

IEEE Std. 1003.1-2008

Finnbarr P. Murphy

(fpm@fpmurphy.com)

I am a big fan of the `POSIX.1` standard. The latest version of this standard (IEEE Std. 1003.1-2008) was released in December 1st 2008. The formal name for this version of the standard is *Portable Operating System Interface (POSIX) Base Specifications, Issue 7*.

Like its predecessors, IEEE Std. 1003.1-2004 and IEEE Std. 1003.1-2001, it was developed by the [Austin Common Standards Revision Group](#) (CSRG), more commonly called the Austin Working Group, which is a joint technical working group established in 1999 to develop and maintain the core open systems interfaces. The working group's stated approach to specification development is *write once, adopt everywhere*, with the deliverables being a set of specifications that simultaneously carry the IEEE POSIX designation, The Open Group's Technical Standard designation, and the ISO/IEC designation. Thus the new version of the specifications is simultaneously ISO/IEC 9945, IEEE Std. 1003.1-2008 and The Open Group Base Technical Standard Specifications, Issue 7.

The standard describes the characteristics and facilities of an operating system that are of importance to application developers. It intended, however, to be used by both application developers and system implementors. More precisely, it defines a standard operating system interface and environment, including a shell and utilities to support application portability at the source code level. It contains approximately 4000 pages and is divided into 4 volumes. General definitions, terms, concepts, and interfaces common to all volumes of this standard, including utility conventions and C programming language header definitions, are included in the *Base Definitions* volume. Definitions for system service functions and subroutines, language-specific system services for the C programming language, function issues, including portability, error handling, and error recovery, are included in the *System Interfaces* volume. Definitions for a command interpreter (shell) language and various utilities for application development and support are included in the *Shell and Utilities* volume. The *Rational* volume contains information that did not fit well into the other volumes, including historical information about the standard and why particular features were included, not included or removed by the working group.

The standard contains a base set of interfaces and functionality which is required in all compliant systems, together with a number of option groups. Based on these options and option groups, profiles can be defined which further group the options which are needed for a particular application domain. Some of the option groups contain a lot of functionality, others very little. The size of the mandatory base set was significantly increased in this revision. Ulrich Drepper has put up Web page about [POSIX options](#). It is Linux-centric but nevertheless is quite useful and worth looking at.

The following summarizes the major changes that were incorporated in the standard: Austin Group defect reports, IEEE Interpretations against IEEE Std 1003.1-2004, responses to ISO/IEC 9945 defect reports, Open Group corrigenda and resolutions, the Batch Environment Services and Utilities option was marked obsolescent, the `and` headers were removed, the Open Group Technical Standard, 2006, Extended API Set, parts 1 to 4 was incorporated and existing functionality was aligned with ISO/IEC 9899:1999 and ISO/IEC 9899:1999 Corregenda 2:2004.

The following functions were added:

<code>ialphasort()</code>	<code>dirfd()</code>	<code>dprintf()</code>
<code>duplocale()</code>	<code>faccessat()</code>	<code>fchmodat()</code>
<code>fchownat()</code>	<code>fdopendir()</code>	<code>fexecve()</code>
<code>fmemopen()</code>	<code>freelocale()</code>	<code>fstatat()</code>

futimens()	getdelim()	getline()
isalnum_l()	isalpha_l()	isblank_l()
iscntrl_l()	isdigit_l()	isgraph_l()
islower_l()	isprint_l()	ispunct_l()
isspace_l()	isupper_l()	iswalnum_l()
iswalpha_l()	iswblank_l()	iswcntrl_l()
iswctype_l()	iswdigit_l()	iswgraph_l()
iswlower_l()	iswprint_l()	iswpunct_l()
iswspace_l()	iswupper_l()	iswxdigit_l()
isxdigit_l()	linkat()	mbsnrtowcs()
mkdirat()	mkdtemp()	mkfifoat()
mknodat()	newlocale()	openat()
open_memstream()	open_wmemstream()	psiginfo()
psignal()	renameat()	scandir()
stpncpy()	stpncpy()	strcasecmp_l()
strcoll_l()	strfmon_l()	strncasecmp_l()
strndup()	strlen()	strsignal()
strxfrm_l()	symlinkat()	tolower_l()
toupper_l()	towctrans_l()	towlower_l()
towupper_l()	unlin	wcsncasecmp()
wcsncasecmp_l()	wcsnlen()	wcsnrtombs()
wcsxfrm_l()	wctrans_l()	wctype_l()

The following functions were previously part of an option group but are now mandatory:

aio_cancel()	aio_error()	aio_fsync()
aio_read()	aio_return()	aio_suspend()
aio_write()	asctime_r()	catclose()
catgets()	catopen()	clock_getres()
clock_gettime()	clock_nanosleep()	clock_settime()
ctime_r()	dlclose()	dLError()
dlopen()	dlsym()	fchdir()
flockfile()	fstatvfs()	ftrylockfile()
funlockfile()	getc_unlocked()	getchar_unlocked()
getgrgid_r()	getgrnam_r()	getlogin_r()
getpgid()	getpwnam_r()	getpwuid_r()
getsid()	getsubopt()	gmtime_r()
iconv()	iconv_close()	iconv_open()
lchown()	lio_listio()	localtime_r()
mkstemp()	mmap()	mprotect()
munmap()	nanosleep()	nl_langinfo()
poll()	posix_trace_timedgetnext_event()	
pread()	pthread_atfork()	pthread_exit()
pthread_attr_destroy()	pthread_attr_getdetachstate()	
pthread_attr_getguardsize()	pthread_attr_getschedparam()	

pthread_attr_init()	pthread_attr_setdetachstate()
pthread_attr_setguardsize()	pthread_attr_setschedparam()
pthread_barrier_destroy()	pthread_barrier_init()
pthread_barrierd_cond_wait()	pthread_condattr_destroy()
pthread_condattr_getclock()	pthread_condattr_init()
pthread_condattr_setclock()	pthread_create()
pthread_detach()	pthread_equal()
pthread_getspecific()	pthread_join()
pthread_key_create()	pthread_keyread_mutex_timedlock()
pthread_mutex_trylock()	pthread_mutex_unlock()
pthread_mutexattr_destroy()	pthread_mutexattr_gettype()
pthread_mutexattr_init()	pthread_mutexattr_settype()
pthread_once()	pthread_rwlock_destroy()
pthread_rwlock_init()	pthread_rwlockcrydlock()
pthread_rwlock_trywrlock()	pthread_rwlock_unlock()
pthread_rwlock_wrlock()	pthread_rwlockattr_destroy()
pthread_rwlockattr_init()	pthread_self()
pthread_setcancelstate()	pthread_setcanceltype()
pthread_setspecific()	pthread_spin_destroy()
pthre_unlock()	pthread_testcancel()
putc_unlocked()	putchar_unlocked()

pwrite()	rand_r()	readdir_r()
sem_close()	sem_destroy()	sem_getvalue()
sem_init()	sem_open()	sem_post()
sem_timedwait()	sem_trywait()	sem_unlink()
sem_wait()	sigqueue()	sigqueue()
sigtimedwcmp()	strdup()	strerror_r()
strfmon()	strncasecmp()	strtok_r()
tcgetsid()	timer_create()	timer_delete()
timer_getoverrun()	timer_gettime()	timer_settime()
truncate()	ttyname_r()	waitid()

The following functions were marked as obsolescent:

asctime()	asctime_r()	ctime()
ctime_r()	gets()	rand_r()
tmpnam()	utime()	_longjmp()
_setjmp()	_tolower()	_toupper()
ftw()	getitimer()	gettimeofday()
isascii()	setitimer()	setpgrp()
sighold()	sigignore()	sigpause()
sigrelse()	sigset()	siginterrupt()
tempnam()	toascii()	ulimit()

The following functions and symbols were removed:

bcmp()	bcopy()	bsd_signal()
bzero()	ecvt()	fcvt()
ftime()	gcvt()	getcontext()
gethostbyaddr()	gethostbyname()	getwd()
h_errno	index()	makecontext()
mktemp()	rindex()	scalb()
setcontext()	swapcontext()	ualarm()
usleep()	vfork()	wcs wcs()

Finally, if you want easy access to the specifications from your [FireFox](#) browser, you can install the [Firefox Search Bar](#) plugin which I have posted at [Mycroft Project](#). It searches the online version of [Base Specification Issue 7](#) at the [Open Group](#). Like [ECMA](#) and a number of other standards development organizations, the [Open Group](#) has a policy of free access to their specifications online whereas the [IEEE](#) charges for access to their standards.

Well, this is my last post for 2008. What a year it has been! A Happy and Prosperous New Year to all.