②

UNIX Standardization and Implementation

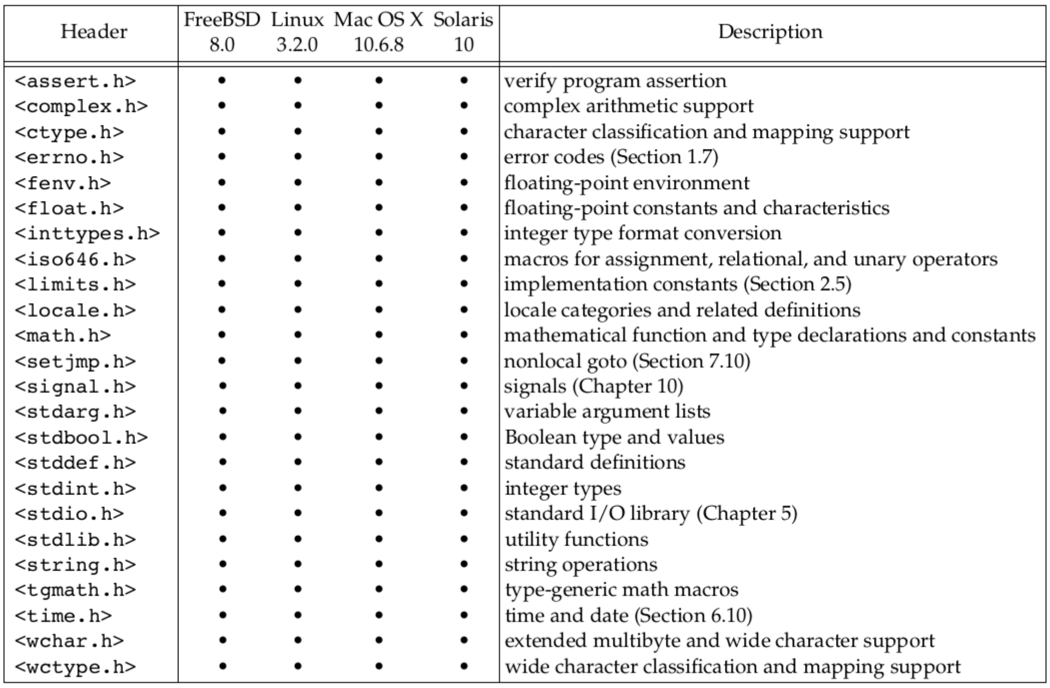
★ Unix Standardization

• ISO C

⇛ The intent of the ISO C standard is to provide portability of conforming C programs to a wide variety of operating systems, not only the UNIX System.

⇛ This standard defines not only the syntax and semantics of the programming language but also a standard library.

⇛ The ISO C library can be divided into 24 areas, based on the headers defined by the standard.



Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
`<assert.h>`	•	•	•	•	verify program assertion
`<complex.h>`	•	•	•	•	complex arithmetic support
`<ctype.h>`	•	•	•	•	character classification and mapping support
`<errno.h>`	•	•	•	•	error codes (Section 1.7)
`<fenv.h>`	•	•	•	•	floating-point environment
`<float.h>`	•	•	•	•	floating-point constants and characteristics
`<inttypes.h>`	•	•	•	•	integer type format conversion
`<iso646.h>`	•	•	•	•	macros for assignment, relational, and unary operators
`<limits.h>`	•	•	•	•	implementation constants (Section 2.5)
`<locale.h>`	•	•	•	•	locale categories and related definitions
`<math.h>`	•	•	•	•	mathematical function and type declarations and constants
`<setjmp.h>`	•	•	•	•	nonlocal goto (Section 7.10)
`<signal.h>`	•	•	•	•	signals (Chapter 10)
`<stdarg.h>`	•	•	•	•	variable argument lists
`<stdbool.h>`	•	•	•	•	Boolean type and values
`<stddef.h>`	•	•	•	•	standard definitions
`<stdint.h>`	•	•	•	•	integer types
`<stdio.h>`	•	•	•	•	standard I/O library (Chapter 5)
`<stdlib.h>`	•	•	•	•	utility functions
`<string.h>`	•	•	•	•	string operations
`<tgmath.h>`	•	•	•	•	type-generic math macros
`<time.h>`	•	•	•	•	time and date (Section 6.10)
`<wchar.h>`	•	•	•	•	extended multibyte and wide character support
`<wctype.h>`	•	•	•	•	wide character classification and mapping support

{Figure 2.1}

• IEEE POSIX {Portable Operating System Interface}

⇛ POSIX is a family of standards initially developed by the IEEE (Institute of Electrical and Electronics Engineers).

⇛ Goal is to promote the portability of applications among various UNIX System environments.

⇛ This standard defines the services that an operating system must provide if it is to be ‘‘POSIX compliant,’’ and has been adopted by most computer vendors.

⇛ Because the 1003.1 standard specifies an interface and not an implementation, no distinction is made between system calls and library functions.

↳ All the routines in the standard are called functions.

⇛ The required and optional headers as specified by POSIX.1

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<aio.h>	•	•	•	•	asynchronous I/O
<cpio.h>	•	•	•	•	cpio archive values
<dirent.h>	•	•	•	•	directory entries (Section 4.22)
<dlfcn.h>	•	•	•	•	dynamic linking
<fcntl.h>	•	•	•	•	file control (Section 3.14)
<fnmatch.h>	•	•	•	•	filename-matching types
<glob.h>	•	•	•	•	pathname pattern-matching and generation
<grp.h>	•	•	•	•	group file (Section 6.4)
<iconv.h>	•	•	•	•	codeset conversion utility
<langinfo.h>	•	•	•	•	language information constants
<monetary.h>	•	•	•	•	monetary types and functions
<netdb.h>	•	•	•	•	network database operations
<nl_types.h>	•	•	•	•	message catalogs
<poll.h>	•	•	•	•	poll function (Section 14.4.2)
<pthread.h>	•	•	•	•	threads (Chapters 11 and 12)
<pwd.h>	•	•	•	•	password file (Section 6.2)
<regex.h>	•	•	•	•	regular expressions
<sched.h>	•	•	•	•	execution scheduling
<semaphore.h>	•	•	•	•	semaphores
<strings.h>	•	•	•	•	string operations
<tar.h>	•	•	•	•	tar archive values
<termios.h>	•	•	•	•	terminal I/O (Chapter 18)
<unistd.h>	•	•	•	•	symbolic constants
<wordexp.h>	•	•	•	•	word-expansion definitions
<arpa/inet.h>	•	•	•	•	Internet definitions (Chapter 16)
<net/if.h>	•	•	•	•	socket local interfaces (Chapter 16)
<netinet/in.h>	•	•	•	•	Internet address family (Section 16.3)
<netinet/tcp.h>	•	•	•	•	Transmission Control Protocol definitions
<sys/mman.h>	•	•	•	•	memory management declarations
<sys/select.h>	•	•	•	•	select function (Section 14.4.1)
<sys/socket.h>	•	•	•	•	sockets interface (Chapter 16)
<sys/stat.h>	•	•	•	•	file status (Chapter 4)
<sys/statvfs.h>	•	•	•	•	file system information
<sys/times.h>	•	•	•	•	process times (Section 8.17)
<sys/types.h>	•	•	•	•	primitive system data types (Section 2.8)
<sys/un.h>	•	•	•	•	UNIX domain socket definitions (Section 17.2)
<sys/utsname.h>	•	•	•	•	system name (Section 6.9)
<sys/wait.h>	•	•	•	•	process control (Section 8.6)

Figure 2.2 Required headers defined by the POSIX standard

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<fmtmsg.h>	•	•	•	•	message display structures
<ftw.h>	•	•	•	•	file tree walking (Section 4.22)
<libgen.h>	•	•	•	•	pathname management functions
<ndbm.h>	•		•	•	database operations
<search.h>	•	•	•	•	search tables
<syslog.h>	•	•	•	•	system error logging (Section 13.4)
<utmpx.h>		•	•	•	user accounting database
<sys/ipc.h>	•	•	•	•	IPC (Section 15.6)
<sys/msg.h>	•	•	•	•	XSI message queues (Section 15.7)
<sys/resource.h>	•	•	•	•	resource operations (Section 7.11)
<sys/sem.h>	•	•	•	•	XSI semaphores (Section 15.8)
<sys/shm.h>	•	•	•	•	XSI shared memory (Section 15.9)
<sys/time.h>	•	•	•	•	time types
<sys/uio.h>	•	•	•	•	vector I/O operations (Section 14.6)

Figure 2.3 XSI option headers defined by the POSIX standard

Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<mqueue.h>	•	•		•	message queues
<spawn.h>	•	•	•	•	real-time spawn interface

Figure 2.4 Optional headers defined by the POSIX standard

Because POSIX.1 includes the ISO C standard library functions, it also requires the headers listed in Figure 2.1.

POSIX.1 standard is maintained by an open working group known as the Austin Group.