype.h>	189
4.69	<wordexp.h>	191
<b>5</b>	<b>POSIX-2017 (Issue 7 TC2)</b>	<b>193</b>
5.1	<aio.h>	194
5.2	<arpa/inet.h>	195
5.3	<assert.h>	196
5.4	<complex.h>	197
5.5	<ctype.h>	199
5.6	<dirent.h>	200
5.7	<dlfcn.h>	201
5.8	<errno.h>	202
5.9	<fcntl.h>	203
5.10	<fenv.h>	204
5.11	<fmtmsg.h>	205
5.12	<fnmatch.h>	206
5.13	<ftw.h>	207
5.14	<glob.h>	208
5.15	<grp.h>	209
5.16	<iconv.h>	210
5.17	<inttypes.h>	211
5.18	<langinfo.h>	212
5.19	<libgen.h>	213
5.20	<locale.h>	214
5.21	<math.h>	215
5.22	<monetary.h>	221
5.23	<mqueue.h>	222
5.24	<ndbm.h>	223
5.25	<net/if.h>	224
5.26	<netdb.h>	225
5.27	<nl_types.h>	226
5.28	<poll.h>	227
5.29	<pthread.h>	228
5.30	<pwd.h>	232
5.31	<regex.h>	233
5.32	<sched.h>	234
5.33	<search.h>	235
5.34	<semaphore.h>	236
5.35	<setjmp.h>	237
5.36	<signal.h>	238
5.37	<spawn.h>	239
5.38	<stdarg.h>	240
5.39	<stddef.h>	241

5.40	<stdint.h>	242
5.41	<stdio.h>	243
5.42	<stdlib.h>	246
5.43	<string.h>	248
5.44	<strings.h>	250
5.45	<stropts.h>	251
5.46	<sys/ipc.h>	252
5.47	<sys/mman.h>	253
5.48	<sys/msg.h>	254
5.49	<sys/resource.h>	255
5.50	<sys/select.h>	256
5.51	<sys/sem.h>	257
5.52	<sys/shm.h>	258
5.53	<sys/socket.h>	259
5.54	<sys/stat.h>	260
5.55	<sys/statvfs.h>	261
5.56	<sys/time.h>	262
5.57	<sys/uio.h>	263
5.58	<sys/utsname.h>	264
5.59	<sys/wait.h>	265
5.60	<syslog.h>	266
5.61	<termios.h>	267
5.62	<time.h>	268
5.63	<trace.h>	270
5.64	<ulimit.h>	272
5.65	<unistd.h>	273
5.66	<utime.h>	276
5.67	<utmpx.h>	277
5.68	<wchar.h>	278
5.69	<wctype.h>	281
5.70	<wordexp.h>	283
<b>6</b>	<b>POSIX-2008 (Issue 7)</b>	<b>285</b>
6.1	<aio.h>	286
6.2	<arpa/inet.h>	287
6.3	<assert.h>	288
6.4	<complex.h>	289
6.5	<ctype.h>	291
6.6	<dirent.h>	292
6.7	<dlfcn.h>	293
6.8	<errno.h>	294
6.9	<fcntl.h>	295
6.10	<fenv.h>	296
6.11	<fmtmsg.h>	297
6.12	<fnmatch.h>	298
6.13	<ftw.h>	299
6.14	<glob.h>	300
6.15	<grp.h>	301
6.16	<iconv.h>	302
6.17	<inttypes.h>	303
6.18	<langinfo.h>	304

6.19	<libgen.h>	305
6.20	<locale.h>	306
6.21	<math.h>	307
6.22	<monetary.h>	313
6.23	<mqueue.h>	314
6.24	<ndbm.h>	315
6.25	<net/if.h>	316
6.26	<netdb.h>	317
6.27	<nl_types.h>	318
6.28	<poll.h>	319
6.29	<pthread.h>	320
6.30	<pwd.h>	324
6.31	<regex.h>	325
6.32	<sched.h>	326
6.33	<search.h>	327
6.34	<semaphore.h>	328
6.35	<setjmp.h>	329
6.36	<signal.h>	330
6.37	<spawn.h>	331
6.38	<stdarg.h>	332
6.39	<stddef.h>	333
6.40	<stdint.h>	334
6.41	<stdio.h>	335
6.42	<stdlib.h>	338
6.43	<string.h>	340
6.44	<strings.h>	342
6.45	<stropts.h>	343
6.46	<sys/ipc.h>	344
6.47	<sys/mman.h>	345
6.48	<sys/msg.h>	346
6.49	<sys/resource.h>	347
6.50	<sys/select.h>	348
6.51	<sys/sem.h>	349
6.52	<sys/shm.h>	350
6.53	<sys/socket.h>	351
6.54	<sys/stat.h>	352
6.55	<sys/statvfs.h>	353
6.56	<sys/time.h>	354
6.57	<sys/uio.h>	355
6.58	<sys/utsname.h>	356
6.59	<sys/wait.h>	357
6.60	<syslog.h>	358
6.61	<termios.h>	359
6.62	<time.h>	360
6.63	<trace.h>	362
6.64	<ulimit.h>	364
6.65	<unistd.h>	365
6.66	<utime.h>	368
6.67	<utmpx.h>	369
6.68	<wchar.h>	370
6.69	<wctype.h>	373



6.70	<wordexp.h>	375
<b>7</b>	<b>POSIX-2003 (Issue 6)</b>	<b>377</b>
7.1	<aio.h>	378
7.2	<arpa/inet.h>	379
7.3	<assert.h>	380
7.4	<complex.h>	381
7.5	<ctype.h>	383
7.6	<dirent.h>	384
7.7	<dlfcn.h>	385
7.8	<errno.h>	386
7.9	<fcntl.h>	387
7.10	<fenv.h>	388
7.11	<fmtmsg.h>	389
7.12	<fnmatch.h>	390
7.13	<ftw.h>	391
7.14	<glob.h>	392
7.15	<grp.h>	393
7.16	<iconv.h>	394
7.17	<inttypes.h>	395
7.18	<langinfo.h>	396
7.19	<libgen.h>	397
7.20	<locale.h>	398
7.21	<math.h>	399
7.22	<monetary.h>	405
7.23	<mqueue.h>	406
7.24	<ndbm.h>	407
7.25	<net/if.h>	408
7.26	<netdb.h>	409
7.27	<nl_types.h>	410
7.28	<poll.h>	411
7.29	<pthread.h>	412
7.30	<pwd.h>	415
7.31	<regex.h>	416
7.32	<sched.h>	417
7.33	<search.h>	418
7.34	<semaphore.h>	419
7.35	<setjmp.h>	420
7.36	<signal.h>	421
7.37	<spawn.h>	422
7.38	<stdarg.h>	423
7.39	<stddef.h>	424
7.40	<stdint.h>	425
7.41	<stdio.h>	426
7.42	<stdlib.h>	428
7.43	<string.h>	431
7.44	<strings.h>	432
7.45	<stropts.h>	433
7.46	<sys/ipc.h>	434
7.47	<sys/mman.h>	435
7.48	<sys/msg.h>	436

7.49	<sys/resource.h>	437
7.50	<sys/select.h>	438
7.51	<sys/sem.h>	439
7.52	<sys/shm.h>	440
7.53	<sys/socket.h>	441
7.54	<sys/stat.h>	442
7.55	<sys/statvfs.h>	443
7.56	<sys/time.h>	444
7.57	<sys/uio.h>	445
7.58	<sys/utsname.h>	446
7.59	<sys/wait.h>	447
7.60	<syslog.h>	448
7.61	<termios.h>	449
7.62	<time.h>	450
7.63	<trace.h>	452
7.64	<ulimit.h>	454
7.65	<unistd.h>	455
7.66	<utime.h>	458
7.67	<utmpx.h>	459
7.68	<wchar.h>	460
7.69	<wctype.h>	462
7.70	<wordexp.h>	463
<b>8</b>	<b>POSIX PSE51 - Minimal</b>	<b>465</b>
8.1	<ctype.h>	466
8.2	<errno.h>	467
8.3	<fcntl.h>	468
8.4	<fenv.h>	469
8.5	<inttypes.h>	470
8.6	<locale.h>	471
8.7	<pthread.h>	472
8.8	<sched.h>	475
8.9	<semaphore.h>	476
8.10	<setjmp.h>	477
8.11	<signal.h>	478
8.12	<stdarg.h>	479
8.13	<stddef.h>	480
8.14	<stdio.h>	481
8.15	<stdlib.h>	483
8.16	<string.h>	484
8.17	<sys/mman.h>	485
8.18	<sys/utsname.h>	486
8.19	<time.h>	487
8.20	<unistd.h>	488
<b>9</b>	<b>POSIX PSE52 - Real-Time Controller</b>	<b>489</b>
9.1	<complex.h>	490
9.2	<ctype.h>	492
9.3	<dirent.h>	493
9.4	<errno.h>	494
9.5	<fcntl.h>	495
9.6	<fenv.h>	496

9.7	<inttypes.h>	497
9.8	<locale.h>	498
9.9	<math.h>	499
9.10	<mqueue.h>	505
9.11	<pthread.h>	506
9.12	<sched.h>	509
9.13	<semaphore.h>	510
9.14	<setjmp.h>	511
9.15	<signal.h>	512
9.16	<stdarg.h>	513
9.17	<stddef.h>	514
9.18	<stdio.h>	515
9.19	<stdlib.h>	517
9.20	<string.h>	518
9.21	<sys/mman.h>	519
9.22	<sys/stat.h>	520
9.23	<sys/utsname.h>	521
9.24	<time.h>	522
9.25	<trace.h>	523
9.26	<unistd.h>	525
9.27	<utime.h>	526
<b>10</b>	<b>POSIX PSE53 - Dedicated</b>	<b>527</b>
10.1	<aio.h>	528
10.2	<arpa/inet.h>	529
10.3	<assert.h>	530
10.4	<complex.h>	531
10.5	<ctype.h>	533
10.6	<dirent.h>	534
10.7	<errno.h>	535
10.8	<fcntl.h>	536
10.9	<fenv.h>	537
10.10	<inttypes.h>	538
10.11	<locale.h>	539
10.12	<math.h>	540
10.13	<mqueue.h>	546
10.14	<net/if.h>	547
10.15	<netdb.h>	548
10.16	<pthread.h>	549
10.17	<sched.h>	552
10.18	<semaphore.h>	553
10.19	<setjmp.h>	554
10.20	<signal.h>	555
10.21	<spawn.h>	556
10.22	<stdarg.h>	557
10.23	<stddef.h>	558
10.24	<stdio.h>	559
10.25	<stdlib.h>	561
10.26	<string.h>	563
10.27	<sys/mman.h>	564
10.28	<sys/select.h>	565

10.29	<sys/socket.h>	566
10.30	<sys/stat.h>	567
10.31	<sys/time.h>	568
10.32	<sys/utsname.h>	569
10.33	<sys/wait.h>	570
10.34	<time.h>	571
10.35	<trace.h>	572
10.36	<unistd.h>	574
10.37	<utime.h>	576
<b>11</b>	<b>POSIX PSE54 - Multipurpose</b>	<b>577</b>
11.1	<aio.h>	578
11.2	<arpa/inet.h>	579
11.3	<assert.h>	580
11.4	<complex.h>	581
11.5	<ctype.h>	583
11.6	<dirent.h>	584
11.7	<dlfcn.h>	585
11.8	<errno.h>	586
11.9	<fcntl.h>	587
11.10	<fenv.h>	588
11.11	<fnmatch.h>	589
11.12	<glob.h>	590
11.13	<grp.h>	591
11.14	<inttypes.h>	592
11.15	<locale.h>	593
11.16	<math.h>	594
11.17	<mqueue.h>	600
11.18	<net/if.h>	601
11.19	<netdb.h>	602
11.20	<pthread.h>	603
11.21	<pwd.h>	606
11.22	<regex.h>	607
11.23	<sched.h>	608
11.24	<semaphore.h>	609
11.25	<setjmp.h>	610
11.26	<signal.h>	611
11.27	<spawn.h>	612
11.28	<stdarg.h>	613
11.29	<stddef.h>	614
11.30	<stdio.h>	615
11.31	<stdlib.h>	617
11.32	<string.h>	619
11.33	<sys/mman.h>	620
11.34	<sys/select.h>	621
11.35	<sys/socket.h>	622
11.36	<sys/stat.h>	623
11.37	<sys/time.h>	624
11.38	<sys/utsname.h>	625
11.39	<sys/wait.h>	626
11.40	<syslog.h>	627

11.41	<termios.h>	628
11.42	<time.h>	629
11.43	<trace.h>	630
11.44	<unistd.h>	632
11.45	<utime.h>	634
11.46	<wchar.h>	635
11.47	<wctype.h>	637
11.48	<wordexp.h>	638
<b>12</b>	<b>C99 Standard Library</b>	<b>639</b>
12.1	<assert.h>	640
12.2	<complex.h>	641
12.3	<ctype.h>	643
12.4	<errno.h>	644
12.5	<fenv.h>	645
12.6	<inttypes.h>	646
12.7	<locale.h>	647
12.8	<math.h>	648
12.9	<setjmp.h>	654
12.10	<signal.h>	655
12.11	<stdarg.h>	656
12.12	<stddef.h>	657
12.13	<stdint.h>	658
12.14	<stdio.h>	659
12.15	<stdlib.h>	661
12.16	<string.h>	663
12.17	<time.h>	664
12.18	<wchar.h>	665
12.19	<wctype.h>	667
<b>13</b>	<b>C11 Standard Library</b>	<b>669</b>
13.1	<assert.h>	670
13.2	<complex.h>	671
13.3	<ctype.h>	673
13.4	<errno.h>	674
13.5	<fenv.h>	675
13.6	<inttypes.h>	676
13.7	<locale.h>	677
13.8	<math.h>	678
13.9	<setjmp.h>	684
13.10	<signal.h>	685
13.11	<stdarg.h>	686
13.12	<stdatomic.h>	687
13.13	<stddef.h>	688
13.14	<stdint.h>	689
13.15	<stdio.h>	690
13.16	<stdlib.h>	692
13.17	<string.h>	694
13.18	<threads.h>	695
13.19	<time.h>	696
13.20	<wchar.h>	697
13.21	<wctype.h>	699

<b>14 C17 Standard Library</b>	<b>701</b>
14.1 <assert.h> . . . . .	702
14.2 <complex.h> . . . . .	703
14.3 <ctype.h> . . . . .	705
14.4 <errno.h> . . . . .	706
14.5 <fenv.h> . . . . .	707
14.6 <inttypes.h> . . . . .	708
14.7 <locale.h> . . . . .	709
14.8 <math.h> . . . . .	710
14.9 <setjmp.h> . . . . .	716
14.10<signal.h> . . . . .	717
14.11<stdarg.h> . . . . .	718
14.12<stdatomic.h> . . . . .	719
14.13<stddef.h> . . . . .	720
14.14<stdint.h> . . . . .	721
14.15<stdio.h> . . . . .	722
14.16<stdlib.h> . . . . .	724
14.17<string.h> . . . . .	726
14.18<threads.h> . . . . .	727
14.19<time.h> . . . . .	728
14.20<wchar.h> . . . . .	729
14.21<wctype.h> . . . . .	731
 <b>15 FACE Technical Standard, Edition 2.1 Security</b>	 <b>733</b>
15.1 <arpa/inet.h> . . . . .	734
15.2 <ctype.h> . . . . .	735
15.3 <devctl.h> . . . . .	736
15.4 <errno.h> . . . . .	737
15.5 <math.h> . . . . .	738
15.6 <netdb.h> . . . . .	739
15.7 <pthread.h> . . . . .	740
15.8 <sched.h> . . . . .	742
15.9 <semaphore.h> . . . . .	743
15.10<signal.h> . . . . .	744
15.11<stdlib.h> . . . . .	745
15.12<string.h> . . . . .	746
15.13<sys/mman.h> . . . . .	747
15.14<sys/socket.h> . . . . .	748
15.15<sys/stat.h> . . . . .	749
15.16<time.h> . . . . .	750
15.17<unistd.h> . . . . .	751
 <b>16 FACE Technical Standard, Edition 2.1 Safety Base</b>	 <b>753</b>
16.1 <arpa/inet.h> . . . . .	754
16.2 <ctype.h> . . . . .	755
16.3 <devctl.h> . . . . .	756
16.4 <dirent.h> . . . . .	757
16.5 <errno.h> . . . . .	758
16.6 <fcntl.h> . . . . .	759
16.7 <math.h> . . . . .	760
16.8 <mqueue.h> . . . . .	761
16.9 <netdb.h> . . . . .	762

16.10	<pthread.h>	763
16.11	<sched.h>	765
16.12	<semaphore.h>	766
16.13	<signal.h>	767
16.14	<stdio.h>	768
16.15	<stdlib.h>	769
16.16	<string.h>	770
16.17	<sys/mman.h>	771
16.18	<sys/select.h>	772
16.19	<sys/socket.h>	773
16.20	<sys/stat.h>	774
16.21	<time.h>	775
16.22	<unistd.h>	776
<b>17</b>	<b>FACE Technical Standard, Edition 2.1 Safety Extended</b>	<b>777</b>
17.1	<arpa/inet.h>	778
17.2	<ctype.h>	779
17.3	<devctl.h>	780
17.4	<dirent.h>	781
17.5	<errno.h>	782
17.6	<fcntl.h>	783
17.7	<math.h>	784
17.8	<mqueue.h>	785
17.9	<netdb.h>	786
17.10	<pthread.h>	787
17.11	<sched.h>	789
17.12	<semaphore.h>	790
17.13	<setjmp.h>	791
17.14	<signal.h>	792
17.15	<spawn.h>	793
17.16	<stdarg.h>	794
17.17	<stdio.h>	795
17.18	<stdlib.h>	796
17.19	<string.h>	797
17.20	<sys/mman.h>	798
17.21	<sys/select.h>	799
17.22	<sys/socket.h>	800
17.23	<sys/stat.h>	801
17.24	<sys/time.h>	802
17.25	<sys/utsname.h>	803
17.26	<sys/wait.h>	804
17.27	<time.h>	805
17.28	<unistd.h>	806
<b>18</b>	<b>FACE Technical Standard, Edition 2.1 General Purpose</b>	<b>809</b>
18.1	<aio.h>	810
18.2	<arpa/inet.h>	811
18.3	<assert.h>	812
18.4	<complex.h>	813
18.5	<ctype.h>	815
18.6	<devctl.h>	816
18.7	<dirent.h>	817

18.8	<errno.h>	818
18.9	<fcntl.h>	819
18.10	<fcntl.h>	820
18.11	<inttypes.h>	821
18.12	<locale.h>	822
18.13	<math.h>	823
18.14	<mqueue.h>	829
18.15	<net/if.h>	830
18.16	<netdb.h>	831
18.17	<pthread.h>	832
18.18	<sched.h>	835
18.19	<semaphore.h>	836
18.20	<setjmp.h>	837
18.21	<signal.h>	838
18.22	<spawn.h>	839
18.23	<stdarg.h>	840
18.24	<stdio.h>	841
18.25	<stdlib.h>	843
18.26	<string.h>	845
18.27	<sys/mman.h>	846
18.28	<sys/select.h>	847
18.29	<sys/socket.h>	848
18.30	<sys/stat.h>	849
18.31	<sys/time.h>	850
18.32	<sys/utsname.h>	851
18.33	<sys/wait.h>	852
18.34	<time.h>	853
18.35	<unistd.h>	854
18.36	<wchar.h>	856
18.37	<wctype.h>	858
<b>19</b>	<b>FACE Technical Standard, Edition 3.0 Security</b>	<b>859</b>
19.1	<arpa/inet.h>	860
19.2	<ctype.h>	861
19.3	<devctl.h>	862
19.4	<errno.h>	863
19.5	<math.h>	864
19.6	<netdb.h>	865
19.7	<pthread.h>	866
19.8	<sched.h>	868
19.9	<semaphore.h>	869
19.10	<signal.h>	870
19.11	<stddef.h>	871
19.12	<stdint.h>	872
19.13	<stdlib.h>	873
19.14	<string.h>	874
19.15	<sys/mman.h>	875
19.16	<sys/socket.h>	876
19.17	<sys/stat.h>	877
19.18	<time.h>	878
19.19	<unistd.h>	879



<b>20 FACE Technical Standard, Edition 3.0 Safety Base</b>	<b>881</b>
20.1 <arpa/inet.h> . . . . .	882
20.2 <ctype.h> . . . . .	883
20.3 <devctl.h> . . . . .	884
20.4 <dirent.h> . . . . .	885
20.5 <errno.h> . . . . .	886
20.6 <fcntl.h> . . . . .	887
20.7 <math.h> . . . . .	888
20.8 <mqueue.h> . . . . .	889
20.9 <netdb.h> . . . . .	890
20.10<pthread.h> . . . . .	891
20.11<sched.h> . . . . .	893
20.12<semaphore.h> . . . . .	894
20.13<signal.h> . . . . .	895
20.14<stddef.h> . . . . .	896
20.15<stdint.h> . . . . .	897
20.16<stdio.h> . . . . .	898
20.17<stdlib.h> . . . . .	899
20.18<string.h> . . . . .	900
20.19<sys/mman.h> . . . . .	901
20.20<sys/select.h> . . . . .	902
20.21<sys/socket.h> . . . . .	903
20.22<sys/stat.h> . . . . .	904
20.23<time.h> . . . . .	905
20.24<unistd.h> . . . . .	906
 <b>21 FACE Technical Standard, Edition 3.0 Safety Extended</b>	 <b>907</b>
21.1 <arpa/inet.h> . . . . .	908
21.2 <ctype.h> . . . . .	909
21.3 <devctl.h> . . . . .	910
21.4 <dirent.h> . . . . .	911
21.5 <errno.h> . . . . .	912
21.6 <fcntl.h> . . . . .	913
21.7 <math.h> . . . . .	914
21.8 <mqueue.h> . . . . .	915
21.9 <netdb.h> . . . . .	916
21.10<pthread.h> . . . . .	917
21.11<sched.h> . . . . .	919
21.12<semaphore.h> . . . . .	920
21.13<setjmp.h> . . . . .	921
21.14<signal.h> . . . . .	922
21.15<stdarg.h> . . . . .	923
21.16<stddef.h> . . . . .	924
21.17<stdint.h> . . . . .	925
21.18<stdio.h> . . . . .	926
21.19<stdlib.h> . . . . .	927
21.20<string.h> . . . . .	928
21.21<sys/mman.h> . . . . .	929
21.22<sys/select.h> . . . . .	930
21.23<sys/socket.h> . . . . .	931
21.24<sys/stat.h> . . . . .	932

21.25	<sys/utsname.h>	933
21.26	<time.h>	934
21.27	<unistd.h>	935
<b>22</b>	<b>FACE Technical Standard, Edition 3.0 General Purpose</b>	<b>937</b>
22.1	<aio.h>	938
22.2	<arpa/inet.h>	939
22.3	<complex.h>	940
22.4	<ctype.h>	942
22.5	<devctl.h>	943
22.6	<dirent.h>	944
22.7	<errno.h>	945
22.8	<fcntl.h>	946
22.9	<fenv.h>	947
22.10	<inttypes.h>	948
22.11	<locale.h>	949
22.12	<math.h>	950
22.13	<mqueue.h>	956
22.14	<net/if.h>	957
22.15	<netdb.h>	958
22.16	<pthread.h>	959
22.17	<sched.h>	962
22.18	<semaphore.h>	963
22.19	<setjmp.h>	964
22.20	<signal.h>	965
22.21	<stdarg.h>	966
22.22	<stddef.h>	967
22.23	<stdint.h>	968
22.24	<stdio.h>	969
22.25	<stdlib.h>	971
22.26	<string.h>	973
22.27	<sys/mman.h>	974
22.28	<sys/select.h>	975
22.29	<sys/socket.h>	976
22.30	<sys/stat.h>	977
22.31	<sys/utsname.h>	978
22.32	<time.h>	979
22.33	<unistd.h>	980
22.34	<wchar.h>	982
22.35	<wctype.h>	983
<b>23</b>	<b>FACE Technical Standard, Edition 3.1 Security</b>	<b>985</b>
23.1	<arpa/inet.h>	986
23.2	<ctype.h>	987
23.3	<devctl.h>	988
23.4	<errno.h>	989
23.5	<math.h>	990
23.6	<netdb.h>	991
23.7	<pthread.h>	992
23.8	<sched.h>	994
23.9	<semaphore.h>	995
23.10	<signal.h>	996

23.11	<stddef.h>	997
23.12	<stdint.h>	998
23.13	<stdlib.h>	999
23.14	<string.h>	1000
23.15	<sys/mman.h>	1001
23.16	<sys/socket.h>	1002
23.17	<sys/stat.h>	1003
23.18	<time.h>	1004
23.19	<unistd.h>	1005
<b>24</b>	<b>FACE Technical Standard, Edition 3.1 Safety Base</b>	<b>1007</b>
24.1	<arpa/inet.h>	1008
24.2	<ctype.h>	1009
24.3	<devctl.h>	1010
24.4	<dirent.h>	1011
24.5	<errno.h>	1012
24.6	<fcntl.h>	1013
24.7	<math.h>	1014
24.8	<mqueue.h>	1015
24.9	<netdb.h>	1016
24.10	<pthread.h>	1017
24.11	<sched.h>	1019
24.12	<semaphore.h>	1020
24.13	<signal.h>	1021
24.14	<stddef.h>	1022
24.15	<stdint.h>	1023
24.16	<stdio.h>	1024
24.17	<stdlib.h>	1025
24.18	<string.h>	1026
24.19	<sys/mman.h>	1027
24.20	<sys/select.h>	1028
24.21	<sys/socket.h>	1029
24.22	<sys/stat.h>	1030
24.23	<time.h>	1031
24.24	<unistd.h>	1032
<b>25</b>	<b>FACE Technical Standard, Edition 3.1 Safety Extended</b>	<b>1033</b>
25.1	<arpa/inet.h>	1034
25.2	<ctype.h>	1035
25.3	<devctl.h>	1036
25.4	<dirent.h>	1037
25.5	<errno.h>	1038
25.6	<fcntl.h>	1039
25.7	<math.h>	1040
25.8	<mqueue.h>	1041
25.9	<netdb.h>	1042
25.10	<pthread.h>	1043
25.11	<sched.h>	1045
25.12	<semaphore.h>	1046
25.13	<setjmp.h>	1047
25.14	<signal.h>	1048
25.15	<stdarg.h>	1049

25.16	<stddef.h>	1050
25.17	<stdint.h>	1051
25.18	<stdio.h>	1052
25.19	<stdlib.h>	1053
25.20	<string.h>	1054
25.21	<sys/mman.h>	1055
25.22	<sys/select.h>	1056
25.23	<sys/socket.h>	1057
25.24	<sys/stat.h>	1058
25.25	<sys/utsname.h>	1059
25.26	<time.h>	1060
25.27	<unistd.h>	1061
<b>26</b>	<b>FACE Technical Standard, Edition 3.1 General Purpose</b>	<b>1063</b>
26.1	<aio.h>	1064
26.2	<arpa/inet.h>	1065
26.3	<complex.h>	1066
26.4	<ctype.h>	1068
26.5	<devctl.h>	1069
26.6	<dirent.h>	1070
26.7	<errno.h>	1071
26.8	<fcntl.h>	1072
26.9	<fenv.h>	1073
26.10	<inttypes.h>	1074
26.11	<locale.h>	1075
26.12	<math.h>	1076
26.13	<mqueue.h>	1082
26.14	<net/if.h>	1083
26.15	<netdb.h>	1084
26.16	<pthread.h>	1085
26.17	<sched.h>	1088
26.18	<semaphore.h>	1089
26.19	<setjmp.h>	1090
26.20	<signal.h>	1091
26.21	<stdarg.h>	1092
26.22	<stddef.h>	1093
26.23	<stdint.h>	1094
26.24	<stdio.h>	1095
26.25	<stdlib.h>	1097
26.26	<string.h>	1098
26.27	<sys/mman.h>	1099
26.28	<sys/select.h>	1100
26.29	<sys/socket.h>	1101
26.30	<sys/stat.h>	1102
26.31	<sys/utsname.h>	1103
26.32	<time.h>	1104
26.33	<unistd.h>	1105
<b>27</b>	<b>FACE Technical Standard, Edition 3.2 Security</b>	<b>1107</b>
27.1	<arpa/inet.h>	1108
27.2	<ctype.h>	1109
27.3	<devctl.h>	1110

27.4	<errno.h>	1111
27.5	<math.h>	1112
27.6	<netdb.h>	1113
27.7	<pthread.h>	1114
27.8	<sched.h>	1116
27.9	<semaphore.h>	1117
27.10	<signal.h>	1118
27.11	<stddef.h>	1119
27.12	<stdint.h>	1120
27.13	<stdlib.h>	1121
27.14	<string.h>	1122
27.15	<sys/mman.h>	1123
27.16	<sys/socket.h>	1124
27.17	<sys/stat.h>	1125
27.18	<time.h>	1126
27.19	<unistd.h>	1127
<b>28</b>	<b>FACE Technical Standard, Edition 3.2 Safety Base</b>	<b>1129</b>
28.1	<arpa/inet.h>	1130
28.2	<ctype.h>	1131
28.3	<devctl.h>	1132
28.4	<dirent.h>	1133
28.5	<errno.h>	1134
28.6	<fcntl.h>	1135
28.7	<math.h>	1136
28.8	<mqueue.h>	1137
28.9	<netdb.h>	1138
28.10	<pthread.h>	1139
28.11	<sched.h>	1141
28.12	<semaphore.h>	1142
28.13	<signal.h>	1143
28.14	<stddef.h>	1144
28.15	<stdint.h>	1145
28.16	<stdio.h>	1146
28.17	<stdlib.h>	1147
28.18	<string.h>	1148
28.19	<sys/mman.h>	1149
28.20	<sys/select.h>	1150
28.21	<sys/socket.h>	1151
28.22	<sys/stat.h>	1152
28.23	<time.h>	1153
28.24	<unistd.h>	1154
<b>29</b>	<b>FACE Technical Standard, Edition 3.2 Safety Extended</b>	<b>1155</b>
29.1	<arpa/inet.h>	1156
29.2	<ctype.h>	1157
29.3	<devctl.h>	1158
29.4	<dirent.h>	1159
29.5	<errno.h>	1160
29.6	<fcntl.h>	1161
29.7	<math.h>	1162
29.8	<mqueue.h>	1163

29.9	<netdb.h>	1164
29.10	<pthread.h>	1165
29.11	<sched.h>	1167
29.12	<semaphore.h>	1168
29.13	<setjmp.h>	1169
29.14	<signal.h>	1170
29.15	<stdarg.h>	1171
29.16	<stddef.h>	1172
29.17	<stdint.h>	1173
29.18	<stdio.h>	1174
29.19	<stdlib.h>	1175
29.20	<string.h>	1176
29.21	<sys/mman.h>	1177
29.22	<sys/select.h>	1178
29.23	<sys/socket.h>	1179
29.24	<sys/stat.h>	1180
29.25	<sys/utsname.h>	1181
29.26	<time.h>	1182
29.27	<unistd.h>	1183

### **30 FACE Technical Standard, Edition 3.2 General Purpose 1185**

30.1	<aio.h>	1186
30.2	<arpa/inet.h>	1187
30.3	<complex.h>	1188
30.4	<ctype.h>	1190
30.5	<devctl.h>	1191
30.6	<dirent.h>	1192
30.7	<errno.h>	1193
30.8	<fcntl.h>	1194
30.9	<fenv.h>	1195
30.10	<inttypes.h>	1196
30.11	<locale.h>	1197
30.12	<math.h>	1198
30.13	<mqueue.h>	1204
30.14	<net/if.h>	1205
30.15	<netdb.h>	1206
30.16	<pthread.h>	1207
30.17	<sched.h>	1210
30.18	<semaphore.h>	1211
30.19	<setjmp.h>	1212
30.20	<signal.h>	1213
30.21	<stdarg.h>	1214
30.22	<stddef.h>	1215
30.23	<stdint.h>	1216
30.24	<stdio.h>	1217
30.25	<stdlib.h>	1219
30.26	<string.h>	1220
30.27	<sys/mman.h>	1221
30.28	<sys/select.h>	1222
30.29	<sys/socket.h>	1223
30.30	<sys/stat.h>	1224

30.31	<sys/utsname.h>	1225
30.32	<time.h>	1226
30.33	<unistd.h>	1227
<b>31</b>	<b>Software Communications Architecture 2.2.2 AEP</b>	<b>1229</b>
31.1	<ctype.h>	1230
31.2	<dirent.h>	1231
31.3	<fcntl.h>	1232
31.4	<locale.h>	1233
31.5	<math.h>	1234
31.6	<pthread.h>	1235
31.7	<semaphore.h>	1238
31.8	<setjmp.h>	1239
31.9	<signal.h>	1240
31.10	<stdio.h>	1241
31.11	<stdlib.h>	1243
31.12	<string.h>	1244
31.13	<sys/stat.h>	1245
31.14	<time.h>	1246
31.15	<unistd.h>	1247
31.16	<utime.h>	1248
<b>32</b>	<b>Software Communications Architecture 4.1 Ultra Lightweight Application Environ- ment Profile</b>	<b>1249</b>
32.1	<math.h>	1250
32.2	<mqueue.h>	1251
32.3	<pthread.h>	1252
32.4	<semaphore.h>	1253
32.5	<time.h>	1254
<b>33</b>	<b>Software Communications Architecture 4.1 Lightweight Application Environment Profile</b>	<b>1255</b>
33.1	<ctype.h>	1256
33.2	<fcntl.h>	1257
33.3	<math.h>	1258
33.4	<mqueue.h>	1259
33.5	<pthread.h>	1260
33.6	<semaphore.h>	1261
33.7	<stdio.h>	1262
33.8	<stdlib.h>	1263
33.9	<string.h>	1264
33.10	<time.h>	1265
33.11	<unistd.h>	1266
<b>34</b>	<b>Software Communications Architecture 4.1 [Full] Appliation Environment Profile</b>	<b>1267</b>
34.1	<arpa/inet.h>	1268
34.2	<ctype.h>	1269
34.3	<dirent.h>	1270
34.4	<errno.h>	1271
34.5	<fcntl.h>	1272
34.6	<math.h>	1273
34.7	<mqueue.h>	1274

34.8 <pthread.h> . . . . .	1275
34.9 <semaphore.h> . . . . .	1277
34.10<signal.h> . . . . .	1278
34.11<stdarg.h> . . . . .	1279
34.12<stdio.h> . . . . .	1280
34.13<stdlib.h> . . . . .	1282
34.14<string.h> . . . . .	1283
34.15<sys/select.h> . . . . .	1284
34.16<sys/socket.h> . . . . .	1285
34.17<sys/stat.h> . . . . .	1286
34.18<time.h> . . . . .	1287
34.19<unistd.h> . . . . .	1288
<b>35 Glossary</b>	<b>1289</b>
<b>36 References</b>	<b>1291</b>
<b>Index</b>	<b>1293</b>



### Copyrights and License

© 2017 Chris Johns

© 1988, 2025 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 <https://www.rtems.org>
- Documentation <https://docs.rtems.org>
- Mailing Lists <https://lists.rtems.org>
- Bug Reporting <https://gitlab.rtems.org>
- Git Repositories <https://gitlab.rtems.org>
- Developers <https://gitlab.rtems.org>



## 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 version 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.

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

Support	Amount
Supported	1064
ENOSYS	18
Not supported	192

### 3.1 <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()



## 3.2 <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()`

### 3.3 `<assert.h>`

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

- `assert()`

## 3.4 <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()`

### 3.5 <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()`

## 3.6 <devctl.h>

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

- posix\_devctl()

### 3.7 <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()`

### 3.8 <dlfcn.h>

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

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



## 3.9 <errno.h>

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

- errno

### 3.10 <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()

## 3.11 <fenv.h>

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

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

### 3.12 <fmtmsg.h>

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

- `fmtmsg()`

### 3.13 <fnmatch.h>

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

- fnmatch()

### 3.14 <ftw.h>

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

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

### 3.15 <glob.h>

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

- glob()
- globfree()

### 3.16 <grp.h>

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

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



### 3.17 <iconv.h>

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

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

### 3.18 <inttypes.h>

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

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

### 3.19 <langinfo.h>

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

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

## 3.20 <libgen.h>

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

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

### 3.21 <libintl.h>

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

- `textdomain()`

## 3.22 <locale.h>

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

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

### 3.23 <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()
- fabsl()
- fdim()
- fdimf()
- fdiml()
- floor()
- floorf()
- floorl()
- fma()
- maf()
- 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()`
- `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()`

## 3.24 <monetary.h>

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

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

### 3.25 <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()

## 3.26 <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()

### 3.27 <net/if.h>

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

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



## 3.28 <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()`

### 3.29 <nl\_types.h>

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

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

### 3.30 <poll.h>

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

- poll()

### 3.31 <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\_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()`

### 3.32 <pwd.h>

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

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



### 3.33 <regex.h>

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

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

### 3.34 <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()

### 3.35 <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()`

### 3.36 <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()`

### 3.37 <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()

### 3.38 <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()`

### 3.39 <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()

## 3.40 <stdarg.h>

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

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



### 3.41 <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()`

## 3.42 <stddef.h>

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

- `offsetof()`

### 3.43 <stdint.h>

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

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

## 3.44 <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()
- 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()`

### 3.45 <stdlib.h>

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

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

- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`
- `mktemp()`
- `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()`

## 3.46 <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()`

### 3.47 <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()`

### 3.48 <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()`

### 3.49 <sys/ipc.h>

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

- ftok()

### 3.50 <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()`

### 3.51 <sys/msg.h>

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

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



## 3.52 <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()

### 3.53 <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()

### 3.54 <sys/sem.h>

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

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

### 3.55 <sys/shm.h>

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

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

## 3.56 <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()`

### 3.57 <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()

### 3.58 <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()

### 3.59 <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()



## 3.60 <sys/uio.h>

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

- readv()
- writev()

### 3.61 <sys/utsname.h>

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

- `uname()`

## 3.62 <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()

### 3.63 <syslog.h>

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

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

### 3.64 <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()

### 3.65 <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()`

## 3.66 <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()
- mktime()
- 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



### 3.67 <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()`

### 3.68 <ulimit.h>

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

- `ulimit()`

## 3.69 <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()`

### 3.70 <utime.h>

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

- `utime()`

### 3.71 <utmpx.h>

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

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



## 3.72 <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()`
- `wcrtomb()`
- `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()`

### 3.73 <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()`

### 3.74 <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.

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

Support	Amount
Supported	1031
ENOSYS	16
Not supported	125

## 4.1 <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()



## 4.2 <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()`

## 4.3 <assert.h>

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

- `assert()`

## 4.4 <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()`

## 4.5 <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()

## 4.6 <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()`

## 4.7 <dlfcn.h>

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

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

## 4.8 <errno.h>

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

- errno



## 4.9 <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()

## 4.10 <fenv.h>

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

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

## 4.11 <fmtmsg.h>

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

- `fmtmsg()`

## 4.12 <fnmatch.h>

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

- fnmatch()

## 4.13 <ftw.h>

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

- `nftw()`

## 4.14 <glob.h>

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

- glob()
- globfree()

## 4.15 <grp.h>

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

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

## 4.16 <iconv.h>

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

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



## 4.17 <inttypes.h>

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

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

## 4.18 <langinfo.h>

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

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

## 4.19 <libgen.h>

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

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

## 4.20 <libintl.h>

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

- `textdomain()`

## 4.21 <locale.h>

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

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

## 4.22 <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()`
- `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()`

- `sqrthf()`
- `sqrthl()`
- `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()`

## 4.23 <monetary.h>

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

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

## 4.24 <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()

## 4.25 <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()

## 4.26 <net/if.h>

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

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

## 4.27 <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()`



## 4.28 <nl\_types.h>

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

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

## 4.29 <poll.h>

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

- poll()

## 4.30 <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_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()`

## 4.31 <pwd.h>

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

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

## 4.32 <regex.h>

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

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

## 4.33 <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()



## 4.34 <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()`

## 4.35 <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()`

## 4.36 <setjmp.h>

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

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

## 4.37 <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()

## 4.38 <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()

## 4.39 <stdarg.h>

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

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

## 4.40 <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()`

## 4.41 <stddef.h>

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

- `offsetof()`



## 4.42 <stdint.h>

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

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

## 4.43 <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()`
- `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()`

## 4.44 <stdlib.h>

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

- `_Exit()`
- `a64l()`
- `abort()`
- `abs()`
- `aligned_alloc()`
- `at_quick_exit()`
- `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()`
- `mrnd48()`
- `nrnd48()`
- `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()

## 4.45 <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()`

## 4.46 <strings.h>

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

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

## 4.47 <sys/ipc.h>

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

- ftok()

## 4.48 <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()`

## 4.49 <sys/msg.h>

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

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

## 4.50 <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()

## 4.51 <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()

## 4.52 <sys/sem.h>

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

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



## 4.53 <sys/shm.h>

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

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

## 4.54 <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()`

## 4.55 <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()

## 4.56 <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()

## 4.57 <sys/time.h>

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

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

## 4.58 <sys/uio.h>

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

- readv()
- writev()

## 4.59 <sys/utsname.h>

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

- `uname()`

## 4.60 <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()



## 4.61 <syslog.h>

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

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

## 4.62 <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()

## 4.63 <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()`

## 4.64 <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()
- mktime()
- 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

## 4.65 <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()`



## 4.66 <utmpx.h>

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

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

## 4.67 <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()
- wcpncpy()
- wcpncpy()
- wcrctomb()
- 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()`

## 4.68 <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()`

## 4.69 <wordexp.h>

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

- wordexp()
- wordfree()





## POSIX-2017 (ISSUE 7 TC2)

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

The follow table summarizes alignment with the POSIX-2017 (Issue 7 TC2) standard:

Support	Amount
Supported	986
ENOSYS	18
Not supported	192

## 5.1 <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()

## 5.2 <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()`

### 5.3 <assert.h>

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

- `assert()`

## 5.4 <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()`

## 5.5 <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()`

## 5.6 <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()`



## 5.7 <dlfcn.h>

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

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

## 5.8 <errno.h>

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

- errno

## 5.9 <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()

## 5.10 <fenv.h>

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

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

## 5.11 <fmtmsg.h>

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

- `fmtmsg()`

## 5.12 <fnmatch.h>

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

- fnmatch()

## 5.13 <ftw.h>

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

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

## 5.14 <glob.h>

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

- glob()
- globfree()



## 5.15 <grp.h>

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

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

## 5.16 <iconv.h>

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

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

## 5.17 <inttypes.h>

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

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

## 5.18 <langinfo.h>

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

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

## 5.19 <libgen.h>

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

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

## 5.20 <locale.h>

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

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

## 5.21 <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:

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

## 5.22 <monetary.h>

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

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

## 5.23 <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()

## 5.24 <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()

## 5.25 <net/if.h>

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

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



## 5.26 <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()`

## 5.27 <nl\_types.h>

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

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

## 5.28 <poll.h>

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

- poll()

## 5.29 <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()`

## 5.30 <pwd.h>

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

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



## 5.31 <regex.h>

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

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

## 5.32 <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()

### 5.33 <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()`

## 5.34 <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()`

## 5.35 <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()

## 5.36 <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()

## 5.37 <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()`

## 5.38 <stdarg.h>

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

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



## 5.39 <stddef.h>

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

- `offsetof()`

## 5.40 <stdint.h>

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

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

## 5.41 <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()`

## 5.42 <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()`

- `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()`

## 5.43 <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()`

## 5.44 <strings.h>

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

- `ffs()`
- `strcasecmp()`
- `strcasecmp_l()`
- `strncasecmp()`
- `strncasecmp_l()`

## 5.45 <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()`

## 5.46 <sys/ipc.h>

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

- `ftok()`

## 5.47 <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()`

## 5.48 <sys/msg.h>

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

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

## 5.49 <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()

## 5.50 <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()



## 5.51 <sys/sem.h>

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

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

## 5.52 <sys/shm.h>

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

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

## 5.53 <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()`

## 5.54 <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()

## 5.55 <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()

## 5.56 <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()

## 5.57 <sys/uio.h>

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

- readv()
- writev()

## 5.58 <sys/utsname.h>

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

- `uname()`



## 5.59 <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()

## 5.60 <syslog.h>

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

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

## 5.61 <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()

## 5.62 <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()
- mktime()
- 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

## 5.63 <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()`

## 5.64 <ulimit.h>

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

- `ulimit()`



## 5.65 <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()`

## 5.66 <utime.h>

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

- `utime()`

## 5.67 <utmpx.h>

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

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

## 5.68 <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()
- wcrctomb()
- 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()`



## 5.69 <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()`

## 5.70 <wordexp.h>

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

- wordexp()
- wordfree()



## POSIX-2008 (ISSUE 7)

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

The follow table summarizes alignment with the POSIX-2008 (Issue 7) standard:

Support	Amount
Supported	986
ENOSYS	18
Not supported	192

## 6.1 <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()

## 6.2 <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()`

## 6.3 <assert.h>

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

- `assert()`



## 6.4 <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()`

## 6.5 <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()`

## 6.6 <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()`

## 6.7 <dlfcn.h>

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

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

## 6.8 <errno.h>

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

- errno

## 6.9 <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()

## 6.10 <fenv.h>

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

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



## 6.11 <fmtmsg.h>

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

- `fmtmsg()`

## 6.12 <fnmatch.h>

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

- fnmatch()

## 6.13 <ftw.h>

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

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

## 6.14 <glob.h>

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

- glob()
- globfree()

## 6.15 <grp.h>

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

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

## 6.16 <iconv.h>

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

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

## 6.17 <inttypes.h>

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

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

## 6.18 <langinfo.h>

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

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



## 6.19 <libgen.h>

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

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

## 6.20 <locale.h>

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

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

## 6.21 <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:

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



## 6.22 <monetary.h>

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

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

## 6.23 <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()

## 6.24 <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()

## 6.25 <net/if.h>

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

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

## 6.26 <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()`

## 6.27 <nl\_types.h>

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

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

## 6.28 <poll.h>

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

- poll()

## 6.29 <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()`

## 6.30 <pwd.h>

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

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

## 6.31 <regex.h>

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

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

## 6.32 <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()

## 6.33 <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()`

## 6.34 <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()`



## 6.35 <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()`

## 6.36 <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()

## 6.37 <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()`

## 6.38 <stdarg.h>

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

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

## 6.39 <stddef.h>

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

- `offsetof()`

## 6.40 <stdint.h>

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

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

## 6.41 <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()`
- `vsnprintf()`
- `vsprintf()`
- `vsscanf()`



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

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

## 6.42 <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()`

- `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()`

## 6.43 <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()`

## 6.44 <strings.h>

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

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

## 6.45 <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()`

## 6.46 <sys/ipc.h>

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

- `ftok()`



## 6.47 <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()`

## 6.48 <sys/msg.h>

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

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

## 6.49 <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()

## 6.50 <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()

## 6.51 <sys/sem.h>

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

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

## 6.52 <sys/shm.h>

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

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

## 6.53 <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()`

## 6.54 <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()



## 6.55 <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()

## 6.56 <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()

## 6.57 <sys/uio.h>

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

- readv()
- writev()

## 6.58 <sys/utsname.h>

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

- `uname()`

## 6.59 <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()

## 6.60 <syslog.h>

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

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

## 6.61 <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()

## 6.62 <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()
- mktime()
- 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

## 6.63 <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()`

## 6.64 <ulimit.h>

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

- `ulimit()`

## 6.65 <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()`

## 6.66 <utime.h>

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

- `utime()`



## 6.67 <utmpx.h>

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

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

## 6.68 <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()`
- `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()`

## 6.69 <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()`

## 6.70 <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.

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

Support	Amount
Supported	938
ENOSYS	18
Not supported	166

## 7.1 <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()

## 7.2 <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()`

## 7.3 <assert.h>

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

- `assert()`

## 7.4 <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()`

## 7.5 <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()`

## 7.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`
- `seekdir()`



## 7.7 <dlfcn.h>

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

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

## 7.8 <errno.h>

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

- errno

## 7.9 <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()

## 7.10 <fenv.h>

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

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

## 7.11 <fmtmsg.h>

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

- `fmtmsg()`

## 7.12 <fnmatch.h>

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

- fnmatch()

## 7.13 <ftw.h>

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

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

## 7.14 <glob.h>

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

- glob()
- globfree()



## 7.15 <grp.h>

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

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

## 7.16 <iconv.h>

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

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

## 7.17 <inttypes.h>

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

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

## 7.18 <langinfo.h>

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

- nl\_langinfo()

## 7.19 <libgen.h>

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

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

## 7.20 <locale.h>

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

- localeconv()
- setlocale()

## 7.21 <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()`
- `scalb()`
- `scalbln()`
- `scalblnf()`
- `scalblnl()`
- `scalbn()`
- `scalbnf()`
- `scalbnl()`
- `sin()`
- `sinf()`
- `sinh()`
- `sinhf()`
- `sinhl()`
- `sinl()`

- `sqrt()`
- `sqrtf()`
- `sqrtrl()`
- `tan()`
- `tanf()`
- `tanh()`
- `tanhf()`
- `tanh1()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgamma1()`
- `trunc()`
- `truncf()`
- `truncl()`

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

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

## 7.22 <monetary.h>

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

- `strfmon()`

## 7.23 <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()

## 7.24 <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()

## 7.25 <net/if.h>

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

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



## 7.26 <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()`

## 7.27 <nl\_types.h>

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

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

## 7.28 <poll.h>

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

- poll()

## 7.29 <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()`

## 7.30 <pwd.h>

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

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

## 7.31 <regex.h>

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

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



## 7.32 <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()

## 7.33 <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()`

## 7.34 <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()`

## 7.35 <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()`

## 7.36 <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()`

## 7.37 <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()

## 7.38 <stdarg.h>

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

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

## 7.39 <stddef.h>

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

- `offsetof()`



## 7.40 <stdint.h>

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

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

## 7.41 <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()`

## 7.42 <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()`
- `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()`

## 7.43 <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()

## 7.44 <strings.h>

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

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



## 7.45 <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()`

## 7.46 <sys/ipc.h>

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

- ftok()

## 7.47 <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()`

## 7.48 <sys/msg.h>

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

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

## 7.49 <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()

## 7.50 <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()

## 7.51 <sys/sem.h>

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

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

## 7.52 <sys/shm.h>

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

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



## 7.53 <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()`

## 7.54 <sys/stat.h>

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

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

## 7.55 <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()

## 7.56 <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()

## 7.57 <sys/uio.h>

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

- readv()
- writev()

## 7.58 <sys/utsname.h>

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

- `uname()`

## 7.59 <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()

## 7.60 <syslog.h>

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

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



## 7.61 <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()

## 7.62 <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()
- mktime()
- 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`

## 7.63 <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()`

## 7.64 <ulimit.h>

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

- `ulimit()`

## 7.65 <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()`

## 7.66 <utime.h>

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

- `utime()`

## 7.67 <utmpx.h>

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

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

## 7.68 <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()
- 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()`

## 7.69 <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()

## 7.70 <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.

The follow table summarizes alignment with the POSIX PSE51 - Minimal standard:

Support	Amount
Supported	282
ENOSYS	2
Not supported	1

## 8.1 <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()

## 8.2 <errno.h>

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

- errno

### 8.3 <fcntl.h>

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

- open()

## 8.4 <fenv.h>

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

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

## 8.5 <inttypes.h>

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

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

## 8.6 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 8.7 <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\_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()`

## 8.8 <sched.h>

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

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

## 8.9 <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()`

## 8.10 <setjmp.h>

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

- longjmp()
- setjmp()

## 8.11 <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()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 8.12 <stdarg.h>

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

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

## 8.13 <stddef.h>

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

- `offsetof()`



## 8.14 <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()`

## 8.15 <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()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 8.16 <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()`

## 8.17 <sys/mman.h>

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

- `mlock()`
- `mlockall()`
- `mmap()`
- `munlock()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 8.18 <sys/utsname.h>

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

- `uname()`

## 8.19 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 8.20 <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.

The follow table summarizes alignment with the POSIX PSE52 - Real-Time Controller standard:

Support	Amount
Supported	575
ENOSYS	2
Not supported	52

## 9.1 <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()`

## 9.2 <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()

## 9.3 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 9.4 <errno.h>

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

- errno

## 9.5 <fcntl.h>

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

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

## 9.6 <fenv.h>

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

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



## 9.7 <inttypes.h>

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

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

## 9.8 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 9.9 <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()`
- `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()`



## 9.10 <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()

## 9.11 <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\_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()`

## 9.12 <sched.h>

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

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

## 9.13 <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()`

## 9.14 <setjmp.h>

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

- longjmp()
- setjmp()

## 9.15 <signal.h>

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

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



## 9.16 <stdarg.h>

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

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

## 9.17 <stddef.h>

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

- `offsetof()`

## 9.18 <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()`

## 9.19 <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()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 9.20 <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()`

## 9.21 <sys/mman.h>

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

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

## 9.22 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`



## 9.23 <sys/utsname.h>

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

- `uname()`

## 9.24 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 9.25 <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\_getmaxsystemeventszsize()
- posix\_trace\_attr\_getmaxusereventsizsize()
- 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()`

## 9.26 <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()

## 9.27 <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.

The follow table summarizes alignment with the POSIX PSE53 - Dedicated standard:

Support	Amount
Supported	661
ENOSYS	15
Not supported	75

## 10.1 <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()



## 10.2 <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()`

## 10.3 <assert.h>

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

- `assert()`

## 10.4 <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()`

## 10.5 <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()

## 10.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 10.7 <errno.h>

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

- errno

## 10.8 <fcntl.h>

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

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



## 10.9 <fenv.h>

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

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

## 10.10 <inttypes.h>

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

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

## 10.11 <locale.h>

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

- localeconv()
- setlocale()

## 10.12 <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()`
- `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()`

## 10.13 <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()

## 10.14 <net/if.h>

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

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

## 10.15 <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()`

## 10.16 <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()`

## 10.17 <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()



## 10.18 <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()`

## 10.19 <setjmp.h>

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

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

## 10.20 <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()

## 10.21 <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()

## 10.22 <stdarg.h>

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

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

## 10.23 <stddef.h>

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

- `offsetof()`

## 10.24 <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()`



## 10.25 <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()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`
- `strtoul()`
- `strtoull()`

- `unsetenv()`

## 10.26 <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()`

## 10.27 <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()`

## 10.28 <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()

## 10.29 <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()`

## 10.30 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`

## 10.31 <sys/time.h>

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

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



## 10.32 <sys/utsname.h>

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

- `uname()`

## 10.33 <sys/wait.h>

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

- `wait()`

## 10.34 <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()
- mktime()
- 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()

## 10.35 <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()`

## 10.36 <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()`

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

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

- `confstr()`

## 10.37 <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.

The follow table summarizes alignment with the POSIX PSE54 - Multipurpose standard:

Support	Amount
Supported	805
ENOSYS	16
Not supported	85

## 11.1 <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()

## 11.2 <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()`

## 11.3 <assert.h>

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

- `assert()`

## 11.4 <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()`

## 11.5 <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()

## 11.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`



## 11.7 <dlfcn.h>

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

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

## 11.8 <errno.h>

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

- errno

## 11.9 <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()

## 11.10 <fenv.h>

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

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

## 11.11 <fnmatch.h>

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

- fnmatch()

## 11.12 <glob.h>

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

- glob()
- globfree()

## 11.13 <grp.h>

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

- `getgrgid()`
- `getgrgid_r()`
- `getgrnam()`
- `getgrnam_r()`

## 11.14 <inttypes.h>

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

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



## 11.15 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 11.16 <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()`

## 11.17 <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()



## 11.18 <net/if.h>

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

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

## 11.19 <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()`

## 11.20 <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()`

## 11.21 <pwd.h>

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

- `getpwnam()`
- `getpwnam_r()`
- `getpwuid()`
- `getpwuid_r()`

## 11.22 <regex.h>

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

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

## 11.23 <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()



## 11.24 <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()`

## 11.25 <setjmp.h>

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

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

## 11.26 <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()

## 11.27 <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()

## 11.28 <stdarg.h>

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

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

## 11.29 <stddef.h>

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

- `offsetof()`

## 11.30 <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()`



## 11.31 <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()`
- `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()

## 11.32 <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()

## 11.33 <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()`

## 11.34 <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()

## 11.35 <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()`

## 11.36 <sys/stat.h>

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

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

## 11.37 <sys/time.h>

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

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



## 11.38 <sys/utsname.h>

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

- `uname()`

## 11.39 <sys/wait.h>

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

- `wait()`

## 11.40 <syslog.h>

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

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

## 11.41 <termios.h>

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

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

## 11.42 <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()
- mktime()
- 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()

## 11.43 <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()`

## 11.44 <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()`

## 11.45 <utime.h>

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

- `utime()`

## 11.46 <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()

## 11.47 <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()`

## 11.48 <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.

The follow table summarizes alignment with the C99 Standard Library standard:

Support	Amount
Supported	488
ENOSYS	1
Not supported	1

## 12.1 <assert.h>

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

- `assert()`



## 12.2 <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()`

## 12.3 <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()

## 12.4 <errno.h>

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

- errno

## 12.5 <fenv.h>

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

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

## 12.6 <inttypes.h>

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

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

## 12.7 <locale.h>

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

- localeconv()
- setlocale()

## 12.8 <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()`



- ## REFERENCES

- `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()`

## 12.9 <setjmp.h>

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

- longjmp()
- setjmp()

## 12.10 <signal.h>

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

- raise()
- signal()

## 12.11 <stdarg.h>

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

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



## 12.12 <stddef.h>

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

- `offsetof()`

## 12.13 <stdint.h>

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

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

## 12.14 <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()`

## 12.15 <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()`
- `qsort()`
- `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()`

## 12.16 <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()

## 12.17 <time.h>

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

- asctime()
- clock()
- ctime()
- difftime()
- gmtime()
- localtime()
- mktime()
- strftime()
- time()



## 12.18 <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()`
- `wcrtomb()`
- `wcscat()`
- `wcschr()`
- `wcscmp()`
- `wcscoll()`
- `wcscpy()`
- `wcscspn()`

- `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()`

## 12.19 <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.

The follow table summarizes alignment with the C11 Standard Library standard:

Support	Amount
Supported	535
ENOSYS	1
Not supported	1

## 13.1 <assert.h>

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

- `assert()`

## 13.2 <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()`



## 13.3 <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()

## 13.4 <errno.h>

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

- errno

## 13.5 <fenv.h>

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

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

## 13.6 <inttypes.h>

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

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

## 13.7 <locale.h>

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

- localeconv()
- setlocale()

## 13.8 <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()`
- `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()`

## 13.9 <setjmp.h>

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

- longjmp()
- setjmp()

## 13.10 <signal.h>

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

- raise()
- signal()

## 13.11 <stdarg.h>

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

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

## 13.12 <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()`

## 13.13 <stddef.h>

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

- `offsetof()`



## 13.14 <stdint.h>

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

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

## 13.15 <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()
- 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()`

## 13.16 <stdlib.h>

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

- `_Exit()`
- `abs()`
- `aligned_alloc()`
- `at_quick_exit()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `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()`

## 13.17 <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()`

## 13.18 <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()`

## 13.19 <time.h>

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

- asctime()
- clock()
- ctime()
- difftime()
- gmtime()
- localtime()
- mktime()
- strftime()
- time()



## 13.20 <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()`

## 13.21 <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()`
- `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.

The follow table summarizes alignment with the C17 Standard Library standard:

Support	Amount
Supported	531
ENOSYS	1
Not supported	1

## 14.1 <assert.h>

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

- `assert()`

## 14.2 <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()`



## 14.3 <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()

## 14.4 <errno.h>

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

- errno

## 14.5 <fenv.h>

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

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

## 14.6 <inttypes.h>

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

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

## 14.7 <locale.h>

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

- localeconv()
- setlocale()

## 14.8 <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()`

- ## REFERENCES

- `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()`





- `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()`

## 14.9 <setjmp.h>

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

- longjmp()
- setjmp()

## 14.10 <signal.h>

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

- raise()
- signal()

## 14.11 <stdarg.h>

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

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

## 14.12 <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()`

## 14.13 <stddef.h>

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

- `offsetof()`



## 14.14 <stdint.h>

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

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

## 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()`
- `at_quick_exit()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `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()
- mktime()
- 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()`
- `towlower()`
- `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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Security standard:

Support	Amount
Supported	162
ENOSYS	1
Not supported	0

## 15.1 <arpa/inet.h>

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

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

## 15.2 <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()

## 15.3 <devctl.h>

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

- posix\_devctl()



## 15.4 <errno.h>

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

- errno

## 15.5 <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()`

## 15.6 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 15.7 <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()`

## 15.8 <sched.h>

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

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

## 15.9 <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()`

## 15.10 <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()



## 15.11 <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()`

## 15.12 <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()

## 15.13 <sys/mman.h>

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

- `mmap()`
- `shm_open()`

## 15.14 <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()`

## 15.15 <sys/stat.h>

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

- stat()

## 15.16 <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()

## 15.17 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Safety Base standard:

Support	Amount
Supported	246
ENOSYS	1
Not supported	0

## 16.1 <arpa/inet.h>

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

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

## 16.2 <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()

## 16.3 <devctl.h>

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

- posix\_devctl()

## 16.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 16.5 <errno.h>

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

- errno

## 16.6 <fcntl.h>

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

- creat()
- open()

## 16.7 <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()`



## 16.8 <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()

## 16.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 16.10 <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()`

## 16.11 <sched.h>

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

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

## 16.12 <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()`

## 16.13 <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()

## 16.14 <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()



## 16.15 <stdlib.h>

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

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 16.16 <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()

## 16.17 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `shm_open()`

## 16.18 <sys/select.h>

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

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

## 16.19 <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()

## 16.20 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 16.21 <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()
- mktime()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 16.22 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 Safety Extended standard:

Support	Amount
Supported	316
ENOSYS	11
Not supported	9

## 17.1 <arpa/inet.h>

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

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

## 17.2 <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()

## 17.3 <devctl.h>

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

- posix\_devctl()

## 17.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 17.5 <errno.h>

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

- errno

## 17.6 <fcntl.h>

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

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

## 17.7 <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()`



## 17.8 <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()

## 17.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 17.10 <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()`

## 17.11 <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()

## 17.12 <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()`

## 17.13 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()

## 17.14 <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()



## 17.15 <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()`

## 17.16 <stdarg.h>

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

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

## 17.17 <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()`

## 17.18 <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()`
- `rand_r()`
- `realloc()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 17.19 <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()

## 17.20 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `shm_open()`

## 17.21 <sys/select.h>

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

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

## 17.22 <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()`



## 17.23 <sys/stat.h>

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

- `chmod()`
- `fstat()`
- `lstat()`
- `mkdir()`
- `mkfifo()`
- `stat()`
- `umask()`

## 17.24 <sys/time.h>

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

- `times()`

## 17.25 <sys/utsname.h>

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

- `uname()`

## 17.26 <sys/wait.h>

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

- waitpid()

## 17.27 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 17.28 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 2.1 General Purpose standard:

Support	Amount
Supported	773
ENOSYS	13
Not supported	25

## 18.1 <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()

## 18.2 <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()`

## 18.3 <assert.h>

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

- `assert()`

## 18.4 <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()`

## 18.5 <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()

## 18.6 <devctl.h>

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

- posix\_devctl()



## 18.7 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 18.8 <errno.h>

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

- errno

## 18.9 <fcntl.h>

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

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

## 18.10 <fenv.h>

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

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

## 18.11 <inttypes.h>

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

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

## 18.12 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 18.13 <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()`
- `tanl()`
- `tgamma()`
- `tgammaf()`
- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `signbit()`

## 18.14 <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()

## 18.15 <net/if.h>

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

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

## 18.16 <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()`

## 18.17 <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_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()`

## 18.18 <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()

## 18.19 <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()`

## 18.20 <setjmp.h>

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

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

## 18.21 <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()

## 18.22 <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()`

## 18.23 <stdarg.h>

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

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



## 18.24 <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()`

## 18.25 <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()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`
- `strtoll()`

- strtoul()
- strtoull()
- unsetenv()
- wcstombs()
- wctomb()

## 18.26 <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()`

## 18.27 <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()`

## 18.28 <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()

## 18.29 <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()



## 18.30 <sys/stat.h>

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

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

## 18.31 <sys/time.h>

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

- `times()`

## 18.32 <sys/utsname.h>

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

- `uname()`

## 18.33 <sys/wait.h>

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

- wait()
- waitpid()

## 18.34 <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()
- mktime()
- 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()

## 18.35 <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()`

## 18.36 <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()`

## 18.37 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Security standard:

Support	Amount
Supported	172
ENOSYS	1
Not supported	0

## 19.1 <arpa/inet.h>

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

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

## 19.2 <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()

## 19.3 <devctl.h>

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

- posix\_devctl()

## 19.4 <errno.h>

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

- errno

## 19.5 <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()`



## 19.6 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 19.7 <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()`

## 19.8 <sched.h>

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

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

## 19.9 <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()`

## 19.10 <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()

## 19.11 <stddef.h>

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

- `offsetof()`

## 19.12 <stdint.h>

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

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



## 19.13 <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()`

## 19.14 <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()

## 19.15 <sys/mman.h>

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

- `mmap()`
- `shm_open()`

## 19.16 <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()

## 19.17 <sys/stat.h>

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

- stat()

## 19.18 <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()

## 19.19 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Safety Base standard:

Support	Amount
Supported	254
ENOSYS	1
Not supported	0

## 20.1 <arpa/inet.h>

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

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

## 20.2 <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()

## 20.3 <devctl.h>

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

- posix\_devctl()

## 20.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 20.5 <errno.h>

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

- errno

## 20.6 <fcntl.h>

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

- creat()
- open()

## 20.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`



## 20.8 <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()

## 20.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 20.10 <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()`

## 20.11 <sched.h>

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

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

## 20.12 <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()`

## 20.13 <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()

## 20.14 <stddef.h>

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

- `offsetof()`



## 20.15 <stdint.h>

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

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

## 20.16 <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()

## 20.17 <stdlib.h>

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

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 20.18 <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()

## 20.19 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `shm_open()`

## 20.20 <sys/select.h>

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

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

## 20.21 <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()`

## 20.22 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`



## 20.23 <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()
- mktime()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 20.24 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 Safety Extended standard:

Support	Amount
Supported	319
ENOSYS	2
Not supported	0

## 21.1 <arpa/inet.h>

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

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

## 21.2 <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()

## 21.3 <devctl.h>

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

- posix\_devctl()

## 21.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 21.5 <errno.h>

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

- errno



## 21.6 <fcntl.h>

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

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

## 21.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 21.8 <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\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 21.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 21.10 <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()`

## 21.11 <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()

## 21.12 <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()`



## 21.13 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()

## 21.14 <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()

## 21.15 <stdarg.h>

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

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

## 21.16 <stddef.h>

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

- `offsetof()`

## 21.17 <stdint.h>

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

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

## 21.18 <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()

## 21.19 <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()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 21.20 <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()



## 21.21 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `mprotect()`
- `shm_open()`

## 21.22 <sys/select.h>

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

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

## 21.23 <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()`

## 21.24 <sys/stat.h>

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

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

## 21.25 <sys/utsname.h>

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

- `uname()`

## 21.26 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 21.27 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.0 General Purpose standard:

Support	Amount
Supported	704
ENOSYS	4
Not supported	3

## 22.1 <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()

## 22.2 <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()`

## 22.3 <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()`

## 22.4 <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()`

## 22.5 <devctl.h>

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

- posix\_devctl()

## 22.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`



## 22.7 <errno.h>

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

- errno

## 22.8 <fcntl.h>

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

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

## 22.9 <fenv.h>

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

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

## 22.10 <inttypes.h>

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

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

## 22.11 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 22.12 <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()`
- `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()`

## 22.13 <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()

## 22.14 <net/if.h>

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

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

## 22.15 <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()`

## 22.16 <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()`

## 22.17 <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()

## 22.18 <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()`

## 22.19 <setjmp.h>

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

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

## 22.20 <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()

## 22.21 <stdarg.h>

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

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

## 22.22 <stddef.h>

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

- `offsetof()`

## 22.23 <stdint.h>

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

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



## 22.24 <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()`

## 22.25 <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()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

- `wcstombs()`
- `wctomb()`

## 22.26 <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()`

## 22.27 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `mprotect()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 22.28 <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()

## 22.29 <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()



## 22.30 <sys/stat.h>

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

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

## 22.31 <sys/utsname.h>

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

- `uname()`

## 22.32 <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()
- mktime()
- 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()

## 22.33 <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()`

## 22.34 <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()

## 22.35 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Security standard:

Support	Amount
Supported	174
ENOSYS	1
Not supported	0

## 23.1 <arpa/inet.h>

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

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

## 23.2 <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()

## 23.3 <devctl.h>

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

- posix\_devctl()

## 23.4 <errno.h>

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

- errno

## 23.5 <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()`

## 23.6 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 23.7 <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()`

## 23.8 <sched.h>

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

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

## 23.9 <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()`

## 23.10 <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()

## 23.11 <stddef.h>

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

- `offsetof()`

## 23.12 <stdint.h>

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

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

## 23.13 <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()`

## 23.14 <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()`



## 23.15 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `shm_open()`

## 23.16 <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()

## 23.17 <sys/stat.h>

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

- stat()

## 23.18 <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()

## 23.19 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Safety Base standard:

Support	Amount
Supported	254
ENOSYS	1
Not supported	0

## 24.1 <arpa/inet.h>

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

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



## 24.2 <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()

## 24.3 <devctl.h>

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

- posix\_devctl()

## 24.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 24.5 <errno.h>

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

- errno

## 24.6 <fcntl.h>

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

- creat()
- open()

## 24.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 24.8 <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()

## 24.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()



## 24.10 <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()`

## 24.11 <sched.h>

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

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

## 24.12 <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()`

## 24.13 <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()

## 24.14 <stddef.h>

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

- `offsetof()`

## 24.15 <stdint.h>

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

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

## 24.16 <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()



## 24.17 <stdlib.h>

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

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 24.18 <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()

## 24.19 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `shm_open()`

## 24.20 <sys/select.h>

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

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

## 24.21 <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()

## 24.22 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`

## 24.23 <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()
- mktime()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 24.24 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 Safety Extended standard:

Support	Amount
Supported	320
ENOSYS	2
Not supported	0

## 25.1 <arpa/inet.h>

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

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

## 25.2 <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()

## 25.3 <devctl.h>

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

- posix\_devctl()

## 25.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 25.5 <errno.h>

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

- errno

## 25.6 <fcntl.h>

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

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

## 25.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`



## 25.8 <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\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 25.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 25.10 <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()`

## 25.11 <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()

## 25.12 <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()`

## 25.13 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()

## 25.14 <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()



## 25.15 <stdarg.h>

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

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

## 25.16 <stddef.h>

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

- `offsetof()`

## 25.17 <stdint.h>

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

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

## 25.18 <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()

## 25.19 <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()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 25.20 <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()

## 25.21 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `mprotect()`
- `shm_open()`

## 25.22 <sys/select.h>

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

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



## 25.23 <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()`

## 25.24 <sys/stat.h>

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

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

## 25.25 <sys/utsname.h>

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

- `uname()`

## 25.26 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 25.27 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.1 General Purpose standard:

Support	Amount
Supported	673
ENOSYS	4
Not supported	5

## 26.1 <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()



## 26.2 <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()`

## 26.3 <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()`

## 26.4 <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()`

## 26.5 <devctl.h>

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

- posix\_devctl()

## 26.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 26.7 <errno.h>

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

- errno

## 26.8 <fcntl.h>

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

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



## 26.9 <fenv.h>

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

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

## 26.10 <inttypes.h>

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

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

## 26.11 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 26.12 <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()`
- `tgamma1()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `signbit()`

## 26.13 <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()

## 26.14 <net/if.h>

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

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

## 26.15 <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()`

## 26.16 <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()`

## 26.17 <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()



## 26.18 <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()`

## 26.19 <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()

## 26.20 <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()

## 26.21 <stdarg.h>

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

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

## 26.22 <stddef.h>

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

- `offsetof()`

## 26.23 <stdint.h>

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

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

## 26.24 <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()`



## 26.25 <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()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 26.26 <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()`

## 26.27 <sys/mman.h>

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

- `mlockall()`
- `mmap()`
- `mprotect()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 26.28 <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()

## 26.29 <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()`

## 26.30 <sys/stat.h>

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

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

## 26.31 <sys/utsname.h>

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

- `uname()`

## 26.32 <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()
- mktime()
- 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()



## 26.33 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Security standard:

Support	Amount
Supported	172
ENOSYS	1
Not supported	0

## 27.1 <arpa/inet.h>

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

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

## 27.2 <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()

## 27.3 <devctl.h>

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

- posix\_devctl()

## 27.4 <errno.h>

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

- errno

## 27.5 <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()`



## 27.6 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 27.7 <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()`

## 27.8 <sched.h>

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

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

## 27.9 <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()`

## 27.10 <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()

## 27.11 <stddef.h>

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

- `offsetof()`

## 27.12 <stdint.h>

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

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



## 27.13 <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()`

## 27.14 <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()

## 27.15 <sys/mman.h>

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

- `mmap()`
- `shm_open()`

## 27.16 <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()`

## 27.17 <sys/stat.h>

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

- stat()

## 27.18 <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()

## 27.19 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Safety Base standard:

Support	Amount
Supported	253
ENOSYS	1
Not supported	0

## 28.1 <arpa/inet.h>

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

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

## 28.2 <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()

## 28.3 <devctl.h>

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

- posix\_devctl()

## 28.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 28.5 <errno.h>

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

- errno

## 28.6 <fcntl.h>

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

- creat()
- open()

## 28.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`



## 28.8 <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()

## 28.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 28.10 <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()

## 28.11 <sched.h>

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

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

## 28.12 <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()`

## 28.13 <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()

## 28.14 <stddef.h>

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

- `offsetof()`



## 28.15 <stdint.h>

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

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

## 28.16 <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()

## 28.17 <stdlib.h>

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

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `calloc()`
- `div()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 28.18 <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()

## 28.19 <sys/mman.h>

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

- `mmap()`
- `shm_open()`

## 28.20 <sys/select.h>

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

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

## 28.21 <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()`

## 28.22 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`
- `umask()`



## 28.23 <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()
- mktime()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 28.24 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 Safety Extended standard:

Support	Amount
Supported	319
ENOSYS	2
Not supported	0

## 29.1 <arpa/inet.h>

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

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

## 29.2 <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()

## 29.3 <devctl.h>

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

- posix\_devctl()

## 29.4 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 29.5 <errno.h>

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

- errno



## 29.6 <fcntl.h>

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

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

## 29.7 <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()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 29.8 <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\_timedreceive()
- mq\_timedsend()
- mq\_unlink()

## 29.9 <netdb.h>

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

- freeaddrinfo()
- getaddrinfo()
- getnameinfo()

## 29.10 <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()`

## 29.11 <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()

## 29.12 <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()`



## 29.13 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()

## 29.14 <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()

## 29.15 <stdarg.h>

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

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

## 29.16 <stddef.h>

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

- `offsetof()`

## 29.17 <stdint.h>

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

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

## 29.18 <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()

## 29.19 <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()
- qsort()
- rand\_r()
- realloc()
- strtod()
- strtol()
- strtoul()

## 29.20 <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()



## 29.21 <sys/mman.h>

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

- `mmap()`
- `mprotect()`
- `shm_open()`

## 29.22 <sys/select.h>

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

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

## 29.23 <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()`

## 29.24 <sys/stat.h>

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

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

## 29.25 <sys/utsname.h>

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

- `uname()`

## 29.26 <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()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 29.27 <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.

The follow table summarizes alignment with the FACE Technical Standard, Edition 3.2 General Purpose standard:

Support	Amount
Supported	672
ENOSYS	4
Not supported	5

## 30.1 <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()

## 30.2 <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()`

### 30.3 <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()`

## 30.4 <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()`

## 30.5 <devctl.h>

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

- posix\_devctl()

## 30.6 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`



## 30.7 <errno.h>

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

- errno

## 30.8 <fcntl.h>

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

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

## 30.9 <fenv.h>

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

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

## 30.10 <inttypes.h>

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

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

## 30.11 <locale.h>

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

- `localeconv()`
- `setlocale()`

## 30.12 <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()`
- `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()`

## 30.13 <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()

## 30.14 <net/if.h>

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

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

## 30.15 <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()`

## 30.16 <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()`

## 30.17 <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()

## 30.18 <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()`

## 30.19 <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()

## 30.20 <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()

## 30.21 <stdarg.h>

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

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

## 30.22 <stddef.h>

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

- `offsetof()`

## 30.23 <stdint.h>

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

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



## 30.24 <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()`

## 30.25 <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()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 30.26 <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()`

## 30.27 <sys/mman.h>

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

- `mmap()`
- `mprotect()`
- `munmap()`
- `shm_open()`
- `shm_unlink()`

## 30.28 <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()

## 30.29 <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()`

## 30.30 <sys/stat.h>

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

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



## 30.31 <sys/utsname.h>

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

- `uname()`

## 30.32 <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()
- mktime()
- 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()

### 30.33 <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.

The follow table summarizes alignment with the Software Communications Architecture 2.2.2 AEP standard:

Support	Amount
Supported	243
ENOSYS	0
Not supported	0

## 31.1 <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()`

## 31.2 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

### 31.3 <fcntl.h>

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

- creat()
- open()



## 31.4 <locale.h>

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

- `setlocale()`

## 31.5 <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()`

## 31.6 <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()`

## 31.7 <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()`

## 31.8 <setjmp.h>

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

- longjmp()
- setjmp()

## 31.9 <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()



## 31.10 <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()`

## 31.11 <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()

## 31.12 <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()`

### 31.13 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`

## 31.14 <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()

## 31.15 <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()`

## 31.16 <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.

The follow table summarizes alignment with the Software Communications Architecture 4.1 Ultra Lightweight Application Environment Profile standard:

Support	Amount
Supported	22
ENOSYS	0
Not supported	0

## 32.1 <math.h>

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

- exp()
- exp2()

## 32.2 <mqqueue.h>

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

- mq\_open()
- mq\_receive()
- mq\_send()

## 32.3 <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()

## 32.4 <semaphore.h>

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

- `sem_init()`
- `sem_post()`
- `sem_wait()`

## 32.5 <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.

The follow table summarizes alignment with the Software Communications Architecture 4.1 Lightweight Application Environment Profile standard:

Support	Amount
Supported	110
ENOSYS	0
Not supported	0

## 33.1 <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()



## 33.2 <fcntl.h>

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

- `open()`

### 33.3 <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()`

## 33.4 <mqqueue.h>

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

- mq\_open()
- mq\_receive()
- mq\_send()

## 33.5 <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()

## 33.6 <semaphore.h>

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

- `sem_getvalue()`
- `sem_init()`
- `sem_post()`
- `sem_wait()`

## 33.7 <stdio.h>

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

- `sscanf()`

## 33.8 <stdlib.h>

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

- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `free()`
- `malloc()`
- `qsort()`
- `rand()`
- `realloc()`
- `srand()`

## 33.9 <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()



## 33.10 <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()

## 33.11 <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.

The follow table summarizes alignment with the Software Communications Architecture 4.1 [Full] Appliation Environment Profile standard:

Support	Amount
Supported	255
ENOSYS	0
Not supported	0

## 34.1 <arpa/inet.h>

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

- htonl()
- htons()
- ntohl()
- ntohs()

## 34.2 <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()

### 34.3 <dirent.h>

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

- `closedir()`
- `opendir()`
- `readdir()`
- `readdir_r()`
- `rewinddir()`

## 34.4 <errno.h>

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

- errno

## 34.5 <fcntl.h>

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

- creat()
- open()



## 34.6 <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()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 34.7 <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()

## 34.8 <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()`

## 34.9 <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()`

## 34.10 <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()

## 34.11 <stdarg.h>

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

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

## 34.12 <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()`

## 34.13 <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()

## 34.14 <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()`

## 34.15 <sys/select.h>

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

- `select()`

## 34.16 <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()`

## 34.17 <sys/stat.h>

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

- `fstat()`
- `mkdir()`
- `stat()`

## 34.18 <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()

## 34.19 <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.



## REFERENCES



# INDEX

## Symbols

: Portable Operating System Interface is  
a family of standards specified,  
**1289**

## P

POSIX, **1289**