



# RTEMS POSIX 1003.1 Compliance Guide

*Release 5.0.0-m2004-2 (2929th April 2020)*

© 1988, 2020 RTEMS Project and contributors



# CONTENTS

<b>1</b>	<b>Preface</b>	<b>3</b>
<b>2</b>	<b>Standards</b>	<b>5</b>
<b>3</b>	<b>RTEMS Complete Profile</b>	<b>7</b>
3.1	Summary . . . . .	8
3.2	<aio.h> . . . . .	9
3.3	<arpa/inet.h> . . . . .	10
3.4	<assert.h> . . . . .	11
3.5	<complex.h> . . . . .	12
3.6	<ctype.h> . . . . .	14
3.7	<devctl.h> . . . . .	15
3.8	<dirent.h> . . . . .	16
3.9	<dlfcn.h> . . . . .	17
3.10	<errno.h> . . . . .	18
3.11	<fcntl.h> . . . . .	19
3.12	<fenv.h> . . . . .	20
3.13	<fmtmsg.h> . . . . .	21
3.14	<fnmatch.h> . . . . .	22
3.15	<ftw.h> . . . . .	23
3.16	<glob.h> . . . . .	24
3.17	<grp.h> . . . . .	25
3.18	<iconv.h> . . . . .	26
3.19	<inttypes.h> . . . . .	27
3.20	<langinfo.h> . . . . .	28
3.21	<libgen.h> . . . . .	29
3.22	<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	<stddef.h>	57
3.42	<stdio.h>	58
3.43	<stdlib.h>	61
3.44	<string.h>	64
3.45	<strings.h>	66
3.46	<stropts.h>	67
3.47	<sys/ipc.h>	68
3.48	<sys/mman.h>	69
3.49	<sys/msg.h>	70
3.50	<sys/resource.h>	71
3.51	<sys/select.h>	72
3.52	<sys/sem.h>	73
3.53	<sys/shm.h>	74
3.54	<sys/socket.h>	75
3.55	<sys/stat.h>	76
3.56	<sys/statvfs.h>	77
3.57	<sys/time.h>	78
3.58	<sys/uio.h>	79
3.59	<sys/utsname.h>	80
3.60	<sys/wait.h>	81
3.61	<syslog.h>	82
3.62	<termios.h>	83
3.63	<threads.h>	84
3.64	<time.h>	85
3.65	<trace.h>	87
3.66	<ulimit.h>	89
3.67	<unistd.h>	90
3.68	<utime.h>	93
3.69	<utmpx.h>	94
3.70	<wchar.h>	95
3.71	<wctype.h>	98
3.72	<wordexp.h>	100
<b>4</b>	<b>POSIX-2008</b>	<b>101</b>
4.1	Summary	102
4.2	<aio.h>	103
4.3	<arpa/inet.h>	104
4.4	<assert.h>	105
4.5	<complex.h>	106
4.6	<ctype.h>	108
4.7	<dirent.h>	109
4.8	<dlfcn.h>	110
4.9	<errno.h>	111
4.10	<fcntl.h>	112

4.11	<fenv.h>	113
4.12	<fmtmsg.h>	114
4.13	<fnmatch.h>	115
4.14	<ftw.h>	116
4.15	<glob.h>	117
4.16	<grp.h>	118
4.17	<iconv.h>	119
4.18	<inttypes.h>	120
4.19	<langinfo.h>	121
4.20	<libgen.h>	122
4.21	<locale.h>	123
4.22	<math.h>	124
4.23	<monetary.h>	130
4.24	<mqueue.h>	131
4.25	<ndbm.h>	132
4.26	<net/if.h>	133
4.27	<netdb.h>	134
4.28	<nl_types.h>	135
4.29	<poll.h>	136
4.30	<pthread.h>	137
4.31	<pwd.h>	141
4.32	<regex.h>	142
4.33	<sched.h>	143
4.34	<search.h>	144
4.35	<semaphore.h>	145
4.36	<setjmp.h>	146
4.37	<signal.h>	147
4.38	<spawn.h>	148
4.39	<stdarg.h>	149
4.40	<stddef.h>	150
4.41	<stdio.h>	151
4.42	<stdlib.h>	154
4.43	<string.h>	156
4.44	<strings.h>	158
4.45	<stropts.h>	159
4.46	<sys/ipc.h>	160
4.47	<sys/mman.h>	161
4.48	<sys/msg.h>	162
4.49	<sys/resource.h>	163
4.50	<sys/select.h>	164
4.51	<sys/sem.h>	165
4.52	<sys/shm.h>	166
4.53	<sys/socket.h>	167
4.54	<sys/stat.h>	168
4.55	<sys/statvfs.h>	169
4.56	<sys/time.h>	170
4.57	<sys/uio.h>	171
4.58	<sys/utsname.h>	172
4.59	<sys/wait.h>	173
4.60	<syslog.h>	174
4.61	<termios.h>	175

4.62	<time.h>	176
4.63	<trace.h>	178
4.64	<ulimit.h>	180
4.65	<unistd.h>	181
4.66	<utime.h>	184
4.67	<utmpx.h>	185
4.68	<wchar.h>	186
4.69	<wctype.h>	189
4.70	<wordexp.h>	191
<b>5</b>	<b>POSIX-2003</b>	<b>193</b>
5.1	Summary	194
5.2	<aio.h>	195
5.3	<arpa/inet.h>	196
5.4	<assert.h>	197
5.5	<complex.h>	198
5.6	<ctype.h>	200
5.7	<dirent.h>	201
5.8	<dlfcn.h>	202
5.9	<errno.h>	203
5.10	<fcntl.h>	204
5.11	<fenv.h>	205
5.12	<fmtmsg.h>	206
5.13	<fnmatch.h>	207
5.14	<ftw.h>	208
5.15	<glob.h>	209
5.16	<grp.h>	210
5.17	<iconv.h>	211
5.18	<inttypes.h>	212
5.19	<langinfo.h>	213
5.20	<libgen.h>	214
5.21	<locale.h>	215
5.22	<math.h>	216
5.23	<monetary.h>	222
5.24	<mq.h>	223
5.25	<ndbm.h>	224
5.26	<net/if.h>	225
5.27	<netdb.h>	226
5.28	<nlist.h>	227
5.29	<poll.h>	228
5.30	<pthread.h>	229
5.31	<pwd.h>	232
5.32	<regex.h>	233
5.33	<sched.h>	234
5.34	<search.h>	235
5.35	<semaphore.h>	236
5.36	<setjmp.h>	237
5.37	<signal.h>	238
5.38	<spawn.h>	239
5.39	<stdarg.h>	240
5.40	<stddef.h>	241

5.41	<stdio.h>	242
5.42	<stdlib.h>	244
5.43	<string.h>	247
5.44	<strings.h>	248
5.45	<stropts.h>	249
5.46	<sys/ipc.h>	250
5.47	<sys/mman.h>	251
5.48	<sys/msg.h>	252
5.49	<sys/resource.h>	253
5.50	<sys/select.h>	254
5.51	<sys/sem.h>	255
5.52	<sys/shm.h>	256
5.53	<sys/socket.h>	257
5.54	<sys/stat.h>	258
5.55	<sys/statvfs.h>	259
5.56	<sys/time.h>	260
5.57	<sys/uio.h>	261
5.58	<sys/utsname.h>	262
5.59	<sys/wait.h>	263
5.60	<syslog.h>	264
5.61	<termios.h>	265
5.62	<time.h>	266
5.63	<trace.h>	268
5.64	<ulimit.h>	270
5.65	<unistd.h>	271
5.66	<utime.h>	274
5.67	<utmpx.h>	275
5.68	<wchar.h>	276
5.69	<wctype.h>	278
5.70	<wordexp.h>	279
<b>6</b>	<b>POSIX PSE51 - Minimal</b>	<b>281</b>
6.1	Summary	282
6.2	<ctype.h>	283
6.3	<errno.h>	284
6.4	<fcntl.h>	285
6.5	<fenv.h>	286
6.6	<inttypes.h>	287
6.7	<locale.h>	288
6.8	<pthread.h>	289
6.9	<sched.h>	292
6.10	<semaphore.h>	293
6.11	<setjmp.h>	294
6.12	<signal.h>	295
6.13	<stdarg.h>	296
6.14	<stdio.h>	297
6.15	<stdlib.h>	299
6.16	<string.h>	300
6.17	<sys/mman.h>	301
6.18	<sys/utsname.h>	302
6.19	<time.h>	303

6.20	<unistd.h>	304
<b>7</b>	<b>POSIX PSE52 - Real-Time Controller</b>	<b>305</b>
7.1	Summary	306
7.2	<complex.h>	307
7.3	<ctype.h>	309
7.4	<dirent.h>	310
7.5	<errno.h>	311
7.6	<fcntl.h>	312
7.7	<fenv.h>	313
7.8	<inttypes.h>	314
7.9	<locale.h>	315
7.10	<math.h>	316
7.11	<mqueue.h>	322
7.12	<pthread.h>	323
7.13	<sched.h>	326
7.14	<semaphore.h>	327
7.15	<setjmp.h>	328
7.16	<signal.h>	329
7.17	<stdarg.h>	330
7.18	<stdio.h>	331
7.19	<stdlib.h>	333
7.20	<string.h>	334
7.21	<sys/mman.h>	335
7.22	<sys/stat.h>	336
7.23	<sys/utsname.h>	337
7.24	<time.h>	338
7.25	<trace.h>	339
7.26	<unistd.h>	341
7.27	<utime.h>	342
<b>8</b>	<b>POSIX PSE53 - Dedicated</b>	<b>343</b>
8.1	Summary	344
8.2	<aio.h>	345
8.3	<arpa/inet.h>	346
8.4	<assert.h>	347
8.5	<complex.h>	348
8.6	<ctype.h>	350
8.7	<dirent.h>	351
8.8	<errno.h>	352
8.9	<fcntl.h>	353
8.10	<fenv.h>	354
8.11	<inttypes.h>	355
8.12	<locale.h>	356
8.13	<math.h>	357
8.14	<mqueue.h>	363
8.15	<net/if.h>	364
8.16	<netdb.h>	365
8.17	<pthread.h>	366
8.18	<sched.h>	369
8.19	<semaphore.h>	370
8.20	<setjmp.h>	371



8.21	<signal.h>	372
8.22	<spawn.h>	373
8.23	<stdarg.h>	374
8.24	<stdio.h>	375
8.25	<stdlib.h>	377
8.26	<string.h>	379
8.27	<sys/mman.h>	380
8.28	<sys/select.h>	381
8.29	<sys/socket.h>	382
8.30	<sys/stat.h>	383
8.31	<sys/time.h>	384
8.32	<sys/utsname.h>	385
8.33	<sys/wait.h>	386
8.34	<time.h>	387
8.35	<trace.h>	388
8.36	<unistd.h>	390
8.37	<utime.h>	392

<b>9</b>	<b>POSIX PSE54 - Multipurpose</b>	<b>393</b>
9.1	Summary	394
9.2	<aio.h>	395
9.3	<arpa/inet.h>	396
9.4	<assert.h>	397
9.5	<complex.h>	398
9.6	<ctype.h>	400
9.7	<dirent.h>	401
9.8	<dlfcn.h>	402
9.9	<errno.h>	403
9.10	<fcntl.h>	404
9.11	<fenv.h>	405
9.12	<fnmatch.h>	406
9.13	<glob.h>	407
9.14	<grp.h>	408
9.15	<inttypes.h>	409
9.16	<locale.h>	410
9.17	<math.h>	411
9.18	<mqueue.h>	417
9.19	<net/if.h>	418
9.20	<netdb.h>	419
9.21	<pthread.h>	420
9.22	<pwd.h>	423
9.23	<regex.h>	424
9.24	<sched.h>	425
9.25	<semaphore.h>	426
9.26	<setjmp.h>	427
9.27	<signal.h>	428
9.28	<spawn.h>	429
9.29	<stdarg.h>	430
9.30	<stdio.h>	431
9.31	<stdlib.h>	433
9.32	<string.h>	435

9.33	<sys/mman.h>	436
9.34	<sys/select.h>	437
9.35	<sys/socket.h>	438
9.36	<sys/stat.h>	439
9.37	<sys/time.h>	440
9.38	<sys/utsname.h>	441
9.39	<sys/wait.h>	442
9.40	<syslog.h>	443
9.41	<termios.h>	444
9.42	<time.h>	445
9.43	<trace.h>	446
9.44	<unistd.h>	448
9.45	<utime.h>	450
9.46	<wchar.h>	451
9.47	<wctype.h>	453
9.48	<wordexp.h>	454
<b>10</b>	<b>C99 Standard Library</b>	<b>455</b>
10.1	Summary	456
10.2	<assert.h>	457
10.3	<complex.h>	458
10.4	<ctype.h>	460
10.5	<errno.h>	461
10.6	<fenv.h>	462
10.7	<inttypes.h>	463
10.8	<locale.h>	464
10.9	<math.h>	465
10.10	<setjmp.h>	471
10.11	<signal.h>	472
10.12	<stdarg.h>	473
10.13	<stddef.h>	474
10.14	<stdio.h>	475
10.15	<stdlib.h>	477
10.16	<string.h>	479
10.17	<time.h>	480
10.18	<wchar.h>	481
10.19	<wctype.h>	483
<b>11</b>	<b>C11 Standard Library</b>	<b>485</b>
11.1	Summary	486
11.2	<assert.h>	487
11.3	<complex.h>	488
11.4	<ctype.h>	490
11.5	<errno.h>	491
11.6	<fenv.h>	492
11.7	<inttypes.h>	493
11.8	<locale.h>	494
11.9	<math.h>	495
11.10	<setjmp.h>	501
11.11	<signal.h>	502
11.12	<stdarg.h>	503
11.13	<stddef.h>	504

11.14	<stdio.h>	505
11.15	<stdlib.h>	507
11.16	<string.h>	509
11.17	<threads.h>	510
11.18	<time.h>	511
11.19	<wchar.h>	512
11.20	<wctype.h>	514
<b>12</b>	<b>FACE 2.1 Security</b>	<b>515</b>
12.1	Summary	516
12.2	<arpa/inet.h>	517
12.3	<ctype.h>	518
12.4	<devctl.h>	519
12.5	<errno.h>	520
12.6	<math.h>	521
12.7	<netdb.h>	522
12.8	<pthread.h>	523
12.9	<sched.h>	525
12.10	<semaphore.h>	526
12.11	<signal.h>	527
12.12	<stdlib.h>	528
12.13	<string.h>	529
12.14	<sys/mman.h>	530
12.15	<sys/socket.h>	531
12.16	<sys/stat.h>	532
12.17	<time.h>	533
12.18	<unistd.h>	534
<b>13</b>	<b>FACE 2.1 Safety Base</b>	<b>535</b>
13.1	Summary	536
13.2	<arpa/inet.h>	537
13.3	<ctype.h>	538
13.4	<devctl.h>	539
13.5	<dirent.h>	540
13.6	<errno.h>	541
13.7	<fcntl.h>	542
13.8	<math.h>	543
13.9	<mqueue.h>	544
13.10	<netdb.h>	545
13.11	<pthread.h>	546
13.12	<sched.h>	548
13.13	<semaphore.h>	549
13.14	<signal.h>	550
13.15	<stdio.h>	551
13.16	<stdlib.h>	552
13.17	<string.h>	553
13.18	<sys/mman.h>	554
13.19	<sys/select.h>	555
13.20	<sys/socket.h>	556
13.21	<sys/stat.h>	557
13.22	<time.h>	558
13.23	<unistd.h>	559

<b>14 FACE 2.1 Safety Extended</b>	<b>561</b>
14.1 Summary . . . . .	562
14.2 <arpa/inet.h> . . . . .	563
14.3 <ctype.h> . . . . .	564
14.4 <devctl.h> . . . . .	565
14.5 <dirent.h> . . . . .	566
14.6 <errno.h> . . . . .	567
14.7 <fcntl.h> . . . . .	568
14.8 <math.h> . . . . .	569
14.9 <mqueue.h> . . . . .	570
14.10<netdb.h> . . . . .	571
14.11<pthread.h> . . . . .	572
14.12<sched.h> . . . . .	574
14.13<semaphore.h> . . . . .	575
14.14<setjmp.h> . . . . .	576
14.15<signal.h> . . . . .	577
14.16<spawn.h> . . . . .	578
14.17<stdarg.h> . . . . .	579
14.18<stdio.h> . . . . .	580
14.19<stdlib.h> . . . . .	581
14.20<string.h> . . . . .	582
14.21<sys/mman.h> . . . . .	583
14.22<sys/select.h> . . . . .	584
14.23<sys/socket.h> . . . . .	585
14.24<sys/stat.h> . . . . .	586
14.25<sys/time.h> . . . . .	587
14.26<sys/utsname.h> . . . . .	588
14.27<sys/wait.h> . . . . .	589
14.28<time.h> . . . . .	590
14.29<unistd.h> . . . . .	591
<b>15 FACE 2.1 General Purpose</b>	<b>593</b>
15.1 Summary . . . . .	594
15.2 <aio.h> . . . . .	595
15.3 <arpa/inet.h> . . . . .	596
15.4 <assert.h> . . . . .	597
15.5 <complex.h> . . . . .	598
15.6 <ctype.h> . . . . .	600
15.7 <devctl.h> . . . . .	601
15.8 <dirent.h> . . . . .	602
15.9 <errno.h> . . . . .	603
15.10<fcntl.h> . . . . .	604
15.11<fenv.h> . . . . .	605
15.12<inttypes.h> . . . . .	606
15.13<locale.h> . . . . .	607
15.14<math.h> . . . . .	608
15.15<mqueue.h> . . . . .	614
15.16<net/if.h> . . . . .	615
15.17<netdb.h> . . . . .	616
15.18<pthread.h> . . . . .	617
15.19<sched.h> . . . . .	620

15.20	<semaphore.h>	621
15.21	<setjmp.h>	622
15.22	<signal.h>	623
15.23	<spawn.h>	624
15.24	<stdarg.h>	625
15.25	<stdio.h>	626
15.26	<stdlib.h>	628
15.27	<string.h>	630
15.28	<sys/mman.h>	631
15.29	<sys/select.h>	632
15.30	<sys/socket.h>	633
15.31	<sys/stat.h>	634
15.32	<sys/time.h>	635
15.33	<sys/utsname.h>	636
15.34	<sys/wait.h>	637
15.35	<time.h>	638
15.36	<unistd.h>	639
15.37	<wchar.h>	641
15.38	<wctype.h>	643
<b>16</b>	<b>FACE 3.0 Security</b>	<b>645</b>
16.1	Summary	646
16.2	<arpa/inet.h>	647
16.3	<ctype.h>	648
16.4	<devctl.h>	649
16.5	<errno.h>	650
16.6	<math.h>	651
16.7	<netdb.h>	652
16.8	<pthread.h>	653
16.9	<sched.h>	655
16.10	<semaphore.h>	656
16.11	<signal.h>	657
16.12	<stdlib.h>	658
16.13	<string.h>	659
16.14	<sys/mman.h>	660
16.15	<sys/socket.h>	661
16.16	<sys/stat.h>	662
16.17	<time.h>	663
16.18	<unistd.h>	664
<b>17</b>	<b>FACE 3.0 Safety Base</b>	<b>665</b>
17.1	Summary	666
17.2	<arpa/inet.h>	667
17.3	<ctype.h>	668
17.4	<devctl.h>	669
17.5	<dirent.h>	670
17.6	<errno.h>	671
17.7	<fcntl.h>	672
17.8	<math.h>	673
17.9	<mq.h>	674
17.10	<netdb.h>	675
17.11	<pthread.h>	676

17.12	<sched.h>	678
17.13	<semaphore.h>	679
17.14	<signal.h>	680
17.15	<stdio.h>	681
17.16	<stdlib.h>	682
17.17	<string.h>	683
17.18	<sys/mman.h>	684
17.19	<sys/select.h>	685
17.20	<sys/socket.h>	686
17.21	<sys/stat.h>	687
17.22	<time.h>	688
17.23	<unistd.h>	689
<b>18</b>	<b>FACE 3.0 Safety Extended</b>	<b>691</b>
18.1	Summary	692
18.2	<arpa/inet.h>	693
18.3	<ctype.h>	694
18.4	<devctl.h>	695
18.5	<dirent.h>	696
18.6	<errno.h>	697
18.7	<fcntl.h>	698
18.8	<math.h>	699
18.9	<mqueue.h>	700
18.10	<netdb.h>	701
18.11	<pthread.h>	702
18.12	<sched.h>	704
18.13	<semaphore.h>	705
18.14	<setjmp.h>	706
18.15	<signal.h>	707
18.16	<spawn.h>	708
18.17	<stdarg.h>	709
18.18	<stdio.h>	710
18.19	<stdlib.h>	711
18.20	<string.h>	712
18.21	<sys/mman.h>	713
18.22	<sys/select.h>	714
18.23	<sys/socket.h>	715
18.24	<sys/stat.h>	716
18.25	<sys/time.h>	717
18.26	<sys/utsname.h>	718
18.27	<sys/wait.h>	719
18.28	<time.h>	720
18.29	<unistd.h>	721
<b>19</b>	<b>FACE 3.0 General Purpose</b>	<b>723</b>
19.1	Summary	724
19.2	<aio.h>	725
19.3	<arpa/inet.h>	726
19.4	<assert.h>	727
19.5	<complex.h>	728
19.6	<ctype.h>	730
19.7	<devctl.h>	731

19.8	<dirent.h>	732
19.9	<errno.h>	733
19.10	<fcntl.h>	734
19.11	<fcntl.h>	735
19.12	<inttypes.h>	736
19.13	<locale.h>	737
19.14	<math.h>	738
19.15	<mqueue.h>	744
19.16	<net/if.h>	745
19.17	<netdb.h>	746
19.18	<pthread.h>	747
19.19	<sched.h>	750
19.20	<semaphore.h>	751
19.21	<setjmp.h>	752
19.22	<signal.h>	753
19.23	<spawn.h>	754
19.24	<stdarg.h>	755
19.25	<stdio.h>	756
19.26	<stdlib.h>	758
19.27	<string.h>	760
19.28	<sys/mman.h>	761
19.29	<sys/select.h>	762
19.30	<sys/socket.h>	763
19.31	<sys/stat.h>	764
19.32	<sys/time.h>	765
19.33	<sys/utsname.h>	766
19.34	<sys/wait.h>	767
19.35	<time.h>	768
19.36	<unistd.h>	769
19.37	<wchar.h>	771
19.38	<wctype.h>	772
<b>20</b>	<b>SCA 2.2.2 AEP</b>	<b>773</b>
20.1	Summary	774
20.2	<ctype.h>	775
20.3	<dirent.h>	776
20.4	<fcntl.h>	777
20.5	<locale.h>	778
20.6	<math.h>	779
20.7	<pthread.h>	780
20.8	<semaphore.h>	783
20.9	<setjmp.h>	784
20.10	<signal.h>	785
20.11	<stdio.h>	786
20.12	<stdlib.h>	788
20.13	<string.h>	789
20.14	<sys/stat.h>	790
20.15	<time.h>	791
20.16	<unistd.h>	792
20.17	<utime.h>	793
<b>21</b>	<b>SCA 4.1 Ultra Lightweight AEP</b>	<b>795</b>

21.1	Summary	796
21.2	<math.h>	797
21.3	<mqueue.h>	798
21.4	<pthread.h>	799
21.5	<semaphore.h>	800
21.6	<time.h>	801
<b>22</b>	<b>SCA 4.1 Lightweight AEP</b>	<b>803</b>
22.1	Summary	804
22.2	<ctype.h>	805
22.3	<fcntl.h>	806
22.4	<math.h>	807
22.5	<mqueue.h>	808
22.6	<pthread.h>	809
22.7	<semaphore.h>	810
22.8	<stdio.h>	811
22.9	<stdlib.h>	812
22.10	<string.h>	813
22.11	<time.h>	814
22.12	<unistd.h>	815
<b>23</b>	<b>SCA 4.1 [Full] AEP</b>	<b>817</b>
23.1	Summary	818
23.2	<arpa/inet.h>	819
23.3	<ctype.h>	820
23.4	<dirent.h>	821
23.5	<errno.h>	822
23.6	<fcntl.h>	823
23.7	<math.h>	824
23.8	<mqueue.h>	825
23.9	<pthread.h>	826
23.10	<semaphore.h>	828
23.11	<signal.h>	829
23.12	<stdarg.h>	830
23.13	<stdio.h>	831
23.14	<stdlib.h>	833
23.15	<string.h>	834
23.16	<sys/select.h>	835
23.17	<sys/socket.h>	836
23.18	<sys/stat.h>	837
23.19	<time.h>	838
23.20	<unistd.h>	839
<b>24</b>	<b>Glossary</b>	<b>841</b>
	<b>Index</b>	<b>843</b>



## Copyrights and License

© 2017 Chris Johns

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

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

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

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

## RTEMS Online Resources

Home	<a href="https://www.rtems.org">https://www.rtems.org</a>
Documentation	<a href="https://docs.rtems.org">https://docs.rtems.org</a>
Mailing Lists	<a href="https://lists.rtems.org">https://lists.rtems.org</a>
Bug Reporting	<a href="https://devel.rtems.org/wiki/Developer/Bug_Reporting">https://devel.rtems.org/wiki/Developer/Bug_Reporting</a>
Git Repositories	<a href="https://git.rtems.org">https://git.rtems.org</a>
Developers	<a href="https://devel.rtems.org">https://devel.rtems.org</a>



## PREFACE

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

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

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

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

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



# STANDARDS

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

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

The IEEE Standard 1003.1 is the POSIX standard. Specifically, IEEE Standard 1003.1-2008 is the 2003 edition of the POSIX standard and IEEE Standard 1003.1-2008 is the 2008 edition. The 2008 is an update from the 2003 edition. Each edition of the POSIX standard tends to add some methods, deprecate some methods, and obsolete (e.g. remove) other methods.

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

The Open Group FACE Consortium (<https://www.opengroup.org/face>) has defined four POSIX profiles targeting 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 are as follows:

- Security - 163 APIs, single process, no FILE \*
- Safety Basic - 246 APIs, single process, some FILE \*
- Safety Extended - 335 APIs, multi-process, more FILE \*
- General Purpose - 812 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()` methods.

RTEMS provides all of the methods required by the FACE Safety BASE profile and all of the methods in the Safety Extended profile which do not require multiple processes. Similarly, RTEMS provides most of the methods in the General Purpose profile which do not require multiple processes.

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 methods required by the SCA POSIX profiles.

## RTEMS COMPLETE PROFILE

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

## 3.1 Summary

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

Supported	999
ENOSYS	19
Not supported	216



## 3.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

### 3.3 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_addr()
- inet\_ntoa()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

### 3.4 <assert.h>

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

- assert()

## 3.5 <complex.h>

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

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

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

## 3.6 <ctype.h>

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

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

### 3.7 <devctl.h>

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

- posix\_devctl()

## 3.8 <dirent.h>

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

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

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

- `dirfd()`



### 3.9 <dlfcn.h>

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

- dlclose()
- dlerror()
- dlopen()
- dlsym()

### 3.10 <errno.h>

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

- errno

### 3.11 <fcntl.h>

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

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

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

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

## 3.12 <fenv.h>

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

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

### 3.13 <fmtmsg.h>

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

- `fmtmsg()`

### 3.14 <fnmatch.h>

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

- fnmatch()

### 3.15 <ftw.h>

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

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

### 3.16 <glob.h>

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

- glob()
- globfree()



### 3.17 <grp.h>

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

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

### 3.18 <iconv.h>

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

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

### 3.19 `<inttypes.h>`

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

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

## 3.20 <langinfo.h>

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

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

### 3.21 <libgen.h>

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

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

## 3.22 <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()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalb()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()

- `tgammaf()`
- `tgamma()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `j0()`
- `j1()`
- `jn()`
- `nexttowardf()`
- `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 not 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()`
- `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 <stddef.h>

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

- `offsetof()`

## 3.42 <stdio.h>

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

- clearerr()
- ctermid()
- dprintf()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgetpos()
- fgets()
- fileno()
- flockfile()
- fmemopen()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- 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()`
- `vsprintf()`
- `vsscanf()`

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

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

### 3.43 <stdlib.h>

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

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

- `mbtowc()`
- `mkdtemp()`
- `mkstemp()`
- `rand48()`
- `rand48()`
- `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()`

## 3.44 <string.h>

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

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



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

### 3.45 <strings.h>

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

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

### 3.46 <stropts.h>

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

- ioctl()

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

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

### 3.47 <sys/ipc.h>

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

- ftok()

### 3.48 <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()

### 3.49 <sys/msg.h>

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

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

### 3.50 <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()

### 3.51 <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()



### 3.52 <sys/sem.h>

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

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

### 3.53 <sys/shm.h>

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

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

### 3.54 <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:

- sockatmark()

### 3.55 <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()

### 3.56 <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()

### 3.57 <sys/time.h>

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

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

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

- `getitimer()`
- `setitimer()`

### 3.58 <sys/uio.h>

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

- readv()
- writev()

### 3.59 <sys/utsname.h>

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

- `uname()`



### 3.60 <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()

### 3.61 <syslog.h>

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

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

## 3.62 <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()

### 3.63 <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()`

## 3.64 `<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

### 3.65 <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()`



### 3.66 <ulimit.h>

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

- ulimit()

## 3.67 <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()`

### 3.68 <utime.h>

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

- `utime()`

### 3.69 <utmpx.h>

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

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

### 3.70 `<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()`
- `mbsnrto wcs()`
- `mbsrtowcs()`
- `open_wmemstream()`
- `putwc()`
- `putwchar()`
- `swprintf()`
- `swscanf()`
- `ungetwc()`
- `vwprintf()`
- `vwscanf()`
- `vswprintf()`
- `vswscanf()`
- `wprintf()`
- `wscanf()`
- `wcpcpy()`
- `wcpncpy()`
- `wcrtomb()`
- `wscasecmp()`
- `wscasecmp_l()`

- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wsdup()`
- `wcsftime()`
- `wcslen()`
- `wscnsecmp()`
- `wscncat()`
- `wscncmp()`
- `wscncpy()`
- `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()`

### 3.71 <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()`

### 3.72 <wordexp.h>

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

- wordexp()
- wordfree()

## POSIX-2008

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

## 4.1 Summary

The follow table summarizes alignment with the POSIX-2008 standard:

Supported	959
ENOSYS	19
Not supported	216

## 4.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 4.3 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_addr()
- inet\_ntoa()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()



## 4.4 <assert.h>

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

- `assert()`

## 4.5 <complex.h>

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

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

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

## 4.6 <ctype.h>

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

- `_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()`

## 4.7 <dirent.h>

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

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

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

- `dirfd()`

## 4.8 <dlfcn.h>

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

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

## 4.9 <errno.h>

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

- errno

## 4.10 <fcntl.h>

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

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

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

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



## 4.11 <fenv.h>

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

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

## 4.12 <fmtmsg.h>

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

- `fmtmsg()`

## 4.13 <fnmatch.h>

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

- fnmatch()

## 4.14 <ftw.h>

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

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

## 4.15 <glob.h>

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

- glob()
- globfree()

## 4.16 <grp.h>

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

- endgrent()
- getgrent()
- getgrgid()
- getgrgid\_r()
- getgrnam()
- getgrnam\_r()
- setgrent()

## 4.17 <iconv.h>

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

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

## 4.18 <inttypes.h>

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

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



## 4.19 <langinfo.h>

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

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

## 4.20 <libgen.h>

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

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

## 4.21 <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()`
- `asinhl()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()



- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `j0()`
- `j1()`
- `jn()`
- `nexttowardf()`
- `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 <queue.h>

The following methods and variables in <queue.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 not 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\_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()

## 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()

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

- \_longjmp()
- \_setjmp()

## 4.37 <signal.h>

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

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

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

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

## 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 <stddef.h>

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

- `offsetof()`

## 4.41 <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()`

## 4.42 <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()`

## 4.43 <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()`

## 4.44 <strings.h>

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

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

## 4.45 <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()

## 4.46 <sys/ipc.h>

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

- ftok()



## 4.47 <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()`

## 4.48 <sys/msg.h>

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

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

## 4.49 <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()

## 4.50 <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()

## 4.51 <sys/sem.h>

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

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

## 4.52 <sys/shm.h>

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

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

## 4.53 <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:

- sockatmark()

## 4.54 <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()



## 4.55 <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()

## 4.56 <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()`

## 4.57 <sys/uio.h>

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

- readv()
- writev()

## 4.58 <sys/utsname.h>

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

- `uname()`

## 4.59 <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()

## 4.60 <syslog.h>

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

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

## 4.61 <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()

## 4.62 <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

## 4.63 <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()`

## 4.64 <ulimit.h>

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

- `ulimit()`

## 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()`
- `fchmodat()`
- `fexecve()`
- `gethostid()`
- `getpgid()`
- `getsid()`
- `linkat()`
- `lockf()`
- `nice()`
- `readlinkat()`
- `setpgrp()`
- `setregid()`
- `setreuid()`
- `symlinkat()`
- `unlinkat()`

## 4.66 <utime.h>

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

- `utime()`



## 4.67 <utmpx.h>

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

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

## 4.68 <wchar.h>

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

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

- `wscat()`
- `wchr()`
- `wscmp()`
- `wscoll()`
- `wscoll_l()`
- `wscopy()`
- `wscspn()`
- `wsdup()`
- `wcsftime()`
- `wcslen()`
- `wscnsecmp()`
- `wscncat()`
- `wscncmp()`
- `wscncpy()`
- `wcsnlen()`
- `wcsnrtoombs()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcswidth()`
- `wcsxfrm()`
- `wcsxfrm_l()`
- `wctob()`
- `wcwidth()`

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

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

- `wcsncasemcp_l()`

## 4.69 <wctype.h>

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

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

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

## 4.70 <wordexp.h>

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

- wordexp()
- wordfree()





## POSIX-2003

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

## 5.1 Summary

The follow table summarizes alignment with the POSIX-2003 standard:

Supported	906
ENOSYS	19
Not supported	190

## 5.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 5.3 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_addr()
- inet\_ntoa()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 5.4 <assert.h>

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

- `assert()`

## 5.5 <complex.h>

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

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

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

## 5.6 <ctype.h>

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

- `_tolower()`
- `_toupper()`
- `isalnum()`
- `isalpha()`
- `isascii()`
- `isblank()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `islower_l()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `toascii()`
- `tolower()`
- `toupper()`



## 5.7 <dirent.h>

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

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

## 5.8 <dlfcn.h>

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

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

## 5.9 <errno.h>

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

- errno

## 5.10 <fcntl.h>

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

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

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

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

## 5.11 <fenv.h>

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

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

## 5.12 <fmtmsg.h>

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

- `fmtmsg()`

## 5.13 <fnmatch.h>

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

- fnmatch()

## 5.14 <ftw.h>

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

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



## 5.15 <glob.h>

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

- glob()
- globfree()

## 5.16 <grp.h>

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

- endgrent()
- getgrent()
- getgrgid()
- getgrgid\_r()
- getgrnam()
- getgrnam\_r()
- setgrent()

## 5.17 <iconv.h>

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

- `iconv()`
- `iconv_close()`
- `iconv_open()`

## 5.18 <inttypes.h>

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

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

## 5.19 <langinfo.h>

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

- nl\_langinfo()

## 5.20 <libgen.h>

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

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

## 5.21 <locale.h>

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

- localeconv()
- setlocale()

## 5.22 <math.h>

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

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `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()`
- `maf()`
- `mal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `nexttowardl()`
- `pow()`
- `powf()`
- `powl()`
- `remainder()`

- remainderf()
- remainderl()
- remquo()
- remquof()
- remquol()
- rint()
- rintf()
- rintl()
- round()
- roundf()
- roundl()
- scalb()
- scalbln()
- scalblnf()
- scalblnl()
- scalbn()
- scalbnf()
- scalbnl()
- sin()
- sinf()
- sinh()
- sinhf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()

- `tgammaf()`
- `tgamma()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `j0()`
- `j1()`
- `jn()`
- `nexttowardf()`
- `signbit()`
- `signgam`
- `y0()`
- `y1()`
- `yn()`

## 5.23 <monetary.h>

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

- strfmon()

## 5.24 <queue.h>

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

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

## 5.25 <ndbm.h>

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

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



## 5.26 <net/if.h>

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

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

## 5.27 <netdb.h>

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

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

## 5.28 <nl\_types.h>

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

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

## 5.29 <poll.h>

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

- poll()

## 5.30 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_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()`

## 5.31 <pwd.h>

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

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



## 5.32 <regex.h>

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

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

### 5.33 <sched.h>

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

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

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

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

## 5.34 <search.h>

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

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

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

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

## 5.35 <semaphore.h>

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

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

## 5.36 <setjmp.h>

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

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

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

- \_longjmp()
- \_setjmp()

## 5.37 <signal.h>

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

- `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()`

## 5.38 <spawn.h>

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

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

## 5.39 <stdarg.h>

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

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



## 5.40 <stddef.h>

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

- `offsetof()`

## 5.41 <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()
- fscanff()
- 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()`

## 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()
- ecvt()
- erand48()
- exit()
- fcvt()
- free()
- gcvt()
- getenv()
- getsubopt()
- jrand48()
- l64a()
- labs()
- lcong48()
- ldiv()
- llabs()
- lldiv()
- lrand48()
- malloc()
- mblen()
- mbstowcs()

- `mbtowc()`
- `mkstemp()`
- `rand48()`
- `rand48()`
- `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()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcoll()`
- `strcpy()`
- `strcspn()`
- `strdup()`
- `strerror()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`
- `strxfrm()`

## 5.44 <strings.h>

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

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



## 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:

- sockatmark()

## 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()

## 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()
- 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()`
- `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()`

## 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()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcsncpy()
- wcsncpy()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcsnlen()
- wcsnrtombs()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcswidth()
- wcsxfrm()
- wctob()
- wcwidth()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

## 5.69 <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()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`

## 5.70 <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.

## 6.1 Summary

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

Supported	270
ENOSYS	2
Not supported	12

## 6.2 `<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()`

## 6.3 <errno.h>

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

- errno

## 6.4 <fcntl.h>

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

- open()

## 6.5 <fenv.h>

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

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

## 6.6 `<inttypes.h>`

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

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

## 6.7 <locale.h>

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

- localeconv()
- setlocale()



## 6.8 <pthread.h>

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

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

- `pthread_condattr_setclock()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

- pthread\_testcancel()

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

- pthread\_atfork()
- pthread\_getcpuclockid()

## 6.9 <sched.h>

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

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

## 6.10 <semaphore.h>

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

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

## 6.11 <setjmp.h>

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

- longjmp()
- setjmp()

## 6.12 <signal.h>

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

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

## 6.13 <stdarg.h>

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

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



## 6.14 <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()
- fscanf()
- 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()

## 6.15 <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()
- strtodf()
- strtol()
- strtold()
- strtoll()
- strtoul()
- strtoull()
- unsetenv()

## 6.16 <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()

## 6.17 <sys/mman.h>

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

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

## 6.18 <sys/utsname.h>

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

- `uname()`

## 6.19 <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()

## 6.20 <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.

## 7.1 Summary

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

Supported	553
ENOSYS	2
Not supported	73

## 7.2 <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()`

## 7.3 `<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()`

## 7.4 <dirent.h>

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

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

## 7.5 <errno.h>

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

- errno

## 7.6 <fcntl.h>

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

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



## 7.7 `<fenv.h>`

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

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

## 7.8 <inttypes.h>

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

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

## 7.9 <locale.h>

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

- localeconv()
- setlocale()

## 7.10 <math.h>

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

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `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()`
- `maf()`
- `fmal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()



- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 7.11 <mqqueue.h>

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

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

## 7.12 <pthread.h>

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

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

- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_detach()
- pthread\_equal()
- pthread\_exit()
- pthread\_getconcurrency()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_join()
- pthread\_key\_create()
- pthread\_key\_delete()
- pthread\_mutex\_destroy()
- pthread\_mutex\_getprioceiling()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_setprioceiling()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_gettype()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_mutexattr\_settype()
- pthread\_once()
- pthread\_self()
- pthread\_setcancelstate()
- pthread\_setcanceltype()
- pthread\_setconcurrency()
- pthread\_setschedparam()
- pthread\_setschedprio()
- pthread\_setspecific()

- `pthread_testcancel()`

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

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

## 7.13 <sched.h>

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

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

## 7.14 <semaphore.h>

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

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

## 7.15 <setjmp.h>

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

- longjmp()
- setjmp()



## 7.16 <signal.h>

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

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

## 7.17 <stdarg.h>

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

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

## 7.18 `<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()`
- `vsprintf()`
- `vsscanf()`

## 7.19 `<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()`

## 7.20 <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()

## 7.21 <sys/mman.h>

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

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

## 7.22 <sys/stat.h>

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

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



## 7.23 <sys/utsname.h>

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

- `uname()`

## 7.24 <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()

## 7.25 <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_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()`

## 7.26 <unistd.h>

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

- access()
- alarm()
- chdir()
- close()
- dup()
- dup2()
- environ
- fdatasync()
- 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()

## 7.27 <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.

## 8.1 Summary

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

Supported	639
ENOSYS	16
Not supported	96



## 8.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 8.3 <arpa/inet.h>

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

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

## 8.4 <assert.h>

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

- `assert()`

## 8.5 <complex.h>

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

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

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

## 8.6 <ctype.h>

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

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 8.7 <dirent.h>

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

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

## 8.8 <errno.h>

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

- errno



## 8.9 <fcntl.h>

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

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

## 8.10 <fenv.h>

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

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

## 8.11 <inttypes.h>

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

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

## 8.12 <locale.h>

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

- localeconv()
- setlocale()

## 8.13 <math.h>

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

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

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

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 8.14 <queue.h>

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

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

## 8.15 <net/if.h>

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

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

## 8.16 <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()`

## 8.17 <pthread.h>

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

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

- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

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

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



## 8.18 <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()

## 8.19 <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()`

## 8.20 <setjmp.h>

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

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

## 8.21 <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()

## 8.22 <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\_spawnnp()

## 8.23 <stdarg.h>

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

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

## 8.24 <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()



## 8.25 <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()

## 8.26 <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()

## 8.27 <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()`

## 8.28 <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()

## 8.29 <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:

- sockatmark()

## 8.30 <sys/stat.h>

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

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

## 8.31 <sys/time.h>

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

- times()
- utimes()



## 8.32 <sys/utsname.h>

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

- `uname()`

### 8.33 <sys/wait.h>

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

- wait()

## 8.34 <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()

## 8.35 <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()`

## 8.36 <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()`
- `setsid()`
- `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()`

## 8.37 <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.

## 9.1 Summary

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

Supported	783
ENOSYS	17
Not supported	106

## 9.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 9.3 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_addr()
- inet\_ntoa()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 9.4 <assert.h>

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

- `assert()`

## 9.5 <complex.h>

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

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

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

## 9.6 <ctype.h>

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

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()



## 9.7 <dirent.h>

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

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

## 9.8 <dlfcn.h>

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

- `dlclose()`
- `dlerror()`
- `dlopen()`
- `dlsym()`

## 9.9 <errno.h>

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

- errno

## 9.10 <fcntl.h>

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

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

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

- posix\_fadvise()
- posix\_fallocate()

## 9.11 <fenv.h>

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

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

## 9.12 <fnmatch.h>

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

- fnmatch()

## 9.13 <glob.h>

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

- glob()
- globfree()

## 9.14 <grp.h>

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

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



## 9.15 <inttypes.h>

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

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

## 9.16 <locale.h>

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

- localeconv()
- setlocale()

## 9.17 <math.h>

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

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinh1()`
- `asinl()`
- `atan()`
- `atan2()`
- `atan2f()`
- `atan2l()`
- `atanf()`
- `atanh()`
- `atanhf()`
- `atanhl()`
- `atanl()`
- `cbrt()`
- `cbrtf()`
- `cbrtl()`
- `ceil()`
- `ceilf()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`



## 9.18 <queue.h>

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

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

## 9.19 <net/if.h>

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

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

## 9.20 <netdb.h>

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

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

## 9.21 <pthread.h>

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

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

- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

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

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

## 9.22 <pwd.h>

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

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

## 9.23 <regex.h>

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

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



## 9.24 <sched.h>

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

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

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

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

## 9.25 <semaphore.h>

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

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

## 9.26 <setjmp.h>

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

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

## 9.27 <signal.h>

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

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

## 9.28 <spawn.h>

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

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

## 9.29 <stdarg.h>

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

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

## 9.30 <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()



## 9.31 <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()`

## 9.32 `<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()`

### 9.33 <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()

## 9.34 <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()

## 9.35 <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:

- sockatmark()

## 9.36 <sys/stat.h>

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

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

## 9.37 <sys/time.h>

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

- times()
- utimes()



## 9.38 <sys/utsname.h>

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

- `uname()`

## 9.39 <sys/wait.h>

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

- wait()

## 9.40 <syslog.h>

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

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

## 9.41 <termios.h>

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

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

## 9.42 <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()

## 9.43 <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()`

## 9.44 <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()
- setsid()
- 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()`

## 9.45 <utime.h>

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

- `utime()`

## 9.46 <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()
- wcrctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcsrbrk()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcsxfrm()
- wctob()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

## 9.47 <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()`

## 9.48 <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.

## 10.1 Summary

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

Supported	463
ENOSYS	0
Not supported	22



## 10.2 <assert.h>

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

- `assert()`

## 10.3 <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()`

## 10.4 <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()`

## 10.5 <errno.h>

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

- errno

## 10.6 <fenv.h>

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

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

## 10.7 <inttypes.h>

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

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

## 10.8 <locale.h>

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

- localeconv()
- setlocale()



## 10.9 <math.h>

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

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

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 10.10 <setjmp.h>

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

- longjmp()
- setjmp()

## 10.11 <signal.h>

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

- raise()
- signal()



## 10.12 `<stdarg.h>`

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

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

## 10.13 <stddef.h>

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

- `offsetof()`

## 10.14 <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()`
- `vsprintf()`
- `vsprintf()`
- `vsscanf()`

## 10.15 <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()`

## 10.16 <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()

## 10.17 <time.h>

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

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



## 10.18 <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()
- wcrctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcscpy()
- wcsncpy()

- wcsftime()
- wcslen()
- wcsncat()
- wcsncmp()
- wcsncpy()
- wcsrchr()
- wcsrchr()
- wcsrtombs()
- wcsspn()
- wcsstr()
- wcstod()
- wcstof()
- wcstok()
- wcstol()
- wcstold()
- wcstoll()
- wcstoul()
- wcstoull()
- wcsxfrm()
- wctob()
- wmemchr()
- wmemcmp()
- wmemcpy()
- wmemmove()
- wmemset()
- wprintf()
- wscanf()

## 10.19 `<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.

## 11.1 Summary

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

Supported	486
ENOSYS	0
Not supported	22

## 11.2 <assert.h>

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

- assert()

## 11.3 <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()`

## 11.4 <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()`

## 11.5 <errno.h>

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

- errno

## 11.6 <fenv.h>

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

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

## 11.7 `<inttypes.h>`

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

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

## 11.8 <locale.h>

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

- localeconv()
- setlocale()

## 11.9 <math.h>

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

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

- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 11.10 `<setjmp.h>`

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

- `longjmp()`
- `setjmp()`

## 11.11 <signal.h>

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

- raise()
- signal()

## 11.12 <stdarg.h>

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

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

## 11.13 <stddef.h>

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

- `offsetof()`



## 11.14 <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()`
- `vsprintf()`
- `vsprintf()`
- `vsscanf()`

## 11.15 <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()`

## 11.16 <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()

## 11.17 <threads.h>

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

- `call_once()`
- `cond_broadcast()`
- `cond_destroy()`
- `cond_init()`
- `cond_signal()`
- `cond_timedwait()`
- `cond_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()`

## 11.18 <time.h>

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

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

## 11.19 <wchar.h>

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

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wctomb()
- wcscat()
- wcschr()
- wcscmp()
- wcscoll()
- wcsncpy()
- wcsncpy()



- `wcsftime()`
- `wcslen()`
- `wcsncat()`
- `wcsncmp()`
- `wcsncpy()`
- `wcspbrk()`
- `wcsrchr()`
- `wcsrtombs()`
- `wcsspn()`
- `wcsstr()`
- `wcstod()`
- `wcstof()`
- `wcstok()`
- `wcstol()`
- `wcstold()`
- `wcstoll()`
- `wcstoul()`
- `wcstoull()`
- `wcsxfrm()`
- `wctob()`
- `wmemchr()`
- `wmemcmp()`
- `wmemcpy()`
- `wmemmove()`
- `wmemset()`
- `wprintf()`
- `wscanf()`

## 11.20 <wctype.h>

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

- iswalnum()
- iswalpha()
- iswblank()
- iswcntrl()
- iswctype()
- iswdigit()
- iswgraph()
- iswlower()
- iswprint()
- iswpunct()
- iswspace()
- iswupper()
- iswxdigit()
- towctrans()
- tolower()
- toupper()
- wctrans()
- wctype()

## FACE 2.1 SECURITY

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

## 12.1 Summary

The follow table summarizes alignment with the FACE 2.1 Security standard:

Supported	162
ENOSYS	1
Not supported	0

## 12.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 12.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 12.4 <devctl.h>

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

- posix\_devctl()

## 12.5 <errno.h>

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

- errno



## 12.6 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 12.7 <netdb.h>

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

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

## 12.8 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

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

- `pthread_getcpuclockid()`

## 12.9 <sched.h>

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

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

## 12.10 <semaphore.h>

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

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

## 12.11 <signal.h>

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

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 12.12 <stdlib.h>

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

- abs()
- atof()
- atoi()
- atol()
- div()
- labs()
- ldiv()
- malloc()
- rand\_r()
- strtod()
- strtol()
- strtoul()



## 12.13 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 12.14 <sys/mman.h>

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

- mmap()
- shm\_open()

## 12.15 <sys/socket.h>

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

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 12.16 <sys/stat.h>

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

- stat()

## 12.17 <time.h>

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

- clock\_getres()
- clock\_gettime()
- clock\_settime()
- nanosleep()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 12.18 <unistd.h>

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

- alarm()
- ftruncate()
- pause()

## FACE 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.

## 13.1 Summary

The follow table summarizes alignment with the FACE 2.1 Safety Base standard:

Supported	245
ENOSYS	1
Not supported	0



## 13.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 13.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 13.4 <devctl.h>

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

- posix\_devctl()

## 13.5 <dirent.h>

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

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

## 13.6 <errno.h>

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

- errno

## 13.7 <fcntl.h>

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

- creat()
- open()

## 13.8 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 13.9 <mqqueue.h>

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

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



## 13.10 `<netdb.h>`

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

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

## 13.11 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()
- pthread\_equal()
- pthread\_getconcurrency()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_key\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()

- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

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

- `pthread_getcpuclockid()`

## 13.12 <sched.h>

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

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

## 13.13 <semaphore.h>

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

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

## 13.14 <signal.h>

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

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 13.15 <stdio.h>

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

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 13.16 <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()



## 13.17 `<string.h>`

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

- `memchr()`
- `memcmp()`
- `memcpy()`
- `memmove()`
- `memset()`
- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strerror_r()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok_r()`

## 13.18 <sys/mman.h>

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

- mmap()
- shm\_open()

## 13.19 `<sys/select.h>`

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

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

## 13.20 <sys/socket.h>

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

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 13.21 <sys/stat.h>

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

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

## 13.22 <time.h>

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

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime\_r()
- localtime\_r()
- mktime()
- nanosleep()
- time()
- timer\_create()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()
- tzname
- tzset()

## 13.23 <unistd.h>

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

- access()
- alarm()
- chdir()
- close()
- fsync()
- ftruncate()
- getcwd()
- gethostname()
- link()
- lseek()
- pause()
- read()
- rmdir()
- unlink()
- write()





## FACE 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.

## 14.1 Summary

The follow table summarizes alignment with the FACE 2.1 Safety Extended standard:

Supported	315
ENOSYS	11
Not supported	9

## 14.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 14.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 14.4 <devctl.h>

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

- posix\_devctl()

## 14.5 <dirent.h>

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

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

## 14.6 `<errno.h>`

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

- `errno`

## 14.7 <fcntl.h>

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

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



## 14.8 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 14.9 <mqqueue.h>

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

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

## 14.10 <netdb.h>

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

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

## 14.11 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()

- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

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

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

## 14.12 <sched.h>

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

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

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

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

## 14.13 <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()`

## 14.14 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()



## 14.15 <signal.h>

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

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

## 14.16 <spawn.h>

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

- `posix_spawn()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`

## 14.17 <stdarg.h>

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

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

## 14.18 <stdio.h>

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

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

## 14.19 `<stdlib.h>`

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

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `realloc()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 14.20 <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()

## 14.21 <sys/mman.h>

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

- mmap()
- shm\_open()

## 14.22 <sys/select.h>

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

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



## 14.23 `<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()`

## 14.24 <sys/stat.h>

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

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

## 14.25 `<sys/time.h>`

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

- `times()`

## 14.26 <sys/utsname.h>

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

- `uname()`

## 14.27 `<sys/wait.h>`

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

- `waitpid()`

## 14.28 <time.h>

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

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

## 14.29 <unistd.h>

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

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup2()`
- `environ`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setuid()`
- `sleep()`
- `sysconf()`

- `unlink()`
- `write()`

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

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`



## FACE 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.

## 15.1 Summary

The follow table summarizes alignment with the FACE 2.1 General Purpose standard:

Supported	752
ENOSYS	14
Not supported	46

## 15.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 15.3 <arpa/inet.h>

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

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

## 15.4 <assert.h>

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

- assert()

## 15.5 <complex.h>

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

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

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

## 15.6 <ctype.h>

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

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()



## 15.7 <devctl.h>

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

- posix\_devctl()

## 15.8 <dirent.h>

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

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

## 15.9 `<errno.h>`

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

- `errno`

## 15.10 <fcntl.h>

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

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

## 15.11 <fenv.h>

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

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

## 15.12 <inttypes.h>

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

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

## 15.13 <locale.h>

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

- localeconv()
- setlocale()

## 15.14 <math.h>

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

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



- `cosh()`
- `coshf()`
- `coshl()`
- `cosl()`
- `erf()`
- `erfc()`
- `erfcf()`
- `erfcl()`
- `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()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 15.15 <mqqueue.h>

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

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

## 15.16 <net/if.h>

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

- `if_freenameindex()`
- `if_indextoname()`
- `if_nameindex()`
- `if_nametoindex()`

## 15.17 <netdb.h>

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

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



## 15.18 <pthread.h>

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

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

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

- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

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

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

## 15.19 <sched.h>

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

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

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

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

## 15.20 <semaphore.h>

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

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

## 15.21 <setjmp.h>

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

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

## 15.22 <signal.h>

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

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

## 15.23 <spawn.h>

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

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



## 15.24 `<stdarg.h>`

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

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

## 15.25 <stdio.h>

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

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

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

## 15.26 <stdlib.h>

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

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `atoll()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `llabs()`
- `lldiv()`
- `malloc()`
- `mblen()`
- `mbstowcs()`
- `mbtowc()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `setenv()`
- `srand()`
- `strtod()`
- `strtof()`
- `strtol()`
- `strtold()`

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

## 15.27 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 15.28 <sys/mman.h>

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

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

## 15.29 <sys/select.h>

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

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

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

- pselect()



## 15.30 `<sys/socket.h>`

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

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

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

- `socketatmark()`

## 15.31 <sys/stat.h>

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

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

## 15.32 `<sys/time.h>`

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

- `times()`

## 15.33 <sys/utsname.h>

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

- `uname()`

## 15.34 `<sys/wait.h>`

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

- `wait()`
- `waitpid()`

## 15.35 <time.h>

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

- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- 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()

## 15.36 <unistd.h>

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

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`

- `setegid()`
- `seteuid()`
- `setgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

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

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

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

- `confstr()`



## 15.37 <wchar.h>

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

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()
- putwc()
- putwchar()
- swprintf()
- swscanf()
- ungetwc()
- vfwprintf()
- vfwscanf()
- vswprintf()
- vswscanf()
- vwprintf()
- vwscanf()
- wcrctomb()
- 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()`

## 15.38 `<wctype.h>`

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

- `iswalnum()`
- `iswalpha()`
- `iswblank()`
- `iswcntrl()`
- `iswctype()`
- `iswdigit()`
- `iswgraph()`
- `iswlower()`
- `iswprint()`
- `iswpunct()`
- `iswspace()`
- `iswupper()`
- `iswxdigit()`
- `towctrans()`
- `towlower()`
- `towupper()`
- `wctrans()`
- `wctype()`



## FACE 3.0 SECURITY

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

## 16.1 Summary

The follow table summarizes alignment with the FACE 3.0 Security standard:

Supported	163
ENOSYS	1
Not supported	0

## 16.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 16.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`



## 16.4 <devctl.h>

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

- posix\_devctl()

## 16.5 <errno.h>

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

- errno

## 16.6 `<math.h>`

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 16.7 <netdb.h>

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

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

## 16.8 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_create()
- pthread\_equal()
- pthread\_getschedparam()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_timedlock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_destroy()
- pthread\_mutexattr\_getprioceiling()
- pthread\_mutexattr\_getprotocol()
- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_setprotocol()
- pthread\_once()
- pthread\_self()
- pthread\_setschedparam()
- pthread\_setschedprio()

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

- `pthread_getcpuclockid()`

## 16.9 `<sched.h>`

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

- `sched_get_priority_max()`
- `sched_get_priority_min()`
- `sched_yield()`

## 16.10 <semaphore.h>

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

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



## 16.11 <signal.h>

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

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()

## 16.12 <stdlib.h>

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

- abs()
- atof()
- atoi()
- atol()
- div()
- labs()
- ldiv()
- malloc()
- rand\_r()
- strtod()
- strtol()
- strtoul()

## 16.13 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 16.14 <sys/mman.h>

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

- mmap()
- shm\_open()

## 16.15 <sys/socket.h>

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

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 16.16 <sys/stat.h>

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

- stat()

## 16.17 `<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()`

## 16.18 <unistd.h>

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

- alarm()
- ftruncate()
- pause()



## FACE 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.

## 17.1 Summary

The follow table summarizes alignment with the FACE 3.0 Safety Base standard:

Supported	246
ENOSYS	1
Not supported	0

## 17.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 17.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 17.4 <devctl.h>

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

- posix\_devctl()

## 17.5 <dirent.h>

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

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

## 17.6 <errno.h>

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

- errno

## 17.7 <fcntl.h>

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

- creat()
- open()



## 17.8 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 17.9 <mqqueue.h>

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

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

## 17.10 <netdb.h>

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

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

## 17.11 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_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()`

## 17.12 <sched.h>

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

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

## 17.13 <semaphore.h>

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

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

## 17.14 <signal.h>

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

- pthread\_sigmask()
- sigaction()
- sigaddset()
- sigdelset()
- sigemptyset()
- sigfillset()
- sigismember()
- sigpending()
- sigqueue()
- sigsuspend()
- sigtimedwait()
- sigwait()
- sigwaitinfo()



## 17.15 <stdio.h>

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

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- remove()
- rename()
- snprintf()

## 17.16 <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()

## 17.17 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcpy()
- strcspn()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok\_r()

## 17.18 <sys/mman.h>

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

- mmap()
- shm\_open()

## 17.19 `<sys/select.h>`

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

- `FD_CLR()`
- `FD_ISSET()`
- `FD_SET()`
- `FD_ZERO()`
- `select()`

## 17.20 <sys/socket.h>

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

- bind()
- connect()
- getpeername()
- getsockname()
- getsockopt()
- recv()
- recvfrom()
- send()
- sendto()
- setsockopt()
- shutdown()
- socket()

## 17.21 <sys/stat.h>

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

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

## 17.22 <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()



## 17.23 <unistd.h>

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

- access()
- alarm()
- chdir()
- close()
- fsync()
- ftruncate()
- getcwd()
- gethostname()
- link()
- lseek()
- pause()
- read()
- rmdir()
- unlink()
- write()



## FACE 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.

## 18.1 Summary

The follow table summarizes alignment with the FACE 3.0 Safety Extended standard:

Supported	316
ENOSYS	11
Not supported	9

## 18.2 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 18.3 <ctype.h>

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

- `isalnum()`
- `isalpha()`
- `iscntrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`
- `tolower()`
- `toupper()`

## 18.4 <devctl.h>

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

- posix\_devctl()

## 18.5 <dirent.h>

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

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



## 18.6 `<errno.h>`

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

- `errno`

## 18.7 <fcntl.h>

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

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

## 18.8 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `isinf()`
- `isnan()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 18.9 <mqqueue.h>

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

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

## 18.10 <netdb.h>

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

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

## 18.11 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getguardsize()
- pthread\_attr\_getinheritsched()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getschedpolicy()
- pthread\_attr\_getscope()
- pthread\_attr\_getstack()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setdetachstate()
- pthread\_attr\_setguardsize()
- pthread\_attr\_setinheritsched()
- pthread\_attr\_setschedparam()
- pthread\_attr\_setschedpolicy()
- pthread\_attr\_setscope()
- pthread\_attr\_setstack()
- pthread\_attr\_setstacksize()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_getclock()
- pthread\_condattr\_init()
- pthread\_condattr\_setclock()
- pthread\_create()

- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getconcurrency()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`

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

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

## 18.12 <sched.h>

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

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

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

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



## 18.13 <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()`

## 18.14 <setjmp.h>

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

- siglongjmp()
- sigsetjmp()

## 18.15 <signal.h>

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

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

## 18.16 <spawn.h>

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

- `posix_spawn()`
- `posix_spawnattr_destroy()`
- `posix_spawnattr_getflags()`
- `posix_spawnattr_getsigdefault()`
- `posix_spawnattr_getsigmask()`
- `posix_spawnattr_init()`
- `posix_spawnattr_setflags()`
- `posix_spawnattr_setsigdefault()`
- `posix_spawnattr_setsigmask()`

## 18.17 `<stdarg.h>`

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

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

## 18.18 <stdio.h>

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

- clearerr()
- fclose()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- flockfile()
- fopen()
- fprintf()
- fread()
- freopen()
- fseek()
- fseeko()
- ftell()
- ftello()
- ftrylockfile()
- funlockfile()
- fwrite()
- remove()
- rename()
- snprintf()
- sscanf()
- vfprintf()
- vsnprintf()

## 18.19 <stdlib.h>

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

- `_Exit()`
- `abort()`
- `abs()`
- `atexit()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `div()`
- `exit()`
- `free()`
- `getenv()`
- `labs()`
- `ldiv()`
- `malloc()`
- `rand_r()`
- `realloc()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 18.20 <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()



## 18.21 <sys/mman.h>

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

- mmap()
- shm\_open()

## 18.22 <sys/select.h>

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

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

## 18.23 `<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()`

## 18.24 <sys/stat.h>

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

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

## 18.25 <sys/time.h>

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

- times()

## 18.26 <sys/utsname.h>

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

- `uname()`

## 18.27 `<sys/wait.h>`

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

- `waitpid()`

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



## 18.29 <unistd.h>

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

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup2()`
- `environ`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`
- `setegid()`
- `seteuid()`
- `setgid()`
- `setuid()`
- `sleep()`
- `sysconf()`

- `unlink()`
- `write()`

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

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

## FACE 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.

## 19.1 Summary

The follow table summarizes alignment with the FACE 3.0 General Purpose standard:

Supported	707
ENOSYS	14
Not supported	46

## 19.2 <aio.h>

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

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

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

- aio\_suspend()
- lio\_listio()

## 19.3 <arpa/inet.h>

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

- htonl()
- htons()
- inet\_addr()
- inet\_ntoa()
- inet\_ntop()
- inet\_pton()
- ntohl()
- ntohs()

## 19.4 <assert.h>

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

- `assert()`

## 19.5 <complex.h>

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

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



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

## 19.6 <ctype.h>

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

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 19.7 <devctl.h>

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

- posix\_devctl()

## 19.8 <dirent.h>

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

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

## 19.9 <errno.h>

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

- errno

## 19.10 <fcntl.h>

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

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

## 19.11 <fenv.h>

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

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

## 19.12 <inttypes.h>

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

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



## 19.13 <locale.h>

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

- localeconv()
- setlocale()

## 19.14 <math.h>

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

- `acos()`
- `acosf()`
- `acosh()`
- `acoshf()`
- `acoshl()`
- `acosl()`
- `asin()`
- `asinf()`
- `asinh()`
- `asinhf()`
- `asinhl()`
- `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()`
- `maf()`
- `mal()`
- `fmax()`
- `fmaxf()`
- `fmaxl()`

- `fmin()`
- `fminf()`
- `fminl()`
- `fmod()`
- `fmodf()`
- `fmodl()`
- `frexp()`
- `frexpf()`
- `frexpl()`
- `hypot()`
- `hypotf()`
- `hypotl()`
- `ilogb()`
- `ilogbf()`
- `ilogbl()`
- `isinf()`
- `isnan()`
- `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()`
- `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()
- sinhlf()
- sinhl()
- sinl()
- sqrt()
- sqrtf()
- sqrtl()
- tan()
- tanf()
- tanh()
- tanhf()
- tanhl()
- tanl()
- tgamma()
- tgammaf()

- `tgammal()`
- `trunc()`
- `truncf()`
- `truncl()`

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

- `fpclassify()`
- `isfinite()`
- `isgreater()`
- `isgreaterequal()`
- `isless()`
- `islessequal()`
- `islessgreater()`
- `isnormal()`
- `isunordered()`
- `nexttowardf()`
- `signbit()`

## 19.15 <mqqueue.h>

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

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



## 19.16 <net/if.h>

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

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

## 19.17 <netdb.h>

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

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

## 19.18 <pthread.h>

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

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

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

- `pthread_once()`
- `pthread_rwlock_destroy()`
- `pthread_rwlock_init()`
- `pthread_rwlock_rdlock()`
- `pthread_rwlock_timedrdlock()`
- `pthread_rwlock_timedwrlock()`
- `pthread_rwlock_tryrdlock()`
- `pthread_rwlock_trywrlock()`
- `pthread_rwlock_unlock()`
- `pthread_rwlock_wrlock()`
- `pthread_rwlockattr_destroy()`
- `pthread_rwlockattr_init()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`
- `pthread_setconcurrency()`
- `pthread_setschedparam()`
- `pthread_setschedprio()`
- `pthread_setspecific()`
- `pthread_testcancel()`

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

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

## 19.19 <sched.h>

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

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

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

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

## 19.20 `<semaphore.h>`

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

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

## 19.21 <setjmp.h>

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

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



## 19.22 <signal.h>

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

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

## 19.23 <spawn.h>

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

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

## 19.24 <stdarg.h>

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

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

## 19.25 <stdio.h>

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

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

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

## 19.26 <stdlib.h>

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

- \_Exit()
- abort()
- abs()
- atexit()
- atof()
- atoi()
- atol()
- atoll()
- bsearch()
- calloc()
- div()
- exit()
- free()
- getenv()
- labs()
- ldiv()
- llabs()
- lldiv()
- malloc()
- mblen()
- mbstowcs()
- mbtowc()
- qsort()
- rand()
- rand\_r()
- realloc()
- setenv()
- srand()
- strtod()
- strtof()
- strtol()
- strtold()

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

## 19.27 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strcat()
- strchr()
- strcmp()
- strcoll()
- strcpy()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()



## 19.28 `<sys/mman.h>`

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

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

## 19.29 <sys/select.h>

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

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

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

- pselect()

## 19.30 `<sys/socket.h>`

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

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

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

- `socketatmark()`

## 19.31 <sys/stat.h>

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

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

## 19.32 `<sys/time.h>`

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

- `times()`

## 19.33 <sys/utsname.h>

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

- `uname()`

## 19.34 <sys/wait.h>

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

- wait()
- waitpid()

## 19.35 <time.h>

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

- asctime\_r()
- clock()
- clock\_getres()
- clock\_gettime()
- clock\_nanosleep()
- clock\_settime()
- ctime\_r()
- difftime()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- 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()



## 19.36 <unistd.h>

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

- `_exit()`
- `access()`
- `alarm()`
- `chdir()`
- `chown()`
- `close()`
- `dup()`
- `dup2()`
- `environ`
- `fchown()`
- `fdatasync()`
- `fpathconf()`
- `fsync()`
- `ftruncate()`
- `getcwd()`
- `getegid()`
- `geteuid()`
- `getgid()`
- `getgroups()`
- `gethostname()`
- `getlogin()`
- `getpgrp()`
- `getpid()`
- `getppid()`
- `getuid()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `pipe()`
- `read()`
- `rmdir()`

- `setegid()`
- `seteuid()`
- `setgid()`
- `setsid()`
- `setuid()`
- `sleep()`
- `sysconf()`
- `unlink()`
- `write()`

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

- `execl()`
- `execle()`
- `execv()`
- `execve()`
- `fork()`

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

- `confstr()`

## 19.37 <wchar.h>

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

- btowc()
- fgetwc()
- fgetws()
- fputwc()
- fputws()
- fwide()
- fwprintf()
- fwscanf()
- getwc()
- getwchar()
- mbrlen()
- mbrtowc()
- mbsinit()
- mbsrtowcs()

## 19.38 <wctype.h>

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

- iswalnum()
- iswalpha()
- iswblank()
- iswcntrl()
- iswctype()
- iswdigit()
- iswgraph()
- iswlower()
- iswprint()
- iswpunct()
- iswspace()
- iswupper()
- iswxdigit()
- towctrans()
- tolower()
- toupper()
- wctrans()
- wctype()

## SCA 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.

## 20.1 Summary

The follow table summarizes alignment with the SCA 2.2.2 AEP standard:

Supported	244
ENOSYS	0
Not supported	0

## 20.2 <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()`

## 20.3 <dirent.h>

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

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



## 20.4 <fcntl.h>

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

- creat()
- open()

## 20.5 <locale.h>

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

- `setlocale()`

## 20.6 <math.h>

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

- `acos()`
- `asin()`
- `atan()`
- `atan2()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `modf()`
- `pow()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`

## 20.7 <pthread.h>

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

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

- `pthread_condattr_init()`
- `pthread_condattr_setclock()`
- `pthread_condattr_setpshared()`
- `pthread_create()`
- `pthread_detach()`
- `pthread_equal()`
- `pthread_exit()`
- `pthread_getschedparam()`
- `pthread_getspecific()`
- `pthread_join()`
- `pthread_key_create()`
- `pthread_key_delete()`
- `pthread_mutex_destroy()`
- `pthread_mutex_getprioceiling()`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_setprioceiling()`
- `pthread_mutex_timedlock()`
- `pthread_mutex_trylock()`
- `pthread_mutex_unlock()`
- `pthread_mutexattr_destroy()`
- `pthread_mutexattr_getprioceiling()`
- `pthread_mutexattr_getprotocol()`
- `pthread_mutexattr_getpshared()`
- `pthread_mutexattr_gettype()`
- `pthread_mutexattr_init()`
- `pthread_mutexattr_setprioceiling()`
- `pthread_mutexattr_setprotocol()`
- `pthread_mutexattr_setpshared()`
- `pthread_mutexattr_settype()`
- `pthread_once()`
- `pthread_self()`
- `pthread_setcancelstate()`
- `pthread_setcanceltype()`

- `pthread_setschedparam()`
- `pthread_setspecific()`
- `pthread_testcancel()`

## 20.8 <semaphore.h>

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

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

## 20.9 <setjmp.h>

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

- longjmp()
- setjmp()



## 20.10 <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()

## 20.11 <stdio.h>

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

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- getc()
- getchar()
- gets()
- perror()
- printf()
- putc()
- putchar()
- puts()
- remove()
- rename()
- rewind()

- `scanf()`
- `setbuf()`
- `setvbuf()`
- `sprintf()`
- `sscanf()`
- `tmpfile()`
- `tmpnam()`
- `ungetc()`

## 20.12 <stdlib.h>

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

- abort()
- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- malloc()
- qsort()
- rand()
- rand\_r()
- realloc()
- srand()

## 20.13 `<string.h>`

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

- `strcat()`
- `strchr()`
- `strcmp()`
- `strcpy()`
- `strcspn()`
- `strlen()`
- `strncat()`
- `strncmp()`
- `strncpy()`
- `strpbrk()`
- `strrchr()`
- `strspn()`
- `strstr()`
- `strtok()`
- `strtok_r()`

## 20.14 <sys/stat.h>

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

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

## 20.15 <time.h>

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

- asctime()
- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime()
- ctime\_r()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 20.16 <unistd.h>

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

- access()
- chdir()
- close()
- fpathconf()
- getcwd()
- link()
- lseek()
- pathconf()
- pause()
- read()
- rmdir()
- unlink()
- write()



## 20.17 <utime.h>

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

- `utime()`



## SCA 4.1 ULTRA LIGHTWEIGHT AEP

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

## 21.1 Summary

The follow table summarizes alignment with the SCA 4.1 Ultra Lightweight AEP standard:

Supported	22
ENOSYS	0
Not supported	0

## 21.2 <math.h>

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

- `exp()`
- `exp2()`

## 21.3 <mqqueue.h>

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

- mq\_open()
- mq\_receive()
- mq\_send()

## 21.4 <pthread.h>

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

- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_settype()
- pthread\_self()

## 21.5 <semaphore.h>

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

- `sem_init()`
- `sem_post()`
- `sem_wait()`



## 21.6 <time.h>

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

- `clock_getres()`
- `clock_gettime()`
- `timer_create()`
- `timer_settime()`



## SCA 4.1 LIGHTWEIGHT AEP

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

## 22.1 Summary

The follow table summarizes alignment with the SCA 4.1 Lightweight AEP standard:

Supported	111
ENOSYS	0
Not supported	0

## 22.2 `<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()`

## 22.3 <fcntl.h>

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

- open()

## 22.4 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`

## 22.5 <mqqueue.h>

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

- mq\_open()
- mq\_receive()
- mq\_send()



## 22.6 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_wait()
- pthread\_create()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_getpshared()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_settype()
- pthread\_self()

## 22.7 <semaphore.h>

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

- `sem_getvalue()`
- `sem_init()`
- `sem_post()`
- `sem_wait()`

## 22.8 `<stdio.h>`

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

- `sscanf()`

## 22.9 <stdlib.h>

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

- abs()
- atof()
- atoi()
- atol()
- bsearch()
- calloc()
- free()
- malloc()
- qsort()
- rand()
- realloc()
- srand()

## 22.10 <string.h>

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

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strchr()
- strcmp()
- strcspn()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()

## 22.11 <time.h>

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

- clock\_getres()
- clock\_gettime()
- gmtime()
- localtime()
- mktime()
- strftime()
- time()
- timer\_create()
- timer\_gettime()
- timer\_settime()

## 22.12 <unistd.h>

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

- close()
- read()
- write()





## SCA 4.1 [FULL] AEP

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

## 23.1 Summary

The follow table summarizes alignment with the SCA 4.1 [Full] AEP standard:

Supported	256
ENOSYS	0
Not supported	0

## 23.2 <arpa/inet.h>

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

- htonl()
- htons()
- ntohl()
- ntohs()

## 23.3 <ctype.h>

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

- isalnum()
- isalpha()
- isblank()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()
- tolower()
- toupper()

## 23.4 <dirent.h>

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

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

## 23.5 <errno.h>

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

- errno

## 23.6 <fcntl.h>

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

- creat()
- open()

## 23.7 <math.h>

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

- `acos()`
- `acosh()`
- `asin()`
- `asinh()`
- `atan()`
- `atan2()`
- `atanh()`
- `ceil()`
- `cos()`
- `cosh()`
- `exp()`
- `exp2()`
- `fabs()`
- `floor()`
- `fmod()`
- `frexp()`
- `ldexp()`
- `log()`
- `log10()`
- `log2()`
- `modf()`
- `pow()`
- `round()`
- `sin()`
- `sinh()`
- `sqrt()`
- `tan()`
- `tanh()`
- `trunc()`



## 23.8 <queue.h>

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

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

## 23.9 <pthread.h>

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

- pthread\_attr\_destroy()
- pthread\_attr\_getdetachstate()
- pthread\_attr\_getschedparam()
- pthread\_attr\_getstacksize()
- pthread\_attr\_init()
- pthread\_attr\_setinheritsched()
- pthread\_cancel()
- pthread\_cleanup\_pop()
- pthread\_cleanup\_push()
- pthread\_cond\_broadcast()
- pthread\_cond\_destroy()
- pthread\_cond\_init()
- pthread\_cond\_signal()
- pthread\_cond\_timedwait()
- pthread\_cond\_wait()
- pthread\_condattr\_destroy()
- pthread\_condattr\_init()
- pthread\_create()
- pthread\_detach()
- pthread\_equal()
- pthread\_exit()
- pthread\_getschedparam()
- pthread\_getspecific()
- pthread\_join()
- pthread\_key\_create()
- pthread\_key\_delete()
- pthread\_mutex\_destroy()
- pthread\_mutex\_init()
- pthread\_mutex\_lock()
- pthread\_mutex\_trylock()
- pthread\_mutex\_unlock()
- pthread\_mutexattr\_getpshared()

- pthread\_mutexattr\_init()
- pthread\_mutexattr\_setprioceiling()
- pthread\_mutexattr\_settype()
- pthread\_once()
- pthread\_self()
- pthread\_setcancelstate()
- pthread\_setcanceltype()
- pthread\_setschedparam()
- pthread\_setspecific()
- pthread\_testcancel()

## 23.10 <semaphore.h>

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

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

## 23.11 <signal.h>

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

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

## 23.12 <stdarg.h>

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

- va\_arg()
- va\_copy()
- va\_end()
- va\_start()

## 23.13 <stdio.h>

The following methods and variables in <stdio.h> are supported:

- clearerr()
- fclose()
- fdopen()
- feof()
- ferror()
- fflush()
- fgetc()
- fgets()
- fileno()
- fopen()
- fprintf()
- fputc()
- fputs()
- fread()
- freopen()
- fscanf()
- fseek()
- fseeko()
- ftell()
- ftello()
- fwrite()
- getc()
- getchar()
- perror()
- printf()
- putc()
- putchar()
- remove()
- rename()
- rewind()
- setbuf()
- setvbuf()

- `snprintf()`
- `sscanf()`
- `ungetc()`
- `vsnprintf()`



## 23.14 `<stdlib.h>`

The following methods and variables in `<stdlib.h>` are supported:

- `abort()`
- `abs()`
- `atof()`
- `atoi()`
- `atol()`
- `bsearch()`
- `calloc()`
- `free()`
- `labs()`
- `malloc()`
- `qsort()`
- `rand()`
- `rand_r()`
- `realloc()`
- `srand()`
- `strtod()`
- `strtol()`
- `strtoul()`

## 23.15 <string.h>

The following methods and variables in <string.h> are supported:

- memchr()
- memcmp()
- memcpy()
- memmove()
- memset()
- strchr()
- strcmp()
- strcoll()
- strcspn()
- strerror()
- strerror\_r()
- strlen()
- strncat()
- strncmp()
- strncpy()
- strpbrk()
- strrchr()
- strspn()
- strstr()
- strtok()
- strtok\_r()
- strxfrm()

## 23.16 <sys/select.h>

The following methods and variables in <sys/select.h> are supported:

- `select()`

## 23.17 <sys/socket.h>

The following methods and variables in <sys/socket.h> are supported:

- `accept()`
- `bind()`
- `connect()`
- `getsockopt()`
- `listen()`
- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`
- `setsockopt()`
- `socket()`

## 23.18 <sys/stat.h>

The following methods and variables in <sys/stat.h> are supported:

- `fstat()`
- `mkdir()`
- `stat()`

## 23.19 <time.h>

The following methods and variables in <time.h> are supported:

- asctime\_r()
- clock\_getres()
- clock\_gettime()
- clock\_settime()
- ctime\_r()
- gmtime()
- gmtime\_r()
- localtime()
- localtime\_r()
- mktime()
- nanosleep()
- strftime()
- time()
- timer\_create()
- timer\_delete()
- timer\_getoverrun()
- timer\_gettime()
- timer\_settime()

## 23.20 `<unistd.h>`

The following methods and variables in `<unistd.h>` are supported:

- `access()`
- `chdir()`
- `close()`
- `fpathconf()`
- `getcwd()`
- `link()`
- `lseek()`
- `pathconf()`
- `pause()`
- `read()`
- `rmdir()`
- `unlink()`
- `write()`





## GLOSSARY

### **POSIX**

Portable Operating System Interface is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.



# INDEX

## P

POSIX, **841**