Linux Standard Base Specification for the z/Architecture 1.3.0

Linux Standard Base Specification for the z/Architecture 1.3.0

Copyright © 2002 Free Standards Group

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Portions of the text were taken from other copyrighted documents in accordance with the respective licenses of those documents.

Linux is a trademark of Linus Torvalds.

UNIX a registered trademark of the Open Group in the United States and other countries.

LSB is a trademark of the Free Standards Group in the USA and other countries.

AMD is a trademark of Advanced Micro Devices, Inc.

Intel386 and Itanium are trademarks of Intel Corporation.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Table of Contents

List of Tables

I. Introduction

Chapter 1. Introduction

1.1. Introduction

This is version 1.3.0 of the Linux Standard Base Specification for the z/Architecture. An implementation of this version of the specification may not claim to be an implementation of the Linux Standard Base unless it has successfully completed the compliance process as defined by the Free Standards Group.

1.2. Purpose

The Linux Standard Base (LSB) defines a system interface for compiled applications and a minimal environment for support of installation scripts. Its purpose is to enable a uniform industry standard environment for high-volume applications conforming to the LSB.

The LSB defines a binary interface for application programs that are compiled and packaged for LSB-conforming implementations on many different hardware architectures. Since a binary specification must include information specific to the computer processor architecture for which it is intended, it is not possible for a single document to specify the interface for all possible LSB-conforming implementations. Therefore, the LSB is a family of specifications, rather than a single one.

The LSB is composed of two basic parts: A common part of the specification describes those parts of the interface that remain constant across all hardware implementations of the LSB, and an architecture-specific part of the specification describes the parts of the specification that are specific to a particular processor architecture. Together, the generic LSB and the architecture-specific supplement for a single hardware architecture provide a complete interface specification for compiled application programs on systems that share a common hardware architecture.

This document is the architecture-specific supplement. It must be used in conjunction with the generic LSB. This document provides architecture-specific information that supplements the generic LSB as well as additional information that is not found in the generic LSB.

This document should be used in conjunction with the documents it references. This document enumerates the system components it includes, but descriptions of those components may be included entirely or partly in this document, partly in other documents, or entirely in other reference documents. For example, the section that describes system service routines includes a list of the system routines supported in this interface, formal declarations of the data structures they use that are visible to applications, and a pointer to the underlying referenced specification for information about the syntax and semantics of each call. Only those routines not described in standards referenced by this document, or extensions to those standards, are described in the detail. Information referenced in this way is as much a part of this document as is the information explicitly included here.

1.3. Related Standards

The specifications listed below are referenced in whole or in part by the Linux Standard Base. Such references may be normative or non-normative; a reference to specification shall only be considered normative if it is explicitly cited as such. The LSB may make normative references to a portion of these specifications (that is, to define a specific function or group of functions); in such cases, only the explicitly referenced portion of the specification is to be considered normative.

Table 1-1. Related Standards

System V Application Binary Interface - DRAFT - 22 June 2000	http://www.caldera.com/developers/gabi/2000-07-17/contents.html
DWARF Debugging Information Format, Revision 2.0.0 (July 27, 1993)	
Filesystem Hierarchy Standard (FHS) 2.2	http://www.pathname.com/fhs/
IEEE Standard for Binary Floating-Point Arithmetic	http://www.ieee.org/
System V Application Binary Interface, Edition 4.1	http://www.caldera.com/developers/devspecs/gabi41.pdf
ISO/IEC 9899: 1990, Programming Languages --C	
ISO/IEC 9899: 1999, Programming Languages --C	
ISO/IEC 14882: 1998(E) Programming languages --C++	
Linux Assigned Names And Numbers Authority	http://www.lanana.org/
Large File Support	http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html
LI18NUX 2000 Globalization Specification, Version 1.0 with Amendment 4	http://www.li18nux.org/docs/html/LI18NUX-2000-amd4.htm
Linux Standard Base	http://www.linuxbase.org/spec/
OpenGL® Application Binary Interface for Linux	http://oss.sgi.com/projects/ogl-sample/ABI/
OSF-RFC 86.0	http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt
IEEE Std POSIX 1003.2-1992 (ISO/IEC 9945-2:1993)	http://www.ieee.org/
POSIX 1003.1c	http://www.ieee.org/
RFC 1952: GZIP file format specification version 4.3	http://www.ietf.org/rfc/rfc1952.txt
RFC 2440: OpenPGP Message Format	
LINUX for zSeries Application Binary Interface Supplement	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
z/Architecture Principles of Operation	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), plus Corrigendum U018	http://www.opengroup.org/publications/catalog/un.htm
CAE Specification, January 1997, System Interface Definitions (XBD),Issue 5 (ISBN: 1-85912-186-1,	http://www.opengroup.org/publications/catalog/un.htm

C605)	
CAE Specification, January 1997, Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)	http://www.opengroup.org/publications/catalog/un.htm
CAE Specification, February 1997, Networking Services (XNS), Issue 5(ISBN: 1-85912-165-9, C523)	http://www.opengroup.org/
CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)	http://www.opengroup.org/publications/catalog/un.htm
The Single UNIX® Specification(SUS) Version 1 (UNIX 95) System Interfaces & Headers	http://www.opengroup.org/publications/catalog/un.htm
The Single UNIX® Specification(SUS) Version 3	http://www.unix.org/version3/
System V Interface Definition, Issue 3 (ISBN 0201566524)	
System V Interface Definition,Fourth Edition	
Double Buffer Extension Library	http://www.x.org/
X Display Power Management Signaling (DPMS) Extension, Library Specification	http://www.x.org/
X Record Extension Library	http://www.x.org/
Security Extension Specification, Version 7.1	http://www.x.org/
X Nonrectangular Window Shape Extension Library Version 1.0	http://www.x.org/
MIT-SHM--The MIT Shared Memory Extension	http://www.x.org/
X Synchronization Extension Library	http://www.x.org/
XTEST Extension Library	http://www.x.org/
X11R6.4 X Inter-Client Exchange (ICE) Protocol	http://www.x.org/
X11R6.4 X11 Input Extension Library	http://www.x.org/
X11R6.4 Xlib - C library	http://www.x.org/
X/Open Portability Guide, Issue 4	http://www.opengroup.org/
X11R6.4 X Session Management Library	http://www.x.org/
X11R6.4 X Toolkit Intrinsics	http://www.x.org/
zlib 1.1.3 Manual	http://www.gzip.org/zlib/

1.4. Relevant Libraries

The libraries listed here shall be available on a Linux Standard Base system. This list is an addition to the list in the general specification.

Table 1-2. Standard Library Names

Library	Runtime Name
libm	libm.so.6
libdl	libdl.so.2
libcrypt	libcrypt.so.1
libc	libc.so.6
libpthread	libpthread.so.0
proginterp	/lib64/ld-lsb-s390x.so.1

These libraries will be in an implementation-dependent directory which the dynamic linker will search by default.

1.5. How to Use this Standard

The complete LSB specification is composed of a generic LSB specification and this supplemental processor-specific specification. These two documents constitute a specification that should be used in conjunction with the publicly-available standards documents it references. The LSB enumerates the system components it includes, but descriptions of those components may be included entirely in the LSB, partly in the LSB and partly in other documents, or entirely in other reference documents.

1.6. Definitions

gLSB

The common part of the LSB Specification that describes those parts of the interface that remain constant across all hardware implementations of the LSB.

archLSB

The architectural part of the LSB Specification which describes the specific parts of the interface that are platform specific. The archLSB is complementary to the gLSB.

LSB Implementation Conformance

An implementation satisfying the following requirements:

1. The implementation shall implement fully the architecture described in the hardware manual for the target processor architecture.
2. The implementation shall be capable of executing compiled applications having the format and using the system interfaces described in this document.

3. The implementation shall provide libraries containing the interfaces specified by this document, and shall provide a dynamic linking mechanism that allows these interfaces to be attached to applications at runtime. All the interfaces shall behave as specified in this document.
4. The map of virtual memory provided by the implementation shall conform to the requirements of this document.
5. The implementation's low-level behavior with respect to function call linkage, system traps, signals, and other such activities shall conform to the formats described in this document.
6. The implementation shall provide all of the mandatory interfaces in their entirety.
7. The implementation may provide one or more of the optional interfaces. Each optional interface that is provided shall be provided in its entirety. The product documentation shall state which optional interfaces are provided.
8. The implementation shall provide all files and utilities specified as part of this document in the format defined here and in other referenced documents. All commands and utilities shall behave as required by this document. The implementation shall also provide all mandatory components of an application's runtime environment that are included or referenced in this document.
9. The implementation, when provided with standard data formats and values at a named interface, shall provide the behavior defined for those values and data formats at that interface. However, a conforming implementation may consist of components which are separately packaged and/or sold. For example, a vendor of a conforming implementation might sell the hardware, operating system, and windowing system as separately packaged items.
10. The implementation may provide additional interfaces with different names. It may also provide additional behavior corresponding to data values outside the standard ranges, for standard named interfaces.

LSB Application Conformance

An application with the following characteristics:

1. Its executable files are either shell scripts or object files in the format defined for the Object File Format system interface.
2. Its object files participate in dynamic linking as defined in the Program Loading and Linking System interface.
3. It employs only the instructions, traps, and other low-level facilities defined in the Low-Level System interface as being for use by applications.
4. If it requires any optional interface defined in this document in order to be installed or to execute successfully, the requirement for that optional interface is stated in the application's documentation.
5. It does not use any interface or data format that is not required to be provided by a conforming implementation, unless:
   - If such an interface or data format is supplied by another application through direct invocation of that application during execution, that application is in turn an LSB conforming application.
   - The use of that interface or data format, as well as its source, is identified in the documentation of the application.
6. It must not use any values for a named interface that are reserved for vendor extensions.

A strictly conforming application does not require or use any interface, facility, or implementation-defined extension that is not defined in this document in order to be installed or to execute successfully.

Rationale

An LSB conforming application is expected to have no dependencies on any vendor extensions to this document. The most common such extensions are additional function entry points and additional libraries other than the ones defined in this document. If an application requires such extensions, it is not portable, since other LSB conforming implementations may not provide those extensions.

An LSB conforming application is required to use system services on the implementation on which it is running, rather than importing system routines from some other implementation. Thus, it must link dynamically to any routines in the implementation that perform system traps to kernel services.

It is to be expected that some applications may be companion applications to other applications. For example, a query application may be a companion to a database application; a preprocessor may be an adjunct to one or more compilers; a data reformatter may convert data from one document manager to another. In such cases, the application may or may not be LSB conforming, regardless of whether the other application on which it is dependent is LSB conforming. If such an application merely uses data produced by another application, the application's compliance is independent of the other application's compliance. If such an application actually invokes another application during execution (as, for example, a third-party math library), the invoking application is LSB conforming only if it also constitutes a LSB conforming application in combination with the invoked application.

Shell Script

A file that is read by an interpreter (e.g., awk). The first line of the shell script includes a reference to its interpreter binary.

1.7. Terminology

can

Describes a permissible optional feature or behavior available to the user or application. The feature or behavior is mandatory for an implementation that conforms to this document. An application can rely on the existence of the feature or behavior.

implementation-defined

Describes a value or behavior that is not defined by this document but is selected by an implementor. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence of the value or behavior. An application that relies on such a value or behavior cannot be assured to be portable across conforming implementations. The implementor shall document such a value or behavior so that it can be used correctly by an application.

may

Describes a feature or behavior that is optional for an implementation that conforms to this document. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations.

To avoid ambiguity, the opposite of may is expressed as need not, instead of may not.

must

Describes a feature or behavior that is mandatory for an application or user. An implementation that conforms to this document shall support this feature or behavior.

shall

Describes a feature or behavior that is mandatory for an implementation that conforms to this document. An application can rely on the existence of the feature or behavior.

should

For an implementation that conforms to this document, describes a feature or behavior that is recommended but not mandatory. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations.

For an application, describes a feature or behavior that is recommended programming practice for optimum portability.

undefined

Describes the nature of a value or behavior not defined by this document which results from use of an invalid program construct or invalid data input. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

unspecified

Describes the nature of a value or behavior not specified by this document which results from use of a valid program construct or valid data input. The value or behavior may vary among implementations that conform to this document. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

will

Same meaning as shall; shall is the preferred term.

II. Low Level System Information

Chapter 2. Machine Interface

2.1. Processor Architecture

The z/Architecture is specified by the following documents:

- LINUX for zSeries Application Binary Interface Supplement
- z/Architecture Principles of Operation

Only the non optional features of z/Architecture processor instruction set may be assumed to be present. An application is responsible for determining if any additional instruction set features are available before using those additional features. If a feature is not present, then the application may not use it.

Applications conforming to this specification must provide feedback to the user if a feature that is required for correct execution of the application is not present. Applications conforming to this specification should attempt to execute in a diminished capacity if a required instruction set feature is not present.

This specfication does not provide any performance guarantees of a conforming system. A system conforming to this specification may be implemented in either hardware or software.

2.2. Data Representation

LSB-conforming applications shall use the data representation as defined in Chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

2.2.1. Byte Ordering

2.2.2. Fundamental Types

2.2.3. Aggregates and Unions

2.2.4. Bit Fields

Chapter 3. Function Calling Sequence

LSB-conforming applications shall use the function calling sequence as defined in Chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.1. Registers

3.2. The Stack Frame

3.3. Parameter Passing

3.4. Variable Argument Lists

3.5. Return Values

Chapter 4. Operating System Interface

LSB-conforming applications shall use the Operating System Interfaces as defined in Chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

4.1. Virtual Address Space

4.2. Page Size

4.3. Virtual Address Assignments

4.4. Managing the Process Stack

4.5. Coding Guidlines

4.6. Processor Execution Mode

4.7. Exception Interface

4.8. Signal Delivery

4.8.1. Signal Handler Interface

Chapter 5. Process Initialization

LSB-conforming applications shall use the Process Initialization as defined in Chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.1. Registers

5.2. Process Stack

Chapter 6. Coding Examples

LSB-conforming applications may implement fundamental operations using the Coding Examples as defined in Chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

6.1. Code Model Overview

6.2. Function Prolog and Epilog

6.3. Profiling

6.4. Data Objects

6.5. Function Calls

6.6. Branching

6.7. Dynamic Stack Space Allocation

Chapter 7. Debug Information

The LSB does not currently specify the format of Debug information.

III. Object Format

LSB-conforming implementations shall support an object file , called Executable and Linking Format (ELF) as defined by the LINUX for zSeries Application Binary Interface Supplement and as supplemented by the Linux Standard Base Specification and this document. LSB-conforming implementations need not support tags related functionality. LSB-conforming applications must not rely on tags related functionality.

Chapter 8. ELF Header

8.1. Machine Information

LSB-conforming applications shall use the Machine Information as defined in LINUX for zSeries Application Binary Interface Supplement, Chapter 2.

Chapter 9. Sections

9.1. Special Sections

The following sections are defined in the LINUX for zSeries Application Binary Interface Supplement.

Table 9-1. ELF Special Sections

Name	Type	Attributes
.got	SHT_PROGBITS	SHF_ALLOC+SHF_WRITE
.plt	SHT_PROGBITS	SHF_ALLOC+SHF_EXECINSTR

.got

This section holds the global offset table

.plt

This section holds the procedure linkage table

9.2. Linux Special Sections

The following Linux S/390 specific sections are defined here.

Table 9-2. Additional Special Sections

Name	Type	Attributes
.rela.dyn	SHT_RELA	SHF_ALLOC
.rela.plt	SHT_RELA	SHF_ALLOC
.sbss	SHT_PROGBITS	SHF_WRITE

.rela.dyn

This section holds RELA type relocation information for all sections of a shared library except the PLT

.rela.plt

This section holds RELA type relocation information for the PLT section of a shared library or dynamically linked application

.sbss

This section holds uninitialized data that contribute to the program's memory image. The system initializes the data with zeroes when the program begins to run.

Chapter 10. Symbol Table

LSB-conforming applications shall use the Symbol Table as defined in Chapter 2 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 11. Relocation

LSB-conforming applications shall use Relocations as defined in Chapter 2 of the LINUX for zSeries Application Binary Interface Supplement.

11.1. Relocation Types

IV. Program Loading and Dynamic Linking

LSB-conforming implementations shall support the object file information and system actions that create running programs as specified in the System V Application Binary Interface, Edition 4.1, LINUX for zSeries Application Binary Interface Supplement and as supplemented by the Linux Standard Base Specification and this document.

Chapter 12. Program Loading

See LINUX for zSeries Application Binary Interface Supplement, Chapter 3.

Chapter 13. Dynamic Linking

See LINUX for zSeries Application Binary Interface Supplement, Chapter 3.

13.1. Program Interpreter/Dynamic Linker

The LSB specifies the Program Interpreter to be /lib64/ld-lsb-s390x.so.1.

13.2. Dynamic Section

The following dynamic entries are defined in the LINUX for zSeries Application Binary Interface Supplement, Chapter 3.

DT_JMPREL

: This entry is associated with a table of relocation entries for the procedure linkage table. This entry is mandatory both for executable and shared object files

DT_PLTGOT

: This entry's d_ptr member gives the address of the first byte in the procedure linkage table

In addtion the following dynamic entries are also supported:

DT_RELACOUNT

: The number of relative relocations in .rela.dyn

13.3. Global Offset Table

See LINUX for zSeries Application Binary Interface Supplement, Chapter 3.

13.4. Function Addresses

13.5. Procedure Linkage Table

V. Base Libraries

Chapter 14. Libraries

An LSB-conforming implementation shall support base libraries which provide interfaces for accessing the operating system, processor and other hardware in the system.

Only those interfaces that are unique to the z/Architecture platform are defined here. This section should be used in conjunction with the corresponding section in the Linux Standard Base Specification.

14.1. Interfaces for libc

Table 14-1. libc Definition

Library:	libc
SONAME:	libc.so.6

The behavior of the interfaces in this library is specified by the following standards.

ISO/IEC 9899: 1999, Programming Languages --C[1]
Large File Support[2]
Linux Standard Base[3]
IEEE Std POSIX.1-1996 [ISO/IEC 9945-1:1996][4]
CAE Specification, February 1997, Networking Services (XNS), Issue 5(ISBN: 1-85912-165-9, C523)[5]
CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)[6]
The Single UNIX® Specification(SUS) Version 3[7]
System V Interface Definition, Issue 3 (ISBN 0201566524)[8]
System V Interface Definition,Fourth Edition[9]

14.1.1. RPC

Table 14-2. libc - RPC Function Interfaces

authnone_create(GLIBC_2.2)[9]	svc_getreqset(GLIBC_2.2)[8]	xdr_bytes(GLIBC_2.2)[8]	xdr_opaque_auth(GLIBC_2.2)[8]	xdr_union(GLIBC_2.2)[8]
clnt_create(GLIBC_2.2)[9]	svcerr_auth(GLIBC_2.2)[8]	xdr_callhdr(GLIBC_2.2)[8]	xdr_pointer(GLIBC_2.2)[8]	xdr_vector(GLIBC_2.2)[8]
clnt_pcreateerror(GLIBC_2.2)[9]	svcerr_decode(GLIBC_2.2)[8]	xdr_callmsg(GLIBC_2.2)[8]	xdr_reference(GLIBC_2.2)[8]	xdr_void(GLIBC_2.2)[8]
clnt_perrno(GLIBC_2.2)[9]	svcerr_noproc(GLIBC_2.2)[8]	xdr_char(GLIBC_2.2)[8]	xdr_rejected_reply(GLIBC_2.2)[8]	xdr_wrapstring(GLIBC_2.2)[8]
clnt_perror(GLIBC_2.2)[9]	svcerr_noprog(GLIBC_2.2)[8]	xdr_double(GLIBC_2.2)[8]	xdr_replymsg(GLIBC_2.2)[8]	xdrmem_create(GLIBC_2.2)[8]
clnt_spcreateerror(GLIBC_2.2)[9]	svcerr_progvers(GLIBC_2.2)[8]	xdr_enum(GLIBC_2.2)[8]	xdr_short(GLIBC_2.2)[8]	xdrrec_create(GLIBC_2.2)[8]

clnt_sperrno(GLIBC_2.2)[9]	svcerr_systemerr(GLIBC_2.2)[8]	xdr_float(GLIBC_2.2)[8]	xdr_string(GLIBC_2.2)[8]	xdrrec_eof(GLIBC_2.2)[8]
clnt_sperror(GLIBC_2.2)[9]	svcerr_weakauth(GLIBC_2.2)[8]	xdr_free(GLIBC_2.2)[8]	xdr_u_char(GLIBC_2.2)[8]	
getdomainname(GLIBC_2.2)[3]	xdr_accepted_reply(GLIBC_2.2)[8]	xdr_int(GLIBC_2.2)[8]	xdr_u_int(GLIBC_2.2)[3]	
key_decryptsession(GLIBC_2.2)[8]	xdr_array(GLIBC_2.2)[8]	xdr_long(GLIBC_2.2)[8]	xdr_u_long(GLIBC_2.2)[8]	
setdomainname(GLIBC_2.2)[3]	xdr_bool(GLIBC_2.2)[8]	xdr_opaque(GLIBC_2.2)[8]	xdr_u_short(GLIBC_2.2)[8]	

14.1.2. System Calls

Table 14-3. libc - System Calls Function Interfaces

__fxstat(GLIBC_2.2)[3]	fchown(GLIBC_2.2)[6]	ioctl(GLIBC_2.2)[3]	readdir(GLIBC_2.2)[6]	setsid(GLIBC_2.2)[6]
__getpgid(GLIBC_2.2)[3]	fcntl(GLIBC_2.2)[3]	kill(GLIBC_2.2)[3]	readdir_r(GLIBC_2.2)[6]	setuid(GLIBC_2.2)[6]
__lxstat(GLIBC_2.2)[3]	fdatasync(GLIBC_2.2)[6]	killpg(GLIBC_2.2)[6]	readlink(GLIBC_2.2)[6]	sleep(GLIBC_2.2)[6]
__xmknod(GLIBC_2.2)[3]	flock(GLIBC_2.2)[3]	lchown(GLIBC_2.2)[6]	readv(GLIBC_2.2)[6]	statvfs(GLIBC_2.2)[6]
__xstat(GLIBC_2.2)[3]	fork(GLIBC_2.2)[6]	link(GLIBC_2.2)[6]	rename(GLIBC_2.2)[6]	stime(GLIBC_2.2)[3]
access(GLIBC_2.2)[6]	fstatvfs(GLIBC_2.2)[6]	lockf(GLIBC_2.2)[6]	rmdir(GLIBC_2.2)[6]	symlink(GLIBC_2.2)[6]
acct(GLIBC_2.2)[3]	fsync(GLIBC_2.2)[6]	lseek(GLIBC_2.2)[6]	sbrk(GLIBC_2.2)[6]	sync(GLIBC_2.2)[6]
alarm(GLIBC_2.2)[6]	ftime(GLIBC_2.2)[6]	mkdir(GLIBC_2.2)[6]	sched_get_priority_max(GLIBC_2.2)[6]	sysconf(GLIBC_2.2)[6]
brk(GLIBC_2.2)[6]	ftruncate(GLIBC_2.2)[6]	mkfifo(GLIBC_2.2)[6]	sched_get_priority_min(GLIBC_2.2)[6]	time(GLIBC_2.2)[6]
chdir(GLIBC_2.2)[6]	getcontext(GLIBC_2.2)[6]	mlock(GLIBC_2.2)[6]	sched_getparam(GLIBC_2.2)[6]	times(GLIBC_2.2)[6]
chmod(GLIBC_2.2)[6]	getegid(GLIBC_2.2)[6]	mlockall(GLIBC_2.2)[6]	sched_getscheduler(GLIBC_2.2)[6]	truncate(GLIBC_2.2)[6]
chown(GLIBC_2.2)[6]	geteuid(GLIBC_2.2)[6]	mmap(GLIBC_2.2)[6]	sched_rr_get_interval(GLIBC_2.2)[6]	ulimit(GLIBC_2.2)[6]

chroot(GLIBC_2.2)[6]	getgid(GLIBC_2.2)[6]	mprotect(GLIBC_2.2)[6]	sched_setparam(GLIBC_2.2)[6]	umask(GLIBC_2.2)[6]
clock(GLIBC_2.2)[6]	getgroups(GLIBC_2.2)[6]	msync(GLIBC_2.2)[6]	sched_setscheduler(GLIBC_2.2)[6]	uname(GLIBC_2.2)[6]
close(GLIBC_2.2)[6]	getitimer(GLIBC_2.2)[6]	munlock(GLIBC_2.2)[6]	sched_yield(GLIBC_2.2)[6]	unlink(GLIBC_2.2)[3]
closedir(GLIBC_2.2)[6]	getloadavg(GLIBC_2.2)[3]	munlockall(GLIBC_2.2)[6]	select(GLIBC_2.2)[5]	utime(GLIBC_2.2)[6]
creat(GLIBC_2.2)[6]	getpagesize(GLIBC_2.2)[6]	munmap(GLIBC_2.2)[6]	setcontext(GLIBC_2.2)[6]	utimes(GLIBC_2.2)[6]
dup(GLIBC_2.2)[6]	getpgid(GLIBC_2.2)[6]	nanosleep(GLIBC_2.2)[6]	setegid(GLIBC_2.2)[3]	vfork(GLIBC_2.2)[6]
dup2(GLIBC_2.2)[6]	getpgrp(GLIBC_2.2)[6]	nice(GLIBC_2.2)[3]	seteuid(GLIBC_2.2)[3]	wait(GLIBC_2.2)[6]
execl(GLIBC_2.2)[6]	getpid(GLIBC_2.2)[6]	open(GLIBC_2.2)[6]	setgid(GLIBC_2.2)[6]	wait3(GLIBC_2.2)[3]
execle(GLIBC_2.2)[6]	getppid(GLIBC_2.2)[6]	opendir(GLIBC_2.2)[6]	setitimer(GLIBC_2.2)[6]	wait4(GLIBC_2.2)[3]
execlp(GLIBC_2.2)[6]	getpriority(GLIBC_2.2)[6]	pathconf(GLIBC_2.2)[6]	setpgid(GLIBC_2.2)[6]	waitpid(GLIBC_2.2)[3]
execv(GLIBC_2.2)[6]	getrlimit(GLIBC_2.2)[6]	pause(GLIBC_2.2)[6]	setpgrp(GLIBC_2.2)[6]	write(GLIBC_2.2)[6]
execve(GLIBC_2.2)[6]	getrusage(GLIBC_2.2)[6]	pipe(GLIBC_2.2)[6]	setpriority(GLIBC_2.2)[6]	writev(GLIBC_2.2)[6]
execvp(GLIBC_2.2)[6]	getsid(GLIBC_2.2)[6]	poll(GLIBC_2.2)[6]	setregid(GLIBC_2.2)[6]	
exit(GLIBC_2.2)[6]	getuid(GLIBC_2.2)[6]	pread(GLIBC_2.2)[6]	setreuid(GLIBC_2.2)[6]	
fchdir(GLIBC_2.2)[6]	getwd(GLIBC_2.2)[6]	pwrite(GLIBC_2.2)[6]	setrlimit(GLIBC_2.2)[6]	
fchmod(GLIBC_2.2)[6]	initgroups(GLIBC_2.2)[3]	read(GLIBC_2.2)[6]	setrlimit64(GLIBC_2.2)[2]	

14.1.3. Standard I/O

Table 14-4. libc - Standard I/O Function Interfaces

_IO_feof(GLIBC_2.2)[3]	fgetpos(GLIBC_2.2)[6]	fsetpos(GLIBC_2.2)[6]	putc_unlocked(GLIBC_2.2)[6]	sprintf(GLIBC_2.2)[6]

_IO_getc(GLIBC_2.2)[3]	fgets(GLIBC_2.2)[6]	ftell(GLIBC_2.2)[6]	putchar(GLIBC_2.2)[6]	sscanf(GLIBC_2.2)[6]
_IO_putc(GLIBC_2.2)[3]	fgetwc_unlocked(GLIBC_2.2)[6]	ftello(GLIBC_2.2)[6]	putchar_unlocked(GLIBC_2.2)[6]	telldir(GLIBC_2.2)[6]
_IO_puts(GLIBC_2.2)[3]	fileno(GLIBC_2.2)[6]	fwrite(GLIBC_2.2)[6]	puts(GLIBC_2.2)[6]	tempnam(GLIBC_2.2)[6]
asprintf(GLIBC_2.2)[3]	flockfile(GLIBC_2.2)[6]	getc(GLIBC_2.2)[6]	putw(GLIBC_2.2)[6]	ungetc(GLIBC_2.2)[6]
clearerr(GLIBC_2.2)[6]	fopen(GLIBC_2.2)[6]	getc_unlocked(GLIBC_2.2)[6]	remove(GLIBC_2.2)[6]	vasprintf(GLIBC_2.2)[3]
ctermid(GLIBC_2.2)[6]	fprintf(GLIBC_2.2)[6]	getchar(GLIBC_2.2)[6]	rewind(GLIBC_2.2)[6]	vdprintf(GLIBC_2.2)[3]
fclose(GLIBC_2.2)[6]	fputc(GLIBC_2.2)[6]	getchar_unlocked(GLIBC_2.2)[6]	rewinddir(GLIBC_2.2)[6]	vfprintf(GLIBC_2.2)[6]
fdopen(GLIBC_2.2)[6]	fputs(GLIBC_2.2)[6]	gets(GLIBC_2.2)[3]	scanf(GLIBC_2.2)[6]	vprintf(GLIBC_2.2)[6]
feof(GLIBC_2.2)[6]	fread(GLIBC_2.2)[6]	getw(GLIBC_2.2)[6]	seekdir(GLIBC_2.2)[6]	vsnprintf(GLIBC_2.2)[6]
ferror(GLIBC_2.2)[6]	freopen(GLIBC_2.2)[6]	pclose(GLIBC_2.2)[6]	setbuf(GLIBC_2.2)[6]	vsprintf(GLIBC_2.2)[6]
fflush(GLIBC_2.2)[6]	fscanf(GLIBC_2.2)[6]	popen(GLIBC_2.2)[6]	setbuffer(GLIBC_2.2)[3]	
fflush_unlocked(GLIBC_2.2)[6]	fseek(GLIBC_2.2)[6]	printf(GLIBC_2.2)[6]	setvbuf(GLIBC_2.2)[6]	
fgetc(GLIBC_2.2)[6]	fseeko(GLIBC_2.2)[6]	putc(GLIBC_2.2)[6]	snprintf(GLIBC_2.2)[6]	

Table 14-5. libc - Standard I/O Data Interfaces

stderr(GLIBC_2.2)[6]	stdin(GLIBC_2.2)[6]	stdout(GLIBC_2.2)[6]		

14.1.4. Signal Handling

Table 14-6. libc - Signal Handling Function Interfaces

__libc_current_sigrtmax(GLIBC_2.2)[3]	sigaddset(GLIBC_2.2)[6]	sighold(GLIBC_2.2)[6]	sigpause(GLIBC_2.2)[6]	sigsuspend(GLIBC_2.2)[6]
__libc_current_sigrtmin(GLIBC_2.2)[3]	sigaltstack(GLIBC_2.2)[6]	sigignore(GLIBC_2.2)[6]	sigpending(GLIBC_2.2)[6]	sigtimedwait(GLIBC_2.2)[6]
__sigsetjmp(GLIBC	sigandset(GLIBC_2	siginterrupt(GLIBC	sigprocmask(GLIB	sigwait(GLIBC_2.2

_2.2)[3]	.2)[3]	_2.2)[6]	C_2.2)[6]	)[6]
__sysv_signal(GLIBC_2.2)[3]	sigblock(GLIBC_2.2)[3]	sigisemptyset(GLIBC_2.2)[3]	sigqueue(GLIBC_2.2)[6]	sigwaitinfo(GLIBC_2.2)[6]
bsd_signal(GLIBC_2.2)[6]	sigdelset(GLIBC_2.2)[6]	sigismember(GLIBC_2.2)[6]	sigrelse(GLIBC_2.2)[6]	
psignal(GLIBC_2.2)[3]	sigemptyset(GLIBC_2.2)[6]	siglongjmp(GLIBC_2.2)[6]	sigreturn(GLIBC_2.2)[3]	
raise(GLIBC_2.2)[6]	sigfillset(GLIBC_2.2)[6]	signal(GLIBC_2.2)[6]	sigset(GLIBC_2.2)[6]	
sigaction(GLIBC_2.2)[6]	siggetmask(GLIBC_2.2)[3]	sigorset(GLIBC_2.2)[3]	sigstack(GLIBC_2.2)[6]	

Table 14-7. libc - Signal Handling Data Interfaces

_sys_siglist(GLIBC_2.2)[3]				

14.1.5. Localization Functions

Table 14-8. libc - Localization Functions Function Interfaces

bind_textdomain_codeset(GLIBC_2.2)[3]	catopen(GLIBC_2.2)[6]	dngettext(GLIBC_2.2)[3]	iconv_open(GLIBC_2.2)[6]	setlocale(GLIBC_2.2)[6]
bindtextdomain(GLIBC_2.2)[3]	dcgettext(GLIBC_2.2)[3]	gettext(GLIBC_2.2)[3]	localeconv(GLIBC_2.2)[6]	textdomain(GLIBC_2.2)[3]
catclose(GLIBC_2.2)[6]	dcngettext(GLIBC_2.2)[3]	iconv(GLIBC_2.2)[6]	ngettext(GLIBC_2.2)[3]	
catgets(GLIBC_2.2)[6]	dgettext(GLIBC_2.2)[3]	iconv_close(GLIBC_2.2)[6]	nl_langinfo(GLIBC_2.2)[6]	

Table 14-9. libc - Localization Functions Data Interfaces

_nl_msg_cat_cntr(GLIBC_2.2)[3]				

14.1.6. Socket Interface

Table 14-10. libc - Socket Interface Function Interfaces

__h_errno_location(GLIBC_2.2)[3]	gethostbyname_r(GLIBC_2.2)[3]	getsockopt(GLIBC_2.2)[5]	send(GLIBC_2.2)[5]	socket(GLIBC_2.2)[5]
accept(GLIBC_2.2)[5]	gethostid(GLIBC_2.2)[6]	listen(GLIBC_2.2)[5]	sendmsg(GLIBC_2.2)[5]	socketpair(GLIBC_2.2)[5]

bind(GLIBC_2.2)[5]	gethostname(GLIBC_2.2)[5]	recv(GLIBC_2.2)[5]	sendto(GLIBC_2.2)[5]	
bindresvport(GLIBC_2.2)[3]	getpeername(GLIBC_2.2)[5]	recvfrom(GLIBC_2.2)[5]	setsockopt(GLIBC_2.2)[5]	
connect(GLIBC_2.2)[5]	getsockname(GLIBC_2.2)[5]	recvmsg(GLIBC_2.2)[5]	shutdown(GLIBC_2.2)[5]	

14.1.7. Wide Characters

Table 14-11. libc - Wide Characters Function Interfaces

__wcstod_internal(GLIBC_2.2)[3]	mbsinit(GLIBC_2.2)[6]	vwscanf(GLIBC_2.2)[1]	wcsnlen(GLIBC_2.2)[3]	wcstoumax(GLIBC_2.2)[1]
__wcstof_internal(GLIBC_2.2)[3]	mbsnrtowcs(GLIBC_2.2)[3]	wcpcpy(GLIBC_2.2)[3]	wcsnrtombs(GLIBC_2.2)[3]	wcstouq(GLIBC_2.2)[3]
__wcstol_internal(GLIBC_2.2)[3]	mbsrtowcs(GLIBC_2.2)[6]	wcpncpy(GLIBC_2.2)[3]	wcspbrk(GLIBC_2.2)[1]	wcswcs(GLIBC_2.2)[6]
__wcstold_internal(GLIBC_2.2)[3]	mbstowcs(GLIBC_2.2)[6]	wcrtomb(GLIBC_2.2)[6]	wcsrchr(GLIBC_2.2)[6]	wcswidth(GLIBC_2.2)[6]
__wcstoul_internal(GLIBC_2.2)[3]	mbtowc(GLIBC_2.2)[6]	wcscasecmp(GLIBC_2.2)[3]	wcsrtombs(GLIBC_2.2)[6]	wcsxfrm(GLIBC_2.2)[6]
btowc(GLIBC_2.2)[6]	putwc(GLIBC_2.2)[1]	wcscat(GLIBC_2.2)[6]	wcsspn(GLIBC_2.2)[6]	wctob(GLIBC_2.2)[6]
fgetwc(GLIBC_2.2)[6]	putwchar(GLIBC_2.2)[1]	wcschr(GLIBC_2.2)[6]	wcsstr(GLIBC_2.2)[6]	wctomb(GLIBC_2.2)[6]
fgetws(GLIBC_2.2)[1]	swprintf(GLIBC_2.2)[6]	wcscmp(GLIBC_2.2)[6]	wcstod(GLIBC_2.2)[6]	wctrans(GLIBC_2.2)[6]
fputwc(GLIBC_2.2)[1]	swscanf(GLIBC_2.2)[1]	wcscoll(GLIBC_2.2)[6]	wcstof(GLIBC_2.2)[1]	wctype(GLIBC_2.2)[6]
fputws(GLIBC_2.2)[1]	towctrans(GLIBC_2.2)[6]	wcscpy(GLIBC_2.2)[6]	wcstoimax(GLIBC_2.2)[1]	wcwidth(GLIBC_2.2)[6]
fwide(GLIBC_2.2)[1]	towlower(GLIBC_2.2)[1]	wcscspn(GLIBC_2.2)[6]	wcstok(GLIBC_2.2)[6]	wmemchr(GLIBC_2.2)[6]
fwprintf(GLIBC_2.2)[6]	towupper(GLIBC_2.2)[6]	wcsdup(GLIBC_2.2)[3]	wcstol(GLIBC_2.2)[6]	wmemcmp(GLIBC_2.2)[6]
fwscanf(GLIBC_2.2)[1]	ungetwc(GLIBC_2.2)[1]	wcsftime(GLIBC_2.2)[1]	wcstold(GLIBC_2.2)[1]	wmemcpy(GLIBC_2.2)[6]
getwc(GLIBC_2.2)[1]	vfwprintf(GLIBC_2.2)[1]	wcslen(GLIBC_2.2)[6]	wcstoll(GLIBC_2.2)[1]	wmemmove(GLIBC_2.2)[6]

getwchar(GLIBC_2.2)[6]	vfwscanf(GLIBC_2.2)[1]	wcsncasecmp(GLIBC_2.2)[3]	wcstombs(GLIBC_2.2)[6]	wmemset(GLIBC_2.2)[6]
mblen(GLIBC_2.2)[6]	vswprintf(GLIBC_2.2)[1]	wcsncat(GLIBC_2.2)[6]	wcstoq(GLIBC_2.2)[3]	wprintf(GLIBC_2.2)[1]
mbrlen(GLIBC_2.2)[6]	vswscanf(GLIBC_2.2)[1]	wcsncmp(GLIBC_2.2)[6]	wcstoul(GLIBC_2.2)[6]	wscanf(GLIBC_2.2)[1]
mbrtowc(GLIBC_2.2)[6]	vwprintf(GLIBC_2.2)[1]	wcsncpy(GLIBC_2.2)[6]	wcstoull(GLIBC_2.2)[1]	

14.1.8. String Functions

Table 14-12. libc - String Functions Function Interfaces

__mempcpy(GLIBC_2.2)[3]	bzero(GLIBC_2.2)[6]	strcasestr(GLIBC_2.2)[3]	strncasecmp(GLIBC_2.2)[6]	strtoimax(GLIBC_2.2)[1]
__rawmemchr(GLIBC_2.2)[3]	ffs(GLIBC_2.2)[6]	strcat(GLIBC_2.2)[6]	strncat(GLIBC_2.2)[6]	strtok(GLIBC_2.2)[6]
__stpcpy(GLIBC_2.2)[3]	index(GLIBC_2.2)[6]	strchr(GLIBC_2.2)[6]	strncmp(GLIBC_2.2)[6]	strtok_r(GLIBC_2.2)[3]
__strdup(GLIBC_2.2)[3]	memccpy(GLIBC_2.2)[6]	strcmp(GLIBC_2.2)[6]	strncpy(GLIBC_2.2)[6]	strtold(GLIBC_2.2)[1]
__strtod_internal(GLIBC_2.2)[3]	memchr(GLIBC_2.2)[6]	strcoll(GLIBC_2.2)[6]	strndup(GLIBC_2.2)[3]	strtoll(GLIBC_2.2)[1]
__strtof_internal(GLIBC_2.2)[3]	memcmp(GLIBC_2.2)[6]	strcpy(GLIBC_2.2)[6]	strnlen(GLIBC_2.2)[3]	strtoq(GLIBC_2.2)[3]
__strtok_r(GLIBC_2.2)[3]	memcpy(GLIBC_2.2)[6]	strcspn(GLIBC_2.2)[6]	strpbrk(GLIBC_2.2)[6]	strtoull(GLIBC_2.2)[1]
__strtol_internal(GLIBC_2.2)[3]	memmove(GLIBC_2.2)[6]	strdup(GLIBC_2.2)[6]	strptime(GLIBC_2.2)[3]	strtoumax(GLIBC_2.2)[1]
__strtold_internal(GLIBC_2.2)[3]	memrchr(GLIBC_2.2)[3]	strerror(GLIBC_2.2)[6]	strrchr(GLIBC_2.2)[6]	strtouq(GLIBC_2.2)[3]
__strtoll_internal(GLIBC_2.2)[3]	memset(GLIBC_2.2)[6]	strerror_r(GLIBC_2.2)[3]	strsep(GLIBC_2.2)[3]	strverscmp(GLIBC_2.2)[3]
__strtoul_internal(GLIBC_2.2)[3]	rindex(GLIBC_2.2)[6]	strfmon(GLIBC_2.2)[6]	strsignal(GLIBC_2.2)[3]	strxfrm(GLIBC_2.2)[6]
__strtoull_internal(GLIBC_2.2)[3]	stpcpy(GLIBC_2.2)[3]	strfry(GLIBC_2.2)[3]	strspn(GLIBC_2.2)[6]	swab(GLIBC_2.2)[6]
bcmp(GLIBC_2.2)[6]	stpncpy(GLIBC_2.2)[3]	strftime(GLIBC_2.2)[6]	strstr(GLIBC_2.2)[6]	

bcopy(GLIBC_2.2)[6]	strcasecmp(GLIBC_2.2)[6]	strlen(GLIBC_2.2)[6]	strtof(GLIBC_2.2)[1]	

14.1.9. IPC Functions

Table 14-13. libc - IPC Functions Function Interfaces

ftok(GLIBC_2.2)[6]	msgrcv(GLIBC_2.2)[6]	semget(GLIBC_2.2)[6]	shmctl(GLIBC_2.2)[6]	
msgctl(GLIBC_2.2)[6]	msgsnd(GLIBC_2.2)[6]	semop(GLIBC_2.2)[6]	shmdt(GLIBC_2.2)[6]	
msgget(GLIBC_2.2)[6]	semctl(GLIBC_2.2)[6]	shmat(GLIBC_2.2)[6]	shmget(GLIBC_2.2)[6]	

14.1.10. Regular Expressions

Table 14-14. libc - Regular Expressions Function Interfaces

advance(GLIBC_2.2)[6]	re_exec(GLIBC_2.2)[6]	regerror(GLIBC_2.2)[6]	regfree(GLIBC_2.2)[6]	
re_comp(GLIBC_2.2)[6]	regcomp(GLIBC_2.2)[6]	regexec(GLIBC_2.2)[6]	step(GLIBC_2.2)[6]	

Table 14-15. libc - Regular Expressions Data Interfaces

loc1(GLIBC_2.2)[6]	loc2(GLIBC_2.2)[6]	locs(GLIBC_2.2)[6]		

14.1.11. Character Type Functions

Table 14-16. libc - Character Type Functions Function Interfaces

__ctype_get_mb_cur_max(GLIBC_2.2)[3]	isdigit(GLIBC_2.2)[6]	iswalnum(GLIBC_2.2)[6]	iswlower(GLIBC_2.2)[6]	toascii(GLIBC_2.2)[6 2)]

	)[6]	2)[6]	.2)[6]	
iscntrl(GLIBC_2.2)[6]	isupper(GLIBC_2.2)[6]	iswgraph(GLIBC_2.2)[6]	isxdigit(GLIBC_2.2)[6]	

Table 14-17. libc - Character Type Functions Data Interfaces

__ctype_b(GLIBC_2.2)[3]	__ctype_tolower(GLIBC_2.2)[3]	__ctype_toupper(GLIBC_2.2)[3]		

14.1.12. Time Manipulation

Table 14-18. libc - Time Manipulation Function Interfaces

adjtime(GLIBC_2.2)[3]	asctime_r(GLIBC_2.2)[6]	difftime(GLIBC_2.2)[6]	localtime(GLIBC_2.2)[6]	tzset(GLIBC_2.2)[6]
adjtimex(GLIBC_2.2)[3]	ctime(GLIBC_2.2)[6]	gmtime(GLIBC_2.2)[6]	localtime_r(GLIBC_2.2)[6]	ualarm(GLIBC_2.2)[6]
asctime(GLIBC_2.2)[6]	ctime_r(GLIBC_2.2)[6]	gmtime_r(GLIBC_2.2)[6]	mktime(GLIBC_2.2)[6]	

Table 14-19. libc - Time Manipulation Data Interfaces

__daylight(GLIBC_2.2)[3]	__tzname(GLIBC_2.2)[3]	timezone(GLIBC_2.2)[6]		
__timezone(GLIBC_2.2)[3]	daylight(GLIBC_2.2)[6]	tzname(GLIBC_2.2)[6]		

14.1.13. Terminal Interface Functions

Table 14-20. libc - Terminal Interface Functions Function Interfaces

cfgetispeed(GLIBC_2.2)[6]	cfsetispeed(GLIBC_2.2)[6]	tcdrain(GLIBC_2.2)[6]	tcgetattr(GLIBC_2.2)[6]	tcsendbreak(GLIBC_2.2)[6]
cfgetospeed(GLIBC_2.2)[6]	cfsetospeed(GLIBC_2.2)[6]	tcflow(GLIBC_2.2)[6]	tcgetpgrp(GLIBC_2.2)[6]	tcsetattr(GLIBC_2.2)[6]
cfmakeraw(GLIBC_2.2)[3]	cfsetspeed(GLIBC_2.2)[3]	tcflush(GLIBC_2.2)[6]	tcgetsid(GLIBC_2.2)[6]	tcsetpgrp(GLIBC_2.2)[6]

14.1.14. System Database Interface

Table 14-21. libc - System Database Interface Function Interfaces

endgrent(GLIBC_2.2)[6]	getgrgid(GLIBC_2.2)[6]	getprotoent(GLIBC_2.2)[6]	getutent(GLIBC_2.2)[3]	setnetent(GLIBC_2.2)[5]

endnetent(GLIBC_2.2)[5]	getgrgid_r(GLIBC_2.2.2)[6]	getpwent(GLIBC_2.2)[6]	getutent_r(GLIBC_2.2.2)[3]	setprotoent(GLIBC_2.2)[5]
endprotoent(GLIBC_2.2)[5]	getgrnam(GLIBC_2.2)[6]	getpwnam(GLIBC_2.2.2)[6]	getutxent(GLIBC_2.2)[6]	setpwent(GLIBC_2.2)[6]
endpwent(GLIBC_2.2)[6]	gethostbyaddr(GLIBC_2.2)[5]	getpwuid(GLIBC_2.2)[6]	getutxid(GLIBC_2.2)[6]	setservent(GLIBC_2.2.2)[5]
endservent(GLIBC_2.2.2)[5]	gethostbyname(GLIBC_2.2)[3]	getpwuid_r(GLIBC_2.2)[6]	getutxline(GLIBC_2.2.2)[6]	setutent(GLIBC_2.2.2)[3]
endutent(GLIBC_2.2)[6]	getnetbyaddr(GLIBC_2.2)[5]	getservbyname(GLIBC_2.2)[3]	pututxline(GLIBC_2.2.2)[6]	setutxent(GLIBC_2.2)[6]
endutxent(GLIBC_2.2.2)[6]	getprotobyname(GLIBC_2.2)[6]	getservbyport(GLIBC_2.2)[5]	setgrent(GLIBC_2.2.2)[6]	
getgrent(GLIBC_2.2)[6]	getprotobynumber(GLIBC_2.2)[5]	getservent(GLIBC_2.2.2)[3]	setgroups(GLIBC_2.2)[3]	

14.1.15. Language Support

Table 14-22. libc - Language Support Function Interfaces

__libc_start_main(GLIBC_2.2)[3]	_obstack_begin(GLIBC_2.2)[3]	_obstack_newchunk(GLIBC_2.2)[3]	obstack_free(GLIBC_2.2)[3]	

14.1.16. Large File Support

Table 14-23. libc - Large File Support Function Interfaces

__fxstat64(GLIBC_2.2.2)[3]	freopen64(GLIBC_2.2.2)[2]	ftw64(GLIBC_2.2.2)[2]	nftw64(GLIBC_2.2.2)[2]	tmpfile64(GLIBC_2.2.2)[2]
__lxstat64(GLIBC_2.2.2)[3]	fseeko64(GLIBC_2.2.2)[2]	getrlimit64(GLIBC_2.2.2)[2]	open64(GLIBC_2.2.2)[2]	truncate64(GLIBC_2.2.2)[2]
__xstat64(GLIBC_2.2.2)[3]	fsetpos64(GLIBC_2.2.2)[2]	lockf64(GLIBC_2.2.2)[2]	pread64(GLIBC_2.2.2)[2]	
creat64(GLIBC_2.2.2)[2]	fstatvfs64(GLIBC_2.2.2)[2]	lseek64(GLIBC_2.2.2)[2]	pwrite64(GLIBC_2.2.2)[2]	
fgetpos64(GLIBC_2.2.2)[2]	ftello64(GLIBC_2.2.2)[2]	mkstemp64(GLIBC_2.2.2)[2]	readdir64(GLIBC_2.2.2)[2]	
fopen64(GLIBC_2.2.2)[2]	ftruncate64(GLIBC_2.2.2)[2]	mmap64(GLIBC_2.2.2)[2]	statvfs64(GLIBC_2.2.2)[2]	

14.1.17. Standard Library

Table 14-24. libc - Standard Library Function Interfaces

_Exit(GLIBC_2.2)[1]	drand48(GLIBC_2.2)[6]	grantpt(GLIBC_2.2)[6]	lrand48(GLIBC_2.2)[6]	srand48(GLIBC_2.2)[6]
__assert_fail(GLIBC_2.2)[3]	ecvt(GLIBC_2.2)[6]	hcreate(GLIBC_2.2)[6]	lsearch(GLIBC_2.2)[6]	srandom(GLIBC_2.2)[6]
__cxa_atexit(GLIBC_2.2)[3]	erand48(GLIBC_2.2)[6]	hdestroy(GLIBC_2.2)[6]	makecontext(GLIBC_2.2)[6]	strtod(GLIBC_2.2)[6]
__errno_location(GLIBC_2.2)[3]	err(GLIBC_2.2)[3]	hsearch(GLIBC_2.2)[6]	malloc(GLIBC_2.2)[6]	strtol(GLIBC_2.2)[6]
__fpending(GLIBC_2.2)[3]	error(GLIBC_2.2)[3]	htonl(GLIBC_2.2)[5]	memmem(GLIBC_2.2)[3]	strtoul(GLIBC_2.2)[6]
__getpagesize(GLIBC_2.2)[3]	errx(GLIBC_2.2)[3]	htons(GLIBC_2.2)[5]	mkstemp(GLIBC_2.2)[6]	swapcontext(GLIBC_2.2)[6]
__isinf(GLIBC_2.2)[3]	fcvt(GLIBC_2.2)[6]	imaxabs(GLIBC_2.2)[1]	mktemp(GLIBC_2.2)[6]	syslog(GLIBC_2.2)[6]
__isinff(GLIBC_2.2)[3]	fmtmsg(GLIBC_2.2)[6]	imaxdiv(GLIBC_2.2)[1]	mrand48(GLIBC_2.2)[6]	system(GLIBC_2.2)[3]
__isinfl(GLIBC_2.2)[3]	fnmatch(GLIBC_2.2)[6]	inet_addr(GLIBC_2.2)[5]	nftw(GLIBC_2.2)[6]	tdelete(GLIBC_2.2)[6]
__isnan(GLIBC_2.2)[3]	fpathconf(GLIBC_2.2)[6]	inet_aton(GLIBC_2.2)[5]	nrand48(GLIBC_2.2)[6]	tfind(GLIBC_2.2)[6]
__isnanf(GLIBC_2.2)[3]	free(GLIBC_2.2)[6]	inet_ntoa(GLIBC_2.2)[5]	ntohl(GLIBC_2.2)[5]	tmpfile(GLIBC_2.2)[6]
__isnanl(GLIBC_2.2)[3]	freeaddrinfo(GLIBC_2.2)[7]	inet_ntop(GLIBC_2.2)[7]	ntohs(GLIBC_2.2)[5]	tmpnam(GLIBC_2.2)[6]
__sysconf(GLIBC_2.2)[3]	ftrylockfile(GLIBC_2.2)[6]	inet_pton(GLIBC_2.2)[7]	openlog(GLIBC_2.2)[6]	tsearch(GLIBC_2.2)[6]
_exit(GLIBC_2.2)[6]	ftw(GLIBC_2.2)[6]	initstate(GLIBC_2.2)[6]	perror(GLIBC_2.2)[6]	ttyname(GLIBC_2.2)[6]
_longjmp(GLIBC_2.2)[6]	funlockfile(GLIBC_2.2)[6]	insque(GLIBC_2.2)[6]	posix_memalign(GLIBC_2.2)[4]	ttyname_r(GLIBC_2.2)[6]
_setjmp(GLIBC_2.2)[6]	gai_strerror(GLIBC_2.2)[7]	isatty(GLIBC_2.2)[6]	ptsname(GLIBC_2.2)[6]	twalk(GLIBC_2.2)[6]
a64l(GLIBC_2.2)[6]	gcvt(GLIBC_2.2)[6]	isblank(GLIBC_2.2)[7]	putenv(GLIBC_2.2)[6]	unlockpt(GLIBC_2.2)[6]

abort(GLIBC_2.2)[6]	getaddrinfo(GLIBC_2.2)[7]	isinf(GLIBC_2.2)[1]	qsort(GLIBC_2.2)[6]	unsetenv(GLIBC_2.2)[3]
abs(GLIBC_2.2)[6]	getcwd(GLIBC_2.2)[6]	isinff(GLIBC_2.2)	rand(GLIBC_2.2)[6]	usleep(GLIBC_2.2)[6]
atof(GLIBC_2.2)[6]	getdate(GLIBC_2.2)[6]	isinfl(GLIBC_2.2)[1]	rand_r(GLIBC_2.2)[6]	verrx(GLIBC_2.2)[3]
atoi(GLIBC_2.2)[6]	getenv(GLIBC_2.2)[6]	isnan(GLIBC_2.2)[1]	random(GLIBC_2.2)[6]	vfscanf(GLIBC_2.2)[1]
atol(GLIBC_2.2)[6]	getlogin(GLIBC_2.2)[6]	isnanf(GLIBC_2.2)[1]	random_r(GLIBC_2.2)[3]	vscanf(GLIBC_2.2)[1]
atoll(GLIBC_2.2)[1]	getnameinfo(GLIBC_2.2)[7]	isnanl(GLIBC_2.2)[1]	realloc(GLIBC_2.2)[6]	vsscanf(GLIBC_2.2)[1]
basename(GLIBC_2.2)[6]	getopt(GLIBC_2.2)[3]	jrand48(GLIBC_2.2)[6]	realpath(GLIBC_2.2)[6]	vsyslog(GLIBC_2.2)[3]
bsearch(GLIBC_2.2)[6]	getopt_long(GLIBC_2.2)[3]	l64a(GLIBC_2.2)[6]	remque(GLIBC_2.2)[6]	warn(GLIBC_2.2)[3]
calloc(GLIBC_2.2)[6]	getopt_long_only(GLIBC_2.2)[3]	labs(GLIBC_2.2)[6]	seed48(GLIBC_2.2)[6]	warnx(GLIBC_2.2)[3]
closelog(GLIBC_2.2)[6]	getsubopt(GLIBC_2.2)[6]	lcong48(GLIBC_2.2)[6]	setenv(GLIBC_2.2)[3]	wordexp(GLIBC_2.2)[6]
confstr(GLIBC_2.2)[6]	gettimeofday(GLIBC_2.2)[6]	ldiv(GLIBC_2.2)[6]	sethostid(GLIBC_2.2)[3]	wordfree(GLIBC_2.2)[6]
cuserid(GLIBC_2.2)[6]	glob(GLIBC_2.2)[6]	lfind(GLIBC_2.2)[6]	sethostname(GLIBC_2.2)[3]	
daemon(GLIBC_2.2)[3]	glob64(GLIBC_2.2)[3]	llabs(GLIBC_2.2)[1]	setlogmask(GLIBC_2.2)[6]	
dirname(GLIBC_2.2)[6]	globfree(GLIBC_2.2)[6]	lldiv(GLIBC_2.2)[1]	setstate(GLIBC_2.2)[6]	
div(GLIBC_2.2)[6]	globfree64(GLIBC_2.2)[3]	longjmp(GLIBC_2.2)[6]	srand(GLIBC_2.2)[6]	

Table 14-25. libc - Standard Library Data Interfaces

__environ(GLIBC_2.2)[3]	_sys_errlist(GLIBC_2.2)[3]	getdate_err(GLIBC_2.2)[6]	opterr(GLIBC_2.2)[3]	optopt(GLIBC_2.2)[3]
_environ(GLIBC_2.2)[3]	environ(GLIBC_2.2)[6]	optarg(GLIBC_2.2)[6]	optind(GLIBC_2.2)[3]	

14.2. Data Definitions for libc

This section contains standard data definitions that describe system data. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content.

ISO C serves as the LSB reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

14.2.1. errno.h

```
#define EDEADLOCK       35
```

14.2.2. limits.h

```
#define ULONG_MAX       0xFFFFFFFFFFFFFFFFUL
#define LONG_MAX        9223372036854775807L
```

```
#define CHAR_MIN        0
#define CHAR_MAX        255
```

14.2.3. signal.h

```
struct sigaction
{
  union
  {
    __sighandler_t _sa_handler;
    void (*_sa_sigaction) (int, siginfo_t *, void *);
  }
  __sigaction_handler;
  unsigned long sa_flags;
  void (*sa_restorer) (void);
  __sigset_t sa_mask;
}
 ;
```

14.2.4. stddef.h

```
typedef unsigned long size_t;
```

14.2.5. sys/ioctl.h

```
#define FIONREAD        21531
#define TIOCNOTTY       21538
```

14.2.6. sys/ipc.h

```
struct ipc_perm
{
  __key_t __key;
  __uid_t uid;
  __gid_t gid;
  __uid_t cuid;
  __gid_t cgid;
  __mode_t mode;
  unsigned short __seq;
  unsigned short __pad2;
  unsigned long __unused1;
  unsigned long __unused2;
}
```

```
 ;
```

14.2.7. sys/mman.h

```
#define MCL_CURRENT     1
#define MCL_FUTURE      2
```

14.2.8. sys/msg.h

```
typedef unsigned long msgqnum_t;
typedef unsigned long msglen_t;


struct msqid_ds
{
  struct ipc_perm msg_perm;
  __time_t msg_stime;
  __time_t msg_rtime;
  __time_t msg_ctime;
  unsigned long __msg_cbytes;
  msgqnum_t msg_qnum;
  msglen_t msg_qbytes;
  __pid_t msg_lspid;
  __pid_t msg_lrpid;
  unsigned long __unused4;
  unsigned long __unused5;
}
 ;
```

14.2.9. sys/sem.h

```
struct semid_ds
{
  struct ipc_perm sem_perm;
  __time_t sem_otime;
  __time_t sem_ctime;
  unsigned long sem_nsems;
  unsigned long __unused3;
  unsigned long __unused4;
}
 ;
```

14.2.10. sys/shm.h

```
#define SHMLBA  4096
```

```
typedef unsigned long shmatt_t;
```

```
struct shmid_ds
{
  struct ipc_perm shm_perm;
  size_t shm_segsz;
  __time_t shm_atime;
  __time_t shm_dtime;
  __time_t shm_ctime;
  __pid_t shm_cpid;
  __pid_t shm_lpid;
  shmatt_t shm_nattch;
  unsigned long __unused4;
  unsigned long __unused5;
}
 ;
```

14.2.11. sys/stat.h

```
#define _STAT_VER       1
```

```
struct stat
{
  __dev_t st_dev;
  __ino_t st_ino;
  __nlink_t st_nlink;
  __mode_t st_mode;
  __uid_t st_uid;
  __gid_t st_gid;
  int pad0;
  __dev_t st_rdev;
  __off_t st_size;
  __time_t st_atime;
  long __reserved0;
  __time_t st_mtime;
  long __reserved1;
  __time_t st_ctime;
  long __reserved2;
  blksize_t st_blksize;
  __blkcnt_t st_blocks;
```

```
}
 ;
struct stat64
{
  __dev_t st_dev;
  __ino64_t st_ino;
  __nlink_t st_nlink;
  __mode_t st_mode;
  __uid_t st_uid;
  __gid_t st_gid;
  int pad0;
  __dev_t st_rdev;
  __off_t st_size;
  __time_t st_atime;
  int __reserved0;
  __time_t st_mtime;
  int __reserved1;
  __time_t st_ctime;
  int __reserved2;
  blksize_t st_blksize;
  __blkcnt64_t st_blocks;
}
 ;
```

14.2.12. sys/statvfs.h

```
struct statvfs
{
  unsigned long f_bsize;
  unsigned long f_frsize;
  __fsblkcnt64_t f_blocks;
  __fsblkcnt64_t f_bfree;
  __fsblkcnt64_t f_bavail;
  __fsfilcnt64_t f_files;
  __fsfilcnt64_t f_ffree;
  __fsfilcnt64_t f_favail;
  unsigned long f_fsid;
  int __f_unused;
  unsigned long f_flag;
  unsigned long f_namemax;
}
 ;
struct statvfs64
{
  unsigned long f_bsize;
  unsigned long f_frsize;
  __fsblkcnt64_t f_blocks;
  __fsblkcnt64_t f_bfree;
  __fsblkcnt64_t f_bavail;
```

```
  __fsfilcnt64_t f_files;
  __fsfilcnt64_t f_ffree;
  __fsfilcnt64_t f_favail;
  unsigned long f_fsid;
  int __f_unused;
  unsigned long f_flag;
  unsigned long f_namemax;
}
 ;
```

14.2.13. termios.h

```
#define CR2     1024
#define CR3     1536
#define CRDLY   1536
#define VT1     16384
#define VTDLY   16384
#define OLCUC   2
#define TAB1    2048
#define NLDLY   256
#define FF1     32768
#define FFDLY   32768
#define ONLCR   4
#define XCASE   4
#define TAB2    4096
#define CR1     512
#define IUCLC   512
#define TAB3    6144
#define TABDLY  6144
#define BS1     8192
#define BSDLY   8192
```

```
#define VSUSP   10
#define VEOL    11
#define VREPRINT        12
#define VDISCARD        13
#define VWERASE 14
#define VEOL2   16
#define VMIN    6
#define VSWTC   7
#define VSTART  8
#define VSTOP   9
```

```
#define IXON    1024
#define IXOFF   4096
```

```
#define HUPCL   1024
#define CREAD   128
#define CS6     16
#define CLOCAL  2048
#define PARENB  256
#define CS7     32
#define CS8     48
#define CSIZE   48
#define VTIME   5
#define PARODD  512
#define CSTOPB  64
```

```
#define ISIG    1
#define ECHOPRT 1024
#define NOFLSH  128
#define ECHOE   16
#define PENDIN  16384
#define ICANON  2
#define ECHOKE  2048
#define TOSTOP  256
#define ECHOK   32
#define IEXTEN  32768
#define FLUSHO  4096
#define ECHOCTL 512
#define ECHONL  64
```

14.2.14. ucontext.h

```
#define NGREG   27
```

```
typedef union
{
  double d;
  float f;
}
fpreg_t;
```

```
typedef struct
{
  unsigned int fpc;
  fpreg_t fprs[16];
}
fpregset_t;
```

```
typedef struct
{
  unsigned long mask;
  unsigned long addr;
}
__psw_t;
```

```
typedef struct
{
  __psw_t psw;
  unsigned long gregs[16];
  unsigned int aregs[16];
  fpregset_t fpregs;
}
mcontext_t;
```

```
struct ucontext
{
  unsigned long uc_flags;
  struct ucontext *uc_link;
  stack_t uc_stack;
  mcontext_t uc_mcontext;
  __sigset_t uc_sigmask;
}
 ;
```

```
typedef struct ucontext
{
  unsigned long uc_flags;
  struct ucontext *uc_link;
  stack_t uc_stack;
  mcontext_t uc_mcontext;
  __sigset_t uc_sigmask;
}
ucontext_t;
```

14.3. Interfaces for libm

Table 14-26. libm Definition

Library:	libm
SONAME:	libm.so.6

The behavior of the interfaces in this library is specified by the following standards.

ISO/IEC 9899: 1999, Programming Languages --C[10]
CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)[11]

14.3.1. Math

Table 14-27. libm - Math Function Interfaces

acos(GLIBC_2.2)[11]	cexp(GLIBC_2.2)[10]	expf(GLIBC_2.2)[10]	jnf(GLIBC_2.2)[10]	remquof(GLIBC_2.2)[10]
acosf(GLIBC_2.2)[10]	cexpf(GLIBC_2.2)[10]	expl(GLIBC_2.2)[10]	jnl(GLIBC_2.2)[10]	remquol(GLIBC_2.2)[10]
acosh(GLIBC_2.2)[11]	cexpl(GLIBC_2.2)[10]	expm1(GLIBC_2.2)[11]	ldexp(GLIBC_2.2)[11]	rint(GLIBC_2.2)[11]
acoshf(GLIBC_2.2)[10]	cimag(GLIBC_2.2)[10]	fabs(GLIBC_2.2)[11]	ldexpf(GLIBC_2.2)[10]	rintf(GLIBC_2.2)[10]
acoshl(GLIBC_2.2)[10]	cimagf(GLIBC_2.2)[10]	fabsf(GLIBC_2.2)[10]	ldexpl(GLIBC_2.2)[10]	rintl(GLIBC_2.2)[10]
acosl(GLIBC_2.2)[10]	cimagl(GLIBC_2.2)[10]	fabsl(GLIBC_2.2)[10]	lgamma(GLIBC_2.2)[11]	round(GLIBC_2.2)[10]
asin(GLIBC_2.2)[11]	clog(GLIBC_2.2)[10]	fdim(GLIBC_2.2)[10]	lgamma_r(GLIBC_2.2)[10]	roundf(GLIBC_2.2)[10]
asinf(GLIBC_2.2)[10]	clog10(GLIBC_2.2)[10]	fdimf(GLIBC_2.2)[10]	lgammaf(GLIBC_2.2)[10]	roundl(GLIBC_2.2)[10]
asinh(GLIBC_2.2)[11]	clog10f(GLIBC_2.2)[10]	fdiml(GLIBC_2.2)[10]	lgammaf_r(GLIBC_2.2)[10]	scalb(GLIBC_2.2)[11]

asinhf(GLIBC_2.2)[10]	clog10l(GLIBC_2.2)[10]	feclearexcept(GLIBC_2.2)[10]	lgammal(GLIBC_2.2)[10]	scalbf(GLIBC_2.2)[10]
asinhl(GLIBC_2.2)[10]	clogf(GLIBC_2.2)[10]	fegetenv(GLIBC_2.2)[10]	lgammal_r(GLIBC_2.2)[10]	scalbl(GLIBC_2.2)[10]
asinl(GLIBC_2.2)[10]	clogl(GLIBC_2.2)[10]	fegetexceptflag(GLIBC_2.2)[10]	llrint(GLIBC_2.2)[10]	scalbln(GLIBC_2.2)[10]
atan(GLIBC_2.2)[11]	conj(GLIBC_2.2)[10]	fegetround(GLIBC_2.2)[10]	llrintf(GLIBC_2.2)[10]	scalblnf(GLIBC_2.2)[10]
atan2(GLIBC_2.2)[11]	conjf(GLIBC_2.2)[10]	feholdexcept(GLIBC_2.2)[10]	llrintl(GLIBC_2.2)[10]	scalblnl(GLIBC_2.2)[10]
atan2f(GLIBC_2.2)[10]	conjl(GLIBC_2.2)[10]	feraiseexcept(GLIBC_2.2)[10]	llround(GLIBC_2.2)[10]	scalbn(GLIBC_2.2)[10]
atan2l(GLIBC_2.2)[10]	copysign(GLIBC_2.2)[10]	fesetenv(GLIBC_2.2)[10]	llroundf(GLIBC_2.2)[10]	scalbnf(GLIBC_2.2)[10]
atanf(GLIBC_2.2)[10]	copysignf(GLIBC_2.2)[10]	fesetexceptflag(GLIBC_2.2)[10]	llroundl(GLIBC_2.2)[10]	scalbnl(GLIBC_2.2)[10]
atanh(GLIBC_2.2)[11]	copysignl(GLIBC_2.2)[10]	fesetround(GLIBC_2.2)[10]	log(GLIBC_2.2)[11]	significand(GLIBC_2.2)[10]
atanhf(GLIBC_2.2)[10]	cos(GLIBC_2.2)[11]	fetestexcept(GLIBC_2.2)[10]	log10(GLIBC_2.2)[11]	significandf(GLIBC_2.2)[10]
atanhl(GLIBC_2.2)[10]	cosf(GLIBC_2.2)[10]	feupdateenv(GLIBC_2.2)[10]	log10f(GLIBC_2.2)[10]	significandl(GLIBC_2.2)[10]
atanl(GLIBC_2.2)[10]	cosh(GLIBC_2.2)[11]	finite(GLIBC_2.2)[11]	log10l(GLIBC_2.2)[10]	sin(GLIBC_2.2)[11]
cabs(GLIBC_2.2)[11]	coshf(GLIBC_2.2)[10]	finitef(GLIBC_2.2)[10]	log1p(GLIBC_2.2)[11]	sincos(GLIBC_2.2)[10]
cabsf(GLIBC_2.2)[10]	coshl(GLIBC_2.2)[10]	finitel(GLIBC_2.2)[10]	logb(GLIBC_2.2)[11]	sincosf(GLIBC_2.2)[10]
cabsl(GLIBC_2.2)[10]	cosl(GLIBC_2.2)[10]	floor(GLIBC_2.2)[11]	logf(GLIBC_2.2)[10]	sincosl(GLIBC_2.2)[10]
cacos(GLIBC_2.2)[10]	cpow(GLIBC_2.2)[10]	floorf(GLIBC_2.2)[10]	logl(GLIBC_2.2)[10]	sinf(GLIBC_2.2)[10]
cacosf(GLIBC_2.2)[10]	cpowf(GLIBC_2.2)[10]	floorl(GLIBC_2.2)[10]	lrint(GLIBC_2.2)[10]	sinh(GLIBC_2.2)[11]
cacosh(GLIBC_2.2)[10]	cpowl(GLIBC_2.2)[10]	fma(GLIBC_2.2)[10]	lrintf(GLIBC_2.2)[10]	sinhf(GLIBC_2.2)[10]
cacoshf(GLIBC_2.2	cproj(GLIBC_2.2)[10]	fmaf(GLIBC_2.2)[10]	lrintl(GLIBC_2.2)[10]	sinhl(GLIBC_2.2)[10]

)[10]				
cacoshl(GLIBC_2.2)[10]	cprojf(GLIBC_2.2)[10]	fmal(GLIBC_2.2)[10]	lround(GLIBC_2.2)[10]	sinl(GLIBC_2.2)[10]
cacosl(GLIBC_2.2)[10]	cprojl(GLIBC_2.2)[10]	fmax(GLIBC_2.2)[10]	lroundf(GLIBC_2.2)[10]	sqrt(GLIBC_2.2)[11]
carg(GLIBC_2.2)[10]	creal(GLIBC_2.2)[10]	fmaxf(GLIBC_2.2)[10]	lroundl(GLIBC_2.2)[10]	sqrtf(GLIBC_2.2)[10]
cargf(GLIBC_2.2)[10]	crealf(GLIBC_2.2)[10]	fmaxl(GLIBC_2.2)[10]	matherr(GLIBC_2.2)[10]	sqrtl(GLIBC_2.2)[10]
cargl(GLIBC_2.2)[10]	creall(GLIBC_2.2)[10]	fmin(GLIBC_2.2)[10]	modf(GLIBC_2.2)[11]	tan(GLIBC_2.2)[11]
casin(GLIBC_2.2)[10]	csin(GLIBC_2.2)[10]	fminf(GLIBC_2.2)[10]	modff(GLIBC_2.2)[10]	tanf(GLIBC_2.2)[10]
casinf(GLIBC_2.2)[10]	csinf(GLIBC_2.2)[10]	fminl(GLIBC_2.2)[10]	modfl(GLIBC_2.2)[10]	tanh(GLIBC_2.2)[11]
casinh(GLIBC_2.2)[10]	csinh(GLIBC_2.2)[10]	fmod(GLIBC_2.2)[11]	nan(GLIBC_2.2)[10]	tanhf(GLIBC_2.2)[10]
casinhf(GLIBC_2.2)[10]	csinhf(GLIBC_2.2)[10]	fmodf(GLIBC_2.2)[10]	nanf(GLIBC_2.2)[10]	tanhl(GLIBC_2.2)[10]
casinhl(GLIBC_2.2)[10]	csinhl(GLIBC_2.2)[10]	fmodl(GLIBC_2.2)[10]	nanl(GLIBC_2.2)[10]	tanl(GLIBC_2.2)[10]
casinl(GLIBC_2.2)[10]	csinl(GLIBC_2.2)[10]	frexp(GLIBC_2.2)[11]	nearbyint(GLIBC_2.2)[10]	tgamma(GLIBC_2.2)[10]
catan(GLIBC_2.2)[10]	csqrt(GLIBC_2.2)[10]	frexpf(GLIBC_2.2)[10]	nearbyintf(GLIBC_2.2)[10]	tgammaf(GLIBC_2.2)[10]
catanf(GLIBC_2.2)[10]	csqrtf(GLIBC_2.2)[10]	frexpl(GLIBC_2.2)[10]	nearbyintl(GLIBC_2.2)[10]	tgammal(GLIBC_2.2)[10]
catanh(GLIBC_2.2)[10]	csqrtl(GLIBC_2.2)[10]	gamma(GLIBC_2.2)[11]	nextafter(GLIBC_2.2)[11]	trunc(GLIBC_2.2)[10]
catanhf(GLIBC_2.2)[10]	ctan(GLIBC_2.2)[10]	gammaf(GLIBC_2.2)[10]	nextafterf(GLIBC_2.2)[10]	truncf(GLIBC_2.2)[10]
catanhl(GLIBC_2.2)[10]	ctanf(GLIBC_2.2)[10]	gammal(GLIBC_2.2)[10]	nextafterl(GLIBC_2.2)[10]	truncl(GLIBC_2.2)[10]
catanl(GLIBC_2.2)[10]	ctanh(GLIBC_2.2)[10]	hypot(GLIBC_2.2)[11]	nexttoward(GLIBC_2.2)[10]	y0(GLIBC_2.2)[11]
cbrt(GLIBC_2.2)[11]	ctanhf(GLIBC_2.2)[10]	hypotf(GLIBC_2.2)[10]	nexttowardf(GLIBC_2.2)[10]	y0f(GLIBC_2.2)[10]
cbrtf(GLIBC_2.2)[10]	ctanhl(GLIBC_2.2)[1]	hypotl(GLIBC_2.2)	nexttowardl(GLIBC	y0l(GLIBC_2.2)[10]

	[10]	[10]	_2.2)[10]	
cbrtl(GLIBC_2.2)[10]	ctanl(GLIBC_2.2)[10]	ilogb(GLIBC_2.2)[11]	pow(GLIBC_2.2)[10]	y1(GLIBC_2.2)[11]
ccos(GLIBC_2.2)[10]	dremf(GLIBC_2.2)[10]	ilogbf(GLIBC_2.2)[10]	pow10(GLIBC_2.2)[10]	y1f(GLIBC_2.2)[10]
ccosf(GLIBC_2.2)[10]	dreml(GLIBC_2.2)[10]	ilogbl(GLIBC_2.2)[10]	pow10f(GLIBC_2.2)[10]	y1l(GLIBC_2.2)[10]
ccosh(GLIBC_2.2)[10]	erf(GLIBC_2.2)[11]	j0(GLIBC_2.2)[11]	pow10l(GLIBC_2.2)[10]	yn(GLIBC_2.2)[11]
ccoshf(GLIBC_2.2)[10]	erfc(GLIBC_2.2)[11]	j0f(GLIBC_2.2)[10]	powf(GLIBC_2.2)[10]	ynf(GLIBC_2.2)[10]
ccoshl(GLIBC_2.2)[10]	erfcf(GLIBC_2.2)[10]	j0l(GLIBC_2.2)[10]	powl(GLIBC_2.2)[10]	ynl(GLIBC_2.2)[10]
ccosl(GLIBC_2.2)[10]	erfcl(GLIBC_2.2)[10]	j1(GLIBC_2.2)[11]	remainder(GLIBC_2.2)[11]	
ceil(GLIBC_2.2)[11]	erff(GLIBC_2.2)[10]	j1f(GLIBC_2.2)[10]	remainderf(GLIBC_2.2)[10]	
ceilf(GLIBC_2.2)[10]	erfl(GLIBC_2.2)[10]	j1l(GLIBC_2.2)[10]	remainderl(GLIBC_2.2)[10]	
ceill(GLIBC_2.2)[10]	exp(GLIBC_2.2)[11]	jn(GLIBC_2.2)[11]	remquo(GLIBC_2.2)[10]	

Table 14-28. libm - Math Data Interfaces

signgam(GLIBC_2.2)[11]				

14.4. Data Definitions for libm

This section contains standard data definitions that describe system data. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content.

ISO C serves as the LSB reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

14.5. Interfaces for libpthread

Table 14-29. libpthread Definition

Library:	libpthread

SONAME:	libpthread.so.0

The behavior of the interfaces in this library is specified by the following standards.

Linux Standard Base[12]
CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)[13]

14.5.1. Posix Threads

Table 14-30. libpthread - Posix Threads Function Interfaces

pthread_attr_destroy(GLIBC_2.2)[13]	pthread_attr_setstackaddr(GLIBC_2.2)[13]	pthread_getspecific(GLIBC_2.2)[13]	pthread_once(GLIBC_2.2)[13]	pthread_setcanceltype(GLIBC_2.2)[13]
pthread_attr_getdetachstate(GLIBC_2.2)[13]	pthread_attr_setstacksize(GLIBC_2.2)[13]	pthread_join(GLIBC_2.2)[13]	pthread_rwlock_destroy(GLIBC_2.2)[13]	pthread_setconcurrency(GLIBC_2.2)[13]
pthread_attr_getguardsize(GLIBC_2.2)[13]	pthread_cancel(GLIBC_2.2)[13]	pthread_key_create(GLIBC_2.2)[13]	pthread_rwlock_init(GLIBC_2.2)[13]	pthread_setschedparam(GLIBC_2.2)[13]
pthread_attr_getinheritsched(GLIBC_2.2)[13]	pthread_cond_broadcast(GLIBC_2.2)[13]	pthread_key_delete(GLIBC_2.2)[13]	pthread_rwlock_rdlock(GLIBC_2.2)[13]	pthread_setspecific(GLIBC_2.2)[13]
pthread_attr_getschedparam(GLIBC_2.2)[13]	pthread_cond_destroy(GLIBC_2.2)[13]	pthread_kill(GLIBC_2.2)[13]	pthread_rwlock_timedrdlock(GLIBC_2.2)[13]	pthread_sigmask(GLIBC_2.2)[13]
pthread_attr_getschedpolicy(GLIBC_2.2)[13]	pthread_cond_init(GLIBC_2.2)[13]	pthread_mutex_destroy(GLIBC_2.2)[13]	pthread_rwlock_timedwrlock(GLIBC_2.2)[13]	pthread_testcancel(GLIBC_2.2)[13]
pthread_attr_getscope(GLIBC_2.2)[13]	pthread_cond_signal(GLIBC_2.2)[13]	pthread_mutex_init(GLIBC_2.2)[13]	pthread_rwlock_tryrdlock(GLIBC_2.2)[13]	sem_close(GLIBC_2.2)[13]
pthread_attr_getstackaddr(GLIBC_2.2)[13]	pthread_cond_timedwait(GLIBC_2.2)[13]	pthread_mutex_lock(GLIBC_2.2)[13]	pthread_rwlock_trywrlock(GLIBC_2.2)[13]	sem_destroy(GLIBC_2.2)[13]
pthread_attr_getstacksize(GLIBC_2.2)[13]	pthread_cond_wait(GLIBC_2.2)[13]	pthread_mutex_trylock(GLIBC_2.2)[13]	pthread_rwlock_unlock(GLIBC_2.2)[13]	sem_getvalue(GLIBC_2.2)[13]
pthread_attr_init(GLIBC_2.2)[13]	pthread_condattr_destroy(GLIBC_2.2)[13]	pthread_mutex_unlock(GLIBC_2.2)[13]	pthread_rwlock_wrlock(GLIBC_2.2)[13]	sem_init(GLIBC_2.2)[13]
pthread_attr_setdetachstate(GLIBC_2.2)[13]	pthread_condattr_init(GLIBC_2.2)[13]	pthread_mutexattr_destroy(GLIBC_2.2)[13]	pthread_rwlockattr_destroy(GLIBC_2.2)[13]	sem_open(GLIBC_2.2)[13]
pthread_attr_setguar	pthread_create(GLI	pthread_mutexattr_	pthread_rwlockattr_	sem_post(GLIBC_2

dsize(GLIBC_2.2)[13]	BC_2.2)[13]	getpshared(GLIBC_2.2)[13]	getpshared(GLIBC_2.2)[13]	.2)[13]
pthread_attr_setinheritsched(GLIBC_2.2)[13]	pthread_detach(GLIBC_2.2)[13]	pthread_mutexattr_gettype(GLIBC_2.2)[13]	pthread_rwlockattr_init(GLIBC_2.2)[13]	sem_timedwait(GLIBC_2.2)[12]
pthread_attr_setschedparam(GLIBC_2.2)[13]	pthread_equal(GLIBC_2.2)[13]	pthread_mutexattr_init(GLIBC_2.2)[13]	pthread_rwlockattr_setpshared(GLIBC_2.2)[13]	sem_trywait(GLIBC_2.2)[13]
pthread_attr_setschedpolicy(GLIBC_2.2)[13]	pthread_exit(GLIBC_2.2)[13]	pthread_mutexattr_setpshared(GLIBC_2.2)[13]	pthread_self(GLIBC_2.2)[13]	sem_unlink(GLIBC_2.2)[13]
pthread_attr_setscope(GLIBC_2.2)[13]	pthread_getschedparam(GLIBC_2.2)[13]	pthread_mutexattr_settype(GLIBC_2.2)[13]	pthread_setcancelstate(GLIBC_2.2)[13]	sem_wait(GLIBC_2.2)[13]

14.6. Data Definitions for libpthread

This section contains standard data definitions that describe system data. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content.

ISO C serves as the LSB reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

14.7. Interfaces for libdl

Table 14-31. libdl Definition

Library:	libdl
SONAME:	libdl.so.2

The behavior of the interfaces in this library is specified by the following standards.

Linux Standard Base[14]
CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)[15]

14.7.1. Dynamic Loader

Table 14-32. libdl - Dynamic Loader Function Interfaces

dladdr(GLIBC_2.2)[14]	dlclose(GLIBC_2.2)[15]	dlerror(GLIBC_2.2)[15]	dlopen(GLIBC_2.2)[15]	dlsym(GLIBC_2.2)[15]

14.8. Data Definitions for libdl

This section contains standard data definitions that describe system data. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content.

ISO C serves as the LSB reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

14.9. Interfaces for libcrypt

Table 14-33. libcrypt Definition

Library:	libcrypt
SONAME:	libcrypt.so.1

The behavior of the interfaces in this library is specified by the following standards.

CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)[16]

14.9.1. Encryption

Table 14-34. libcrypt - Encryption Function Interfaces

crypt(GLIBC_2.2)[16]	encrypt(GLIBC_2.2)[16]	setkey(GLIBC_2.2)[16]		

14.10. Data Definitions for libcrypt

This section contains standard data definitions that describe system data. These definitions are organized into groups that correspond to system headers. This convention is used as a convenience for the reader, and does not imply the existence of these headers, or their content.

ISO C serves as the LSB reference programming language, and data definitions are specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of these data objects does not preclude their use by other programming languages.

Notes

1. ISO/IEC 9899: 1999, Programming Languages --C
2. Large File Support
3. Linux Standard Base
4. IEEE Std POSIX.1-1996 [ISO/IEC 9945-1:1996]
5. CAE Specification, February 1997, Networking Services (XNS), Issue 5(ISBN: 1-85912-165-9, C523)

6. CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)
7. The Single UNIX® Specification(SUS) Version 3
8. System V Interface Definition, Issue 3 (ISBN 0201566524)
9. System V Interface Definition,Fourth Edition
10. ISO/IEC 9899: 1999, Programming Languages --C
11. CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)
12. Linux Standard Base
13. CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)
14. Linux Standard Base
15. CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)
16. CAE Specification, January 1997, System Interfaces and Headers (XSH),Issue 5 (ISBN: 1-85912-181-0, C606)

Appendix A. Alphabetical Listing of Interfaces

Appendix B. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

B.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been

designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

B.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.4. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

B.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

B.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

B.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

B.12. How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.