

**Linux Standard Base Core Module  
Specification for PPC64 2.0.1**

## **Linux Standard Base Core Module Specification for PPC64 2.0.1**

Copyright © 2004 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 are copyrighted by the following parties:

- The Regents of the University of California
- Free Software Foundation
- Ian F. Darwin
- Paul Vixie
- BSDI (now Wind River)
- Andrew G Morgan
- Jean-loup Gailly and Mark Adler
- Massachusetts Institute of Technology

These excerpts are being used in accordance with their respective licenses.

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.

Intel and Itanium are registered trademarks and Intel386 is a trademarks of Intel Corporation.

OpenGL is a registered trademark of Silicon Graphics, Inc.

# **Specification Introduction**

## **Specification Introduction**

# Table of Contents

<b>Foreword .....</b>	<b>i</b>
<b>Introduction .....</b>	<b>ii</b>
<b>I. Introductory Elements.....</b>	<b>3</b>
1. Scope.....	1
1.1. General.....	1
1.2. Module Specific Scope.....	1
2. Normative References .....	2
3. Requirements.....	6
3.1. Relevant Libraries.....	6
3.2. LSB Implementation Conformance .....	6
3.3. LSB Application Conformance .....	7
4. Definitions.....	8
5. Terminology.....	9
6. Documentation Conventions .....	10

# List of Tables

2-1. Normative References .....	2
3-1. Standard Library Names .....	6

# Foreword

- 1 | This is version 2.0.1 of the Linux Standard Base Core Module Specification for PPC64. An implementation of this
- 2 | version of the specification may not claim to be an implementation of the Linux Standard Base unless it has
- 3 | successfully completed the compliance process as defined by the Free Standards Group.

# Introduction

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

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



# **I. Introductory Elements**



# Chapter 1. Scope

## 1.1. General

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

4 These specifications are composed of two basic parts: A common specification ("LSB-generic") describing those parts  
5 of the interface that remain constant across all implementations of the LSB, and an architecture-specific specification  
6 ("LSB-arch") describing the parts of the interface that vary by processor architecture. Together, the LSB-generic and  
7 the architecture-specific supplement for a single hardware architecture provide a complete interface specification for  
8 compiled application programs on systems that share a common hardware architecture.

9 The LSB-generic document shall be used in conjunction with an architecture-specific supplement. Whenever a section  
10 of the LSB-generic specification shall be supplemented by architecture-specific information, the LSB-generic  
11 document includes a reference to the architecture supplement. Architecture supplements may also contain additional  
12 information that is not referenced in the LSB-generic document.

13 The LSB contains both a set of Application Program Interfaces (APIs) and Application Binary Interfaces (ABIs). APIs  
14 may appear in the source code of portable applications, while the compiled binary of that application may use the  
15 larger set of ABIs. A conforming implementation shall provide all of the ABIs listed here. The compilation system  
16 may replace (e.g. by macro definition) certain APIs with calls to one or more of the underlying binary interfaces, and  
17 may insert calls to binary interfaces as needed.

18 The LSB is primarily a binary interface definition. Not all of the source level APIs available to applications may be  
19 contained in this specification.

## 1.2. Module Specific Scope

20 This is the PPC64 architecture specific Core module of the Linux Standards Base (LSB). This module supplements the  
21 generic LSB Core module with those interfaces that differ between architectures.

22 Interfaces described in this module are mandatory except where explicitly listed otherwise. Core interfaces may be  
23 supplemented by other modules; all modules are built upon the core.

## Chapter 2. Normative References

1 The specifications listed below are referenced in whole or in part by the Linux Standard Base. In this specification,  
 2 where only a particular section of one of these references is identified, then the normative reference is to that section  
 3 alone, and the rest of the referenced document is informative.

4 **Table 2-1. Normative References**

System V Application Binary Interface—DRAFT—17 December 2003	<a href="http://www.caldera.com/developers/gabi/2003-12-17/contents.html">http://www.caldera.com/developers/gabi/2003-12-17/contents.html</a>
DWARF Debugging Information Format, Revision 2.0.0 (July 27, 1993)	<a href="http://www.eagereon.com/dwarf/dwarf-2.0.0.pdf">http://www.eagereon.com/dwarf/dwarf-2.0.0.pdf</a>
Filesystem Hierarchy Standard (FHS) 2.3	<a href="http://www.pathname.com/fhs/">http://www.pathname.com/fhs/</a>
IEEE Standard 754 for Binary Floating-Point Arithmetic	<a href="http://www.ieee.org/">http://www.ieee.org/</a>
System V Application Binary Interface, Edition 4.1	<a href="http://www.caldera.com/developers/devspeecs/gabi41.pdf">http://www.caldera.com/developers/devspeecs/gabi41.pdf</a>
ISO/IEC 9899: 1999, Programming Languages—C	
Linux Assigned Names And Numbers Authority	<a href="http://www.lanana.org/">http://www.lanana.org/</a>
Large File Support	<a href="http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html">http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html</a>
LI18NUX 2000 Globalization Specification, Version 1.0 with Amendment 4	<a href="http://www.li18nux.org/docs/html/LI18NUX-2000-amd4.htm">http://www.li18nux.org/docs/html/LI18NUX-2000-amd4.htm</a>
Linux Standard Base	<a href="http://www.linuxbase.org/spec/">http://www.linuxbase.org/spec/</a>
OSF RFC 86.0	<a href="http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt">http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt</a>
64-bit PowerPC ELF ABI Supplement, Version 1.7	<a href="http://www.linuxbase.org/spec/ELF/ppc64/">http://www.linuxbase.org/spec/ELF/ppc64/</a>
The PowerPC™ Architecture: A Specification for a new family of RISC processors	<a href="http://www.austin.ibm.com">http://www.austin.ibm.com</a>
The PowerPC Architecture Book I changes	<a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg1.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg1.html</a>
The PowerPC Architecture Book II changes	<a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg2.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg2.html</a>
The PowerPC Architecture Book III changes	The PowerPC Architecture Book III changes <a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg3.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg3.html</a>
RFC 1833: Binding Protocols for ONC RPC Version 2	<a href="http://www.ietf.org/rfc/rfc1833.txt">http://www.ietf.org/rfc/rfc1833.txt</a>

RFC 1952: GZIP file format specification version 4.3	<a href="http://www.ietf.org/rfc/rfc1952.txt">http://www.ietf.org/rfc/rfc1952.txt</a>	
RFC 2440: OpenPGP Message Format	<a href="http://www.ietf.org/rfc/rfc2440.txt">http://www.ietf.org/rfc/rfc2440.txt</a>	
CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), plus Corrigendum U018	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>	
The Single UNIX® Specification(SUS) Version 2, Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>	
CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>	
ISO/IEC 9945:2003 Portable Operating System(POSIX)and The Single UNIX® Specification(SUS) V3	<a href="http://www.unix.org/version3/">http://www.unix.org/version3/</a>	
System V Interface Definition, Issue 3 (ISBN 0201566524)		
System V Interface Definition, Fourth Edition		
zlib 1.2 Manual	<a href="http://www.gzip.org/zlib/">http://www.gzip.org/zlib/</a>	
<b>Name</b>	<b>Title</b>	<b>URL</b>
64-bit PowerPC ELF ABI Supplement	64-bit PowerPC ELF ABI Supplement, Version 1.7	<a href="http://www.linuxbase.org/spec/ELF/ppc64/">http://www.linuxbase.org/spec/ELF/ppc64/</a>
DWARF Debugging Information Format	DWARF Debugging Information Format, Revision 2.0.0 (July 27, 1993)	<a href="http://www.eagercon.com/dwarf/dwarf-2.0.0.pdf">http://www.eagercon.com/dwarf/dwarf-2.0.0.pdf</a>
Filesystem Hierarchy Standard	Filesystem Hierarchy Standard (FHS) 2.3	<a href="http://www.pathname.com/fhs/">http://www.pathname.com/fhs/</a>
IEEE Std 754-1985	IEEE Standard 754 for Binary Floating-Point Arithmetic	<a href="http://www.ieee.org/">http://www.ieee.org/</a>
ISO C (1999)	ISO/IEC 9899: 1999, Programming Language --C	
ISO POSIX (2003)	ISO/IEC 9945-1:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 1: Base Definitions  ISO/IEC 9945-2:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 2: System Interfaces	<a href="http://www.unix.org/version3/">http://www.unix.org/version3/</a>

	ISO/IEC 9945-3:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 3: Shell and Utilities  ISO/IEC 9945-4:2003 Information technology -- Portable Operating System Interface (POSIX) -- Part 4: Rationale	
Large File Support	Large File Support	<a href="http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html">http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html</a>
Li18nux Globalization Specification	LI18NUX 2000 Globalization Specification, Version 1.0 with Amendment 4	<a href="http://www.li18nux.org/docs/html/LI18NUX-2000-amd4.htm">http://www.li18nux.org/docs/html/LI18NUX-2000-amd4.htm</a>
Linux Allocated Device Registry	LINUX ALLOCATED DEVICES	<a href="http://www.lanana.org/docs/device-list/devices.txt">http://www.lanana.org/docs/device-list/devices.txt</a>
PAM	Open Software Foundation, Request For Comments: 86.0 , October 1995, V. Samar & R.Schemers (SunSoft)	<a href="http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt">http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt</a>
RFC 1321: The MD5 Message-Digest Algorithm	IETF RFC 1321: The MD5 Message-Digest Algorithm	<a href="http://www.ietf.org/rfc/rfc1321.txt">http://www.ietf.org/rfc/rfc1321.txt</a>
RFC 1833: Binding Protocols for ONC RPC Version 2	IETF RFC 1833: Binding Protocols for ONC RPC Version 2	<a href="http://www.ietf.org/rfc/rfc1833.txt">http://www.ietf.org/rfc/rfc1833.txt</a>
RFC 1951: DEFLATE Compressed Data Format Specification	IETF RFC 1951: DEFLATE Compressed Data Format Specification version 1.3	<a href="http://www.ietf.org/rfc/rfc1951.txt">http://www.ietf.org/rfc/rfc1951.txt</a>
RFC 1952: GZIP File Format Specification	IETF RFC 1952: GZIP file format specification version 4.3	<a href="http://www.ietf.org/rfc/rfc1952.txt">http://www.ietf.org/rfc/rfc1952.txt</a>
RFC 2440: OpenPGP Message Format	IETF RFC 2440: OpenPGP Message Format	<a href="http://www.ietf.org/rfc/rfc2440.txt">http://www.ietf.org/rfc/rfc2440.txt</a>
SUSv2	CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>
SUSv2 Command and Utilities	The Single UNIX® Specification(SUS) Version 2, Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>
SVID Issue 3	American Telephone and Telegraph Company, System V Interface	

	Definition, Issue 3 ; Morristown, NJ, UNIX Press, 1989.(ISBN 0201566524)	
SVID Issue 4	System V Interface Definition,Fourth Edition	
System V ABI	System V Application Binary Interface, Edition 4.1	<a href="http://www.caldera.com/developers/devspecs/gabi41.pdf">http://www.caldera.com/developers/devspecs/gabi41.pdf</a>
System V ABI Update	System V Application Binary Interface - DRAFT - 17 December 2003	<a href="http://www.caldera.com/developers/gabi/2003-12-17/contents.html">http://www.caldera.com/developers/gabi/2003-12-17/contents.html</a>
The PowerPC™ Architecture	The PowerPC™ Architecture: A Specification for a new family of RISC processors	<a href="http://www.austin.ibm.com">http://www.austin.ibm.com</a>
The PowerPC™ Architecture, Book I Changes	The PowerPC Architecture Book I changes	<a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg1.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg1.html</a>
The PowerPC™ Architecture, Book II Changes	The PowerPC Architecture Book II changes	<a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg2.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg2.html</a>
The PowerPC™ Architecture, Book III Changes	The PowerPC Architecture Book III changes	The PowerPC Architecture Book III changes <a href="http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg3.html">http://www-1.ibm.com/servers/eserver/pseries/library/ppc_chg3.html</a>
this specification	Linux Standard Base	<a href="http://www.linuxbase.org/spec/">http://www.linuxbase.org/spec/</a>
X/Open Curses	CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), plus Corrigendum U018	<a href="http://www.opengroup.org/publications/catalog/un.htm">http://www.opengroup.org/publications/catalog/un.htm</a>
zlib Manual	zlib 1.2 Manual	<a href="http://www.gzip.org/zlib/">http://www.gzip.org/zlib/</a>

# Chapter 3. Requirements

## 3.1. Relevant Libraries

1 The libraries listed in Table 3-1 shall be available on PPC64 Linux Standard Base systems, with the specified runtime  
2 names. These names override or supplement the names specified in the generic LSB specification. The specified  
3 program interpreter, referred to as proginterp in this table, shall be used to load the shared libraries specified by  
4 DT\_NEEDED entries at run time.

5 **Table 3-1. Standard Library Names**

Library	Runtime Name
libm	libm.so.6
libdl	libdl.so.2
libcrypt	libcrypt.so.1
<del>libz</del>	<del>libz.so.1</del>
<del>libncurses</del>	<del>libncurses.so.5</del>
<del>libutil</del>	<del>libutil.so.1</del>
libc	libc.so.6
libpthread	libpthread.so.0
proginterp	/lib64/ld-lsb-ppc64.so.2
libgcc_s	libgcc_s.so.1
<del>libncurses</del>	<del>libncurses.so.5</del>
libz	libz.so.1
libutil	libutil.so.1

6  
7 These libraries will be in an implementation-defined directory which the dynamic linker shall search by default.

## 3.2. LSB Implementation Conformance

8 ~~An~~A conforming implementation shall satisfy the following requirements:

- 9 • The implementation shall implement fully the architecture described in the hardware manual for the target  
10 processor architecture.
- 11 • The implementation shall be capable of executing compiled applications having the format and using the system  
12 interfaces described in this document.



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

### 3.3. LSB Application Conformance

32 | ~~Ann~~A conforming application shall satisfy the following requirements:

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

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

## Chapter 4. Definitions

1 For the purposes of this document, the following definitions, as specified in the *ISO/IEC Directives, Part 2, 2001, 4th*  
2 *Edition*, apply:

3 can

4 be able to; there is a possibility of; it is possible to

5 cannot

6 be unable to; there is no possibility of; it is not possible to

7 may

8 is permitted; is allowed; is permissible

9 need not

10 it is not required that; no...is required

11 shall

12 is to; is required to; it is required that; has to; only...is permitted; it is necessary

13 shall not

14 is not allowed [permitted] [acceptable] [permissible]; is required to be not; is required that...be not; is not to be

15 should

16 it is recommended that; ought to

17 should not

18 it is not recommended that; ought not to

# Chapter 5. Terminology

1 For the purposes of this document, the following terms apply:

2 archLSB

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

5 Binary Standard

6 The total set of interfaces that are available to be used in the compiled binary code of a conforming application.

7 gLSB

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

10 implementation-defined

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

16 Shell Script

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

19 Source Standard

20 The set of interfaces that are available to be used in the source code of a conforming application.

21 undefined

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

27 unspecified

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

33 Other terms and definitions used in this document shall have the same meaning as defined in Chapter 3 of the Base  
34 Definitions volume of ISO POSIX (2003).

## Chapter 6. Documentation Conventions

Throughout this document, the following typographic conventions are used:

`function()`

the name of a function

**command**

the name of a command or utility

CONSTANT

a constant value

*parameter*

a parameter

variable

a variable

Throughout this specification, several tables of interfaces are presented. Each entry in these tables has the following format:

name

the name of the interface

(symver)

An optional symbol version identifier, if required.

[refno]

A reference number indexing the table of referenced specifications that follows this table.

For example,

forkpty(GLIBC_2.0) [1]
------------------------

refers to the interface named `forkpty` with symbol version `GLIBC_2.0` that is defined in the first of the listed references below the table.

# **ELF Specification**

2

3 **ELF Specification**

# Table of Contents

<b>I. Low Level System Information</b> .....	<b>16</b>
1. Machine Interface.....	1
1.1. Processor Architecture.....	1
1.2. Data Representation.....	1
1.3. Byte Ordering.....	1
1.4. Fundamental Types.....	1
1.5. Aggregates and Unions.....	2
1.6. Bit Fields.....	2
2. Function Calling Sequence.....	3
2.1. Registers.....	3
2.2. Stack Frame.....	3
2.3. Parameter Passing.....	3
2.4. Return Values.....	3
2.5. Function Descriptors.....	3
3. Traceback Tables.....	4
3.1. Mandatory Fields.....	4
3.2. Optional Fields.....	4
4. Process Initialization.....	5
4.1. Registers.....	5
4.2. Process Stack.....	5
5. Coding Examples.....	6
5.1. Code Model Overview.....	6
5.2. The TOC Section.....	6
5.3. TOC Assembly Language Syntax.....	6
5.4. Function Prologue and Epilogue.....	6
5.5. Register Saving and Restoring Functions.....	6
5.6. Saving General Registers Only.....	6
5.7. Saving General Registers and Floating Point Registers.....	6
5.8. Saving Floating Point Registers Only.....	6
5.9. Save and Restore Services.....	6
5.10. Data Objects.....	7
5.11. Function Calls.....	7
5.12. Branching.....	7
5.13. Dynamic Stack Space Allocation.....	7
<b>II. Object Format</b> .....	<b>8</b>
6. ELF Header.....	9
7. Special Sections.....	10
8. TOC.....	11
9. Symbol Table.....	12
9.1. Symbol Values.....	12
10. Relocation.....	13
10.1. Relocation Types.....	13

<b>III. Program Loading and Dynamic Linking .....</b>	<b>14</b>
11. Program Loading.....	15
12. Dynamic Linking.....	16
12.1. Dynamic Section.....	16
12.2. Global Offset Table .....	16
12.3. Function Addresses.....	16
12.4. Procedure Linkage Table .....	16



# List of Tables

7-1. ELF Special Sections.....	10
--------------------------------	----

# **I. Low Level System Information**

# Chapter 1. Machine Interface

## 1.1. Processor Architecture

1 The PowerPC Architecture is specified by the following documents:

- 2 • 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~
- 3 • The PowerPC™ Architecture: ~~A Specification for a new family of RISC processors~~
- 4 • The PowerPC™ Architecture, Book I ~~changes~~Changes
- 5 • The PowerPC™ Architecture, Book II ~~changes~~Changes
- 6 • The PowerPC™ Architecture, Book III ~~changes~~Changes

7 Only the features of the PowerPC processor instruction set may be assumed to be present. An application is  
8 responsible for determining if any additional instruction set features are available before using those additional  
9 features. If a feature is not present, then the application may not use it.

10 Only instructions which do not require elevated privileges may be used.

11 Applications may not make system calls directly. The interfaces in the C library must be used instead.

12 An implementation must support the 64-bit computation mode as described in The PowerPC™ Architecture: ~~A~~  
13 ~~Specification for a new family of RISC processors.~~

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

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

## 1.2. Data Representation

19 LSB-conforming applications shall use the data representation as defined in Chapter 3 of the 64-bit PowerPC ELF  
20 ABI Supplement, ~~Version 1.7~~.

## 1.3. Byte Ordering

21 LSB-conforming applications shall use big-endian byte ordering. LSB-conforming implementations may support  
22 little-endian applications.

## 1.4. Fundamental Types

23 LSB-conforming applications shall use the fundamental types as defined in Chapter 3 of the 64-bit PowerPC ELF ABI  
24 Supplement, ~~Version 1.7~~.

25 LSB-conforming applications shall not use the long double fundamental type.

## 1.5. Aggregates and Unions

26 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 1.6. Bit Fields

27 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

# Chapter 2. Function Calling Sequence

1 LSB-conforming applications shall use the function calling sequence as defined in Chapter 3 of the 64-bit PowerPC  
2 ELF ABI Supplement, ~~Version 1.7~~.

## 2.1. Registers

3 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 2.2. Stack Frame

4 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 2.3. Parameter Passing

5 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 2.4. Return Values

6 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 2.5. Function Descriptors

7 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

# Chapter 3. Traceback Tables

1 | LSB-conforming applications shall use the traceback tables as defined in Chapter 3 of the 64-bit PowerPC ELF ABI  
2 | Supplement, ~~Version 1.7~~.

## 3.1. Mandatory Fields

3 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 3.2. Optional Fields

4 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

# Chapter 4. Process Initialization

- 1 | LSB-conforming applications shall use the Operating System Interfaces as defined in Chapter 3 of the 64-bit PowerPC  
2 | ELF ABI Supplement, ~~Version 1.7~~.

## 4.1. Registers

- 3 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 4.2. Process Stack

- 4 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

# Chapter 5. Coding Examples

1 LSB-conforming applications may implement fundamental operations using the Coding Examples as defined in  
2 Chapter 3 of the 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~.

## 5.1. Code Model Overview

3 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.2. The TOC Section

4 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.3. TOC Assembly Language Syntax

5 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.4. Function Prologue and Epilogue

6 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.5. Register Saving and Restoring Functions

7 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.6. Saving General Registers Only

8 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.7. Saving General Registers and Floating Point Registers

9 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.8. Saving Floating Point Registers Only

10 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.9. Save and Restore Services

11 See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.



## 5.10. Data Objects

12 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.11. Function Calls

13 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.12. Branching

14 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## 5.13. Dynamic Stack Space Allocation

15 | See Chapter 3 of the 64-bit PowerPC ELF ABI Supplement.

## II. Object Format

- 2 LSB-conforming implementations shall support an object file , called Executable and Linking Format (ELF) as
- 3 | defined by the 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~ and as supplemented by the Linux Standard Base
- 4 Specification and this document. LSB-conforming implementations need not support tags related functionality.
- 5 LSB-conforming applications must not rely on tags related functionality.

## Chapter 6. ELF Header

1 | LSB-conforming applications shall use the ELF header as defined in 64-bit PowerPC ELF ABI Supplement, Version  
2 | 1.7, Chapter 4.

# Chapter 7. Special Sections

The following sections are defined in the 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7.~~

**Table 7-1. ELF Special Sections**

Name	Type	Attributes
.glink	SHT_PROGBITS	SHF_ALLOC+SHF_EXECINSTR
.got	SHT_PROGBITS	SHF_ALLOC+SHF_WRITE
.plt	SHT_NOBITS	SHF_ALLOC+SHF_WRITE
.sbss	SHT_NOBITS	SHF_ALLOC+SHF_WRITE
.sdata	SHT_PROGBITS	SHF_ALLOC+SHF_WRITE
.toc	SHT_PROGBITS	SHF_ALLOC+SHF_WRITE
.tocbss	SHT_NOBITS	SHF_ALLOC+SHF_WRITE

.glink

This section may be used to hold the global linkage table which aids the procedure linkage table. See Procedure Linkage Table in Chapter 5 of the processor supplement for more information

.got

This section may be used to hold the Global Offset Table, or GOT. See The Toc Section and Coding Examples in Chapter 3 and Global Offset Table in Chapter 5 of the processor supplement for more information

.plt

This section holds the procedure linkage table. See Procedure Linkage Table in Chapter 5 of the processor supplement for more information

.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.

.sdata

This section holds initialized small data that contribute to the program memory image.

.toc

This section may be used to hold the initialized Table of Contents, or TOC

.tocbss

This section may be used to hold the uninitialized portions of the TOC. This data may also be stored as zero-initialized data in a .toc section

## Chapter 8. TOC

- 1 LSB-conforming applications shall use the Table of Contents (TOC) as defined in 64-bit PowerPC ELF ABI
- 2 | Supplement, ~~Version 1.7~~, Chapter 4.

# Chapter 9. Symbol Table

- 1 | LSB-conforming applications shall use the Symbol Table as defined in Chapter 4 of the 64-bit PowerPC ELF ABI
- 2 | Supplement, ~~Version 1.7.~~

## 9.1. Symbol Values

- 3 | See Chapter 4 of the 64-bit PowerPC ELF ABI Supplement.

# Chapter 10. Relocation

- 1 | LSB-conforming applications shall use Relocations as defined in Chapter 4 of the 64-bit PowerPC ELF ABI
- 2 | Supplement, ~~Version 1.7.~~

## 10.1. Relocation Types

- 3 | See Chapter 4 of the 64-bit PowerPC ELF ABI Supplement.

## III. Program Loading and Dynamic Linking

- 2 LSB-conforming implementations shall support the object file information and system actions that create running  
3 programs as specified in the System V ~~Application Binary Interface, Edition 4.1~~ABI, 64-bit PowerPC ELF ABI  
4 Supplement, ~~Version 1.7~~ and as supplemented by the Linux Standard Base Specification and this document.



# Chapter 11. Program Loading

1 | See 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~, Chapter 5.1.

# Chapter 12. Dynamic Linking

1 | See 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~, Chapter 5.2.

## 12.1. Dynamic Section

2 | The following dynamic entries are defined in the 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~, Chapter 5.2.

3 | DT\_JMPREL

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

6 | DT\_PLTGOT

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

8 | In addition the following dynamic entries are also supported:

9 | DT\_RELACOUNT

10 |     The number of relative relocations in `.rela.dyn`

## 12.2. Global Offset Table

11 | See 64-bit PowerPC ELF ABI Supplement, ~~Version 1.7~~, Chapter 5.2.2.

## 12.3. Function Addresses

12 | See 64-bit PowerPC ELF ABI Supplement, Chapter 5.2.3.

## 12.4. Procedure Linkage Table

13 | See 64-bit PowerPC ELF ABI Supplement, Chapter 5.2.4.

# **Linux Standard Base Specification**

2

3 **Linux Standard Base Specification**

# Table of Contents

<b>I. Base Libraries</b> .....	<b>23</b>
1. Libraries .....	1
1.1. Program Interpreter/Dynamic Linker .....	1
1.2. Interfaces for libc .....	1
1.2.1. RPC .....	1
1.2.1.1. Interfaces for RPC .....	1
1.2.2. System Calls .....	3
1.2.2.1. Interfaces for System Calls .....	3
1.2.3. Standard I/O .....	5
1.2.3.1. Interfaces for Standard I/O .....	5
1.2.4. Signal Handling .....	7
1.2.4.1. Interfaces for Signal Handling .....	7
1.2.5. Localization Functions .....	8
1.2.5.1. Interfaces for Localization Functions .....	8
1.2.6. Socket Interface .....	9
1.2.6.1. Interfaces for Socket Interface .....	9
1.2.7. Wide Characters .....	10
1.2.7.1. Interfaces for Wide Characters .....	10
1.2.8. String Functions .....	12
1.2.8.1. Interfaces for String Functions .....	12
1.2.9. IPC Functions .....	13
1.2.9.1. Interfaces for IPC Functions .....	13
1.2.10. Regular Expressions .....	14
1.2.10.1. Interfaces for Regular Expressions .....	14
1.2.11. Character Type Functions .....	15
1.2.11.1. Interfaces for Character Type Functions .....	15
1.2.12. Time Manipulation .....	16
1.2.12.1. Interfaces for Time Manipulation .....	16
1.2.13. Terminal Interface Functions .....	17
1.2.13.1. Interfaces for Terminal Interface Functions .....	17
1.2.14. System Database Interface .....	17
1.2.14.1. Interfaces for System Database Interface .....	17
1.2.15. Language Support .....	18
1.2.15.1. Interfaces for Language Support .....	18
1.2.16. Large File Support .....	19
1.2.16.1. Interfaces for Large File Support .....	19
1.2.17. Standard Library .....	19
1.2.17.1. Interfaces for Standard Library .....	19
1.3. Data Definitions for libc .....	23
1.3.1. errno.h .....	23
1.3.2. inttypes.h .....	23
1.3.3. limits.h .....	23
1.3.4. setjmp.h .....	23

1.3.5. signal.h .....	23
1.3.6. stddef.h .....	24
1.3.7. sys/ioctl.h .....	24
1.3.8. sys/ipc.h .....	24
1.3.9. sys/mman.h .....	25
1.3.10. sys/msg.h .....	25
1.3.11. sys/sem.h .....	25
1.3.12. sys/shm.h .....	26
1.3.13. sys/socket.h .....	26
1.3.14. sys/stat.h .....	26
1.3.15. sys/statvfs.h .....	27
1.3.16. sys/types.h .....	28
1.3.17. termios.h .....	28
1.3.18. ucontext.h .....	29
1.3.19. unistd.h .....	29
1.3.20. utmp.h .....	30
1.3.21. utmpx.h .....	30
1.4. Interfaces for libm .....	31
1.4.1. Math .....	31
1.4.1.1. Interfaces for Math .....	31
1.5. Interfaces for libpthread .....	36
1.5.1. Realtime Threads .....	36
1.5.1.1. Interfaces for Realtime Threads .....	36
1.5.2. Advanced Realtime Threads .....	36
1.5.2.1. Interfaces for Advanced Realtime Threads .....	36
1.5.3. Posix Threads .....	36
1.5.3.1. Interfaces for Posix Threads .....	36
1.6. Interfaces for libgcc_s .....	39
1.6.1. Unwind Library .....	39
1.6.1.1. Interfaces for Unwind Library .....	39
1.7. Interface Definitions for libgcc_s .....	39
_Unwind_DeleteException .....	40
_Unwind_Find_FDE .....	40
_Unwind_ForcedUnwind .....	41
_Unwind_GetDataRelBase .....	42
_Unwind_GetGR .....	42
_Unwind_GetIP .....	42
_Unwind_GetLanguageSpecificData .....	43
_Unwind_GetRegionStart .....	43
_Unwind_GetTextRelBase .....	43
_Unwind_RaiseException .....	44
_Unwind_Resume .....	45
_Unwind_SetGR .....	45
_Unwind_SetIP .....	45
1.8. Interfaces for libdl .....	45
1.8.1. Dynamic Loader .....	46
1.8.1.1. Interfaces for Dynamic Loader .....	46
1.9. Interfaces for libcrypt .....	46

1.9.1. Encryption .....	46
1.9.1.1. Interfaces for Encryption .....	46
<b>II. Utility Libraries .....</b>	<b>48</b>
2. Libraries .....	49
2.1. Interfaces for libz .....	49
2.1.1. Compression Library .....	49
2.1.1.1. Interfaces for Compression Library .....	49
2.2. Data Definitions for libz .....	49
2.3. Interfaces for libncurses .....	49
2.3.1. Curses .....	49
2.3.1.1. Interfaces for Curses .....	49
2.4. Data Definitions for libncurses .....	49
2.4.1. curses.h .....	50
2.5. Interfaces for libutil .....	50
2.5.1. Utility Functions .....	50
2.5.1.1. Interfaces for Utility Functions .....	50
<b>A. Alphabetical Listing of Interfaces .....</b>	<b>51</b>
A.1. libgcc_s .....	51

# List of Tables

1-1. libc Definition.....	1
1-2. libc - RPC Function Interfaces .....	1
1-3. libc - System Calls Function Interfaces .....	3
1-4. libc - Standard I/O Function Interfaces .....	5
1-5. libc - Standard I/O Data Interfaces .....	7
1-6. libc - Signal Handling Function Interfaces .....	7
1-7. libc - Signal Handling Data Interfaces.....	8
1-8. libc - Localization Functions Function Interfaces .....	8
1-9. libc - Localization Functions Data Interfaces .....	9
1-10. libc - Socket Interface Function Interfaces .....	9
1-11. libc - Socket Interface Deprecated Function Interfaces .....	10
1-12. libc - Wide Characters Function Interfaces .....	10
1-13. libc - String Functions Function Interfaces.....	12
1-14. libc - IPC Functions Function Interfaces .....	13
1-15. libc - Regular Expressions Function Interfaces .....	14
1-16. libc - Regular Expressions Deprecated Function Interfaces .....	14
1-17. libc - Regular Expressions Deprecated Data Interfaces.....	15
1-18. libc - Character Type Functions Function Interfaces.....	15
1-19. libc - Time Manipulation Function Interfaces .....	16
1-20. libc - Time Manipulation Deprecated Function Interfaces .....	16
1-21. libc - Time Manipulation Data Interfaces.....	17
1-22. libc - Terminal Interface Functions Function Interfaces.....	17
1-23. libc - System Database Interface Function Interfaces.....	18
1-24. libc - Language Support Function Interfaces.....	19
1-25. libc - Large File Support Function Interfaces .....	19
1-26. libc - Standard Library Function Interfaces .....	20
1-27. libc - Standard Library Data Interfaces .....	22
1-28. libm Definition .....	31
1-29. libm - Math Function Interfaces .....	31
1-30. libm - Math Data Interfaces.....	36
1-31. libpthread Definition .....	36
1-32. libpthread - Posix Threads Function Interfaces .....	37
1-33. libgcc_s Definition .....	39
1-34. libgcc_s - Unwind Library Function Interfaces .....	39
1-35. libdl Definition .....	46
1-36. libdl - Dynamic Loader Function Interfaces .....	46
1-37. libcrypt Definition .....	46
1-38. libcrypt - Encryption Function Interfaces .....	47
2-1. libz Definition.....	49
2-2. libncurses Definition .....	49
2-3. libutil Definition .....	50
2-4. libutil - Utility Functions Function Interfaces .....	50
A-1. libgcc_s Function Interfaces .....	51



# **I. Base Libraries**



# Chapter 1. Libraries

- 1 An LSB-conforming implementation shall support base libraries which provide interfaces for accessing the operating  
 2 system, processor and other hardware in the system.
- 3 Only those interfaces that are unique to the PowerPC 64 platform are defined here. This section should be used in  
 4 conjunction with the corresponding section in the Linux Standard Base Specification.

## 1.1. Program Interpreter/Dynamic Linker

- 5 The LSB specifies the Program Interpreter to be /lib64/ld-lsb-ppc64.so.2.

## 1.2. Interfaces for libc

- 6 Table 1-1 defines the library name and shared object name for the libc library

7 **Table 1-1. libc Definition**

Library:	libc
SONAME:	libc.so.6

- 9 The behavior of the interfaces in this library is specified by the following specifications:

### Large File Support

~~Linux Standard Base~~ this specification

~~CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1 85912 181 0, C606) SUSv2~~

~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3) System V Interface Definition, SVID Issue 3 (ISBN 0201566524)~~

- 10 ~~System V Interface Definition, Fourth Edition SVID Issue 4~~

### 1.2.1. RPC

#### 1.2.1.1. Interfaces for RPC

- 11 An LSB conforming implementation shall provide the architecture specific functions for RPC specified in Table 1-2,  
 12 with the full functionality as described in the referenced underlying specification.

14 **Table 1-2. libc - RPC Function Interfaces**

<del>authnone_create(GLIBC_2.3)authnone_create(GLIBC_2.3) [1]</del>	<del>pmap_unset(GLIBC_2.3)pmap_unset(GLIBC_2.3) [2]</del>	<del>svcerr_weakauth(GLIBC_2.3)svcerr_weakauth(GLIBC_2.3) [3]</del>	<del>xdr_float(GLIBC_2.3)xdr_float(GLIBC_2.3) [3]</del>	<del>xdr_u_char(GLIBC_2.3)xdr_u_char(GLIBC_2.3) [3]</del>
<del>clnt_create(GLIBC_2.3)clnt_create(GLIBC_2.3)</del>	<del>setdomainname(GLIBC_2.3)setdomainname</del>	<del>svctcp_create(GLIBC_2.3)svctcp_create</del>	<del>xdr_free(GLIBC_2.3)xdr_free(GLIBC_2.3)</del>	<del>xdr_u_int(GLIBC_2.3)xdr_u_int(GLIBC_2.3)</del>

BC_2.3) [1]	name(GLIBC_2.3) [2]	(GLIBC_2.3) [2]	2.3) [3]	C_2.3) [2]
clnt_pcreateerror(GLIBC_2.3)clnt_pcreateerror(GLIBC_2.3) [1]	sve_getreqset(GLIBC_2.3)svc_getreqset(GLIBC_2.3) [3]	sveudp_create(GLIBC_2.3)svcudp_create(GLIBC_2.3) [2]	xdr_int(GLIBC_2.3)xdr_int(GLIBC_2.3) [3]	xdr_u_long(GLIBC_2.3)xdr_u_long(GLIBC_2.3) [3]
clnt_perrno(GLIBC_2.3)clnt_perrno(GLIBC_2.3) [1]	sve_register(GLIBC_2.3)svc_register(GLIBC_2.3) [2]	xdr_accepted_reply(GLIBC_2.3)xdr_accepted_reply(GLIBC_2.3) [3]	xdr_long(GLIBC_2.3)xdr_long(GLIBC_2.3) [3]	xdr_u_short(GLIBC_2.3)xdr_u_short(GLIBC_2.3) [3]
clnt_perror(GLIBC_2.3)clnt_perror(GLIBC_2.3) [1]	sve_run(GLIBC_2.3)svc_run(GLIBC_2.3) [2]	xdr_array(GLIBC_2.3)xdr_array(GLIBC_2.3) [3]	xdr_opaque(GLIBC_2.3)xdr_opaque(GLIBC_2.3) [3]	xdr_union(GLIBC_2.3)xdr_union(GLIBC_2.3) [3]
clnt_screateerror(GLIBC_2.3)clnt_screateerror(GLIBC_2.3) [1]	sve_sendreply(GLIBC_2.3)svc_sendreply(GLIBC_2.3) [2]	xdr_bool(GLIBC_2.3)xdr_bool(GLIBC_2.3) [3]	xdr_opaque_auth(GLIBC_2.3)xdr_opaque_auth(GLIBC_2.3) [3]	xdr_vector(GLIBC_2.3)xdr_vector(GLIBC_2.3) [3]
clnt_sperno(GLIBC_2.3)clnt_sperno(GLIBC_2.3) [1]	sveerr_auth(GLIBC_2.3)svcerr_auth(GLIBC_2.3) [3]	xdr_bytes(GLIBC_2.3)xdr_bytes(GLIBC_2.3) [3]	xdr_pointer(GLIBC_2.3)xdr_pointer(GLIBC_2.3) [3]	xdr_void(GLIBC_2.3)xdr_void(GLIBC_2.3) [3]
clnt_sperror(GLIBC_2.3)clnt_sperror(GLIBC_2.3) [1]	sveerr_decode(GLIBC_2.3)svcerr_decode(GLIBC_2.3) [3]	xdr_callhdr(GLIBC_2.3)xdr_callhdr(GLIBC_2.3) [3]	xdr_reference(GLIBC_2.3)xdr_reference(GLIBC_2.3) [3]	xdr_wrapstring(GLIBC_2.3)xdr_wrapstring(GLIBC_2.3) [3]
getdomainname(GLIBC_2.3)getdomainname(GLIBC_2.3) [2]	sveerr_noproc(GLIBC_2.3)svcerr_noproc(GLIBC_2.3) [3]	xdr_callmsg(GLIBC_2.3)xdr_callmsg(GLIBC_2.3) [3]	xdr_rejected_reply(GLIBC_2.3)xdr_rejected_reply(GLIBC_2.3) [3]	xdrmem_create(GLIBC_2.3)xdrmem_create(GLIBC_2.3) [3]
key_decryptsession(GLIBC_2.3)key_decryptsession(GLIBC_2.3) [3]	sveerr_noprog(GLIBC_2.3)svcerr_noprog(GLIBC_2.3) [3]	xdr_char(GLIBC_2.3)xdr_char(GLIBC_2.3) [3]	xdr_replymsg(GLIBC_2.3)xdr_replymsg(GLIBC_2.3) [3]	xdrrec_create(GLIBC_2.3)xdrrec_create(GLIBC_2.3) [3]
pmap_getport(GLIBC_2.3)pmap_getport(GLIBC_2.3) [2]	sveerr_progvers(GLIBC_2.3)svcerr_progvers(GLIBC_2.3) [3]	xdr_double(GLIBC_2.3)xdr_double(GLIBC_2.3) [3]	xdr_short(GLIBC_2.3)xdr_short(GLIBC_2.3) [3]	xdrrec_eof(GLIBC_2.3)xdrrec_eof(GLIBC_2.3) [3]
pmap_set(GLIBC_2.3)pmap_set(GLIBC_2.3) [2]	sveerr_systemerr(GLIBC_2.3)svcerr_systemerr(GLIBC_2.3) [3]	xdr_enum(GLIBC_2.3)xdr_enum(GLIBC_2.3) [3]	xdr_string(GLIBC_2.3)xdr_string(GLIBC_2.3) [3]	

15

16 *Referenced Specification(s)*

17 [1]. System V Interface Definition, Fourth Edition SVID Issue 4

- 18 [2]. Linux Standard Base this specification  
 19 [3]. System V Interface Definition, SVID Issue 3 (ISBN 0201566524)

## 1.2.2. System Calls

### 1.2.2.1. Interfaces for System Calls

21 An LSB conforming implementation shall provide the architecture specific functions for System Calls specified in  
 22 Table 1-3, with the full functionality as described in the referenced underlying specification.

23 **Table 1-3. libc - System Calls Function Interfaces**

<code>__fxstat(GLIBC_2.3)</code> <code>__fxstat(GLIBC_2.3)</code> [1]	<code>fchmod(GLIBC_2.3)</code> <code>fchmod(GLIBC_2.3)</code> [2]	<code>getwd(GLIBC_2.3)</code> <code>getwd(GLIBC_2.3)</code> [2]	<code>read(GLIBC_2.3)</code> <code>read(GLIBC_2.3)</code> [2]	<code>setrlimit(GLIBC_2.3)</code> <code>setrlimit(GLIBC_2.3)</code> [2]
<code>__getpgid(GLIBC_2.3)</code> <code>__getpgid(GLIBC_2.3)</code> [1]	<code>fchown(GLIBC_2.3)</code> <code>fchown(GLIBC_2.3)</code> [2]	<code>initgroups(GLIBC_2.3)</code> <code>initgroups(GLIBC_2.3)</code> [1]	<code>readdir(GLIBC_2.3)</code> <code>readdir(GLIBC_2.3)</code> [2]	<code>setrlimit64(GLIBC_2.3)</code> <code>setrlimit64(GLIBC_2.3)</code> [3]
<code>__lxstat(GLIBC_2.3)</code> <code>__lxstat(GLIBC_2.3)</code> [1]	<code>fentl(GLIBC_2.3)</code> <code>fentl(GLIBC_2.3)</code> [1]	<code>ioctl(GLIBC_2.3)</code> <code>ioctl(GLIBC_2.3)</code> [1]	<code>readdir_r(GLIBC_2.3)</code> <code>readdir_r(GLIBC_2.3)</code> [2]	<code>setsid(GLIBC_2.3)</code> <code>setsid(GLIBC_2.3)</code> [2]
<code>__xmknod(GLIBC_2.3)</code> <code>__xmknod(GLIBC_2.3)</code> [1]	<code>fdatasync(GLIBC_2.3)</code> <code>fdatasync(GLIBC_2.3)</code> [2]	<code>kill(GLIBC_2.3)</code> <code>kill(GLIBC_2.3)</code> [1]	<code>readlink(GLIBC_2.3)</code> <code>readlink(GLIBC_2.3)</code> [2]	<code>setuid(GLIBC_2.3)</code> <code>setuid(GLIBC_2.3)</code> [2]
<code>__xstat(GLIBC_2.3)</code> <code>__xstat(GLIBC_2.3)</code> [1]	<code>flock(GLIBC_2.3)</code> <code>flock(GLIBC_2.3)</code> [1]	<code>killpg(GLIBC_2.3)</code> <code>killpg(GLIBC_2.3)</code> [2]	<code>readv(GLIBC_2.3)</code> <code>readv(GLIBC_2.3)</code> [2]	<code>sleep(GLIBC_2.3)</code> <code>sleep(GLIBC_2.3)</code> [2]
<code>access(GLIBC_2.3)</code> <code>access(GLIBC_2.3)</code> [2]	<code>fork(GLIBC_2.3)</code> <code>fork(GLIBC_2.3)</code> [2]	<code>lchown(GLIBC_2.3)</code> <code>lchown(GLIBC_2.3)</code> [2]	<code>rename(GLIBC_2.3)</code> <code>rename(GLIBC_2.3)</code> [2]	<code>statvfs(GLIBC_2.3)</code> <code>statvfs(GLIBC_2.3)</code> [2]
<code>aect(GLIBC_2.3)</code> <code>aect(GLIBC_2.3)</code> [1]	<code>fstatvfs(GLIBC_2.3)</code> <code>fstatvfs(GLIBC_2.3)</code> [2]	<code>link(GLIBC_2.3)</code> <code>link(GLIBC_2.3)</code> [2]	<code>rmdir(GLIBC_2.3)</code> <code>rmdir(GLIBC_2.3)</code> [2]	<code>stime(GLIBC_2.3)</code> <code>stime(GLIBC_2.3)</code> [1]
<code>alarm(GLIBC_2.3)</code> <code>alarm(GLIBC_2.3)</code> [2]	<code>fsync(GLIBC_2.3)</code> <code>fsync(GLIBC_2.3)</code> [2]	<code>lockf(GLIBC_2.3)</code> <code>lockf(GLIBC_2.3)</code> [2]	<code>sbrk(GLIBC_2.3)</code> <code>sbrk(GLIBC_2.3)</code> [4]	<code>symlink(GLIBC_2.3)</code> <code>symlink(GLIBC_2.3)</code> [2]
<code>brk(GLIBC_2.3)</code> <code>brk(GLIBC_2.3)</code> [4]	<code>ftime(GLIBC_2.3)</code> <code>ftime(GLIBC_2.3)</code> [2]	<code>lseek(GLIBC_2.3)</code> <code>lseek(GLIBC_2.3)</code> [2]	<code>sched_get_priority_max(GLIBC_2.3)</code> <code>sched_get_priority_max(GLIBC_2.3)</code> [2]	<code>sync(GLIBC_2.3)</code> <code>sync(GLIBC_2.3)</code> [2]
<code>chdir(GLIBC_2.3)</code> <code>chdir(GLIBC_2.3)</code> [2]	<code>ftruncate(GLIBC_2.3)</code> <code>ftruncate(GLIBC_2.3)</code> [2]	<code>mkdir(GLIBC_2.3)</code> <code>mkdir(GLIBC_2.3)</code> [2]	<code>sched_get_priority_min(GLIBC_2.3)</code> <code>sched_get_priority_min(GLIBC_2.3)</code> [2]	<code>sysconf(GLIBC_2.3)</code> <code>sysconf(GLIBC_2.3)</code> [2]

			n(GLIBC_2.3) [2]	
chmod(GLIBC_2.3) chmod(GLIBC_2.3) [2]	getecontext(GLIBC_2.3) getecontext(GLIBC_2.3) [2]	mkfifo(GLIBC_2.3) mkfifo(GLIBC_2.3) [2]	sched_getparam(GLIBC_2.3) sched_getparam(GLIBC_2.3) [2]	time(GLIBC_2.3) time(GLIBC_2.3) [2]
chown(GLIBC_2.3) chown(GLIBC_2.3) [2]	getegid(GLIBC_2.3) getegid(GLIBC_2.3) [2]	mlock(GLIBC_2.3) mlock(GLIBC_2.3) [2]	sched_getscheduler(GLIBC_2.3) sched_getscheduler(GLIBC_2.3) [2]	times(GLIBC_2.3) times(GLIBC_2.3) [2]
chroot(GLIBC_2.3) chroot(GLIBC_2.3) [4]	geteuid(GLIBC_2.3) geteuid(GLIBC_2.3) [2]	mlockall(GLIBC_2.3) mlockall(GLIBC_2.3) [2]	sched_rr_get_interval(GLIBC_2.3) sched_rr_get_interval(GLIBC_2.3) [2]	truncate(GLIBC_2.3) truncate(GLIBC_2.3) [2]
clock(GLIBC_2.3) clock(GLIBC_2.3) [2]	getgid(GLIBC_2.3) getgid(GLIBC_2.3) [2]	mmap(GLIBC_2.3) mmap(GLIBC_2.3) [2]	sched_setparam(GLIBC_2.3) sched_setparam(GLIBC_2.3) [2]	ulimit(GLIBC_2.3) ulimit(GLIBC_2.3) [2]
close(GLIBC_2.3) close(GLIBC_2.3) [2]	getgroups(GLIBC_2.3) getgroups(GLIBC_2.3) [2]	mprotect(GLIBC_2.3) mprotect(GLIBC_2.3) [2]	sched_setscheduler(GLIBC_2.3) sched_setscheduler(GLIBC_2.3) [2]	umask(GLIBC_2.3) umask(GLIBC_2.3) [2]
closedir(GLIBC_2.3) closedir(GLIBC_2.3) [2]	getitimer(GLIBC_2.3) getitimer(GLIBC_2.3) [2]	msync(GLIBC_2.3) msync(GLIBC_2.3) [2]	sched_yield(GLIBC_2.3) sched_yield(GLIBC_2.3) [2]	uname(GLIBC_2.3) uname(GLIBC_2.3) [2]
creat(GLIBC_2.3) creat(GLIBC_2.3) [1]	getloadavg(GLIBC_2.3) getloadavg(GLIBC_2.3) [1]	munlock(GLIBC_2.3) munlock(GLIBC_2.3) [2]	select(GLIBC_2.3) select(GLIBC_2.3) [2]	unlink(GLIBC_2.3) unlink(GLIBC_2.3) [1]
dup(GLIBC_2.3) dup(GLIBC_2.3) [2]	getpagesize(GLIBC_2.3) getpagesize(GLIBC_2.3) [4]	munlockall(GLIBC_2.3) munlockall(GLIBC_2.3) [2]	setecontext(GLIBC_2.3) setecontext(GLIBC_2.3) [2]	utime(GLIBC_2.3) utime(GLIBC_2.3) [2]
dup2(GLIBC_2.3) dup2(GLIBC_2.3) [2]	getpgid(GLIBC_2.3) getpgid(GLIBC_2.3) [2]	munmap(GLIBC_2.3) munmap(GLIBC_2.3) [2]	setegid(GLIBC_2.3) setegid(GLIBC_2.3) [2]	utimes(GLIBC_2.3) utimes(GLIBC_2.3) [2]
execl(GLIBC_2.3) execl(GLIBC_2.3) [2]	getpgrp(GLIBC_2.3) getpgrp(GLIBC_2.3) [2]	nanosleep(GLIBC_2.3) nanosleep(GLIBC_2.3) [2]	seteuid(GLIBC_2.3) seteuid(GLIBC_2.3) [2]	vfork(GLIBC_2.3) vfork(GLIBC_2.3) [2]
execle(GLIBC_2.3) execle(GLIBC_2.3) [2]	getpid(GLIBC_2.3) getpid(GLIBC_2.3) [2]	nice(GLIBC_2.3) nice(GLIBC_2.3) [2]	setgid(GLIBC_2.3) setgid(GLIBC_2.3) [2]	wait(GLIBC_2.3) wait(GLIBC_2.3) [2]
execlp(GLIBC_2.3) execlp(GLIBC_2.3)	getppid(GLIBC_2.3) getppid(GLIBC_2.3)	open(GLIBC_2.3) open(GLIBC_2.3) [1]	setitimer(GLIBC_2.3) setitimer(GLIBC_2.3)	wait3(GLIBC_2.3) wait3(GLIBC_2.3)

[2]	3) [2]		2.3) [2]	[1]
<code>execv(GLIBC_2.3)</code> <code>execv(GLIBC_2.3)</code> [2]	<code>getpriority(GLIBC_2.3)</code> <code>getpriority(GLIBC_2.3)</code> [2]	<code>opendir(GLIBC_2.3)</code> <code>opendir(GLIBC_2.3)</code> [2]	<code>setpgid(GLIBC_2.3)</code> <code>setpgid(GLIBC_2.3)</code> [2]	<code>wait4(GLIBC_2.3)</code> <code>wait4(GLIBC_2.3)</code> [1]
<code>execve(GLIBC_2.3)</code> <code>execve(GLIBC_2.3)</code> [2]	<code>getrlimit(GLIBC_2.3)</code> <code>getrlimit(GLIBC_2.3)</code> [2]	<code>pathconf(GLIBC_2.3)</code> <code>pathconf(GLIBC_2.3)</code> [2]	<code>setpgrp(GLIBC_2.3)</code> <code>setpgrp(GLIBC_2.3)</code> [2]	<code>waitpid(GLIBC_2.3)</code> <code>waitpid(GLIBC_2.3)</code> [1]
<code>execvp(GLIBC_2.3)</code> <code>execvp(GLIBC_2.3)</code> [2]	<code>getrusage(GLIBC_2.3)</code> <code>getrusage(GLIBC_2.3)</code> [2]	<code>pause(GLIBC_2.3)</code> <code>pause(GLIBC_2.3)</code> [2]	<code>setpriority(GLIBC_2.3)</code> <code>setpriority(GLIBC_2.3)</code> [2]	<code>write(GLIBC_2.3)</code> <code>write(GLIBC_2.3)</code> [2]
<code>exit(GLIBC_2.3)</code> <code>exit(GLIBC_2.3)</code> [2]	<code>getsid(GLIBC_2.3)</code> <code>getsid(GLIBC_2.3)</code> [2]	<code>pipe(GLIBC_2.3)</code> <code>pipe(GLIBC_2.3)</code> [2]	<code>setregid(GLIBC_2.3)</code> <code>setregid(GLIBC_2.3)</code> [2]	<code>writew(GLIBC_2.3)</code> <code>writew(GLIBC_2.3)</code> [2]
<code>fcntl(GLIBC_2.3)</code> <code>fcntl(GLIBC_2.3)</code> [2]	<code>getuid(GLIBC_2.3)</code> <code>getuid(GLIBC_2.3)</code> [2]	<code>poll(GLIBC_2.3)</code> <code>poll(GLIBC_2.3)</code> [2]	<code>setreuid(GLIBC_2.3)</code> <code>setreuid(GLIBC_2.3)</code> [2]	

24

25 *Referenced Specification(s)*

26 [1]. Linux Standard Base this specification

27 [2]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))  
28 V3)

29 [3]. Large File Support

30 [4]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1 85912 181 0,  
31 €606) SUSv2

## 1.2.3. Standard I/O

## 32 1.2.3.1. Interfaces for Standard I/O

33 An LSB conforming implementation shall provide the architecture specific functions for Standard I/O specified in  
34 Table 1-4, with the full functionality as described in the referenced underlying specification.35 **Table 1-4. libc - Standard I/O Function Interfaces**

<code>_IO_feof(GLIBC_2.3)</code> <code>_IO_feof(GLIBC_2.3)</code> [1]	<code>fgetpos(GLIBC_2.3)</code> <code>fgetpos(GLIBC_2.3)</code> [2]	<code>fsetpos(GLIBC_2.3)</code> <code>fsetpos(GLIBC_2.3)</code> [2]	<code>putchar(GLIBC_2.3)</code> <code>putchar(GLIBC_2.3)</code> [2]	<code>sscanf(GLIBC_2.3)</code> <code>sscanf(GLIBC_2.3)</code> [2]
<code>_IO_getc(GLIBC_2.3)</code> <code>_IO_getc(GLIBC_2.3)</code> [1]	<code>fgets(GLIBC_2.3)</code> <code>gets(GLIBC_2.3)</code> [2]	<code>ftell(GLIBC_2.3)</code> <code>ftell(GLIBC_2.3)</code> [2]	<code>putchar_unlocked(GLIBC_2.3)</code> <code>putchar_unlocked(GLIBC_2.3)</code> [2]	<code>telldir(GLIBC_2.3)</code> <code>telldir(GLIBC_2.3)</code> [2]
<code>_IO_putc(GLIBC_2.3)</code> <code>_IO_putc(GLIBC_2.3)</code> [1]	<code>fgetwc_unlocked(GLIBC_2.3)</code> <code>fgetwc_u</code>	<code>ftello(GLIBC_2.3)</code> <code>ftello(GLIBC_2.3)</code>	<code>puts(GLIBC_2.3)</code> <code>puts(GLIBC_2.3)</code> [2]	<code>tempnam(GLIBC_2.3)</code> <code>tempnam(GLIBC_2.3)</code>

<del>_2.3) [1]</del>	<del>nlocked(GLIBC_2.3) [1]</del>	<del>[2]</del>		<del>_2.3) [2]</del>
<del>_IO_puts(GLIBC_2.3)_IO_puts(GLIBC_2.3) [1]</del>	<del>fileno(GLIBC_2.3)fileno(GLIBC_2.3) [2]</del>	<del>fwrite(GLIBC_2.3)fwrite(GLIBC_2.3) [2]</del>	<del>putw(GLIBC_2.3)putw(GLIBC_2.3) [3]</del>	<del>ungetc(GLIBC_2.3)ungetc(GLIBC_2.3) [2]</del>
<del>asprintf(GLIBC_2.3)asprintf(GLIBC_2.3) [1]</del>	<del>flockfile(GLIBC_2.3)flockfile(GLIBC_2.3) [2]</del>	<del>getc(GLIBC_2.3)getc(GLIBC_2.3) [2]</del>	<del>remove(GLIBC_2.3)remove(GLIBC_2.3) [2]</del>	<del>vasprintf(GLIBC_2.3)vasprintf(GLIBC_2.3) [1]</del>
<del>clearerr(GLIBC_2.3)clearerr(GLIBC_2.3) [2]</del>	<del>fopen(GLIBC_2.3)fopen(GLIBC_2.3) [1]</del>	<del>getc_unlocked(GLIBC_2.3)getc_unlocked(GLIBC_2.3) [2]</del>	<del>rewind(GLIBC_2.3)rewind(GLIBC_2.3) [2]</del>	<del>vdprintf(GLIBC_2.3)vdprintf(GLIBC_2.3) [1]</del>
<del>etermid(GLIBC_2.3)ctermid(GLIBC_2.3) [2]</del>	<del>fprintf(GLIBC_2.3)fprintf(GLIBC_2.3) [2]</del>	<del>getchar(GLIBC_2.3)getchar(GLIBC_2.3) [2]</del>	<del>rewinddir(GLIBC_2.3)rewinddir(GLIBC_2.3) [2]</del>	<del>vfprintf(GLIBC_2.3)vfprintf(GLIBC_2.3) [2]</del>
<del>fclose(GLIBC_2.3)fclose(GLIBC_2.3) [2]</del>	<del>fputc(GLIBC_2.3)putc(GLIBC_2.3) [2]</del>	<del>getchar_unlocked(GLIBC_2.3)getchar_unlocked(GLIBC_2.3) [2]</del>	<del>scanf(GLIBC_2.3)scanf(GLIBC_2.3) [2]</del>	<del>vprintf(GLIBC_2.3)vprintf(GLIBC_2.3) [2]</del>
<del>fdopen(GLIBC_2.3)fdopen(GLIBC_2.3) [2]</del>	<del>fputs(GLIBC_2.3)fputs(GLIBC_2.3) [2]</del>	<del>getw(GLIBC_2.3)getw(GLIBC_2.3) [3]</del>	<del>seekdir(GLIBC_2.3)seekdir(GLIBC_2.3) [2]</del>	<del>vsprintf(GLIBC_2.3)vsprintf(GLIBC_2.3) [2]</del>
<del>feof(GLIBC_2.3)feof(GLIBC_2.3) [2]</del>	<del>fread(GLIBC_2.3)fread(GLIBC_2.3) [2]</del>	<del>pclose(GLIBC_2.3)pclose(GLIBC_2.3) [2]</del>	<del>setbuf(GLIBC_2.3)setbuf(GLIBC_2.3) [2]</del>	<del>vsprintf(GLIBC_2.3)vsprintf(GLIBC_2.3) [2]</del>
<del>ferror(GLIBC_2.3)ferror(GLIBC_2.3) [2]</del>	<del>freopen(GLIBC_2.3)freopen(GLIBC_2.3) [1]</del>	<del>popen(GLIBC_2.3)popen(GLIBC_2.3) [2]</del>	<del>setbuffer(GLIBC_2.3)setbuffer(GLIBC_2.3) [1]</del>	
<del>fflush(GLIBC_2.3)fflush(GLIBC_2.3) [2]</del>	<del>fscanf(GLIBC_2.3)fscanf(GLIBC_2.3) [2]</del>	<del>printf(GLIBC_2.3)printf(GLIBC_2.3) [2]</del>	<del>setvbuf(GLIBC_2.3)setvbuf(GLIBC_2.3) [2]</del>	
<del>fflush_unlocked(GLIBC_2.3)fflush_unlocked(GLIBC_2.3) [1]</del>	<del>fseek(GLIBC_2.3)fseek(GLIBC_2.3) [2]</del>	<del>putc(GLIBC_2.3)putc(GLIBC_2.3) [2]</del>	<del>snprintf(GLIBC_2.3)snprintf(GLIBC_2.3) [2]</del>	
<del>fgetc(GLIBC_2.3)fgetc(GLIBC_2.3) [2]</del>	<del>fseeko(GLIBC_2.3)fseeko(GLIBC_2.3) [2]</del>	<del>putc_unlocked(GLIBC_2.3)putc_unlocked(GLIBC_2.3) [2]</del>	<del>sprintf(GLIBC_2.3)sprintf(GLIBC_2.3) [2]</del>	

36

37 *Referenced Specification(s)*38 [1]. ~~Linux Standard Bas~~this specification



39 [2]. ISO/IEC 9945: POSIX (2003-Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)  
40 V3)

41 [3]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1 85912 181 0,  
42 €606) SUSv2

43 An LSB conforming implementation shall provide the architecture specific data interfaces for Standard I/O specified  
44 in Table 1-5, with the full functionality as described in the referenced underlying specification.

45 **Table 1-5. libc - Standard I/O Data Interfaces**

stderr(GLIBC_2.3)s tderr(GLIBC_2.3) [1]	stdin(GLIBC_2.3)st din(GLIBC_2.3) [1]	stdout(GLIBC_2.3)s tdout(GLIBC_2.3) [1]		
---	--	---	--	--

47 *Referenced Specification(s)*

48 [1]. ISO/IEC 9945: POSIX (2003-Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)  
49 V3)

## 1.2.4. Signal Handling

### 1.2.4.1. Interfaces for Signal Handling

51 An LSB conforming implementation shall provide the architecture specific functions for Signal Handling specified in  
52 Table 1-6, with the full functionality as described in the referenced underlying specification.

53 **Table 1-6. libc - Signal Handling Function Interfaces**

__libc_current_sigrt max(GLIBC_2.3) __ libc_current_sigrtm ax(GLIBC_2.3) [1]	sigaddset(GLIBC_2 )sigaddset(GLIBC _2.3) [2]	sighold(GLIBC_2.3 )sighold(GLIBC_2. 3) [2]	sigpause(GLIBC_2. )sigpause(GLIBC_ 2.3) [2]	sigsuspend(GLIBC_ 2.3)sigsuspend(GLI BC_2.3) [2]
__libc_current_sigrt min(GLIBC_2.3) __l ibc_current_sigrtmi n(GLIBC_2.3) [1]	sigaltstack(GLIBC_ 2.3)sigaltstack(GLI BC_2.3) [2]	sigignore(GLIBC_2 )sigignore(GLIBC _2.3) [2]	sigpending(GLIBC_ 2.3)sigpending(GLI BC_2.3) [2]	sigtimedwait(GLIB C_2.3)sigtimedwait( GLIBC_2.3) [2]
__sigsetjmp(GLIBC _2.3) __sigsetjmp(G LIBC_2.3) [1]	sigandset(GLIBC_2 )sigandset(GLIBC _2.3) [1]	siginterrupt(GLIBC _2.3)siginterrupt(G LIBC_2.3) [2]	sigprocmask(GLIB C_2.3)sigprocmask( GLIBC_2.3) [2]	sigwait(GLIBC_2.3 )sigwait(GLIBC_2. 3) [2]
__sysv_signal(GLI BC_2.3) __sysv_sig nal(GLIBC_2.3) [1]	sigblock(GLIBC_2. )sigblock(GLIBC_ 2.3) [1]	sigisemptyset(GLIB C_2.3)sigisemptyset (GLIBC_2.3) [1]	sigqueue(GLIBC_2. )sigqueue(GLIBC_ 2.3) [2]	sigwaitinfo(GLIBC _2.3)sigwaitinfo(GL IBC_2.3) [2]
bsd_signal(GLIBC_ 2.3)bsd_signal(GLI BC_2.3) [2]	sigdelset(GLIBC_2. )sigdelset(GLIBC_ 2.3) [2]	sigismember(GLIB C_2.3)sigismember( GLIBC_2.3) [2]	sigrelse(GLIBC_2.3 )sigrelse(GLIBC_2. 3) [2]	
psignal(GLIBC_2.3 )psignal(GLIBC_2.	sigemptyset(GLIBC _2.3)sigemptyset(G	siglongjmp(GLIBC _2.3)siglongjmp(GL	sigreturn(GLIBC_2. )sigreturn(GLIBC_	

3) [1]	LIBC_2.3) [2]	IBC_2.3) [2]	2.3) [1]	
<del>raise(GLIBC_2.3)</del> raise(GLIBC_2.3) [2]	<del>sigfillset(GLIBC_2.3)</del> sigfillset(GLIBC_2.3) [2]	<del>signal(GLIBC_2.3)</del> signal(GLIBC_2.3) [2]	<del>sigset(GLIBC_2.3)</del> sigset(GLIBC_2.3) [2]	
<del>sigaction(GLIBC_2.3)</del> sigaction(GLIBC_2.3) [2]	<del>siggetmask(GLIBC_2.3)</del> siggetmask(GLIBC_2.3) [1]	<del>sigorset(GLIBC_2.3)</del> sigorset(GLIBC_2.3) [1]	<del>sigstack(GLIBC_2.3)</del> sigstack(GLIBC_2.3) [3]	

54

55 *Referenced Specification(s)*56 [1]. ~~Linux Standard Base~~this specification57 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~58 ~~V3)~~59 [3]. ~~CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1 85912 181 0,~~60 ~~€606)SUSv2~~

61 An LSB conforming implementation shall provide the architecture specific data interfaces for Signal Handling

62 specified in Table 1-7, with the full functionality as described in the referenced underlying specification.

63 **Table 1-7. libc - Signal Handling Data Interfaces**

<del>_sys_siglist(GLIBC_2.3)</del> _sys_siglist(GLIBC_2.3) [1]				
--	--	--	--	--

64

65 *Referenced Specification(s)*66 [1]. ~~Linux Standard Base~~this specification

## 1.2.5. Localization Functions

### 1.2.5.1. Interfaces for Localization Functions

68 An LSB conforming implementation shall provide the architecture specific functions for Localization Functions

69 specified in Table 1-8, with the full functionality as described in the referenced underlying specification.

70 **Table 1-8. libc - Localization Functions Function Interfaces**

<del>bind_textdomain_codeset(GLIBC_2.3)</del> bind_textdomain_codeset(GLIBC_2.3) [1]	<del>catopen(GLIBC_2.3)</del> catopen(GLIBC_2.3) [2]	<del>dngettext(GLIBC_2.3)</del> dngettext(GLIBC_2.3) [1]	<del>iconv_open(GLIBC_2.3)</del> iconv_open(GLIBC_2.3) [2]	<del>setlocale(GLIBC_2.3)</del> setlocale(GLIBC_2.3) [2]
<del>bindtextdomain(GLIBC_2.3)</del> bindtextdomain(GLIBC_2.3) [1]	<del>dcgettext(GLIBC_2.3)</del> dcgettext(GLIBC_2.3) [1]	<del>gettext(GLIBC_2.3)</del> gettext(GLIBC_2.3) [1]	<del>localeconv(GLIBC_2.3)</del> localeconv(GLIBC_2.3) [2]	<del>textdomain(GLIBC_2.3)</del> textdomain(GLIBC_2.3) [1]

<del>eatclose(GLIBC_2.3)</del> catclose(GLIBC_2.3) [2]	<del>dcgettext(GLIBC_2.3)</del> dcgettext(GLIBC_2.3) [1]	<del>iconv(GLIBC_2.3)</del> iconv(GLIBC_2.3) [2]	<del>ngettext(GLIBC_2.3)</del> ngettext(GLIBC_2.3) [1]	
<del>catgets(GLIBC_2.3)</del> catgets(GLIBC_2.3) [2]	<del>dgettext(GLIBC_2.3)</del> dgettext(GLIBC_2.3) [1]	<del>iconv_close(GLIBC_2.3)</del> iconv_close(GLIBC_2.3) [2]	<del>nl_langinfo(GLIBC_2.3)</del> nl_langinfo(GLIBC_2.3) [2]	

71

72 *Referenced Specification(s)*73 [1]. ~~Linux Standard Base~~this specification74 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~75 ~~VS3)~~76 An LSB conforming implementation shall provide the architecture specific data interfaces for Localization Functions  
77 specified in Table 1-9, with the full functionality as described in the referenced underlying specification.78 **Table 1-9. libc - Localization Functions Data Interfaces**

<del>_nl_msg_cat_cntr(GLIBC_2.3)</del> _nl_msg_cat_cntr(GLIBC_2.3) [1]				
---	--	--	--	--

79

80 *Referenced Specification(s)*81 [1]. ~~Linux Standard Base~~this specification

## 1.2.6. Socket Interface

### 1.2.6.1. Interfaces for Socket Interface

83 An LSB conforming implementation shall provide the architecture specific functions for Socket Interface specified in  
84 Table 1-10, with the full functionality as described in the referenced underlying specification.85 **Table 1-10. libc - Socket Interface Function Interfaces**

<del>__h_errno_location(GLIBC_2.3)</del> __h_errno_location(GLIBC_2.3) [1]	<del>gethostid(GLIBC_2.3)</del> gethostid(GLIBC_2.3) [2]	<del>listen(GLIBC_2.3)</del> listen(GLIBC_2.3) [2]	<del>sendmsg(GLIBC_2.3)</del> sendmsg(GLIBC_2.3) [2]	<del>socketpair(GLIBC_2.3)</del> socketpair(GLIBC_2.3) [2]
<del>accept(GLIBC_2.3)</del> accept(GLIBC_2.3) [2]	<del>gethostname(GLIBC_2.3)</del> gethostname(GLIBC_2.3) [2]	<del>recv(GLIBC_2.3)</del> recv(GLIBC_2.3) [2]	<del>sendto(GLIBC_2.3)</del> sendto(GLIBC_2.3) [2]	
<del>bind(GLIBC_2.3)</del> bind(GLIBC_2.3) [2]	<del>getpeername(GLIBC_2.3)</del> getpeername(GLIBC_2.3) [2]	<del>recvfrom(GLIBC_2.3)</del> recvfrom(GLIBC_2.3) [2]	<del>setsockopt(GLIBC_2.3)</del> setsockopt(GLIBC_2.3) [1]	
<del>bindresvport(GLIBC_2.3)</del> bindresvport(GLIBC_2.3) [2]	<del>getsockname(GLIBC_2.3)</del> getsockname(GLIBC_2.3) [2]	<del>recvmsg(GLIBC_2.3)</del> recvmsg(GLIBC_2.3) [2]	<del>shutdown(GLIBC_2.3)</del> shutdown(GLIBC_2.3) [2]	

GLIBC_2.3) [1]	(GLIBC_2.3) [2]	2.3) [2]	C_2.3) [2]	
<del>econnect</del> (GLIBC_2.3) <del>connect</del> (GLIBC_2.3) [2]	<del>getsockopt</del> (GLIBC_2.3) <del>getsockopt</del> (GLIBC_2.3) [2]	<del>send</del> (GLIBC_2.3) <del>send</del> (GLIBC_2.3) [2]	<del>socket</del> (GLIBC_2.3) <del>socket</del> (GLIBC_2.3) [2]	

86

87 *Referenced Specification(s)*88 [1]. ~~Linux Standard Base~~this specification89 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

90

91 An LSB conforming implementation shall provide the architecture specific deprecated functions for Socket Interface  
92 specified in Table 1-11, with the full functionality as described in the referenced underlying specification.93 These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn  
94 in future releases of this specification.95 **Table 1-11. libc - Socket Interface Deprecated Function Interfaces**

<del>gethostbyname_r</del> (GLIBC_2.3) <del>gethostbyname_r</del> (GLIBC_2.3) [1]				
--	--	--	--	--

96

97 *Referenced Specification(s)*98 [1]. ~~Linux Standard Base~~this specification

## 1.2.7. Wide Characters

### 1.2.7.1. Interfaces for Wide Characters

99 An LSB conforming implementation shall provide the architecture specific functions for Wide Characters specified in  
100 Table 1-12, with the full functionality as described in the referenced underlying specification.  
101102 **Table 1-12. libc - Wide Characters Function Interfaces**

<del>__westod_internal</del> (GLIBC_2.3) <del>__westod_internal</del> (GLIBC_2.3) [1]	<del>mbsinit</del> (GLIBC_2.3) <del>mbsinit</del> (GLIBC_2.3) [2]	<del>vwscanf</del> (GLIBC_2.3) <del>vwscanf</del> (GLIBC_2.3) [2]	<del>wesnlen</del> (GLIBC_2.3) <del>wcsnlen</del> (GLIBC_2.3) [1]	<del>westoumax</del> (GLIBC_2.3) <del>wcstoumax</del> (GLIBC_2.3) [2]
<del>__westof_internal</del> (GLIBC_2.3) <del>__westof_internal</del> (GLIBC_2.3) [1]	<del>mbsnrtowcs</del> (GLIBC_2.3) <del>mbsnrtowcs</del> (GLIBC_2.3) [1]	<del>wepcpy</del> (GLIBC_2.3) <del>wcpcpy</del> (GLIBC_2.3) [1]	<del>wesnrtombs</del> (GLIBC_2.3) <del>wcsnrtombs</del> (GLIBC_2.3) [1]	<del>westouq</del> (GLIBC_2.3) <del>wcstouq</del> (GLIBC_2.3) [1]
<del>__westol_internal</del> (GLIBC_2.3) <del>__westol_internal</del> (GLIBC_2.3) [1]	<del>mbsrtowcs</del> (GLIBC_2.3) <del>mbsrtowcs</del> (GLIBC_2.3) [2]	<del>wepncpy</del> (GLIBC_2.3) <del>wcpcpy</del> (GLIBC_2.3) [1]	<del>wesprbrk</del> (GLIBC_2.3) <del>wcspbrk</del> (GLIBC_2.3) [2]	<del>weswcs</del> (GLIBC_2.3) <del>wcswcs</del> (GLIBC_2.3) [2]

<code>__westold_internal( GLIBC_2.3) __wcst old_internal(GLIBC _2.3) [1]</code>	<code>mbstowes(GLIBC_ 2.3)mbstowcs(GLIB C_2.3) [2]</code>	<code>wertomb(GLIBC_2. 3)wrtomb(GLIBC_ 2.3) [2]</code>	<code>wesrehr(GLIBC_2.3 )wscrchr(GLIBC_2. 3) [2]</code>	<code>weswidth(GLIBC_2 3)wcswidth(GLIBC _2.3) [2]</code>
<code>__westoul_internal( GLIBC_2.3) __wcst oul_internal(GLIBC _2.3) [1]</code>	<code>mbtowe(GLIBC_2. 3)mbtowc(GLIBC_ 2.3) [2]</code>	<code>wescaseemp(GLIB C_2.3)wscasecmp( GLIBC_2.3) [1]</code>	<code>wesrtombs(GLIBC_ 2.3)wscrombs(GLI BC_2.3) [2]</code>	<code>wesxfrm(GLIBC_2. 3)wcsxfrm(GLIBC_ 2.3) [2]</code>
<code>btowe(GLIBC_2.3) btowc(GLIBC_2.3) [2]</code>	<code>putwe(GLIBC_2.3) putwc(GLIBC_2.3) [2]</code>	<code>wesecat(GLIBC_2.3) wscat(GLIBC_2.3) [2]</code>	<code>wesspn(GLIBC_2.3 )wcsspn(GLIBC_2. 3) [2]</code>	<code>wetob(GLIBC_2.3) wctob(GLIBC_2.3) [2]</code>
<code>fgetwe(GLIBC_2.3) fgetwc(GLIBC_2.3) [2]</code>	<code>putwehar(GLIBC_2 3)putwchar(GLIBC _2.3) [2]</code>	<code>weschr(GLIBC_2.3) wchr(GLIBC_2.3) [2]</code>	<code>wesstr(GLIBC_2.3) wcsstr(GLIBC_2.3) [2]</code>	<code>wetomb(GLIBC_2. 3)wctomb(GLIBC_ 2.3) [2]</code>
<code>fgetws(GLIBC_2.3) fgetws(GLIBC_2.3) [2]</code>	<code>swprintf(GLIBC_2. 3)swprintf(GLIBC_ 2.3) [2]</code>	<code>wesemp(GLIBC_2. 3)wscmp(GLIBC_ 2.3) [2]</code>	<code>westod(GLIBC_2.3) westod(GLIBC_2.3) [2]</code>	<code>wetrans(GLIBC_2.3 )wctrans(GLIBC_2. 3) [2]</code>
<code>fputwe(GLIBC_2.3) fputwc(GLIBC_2.3) [2]</code>	<code>swscanf(GLIBC_2. 3)swscanf(GLIBC_ 2.3) [2]</code>	<code>wescoll(GLIBC_2.3 )wscoll(GLIBC_2. 3) [2]</code>	<code>westof(GLIBC_2.3) westof(GLIBC_2.3) [2]</code>	<code>wetype(GLIBC_2.3 )wctype(GLIBC_2. 3) [2]</code>
<code>fputws(GLIBC_2.3) fputws(GLIBC_2.3) [2]</code>	<code>towetrans(GLIBC_2 3)towctrans(GLIB C_2.3) [2]</code>	<code>wesepy(GLIBC_2.3 )wscpy(GLIBC_2. 3) [2]</code>	<code>westoimax(GLIBC_ 2.3)wcstoimax(GLI BC_2.3) [2]</code>	<code>wewidth(GLIBC_2. 3)wcwidth(GLIBC_ 2.3) [2]</code>
<code>fwide(GLIBC_2.3)f wide(GLIBC_2.3) [2]</code>	<code>towlower(GLIBC_2 3)towlower(GLIBC _2.3) [2]</code>	<code>wesespn(GLIBC_2. 3)wscspn(GLIBC_ 2.3) [2]</code>	<code>westok(GLIBC_2.3) westok(GLIBC_2.3) [2]</code>	<code>wmemchr(GLIBC_ 2.3)wmemchr(GLIB C_2.3) [2]</code>
<code>fwprintf(GLIBC_2. 3)fwprintf(GLIBC_ 2.3) [2]</code>	<code>towupper(GLIBC_2 3)towupper(GLIBC _2.3) [2]</code>	<code>wesdup(GLIBC_2.3 )wscdup(GLIBC_2. 3) [1]</code>	<code>westol(GLIBC_2.3) westol(GLIBC_2.3) [2]</code>	<code>wmemcmp(GLIBC_ _2.3)wmemcmp(GLI BC_2.3) [2]</code>
<code>fwscanf(GLIBC_2.3 )fwscanf(GLIBC_2. 3) [2]</code>	<code>ungetwe(GLIBC_2. 3)ungetwc(GLIBC_ 2.3) [2]</code>	<code>wesftime(GLIBC_2. 3)wscftime(GLIBC _2.3) [2]</code>	<code>westold(GLIBC_2.3 )wcstold(GLIBC_2. 3) [2]</code>	<code>wmemcpy(GLIBC_ 2.3)wmemcpy(GLI BC_2.3) [2]</code>
<code>getwe(GLIBC_2.3) getwc(GLIBC_2.3) [2]</code>	<code>vfwprintf(GLIBC_2 3)vfwprintf(GLIBC _2.3) [2]</code>	<code>weslen(GLIBC_2.3) wscnlen(GLIBC_2.3) [2]</code>	<code>westoll(GLIBC_2.3 )wcstoll(GLIBC_2. 3) [2]</code>	<code>wmemmove(GLIB C_2.3)wmemmove( GLIBC_2.3) [2]</code>
<code>getwehar(GLIBC_2. 3)getwchar(GLIBC _2.3) [2]</code>	<code>vfwscanf(GLIBC_2. 3)vfwscanf(GLIBC _2.3) [2]</code>	<code>wescaseemp(GLIB C_2.3)wscnecmp( GLIBC_2.3) [1]</code>	<code>westombs(GLIBC_ 2.3)wcstombs(GLIB C_2.3) [2]</code>	<code>wmemset(GLIBC_2 3)wmemset(GLIBC _2.3) [2]</code>
<code>mblen(GLIBC_2.3) mblen(GLIBC_2.3)</code>	<code>vswprintf(GLIBC_2 3)vswprintf(GLIBC</code>	<code>wesncat(GLIBC_2. 3)wscncat(GLIBC_</code>	<code>westoq(GLIBC_2.3) westoq(GLIBC_2.3)</code>	<code>wprintf(GLIBC_2.3 )wprintf(GLIBC_2.</code>

[2]	_2.3) [2]	2.3) [2]	[1]	3) [2]
<del>mbrlen(GLIBC_2.3)</del> mbrlen(GLIBC_2.3) [2]	<del>vswscanf(GLIBC_2.3)</del> vswscanf(GLIBC_2.3) [2]	<del>wesnemp(GLIBC_2.3)</del> wcsncmp(GLIBC_2.3) [2]	<del>westoul(GLIBC_2.3)</del> wcstoul(GLIBC_2.3) [2]	<del>wscanf(GLIBC_2.3)</del> wscanf(GLIBC_2.3) [2]
<del>mbrtowc(GLIBC_2.3)</del> mbrtowc(GLIBC_2.3) [2]	<del>vwprintf(GLIBC_2.3)</del> vwprintf(GLIBC_2.3) [2]	<del>wesnepy(GLIBC_2.3)</del> wcsncpy(GLIBC_2.3) [2]	<del>westoull(GLIBC_2.3)</del> wcstoull(GLIBC_2.3) [2]	

103

104 *Referenced Specification(s)*105 [1]. ~~Linux Standard Base~~this specification106 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))~~107 ~~VS3)~~

## 1.2.8. String Functions

### 1.2.8.1. Interfaces for String Functions

109 An LSB conforming implementation shall provide the architecture specific functions for String Functions specified in  
110 Table 1-13, with the full functionality as described in the referenced underlying specification.

111 **Table 1-13. libc - String Functions Function Interfaces**

<del>__mempepy(GLIBC_2.3)</del> __mempcpy(GLIBC_2.3) [1]	<del>bzero(GLIBC_2.3)</del> bzero(GLIBC_2.3) [2]	<del>streasestr(GLIBC_2.3)</del> strcasestr(GLIBC_2.3) [1]	<del>strncasecmp(GLIBC_2.3)</del> strncasecmp(GLIBC_2.3) [2]	<del>strtoimax(GLIBC_2.3)</del> strtoimax(GLIBC_2.3) [2]
<del>__rawmemchr(GLIBC_2.3)</del> __rawmemchr(GLIBC_2.3) [1]	<del>ffs(GLIBC_2.3)</del> ffs(GLIBC_2.3) [2]	<del>streat(GLIBC_2.3)</del> strcat(GLIBC_2.3) [2]	<del>strncat(GLIBC_2.3)</del> strncat(GLIBC_2.3) [2]	<del>strtok(GLIBC_2.3)</del> strtok(GLIBC_2.3) [2]
<del>__stpepy(GLIBC_2.3)</del> __stpcpy(GLIBC_2.3) [1]	<del>index(GLIBC_2.3)</del> index(GLIBC_2.3) [2]	<del>strchr(GLIBC_2.3)</del> strchr(GLIBC_2.3) [2]	<del>strnemp(GLIBC_2.3)</del> strncmp(GLIBC_2.3) [2]	<del>strtok_r(GLIBC_2.3)</del> strtok_r(GLIBC_2.3) [4+2]
<del>__strdup(GLIBC_2.3)</del> __strdup(GLIBC_2.3) [1]	<del>memcpy(GLIBC_2.3)</del> memcpy(GLIBC_2.3) [2]	<del>strcmp(GLIBC_2.3)</del> strcmp(GLIBC_2.3) [2]	<del>strncpy(GLIBC_2.3)</del> strncpy(GLIBC_2.3) [2]	<del>strtol(GLIBC_2.3)</del> strtol(GLIBC_2.3) [2]
<del>__strtod_internal(GLIBC_2.3)</del> __strtod_internal(GLIBC_2.3) [1]	<del>memchr(GLIBC_2.3)</del> memchr(GLIBC_2.3) [2]	<del>streq(GLIBC_2.3)</del> strcoll(GLIBC_2.3) [2]	<del>strndup(GLIBC_2.3)</del> strndup(GLIBC_2.3) [1]	<del>strtoll(GLIBC_2.3)</del> strtoll(GLIBC_2.3) [2]
<del>__strtof_internal(GLIBC_2.3)</del> __strtof_internal(GLIBC_2.3) [1]	<del>mememp(GLIBC_2.3)</del> memcmp(GLIBC_2.3) [2]	<del>streqy(GLIBC_2.3)</del> strcpy(GLIBC_2.3) [2]	<del>strnlen(GLIBC_2.3)</del> strnlen(GLIBC_2.3) [1]	<del>strtoq(GLIBC_2.3)</del> strtoq(GLIBC_2.3) [1]
<del>__strtok_r(GLIBC_2.3)</del>	<del>mempepy(GLIBC_2.3)</del>	<del>strespn(GLIBC_2.3)</del>	<del>strpbrk(GLIBC_2.3)</del>	<del>strtoull(GLIBC_2.3)</del>

<code>__strtok_r(GLIBC_2.3)</code> [1]	<code>memcpy(GLIBC_2.3)</code> [2]	<code>strcspn(GLIBC_2.3)</code> [2]	<code>strpbrk(GLIBC_2.3)</code> [2]	<code>strtoull(GLIBC_2.3)</code> [2]
<code>__strtol_internal(GLIBC_2.3)</code> [1]	<code>memmove(GLIBC_2.3)</code> [2]	<code>strdup(GLIBC_2.3)</code> [2]	<code>strptime(GLIBC_2.3)</code> [1]	<code>strtoumax(GLIBC_2.3)</code> [2]
<code>__strtold_internal(GLIBC_2.3)</code> [1]	<code>memrchr(GLIBC_2.3)</code> [1]	<code>strerror(GLIBC_2.3)</code> [2]	<code>strchr(GLIBC_2.3)</code> [2]	<code>strtouq(GLIBC_2.3)</code> [1]
<code>__strtoll_internal(GLIBC_2.3)</code> [1]	<code>memset(GLIBC_2.3)</code> [2]	<code>strerror_r(GLIBC_2.3)</code> [1]	<code>strsep(GLIBC_2.3)</code> [1]	<code>strverscmp(GLIBC_2.3)</code> [1]
<code>__strtoul_internal(GLIBC_2.3)</code> [1]	<code>rindex(GLIBC_2.3)</code> [2]	<code>strfmon(GLIBC_2.3)</code> [2]	<code>strsignal(GLIBC_2.3)</code> [1]	<code>strxfrm(GLIBC_2.3)</code> [2]
<code>__strtoull_internal(GLIBC_2.3)</code> [1]	<code>stpcpy(GLIBC_2.3)</code> [1]	<code>strfry(GLIBC_2.3)</code> [1]	<code>strspn(GLIBC_2.3)</code> [2]	<code>swab(GLIBC_2.3)</code> [2]
<code>bcmp(GLIBC_2.3)</code> [2]	<code>stpncpy(GLIBC_2.3)</code> [1]	<code>strftime(GLIBC_2.3)</code> [2]	<code>strstr(GLIBC_2.3)</code> [2]	
<code>bcopy(GLIBC_2.3)</code> [2]	<code>strcasecmp(GLIBC_2.3)</code> [2]	<code>strlen(GLIBC_2.3)</code> [2]	<code>strtof(GLIBC_2.3)</code> [2]	

112

113 *Referenced Specification(s)*

114 [1]. Linux Standard Basethis specification

115 [2]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)

116 V3)

## 1.2.9. IPC Functions

### 1.2.9.1. Interfaces for IPC Functions

118 An LSB conforming implementation shall provide the architecture specific functions for IPC Functions specified in  
119 Table 1-14, with the full functionality as described in the referenced underlying specification.

120 **Table 1-14. libc - IPC Functions Function Interfaces**

<code>ftok(GLIBC_2.3)</code>	<code>msgrev(GLIBC_2.3)</code>	<code>semget(GLIBC_2.3)</code>	<code>shmetl(GLIBC_2.3)</code>	
------------------------------	--------------------------------	--------------------------------	--------------------------------	--

k(GLIBC_2.3) [1]	̀msgrcv(GLIBC_2.3) [1]	semget(GLIBC_2.3) [1]	shmctl(GLIBC_2.3) [1]	
msgctl(GLIBC_2.3) [1]	msgsnd(GLIBC_2.3) [1]	semop(GLIBC_2.3) [1]	shmdt(GLIBC_2.3) [1]	
msgget(GLIBC_2.3) [1]	semctl(GLIBC_2.3) [1]	shmat(GLIBC_2.3) [1]	shmget(GLIBC_2.3) [1]	

122 *Referenced Specification(s)*

123 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))  
 124 V3)

## 1.2.10. Regular Expressions

### 1.2.10.1. Interfaces for Regular Expressions

126 An LSB conforming implementation shall provide the architecture specific functions for Regular Expressions  
 127 specified in Table 1-15, with the full functionality as described in the referenced underlying specification.

128 **Table 1-15. libc - Regular Expressions Function Interfaces**

regcomp(GLIBC_2.3) [1]	regerror(GLIBC_2.3) [1]	regexec(GLIBC_2.3) [1]	regfree(GLIBC_2.3) [1]	
------------------------	-------------------------	------------------------	------------------------	--

130 *Referenced Specification(s)*

131 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))  
 132 V3)

133 An LSB conforming implementation shall provide the architecture specific deprecated functions for Regular  
 134 Expressions specified in Table 1-16, with the full functionality as described in the referenced underlying specification.

135 These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn  
 136 in future releases of this specification.

137 **Table 1-16. libc - Regular Expressions Deprecated Function Interfaces**

advance(GLIBC_2.3) [1]	re_comp(GLIBC_2.3) [1]	re_exec(GLIBC_2.3) [1]	step(GLIBC_2.3) [1]	
------------------------	------------------------	------------------------	---------------------	--

139 *Referenced Specification(s)*

140 [1]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,  
 141 C606) SUSv2

142 An LSB conforming implementation shall provide the architecture specific deprecated data interfaces for Regular  
 143 Expressions specified in Table 1-17, with the full functionality as described in the referenced underlying specification.



144 These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn  
 145 in future releases of this specification.

146 **Table 1-17. libc - Regular Expressions Deprecated Data Interfaces**

147	loel(GLIBC_2.3)lo c1(GLIBC_2.3) [1]	loe2(GLIBC_2.3)lo c2(GLIBC_2.3) [1]	loes(GLIBC_2.3)loc s(GLIBC_2.3) [1]		
-----	--	--	--	--	--

148 *Referenced Specification(s)*

149 [1]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,  
 150 €606)SUSv2

## 1.2.11. Character Type Functions

### 1.2.11.1. Interfaces for Character Type Functions

152 An LSB conforming implementation shall provide the architecture specific functions for Character Type Functions  
 153 specified in Table 1-18, with the full functionality as described in the referenced underlying specification.

154 **Table 1-18. libc - Character Type Functions Function Interfaces**

155	__ctype_get_mb_cu r_max(GLIBC_2.3) __ctype_get_mb_cu r_max(GLIBC_2.3) [1]	isdigit(GLIBC_2.3)i sdigit(GLIBC_2.3) [2]	iswalnum(GLIBC_2. 3)iswalnum(GLIB C_2.3) [2]	iswlower(GLIBC_2. 3)iswlower(GLIBC _2.3) [2]	toascii(GLIBC_2.3) toascii(GLIBC_2.3) [2]
	_tolower(GLIBC_2. 3)_tolower(GLIBC_ 2.3) [2]	isgraph(GLIBC_2.3) isgraph(GLIBC_2. 3) [2]	iswalpha(GLIBC_2. 3)iswalpha(GLIBC_ 2.3) [2]	iswprint(GLIBC_2. 3)iswprint(GLIBC_ 2.3) [2]	tolower(GLIBC_2.3) )tolower(GLIBC_2. 3) [2]
	_toupper(GLIBC_2. 3)_toupper(GLIBC_ 2.3) [2]	islower(GLIBC_2.3) islower(GLIBC_2. 3) [2]	iswblank(GLIBC_2. 3)iswblank(GLIBC _2.3) [2]	iswpunct(GLIBC_2. 3)iswpunct(GLIBC _2.3) [2]	toupper(GLIBC_2.3) )toupper(GLIBC_2. 3) [2]
	isalnum(GLIBC_2.3) isalnum(GLIBC_2. 3) [2]	isprint(GLIBC_2.3)i sprint(GLIBC_2.3) [2]	iswcntrl(GLIBC_2. 3)iswcntrl(GLIBC_ 2.3) [2]	iswspace(GLIBC_2. 3)iswspace(GLIBC _2.3) [2]	
	isalpha(GLIBC_2.3) isalpha(GLIBC_2.3) [2]	ispunct(GLIBC_2.3) ispunct(GLIBC_2. 3) [2]	iswctype(GLIBC_2. 3)iswctype(GLIBC_ 2.3) [1+2]	iswupper(GLIBC_2. 3)iswupper(GLIBC _2.3) [2]	
	isascii(GLIBC_2.3)i sascii(GLIBC_2.3) [2]	isspace(GLIBC_2.3) isspace(GLIBC_2. 3) [2]	iswdigit(GLIBC_2. 3)iswdigit(GLIBC_ 2.3) [2]	iswxdigit(GLIBC_2. 3)iswxdigit(GLIBC _2.3) [2]	
	iscntrl(GLIBC_2.3)i scntrl(GLIBC_2.3) [2]	isupper(GLIBC_2.3) isupper(GLIBC_2. 3) [2]	iswgraph(GLIBC_2. 3)iswgraph(GLIBC _2.3) [2]	isxdigit(GLIBC_2.3) isxdigit(GLIBC_2. 3) [2]	

156 *Referenced Specification(s)*

157 [1]. ~~Linux Standard Base~~this specification

158 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~

159 ~~∇3)~~

## 1.2.12. Time Manipulation

### 1.2.12.1. Interfaces for Time Manipulation

161 An LSB conforming implementation shall provide the architecture specific functions for Time Manipulation specified  
162 in Table 1-19, with the full functionality as described in the referenced underlying specification.

163 **Table 1-19. libc - Time Manipulation Function Interfaces**

<del>adjtime(GLIBC_2.3)</del> adjtime(GLIBC_2.3) [1]	<del>etime(GLIBC_2.3)</del> etime(GLIBC_2.3) [2]	<del>gmtime(GLIBC_2.3)</del> gmtime(GLIBC_2.3) [2]	<del>localtime_r(GLIBC_2.3)</del> localtime_r(GLIBC_2.3) [2]	<del>ualarm(GLIBC_2.3)</del> ualarm(GLIBC_2.3) [2]
<del>asctime(GLIBC_2.3)</del> asctime(GLIBC_2.3) [2]	<del>etime_r(GLIBC_2.3)</del> etime_r(GLIBC_2.3) [2]	<del>gmtime_r(GLIBC_2.3)</del> gmtime_r(GLIBC_2.3) [2]	<del>mktime(GLIBC_2.3)</del> mktime(GLIBC_2.3) [2]	
<del>asctime_r(GLIBC_2.3)</del> asctime_r(GLIBC_2.3) [2]	<del>difftime(GLIBC_2.3)</del> difftime(GLIBC_2.3) [2]	<del>localtime(GLIBC_2.3)</del> localtime(GLIBC_2.3) [2]	<del>tzset(GLIBC_2.3)</del> tzset(GLIBC_2.3) [2]	

165 *Referenced Specification(s)*

166 [1]. ~~Linux Standard Base~~this specification

167 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~

168 ~~∇3)~~

169 An LSB conforming implementation shall provide the architecture specific deprecated functions for Time  
170 Manipulation specified in Table 1-20, with the full functionality as described in the referenced underlying  
171 specification.

172 These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn  
173 in future releases of this specification.

174 **Table 1-20. libc - Time Manipulation Deprecated Function Interfaces**

<del>adjtimex(GLIBC_2.3)</del> adjtimex(GLIBC_2.3) [1]				
---	--	--	--	--

176 *Referenced Specification(s)*

177 [1]. ~~Linux Standard Base~~this specification

178 An LSB conforming implementation shall provide the architecture specific data interfaces for Time Manipulation  
179 specified in Table 1-21, with the full functionality as described in the referenced underlying specification.

180 **Table 1-21. libc - Time Manipulation Data Interfaces**

<del>__daylight(GLIBC_2.3)</del> daylight(GLIBC_2.3) [1]	<del>__tzname(GLIBC_2.3)</del> tzname(GLIBC_2.3) [1]	<del>timezone(GLIBC_2.3)</del> timezone(GLIBC_2.3) [2]		
<del>__timezone(GLIBC_2.3)</del> timezone(GLIBC_2.3) [1]	<del>daylight(GLIBC_2.3)</del> daylight(GLIBC_2.3) [2]	<del>tzname(GLIBC_2.3)</del> tzname(GLIBC_2.3) [2]		

182 *Referenced Specification(s)*183 [1]. ~~Linux Standard Base~~this specification184 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

## 1.2.13. Terminal Interface Functions

### 1.2.13.1. Interfaces for Terminal Interface Functions

187 An LSB conforming implementation shall provide the architecture specific functions for Terminal Interface Functions  
188 specified in Table 1-22, with the full functionality as described in the referenced underlying specification.

189 **Table 1-22. libc - Terminal Interface Functions Function Interfaces**

<del>efgetispeed(GLIBC_2.3)</del> cfgetispeed(GLIBC_2.3) [1]	<del>efsetispeed(GLIBC_2.3)</del> cfsetispeed(GLIBC_2.3) [1]	<del>tedrain(GLIBC_2.3)</del> tcdrain(GLIBC_2.3) [1]	<del>tegetattr(GLIBC_2.3)</del> tcgetattr(GLIBC_2.3) [1]	<del>tesendbreak(GLIBC_2.3)</del> tcsendbreak(GLIBC_2.3) [1]
<del>efgetospeed(GLIBC_2.3)</del> cfgetospeed(GLIBC_2.3) [1]	<del>efsetospeed(GLIBC_2.3)</del> cfsetospeed(GLIBC_2.3) [1]	<del>teflow(GLIBC_2.3)</del> tcfloor(GLIBC_2.3) [1]	<del>tegetpgrp(GLIBC_2.3)</del> tcgetpgrp(GLIBC_2.3) [1]	<del>tesetattr(GLIBC_2.3)</del> tcsetattr(GLIBC_2.3) [1]
<del>efmakeraw(GLIBC_2.3)</del> cfmakeraw(GLIBC_2.3) [2]	<del>efsetspeed(GLIBC_2.3)</del> cfsetspeed(GLIBC_2.3) [2]	<del>teflush(GLIBC_2.3)</del> tcfloor(GLIBC_2.3) [1]	<del>tegetsid(GLIBC_2.3)</del> tcgetsid(GLIBC_2.3) [1]	<del>tesetpgrp(GLIBC_2.3)</del> tcsetpgrp(GLIBC_2.3) [1]

191 *Referenced Specification(s)*192 [1]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~194 [2]. ~~Linux Standard Base~~this specification

## 1.2.14. System Database Interface

### 1.2.14.1. Interfaces for System Database Interface

196 An LSB conforming implementation shall provide the architecture specific functions for System Database Interface  
197 specified in Table 1-23, with the full functionality as described in the referenced underlying specification.

198 **Table 1-23. libc - System Database Interface Function Interfaces**

<code>endgrent(GLIBC_2.3)</code> [1]	<code>getgrgid(GLIBC_2.3)</code> [1]	<code>getprotobynumber(GLIBC_2.3)</code> [1]	<code>getservbyport(GLIBC_2.3)</code> [1]	<code>setgrent(GLIBC_2.3)</code> [1]
<code>endnetent(GLIBC_2.3)</code> [1]	<code>getgrgid_r(GLIBC_2.3)</code> [1]	<code>getprotoent(GLIBC_2.3)</code> [1]	<code>getservent(GLIBC_2.3)</code> [1]	<code>setgroups(GLIBC_2.3)</code> [2]
<code>endprotoent(GLIBC_2.3)</code> [1]	<code>getgrnam(GLIBC_2.3)</code> [1]	<code>getpwent(GLIBC_2.3)</code> [1]	<code>getutent(GLIBC_2.3)</code> [2]	<code>setnetent(GLIBC_2.3)</code> [1]
<code>endpwent(GLIBC_2.3)</code> [1]	<code>getgrnam_r(GLIBC_2.3)</code> [1]	<code>getpwnam(GLIBC_2.3)</code> [1]	<code>getutent_r(GLIBC_2.3)</code> [2]	<code>setprotoent(GLIBC_2.3)</code> [1]
<code>endservent(GLIBC_2.3)</code> [1]	<code>gethostbyaddr(GLIBC_2.3)</code> [1]	<code>getpwnam_r(GLIBC_2.3)</code> [1]	<code>getutxent(GLIBC_2.3)</code> [1]	<code>setpwent(GLIBC_2.3)</code> [1]
<code>endutent(GLIBC_2.3)</code> [3]	<code>gethostbyname(GLIBC_2.3)</code> [1]	<code>getpwuid(GLIBC_2.3)</code> [1]	<code>getutxid(GLIBC_2.3)</code> [1]	<code>setservent(GLIBC_2.3)</code> [1]
<code>endutxent(GLIBC_2.3)</code> [1]	<code>getnetbyaddr(GLIBC_2.3)</code> [1]	<code>getpwuid_r(GLIBC_2.3)</code> [1]	<code>getutxline(GLIBC_2.3)</code> [1]	<code>setutent(GLIBC_2.3)</code> [2]
<code>getgrent(GLIBC_2.3)</code> [1]	<code>getprotobyname(GLIBC_2.3)</code> [1]	<code>getservbyname(GLIBC_2.3)</code> [1]	<code>pututxline(GLIBC_2.3)</code> [1]	<code>setutxent(GLIBC_2.3)</code> [1]

199

200 *Referenced Specification(s)*201 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)  
202 V3)

203 [2]. Linux Standard Base this specification

204 [3]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,  
205 C606) SUSv2**1.2.15. Language Support**206 **1.2.15.1. Interfaces for Language Support**207 An LSB conforming implementation shall provide the architecture specific functions for Language Support specified  
208 in Table 1-24, with the full functionality as described in the referenced underlying specification.

209 **Table 1-24. libc - Language Support Function Interfaces**

<code>__libc_start_main( GLIBC_2.3) __libc_ start_main(GLIBC_ 2.3) [1]</code>	<code>__obstack_begin(GLI IBC_2.3) __obstack_ begin(GLIBC_2.3) [1]</code>	<code>__obstack_newchunk (GLIBC_2.3) __obsta ck_newchunk(GLIB C_2.3) [1]</code>	<code>obstack_free(GLIB C_2.3) obstack_free( GLIBC_2.3) [1]</code>	
---	---	---	--	--

211 *Referenced Specification(s)*212 [1]. ~~Linux Standard Base~~this specification

## 1.2.16. Large File Support

### 1.2.16.1. Interfaces for Large File Support

214 An LSB conforming implementation shall provide the architecture specific functions for Large File Support specified  
215 in Table 1-25, with the full functionality as described in the referenced underlying specification.

216 **Table 1-25. libc - Large File Support Function Interfaces**

<code>__fxstat64(GLIBC_ 2.3) __fxstat64(GLI BC_2.3) [1]</code>	<code>fopen64(GLIBC_2. 3) fopen64(GLIBC_ 2.3) [2]</code>	<code>ftello64(GLIBC_2.3 )ftello64(GLIBC_2. 3) [2]</code>	<code>lseek64(GLIBC_2.3 )lseek64(GLIBC_2. 3) [2]</code>	<code>readdir64(GLIBC_2 )readdir64(GLIBC _2.3) [2]</code>
<code>__lxstat64(GLIBC_ 2.3) __lxstat64(GLI BC_2.3) [1]</code>	<code>freopen64(GLIBC_ 2.3) freopen64(GLI BC_2.3) [2]</code>	<code>ftruncate64(GLIBC _2.3) ftruncate64(G LIBC_2.3) [2]</code>	<code>mkstemp64(GLIBC _2.3) mkstemp64(G LIBC_2.3) [2]</code>	<code>statvfs64(GLIBC_2. )statvfs64(GLIBC _2.3) [2]</code>
<code>__xstat64(GLIBC_2 )__xstat64(GLIB C_2.3) [1]</code>	<code>fseeko64(GLIBC_2. )fseeko64(GLIBC _2.3) [2]</code>	<code>ftw64(GLIBC_2.3) f tw64(GLIBC_2.3) [2]</code>	<code>mmap64(GLIBC_2. )mmap64(GLIBC_ 2.3) [2]</code>	<code>tmpfile64(GLIBC_2 )tmpfile64(GLIB C_2.3) [2]</code>
<code>creat64(GLIBC_2.3 )creat64(GLIBC_2. 3) [2]</code>	<code>fsetpos64(GLIBC_2 )fsetpos64(GLIBC _2.3) [2]</code>	<code>getrlimit64(GLIBC _2.3) getrlimit64(GL IBC_2.3) [2]</code>	<code>nftw64(GLIBC_2.3) nftw64(GLIBC_2.3) [2]</code>	<code>truncate64(GLIBC_ 2.3) truncate64(GLI BC_2.3) [2]</code>
<code>fgetpos64(GLIBC_ 2.3) fgetpos64(GLIB C_2.3) [2]</code>	<code>fstatvfs64(GLIBC_ 2.3) fstatvfs64(GLIB C_2.3) [2]</code>	<code>lockf64(GLIBC_2.3 )lockf64(GLIBC_2. 3) [2]</code>	<code>open64(GLIBC_2.3 )open64(GLIBC_2. 3) [2]</code>	

218 *Referenced Specification(s)*219 [1]. ~~Linux Standard Base~~this specification

220 [2]. Large File Support

## 1.2.17. Standard Library

### 1.2.17.1. Interfaces for Standard Library

222 An LSB conforming implementation shall provide the architecture specific functions for Standard Library specified in  
223 Table 1-26, with the full functionality as described in the referenced underlying specification.

Table 1-26. `libc` - Standard Library Function Interfaces

<code>_Exit(GLIBC_2.3)_Exit(GLIBC_2.3)</code> [1]	<code>dirname(GLIBC_2.3)dirname(GLIBC_2.3)</code> [1]	<code>glob(GLIBC_2.3)glob(GLIBC_2.3)</code> [1]	<code>lsearch(GLIBC_2.3)lsearch(GLIBC_2.3)</code> [1]	<code>srand(GLIBC_2.3)srand(GLIBC_2.3)</code> [1]
<code>__assert_fail(GLIBC_2.3)__assert_fail(GLIBC_2.3)</code> [2]	<code>div(GLIBC_2.3)div(GLIBC_2.3)</code> [1]	<code>glob64(GLIBC_2.3)glob64(GLIBC_2.3)</code> [2]	<code>makecontext(GLIBC_2.3)makecontext(GLIBC_2.3)</code> [1]	<code>srand48(GLIBC_2.3)srand48(GLIBC_2.3)</code> [1]
<code>__exa_atexit(GLIBC_2.3)__exa_atexit(GLIBC_2.3)</code> [2]	<code>drand48(GLIBC_2.3)drand48(GLIBC_2.3)</code> [1]	<code>globfree(GLIBC_2.3)globfree(GLIBC_2.3)</code> [1]	<code>malloc(GLIBC_2.3)malloc(GLIBC_2.3)</code> [1]	<code>srandom(GLIBC_2.3)srandom(GLIBC_2.3)</code> [1]
<code>__errno_location(GLIBC_2.3)__errno_location(GLIBC_2.3)</code> [2]	<code>eevt(GLIBC_2.3)ecvt(GLIBC_2.3)</code> [1]	<code>globfree64(GLIBC_2.3)globfree64(GLIBC_2.3)</code> [2]	<code>memmem(GLIBC_2.3)memmem(GLIBC_2.3)</code> [2]	<code>strtod(GLIBC_2.3)strtod(GLIBC_2.3)</code> [1]
<code>__fpending(GLIBC_2.3)__fpending(GLIBC_2.3)</code> [2]	<code>erand48(GLIBC_2.3)erand48(GLIBC_2.3)</code> [1]	<code>grantpt(GLIBC_2.3)grantpt(GLIBC_2.3)</code> [1]	<code>mkstemp(GLIBC_2.3)mkstemp(GLIBC_2.3)</code> [1]	<code>strtol(GLIBC_2.3)strtol(GLIBC_2.3)</code> [1]
<code>__getpagesize(GLIBC_2.3)__getpagesize(GLIBC_2.3)</code> [2]	<code>err(GLIBC_2.3)err(GLIBC_2.3)</code> [2]	<code>hereate(GLIBC_2.3)hcreate(GLIBC_2.3)</code> [1]	<code>mktemp(GLIBC_2.3)mktemp(GLIBC_2.3)</code> [1]	<code>strtoul(GLIBC_2.3)strtoul(GLIBC_2.3)</code> [1]
<code>__isinf(GLIBC_2.3)__isinf(GLIBC_2.3)</code> [2]	<code>error(GLIBC_2.3)error(GLIBC_2.3)</code> [2]	<code>hdestroy(GLIBC_2.3)hdestroy(GLIBC_2.3)</code> [1]	<code>mrnd48(GLIBC_2.3)mrnd48(GLIBC_2.3)</code> [1]	<code>swapecontext(GLIBC_2.3)swapcontext(GLIBC_2.3)</code> [1]
<code>__isinff(GLIBC_2.3)__isinff(GLIBC_2.3)</code> [2]	<code>errx(GLIBC_2.3)errx(GLIBC_2.3)</code> [2]	<code>hsearch(GLIBC_2.3)hsearch(GLIBC_2.3)</code> [1]	<code>nftw(GLIBC_2.3)nftw(GLIBC_2.3)</code> [1]	<code>syslog(GLIBC_2.3)syslog(GLIBC_2.3)</code> [1]
<code>__isinfl(GLIBC_2.3)__isinfl(GLIBC_2.3)</code> [2]	<code>fevt(GLIBC_2.3)fcvt(GLIBC_2.3)</code> [1]	<code>htonl(GLIBC_2.3)htonl(GLIBC_2.3)</code> [1]	<code>nrnd48(GLIBC_2.3)nrnd48(GLIBC_2.3)</code> [1]	<code>system(GLIBC_2.3)system(GLIBC_2.3)</code> [2]
<code>__isnan(GLIBC_2.3)__isnan(GLIBC_2.3)</code> [2]	<code>fmtmsg(GLIBC_2.3)fmtmsg(GLIBC_2.3)</code> [1]	<code>htons(GLIBC_2.3)htons(GLIBC_2.3)</code> [1]	<code>ntohl(GLIBC_2.3)ntohl(GLIBC_2.3)</code> [1]	<code>tdelete(GLIBC_2.3)tdelete(GLIBC_2.3)</code> [1]
<code>__isnanf(GLIBC_2.3)__isnanf(GLIBC_2.3)</code> [2]	<code>fnmatch(GLIBC_2.3)fnmatch(GLIBC_2.3)</code> [1]	<code>imaxabs(GLIBC_2.3)imaxabs(GLIBC_2.3)</code> [1]	<code>ntohs(GLIBC_2.3)ntohs(GLIBC_2.3)</code> [1]	<code>tfind(GLIBC_2.3)tfind(GLIBC_2.3)</code> [1]
<code>__isnank(GLIBC_2.3)__isnank(GLIBC_2.3)</code> [2]	<code>fpathconf(GLIBC_2.3)fpathconf(GLIBC_2.3)</code> [1]	<code>imaxdiv(GLIBC_2.3)imaxdiv(GLIBC_2.3)</code> [1]	<code>openlog(GLIBC_2.3)openlog(GLIBC_2.3)</code> [1]	<code>tmpfile(GLIBC_2.3)tmpfile(GLIBC_2.3)</code> [1]
<code>__sysconf(GLIBC_2.3)__sysconf(GLIBC_2.3)</code> [1]	<code>free(GLIBC_2.3)free(GLIBC_2.3)</code> [1]	<code>inet_addr(GLIBC_2.3)inet_addr(GLIBC_2.3)</code> [1]	<code>perror(GLIBC_2.3)perror(GLIBC_2.3)</code> [1]	<code>tmpnam(GLIBC_2.3)tmpnam(GLIBC_2.3)</code> [1]

BC_2.3) [2]		_2.3) [1]	[1]	2.3) [1]
_exit(GLIBC_2.3) exit(GLIBC_2.3) [1]	freeaddrinfo(GLIBC_2.3) freeaddrinfo(GLIBC_2.3) [1]	inet_ntoa(GLIBC_2.3) inet_ntoa(GLIBC_2.3) [1]	posix_memalign(GLIBC_2.3) posix_memalign(GLIBC_2.3) [1]	tsearch(GLIBC_2.3) tsearch(GLIBC_2.3) [1]
_longjmp(GLIBC_2.3) _longjmp(GLIBC_2.3) [1]	ftw(GLIBC_2.3) ftw(GLIBC_2.3) [1]	inet_ntop(GLIBC_2.3) inet_ntop(GLIBC_2.3) [1]	ptsname(GLIBC_2.3) ptsname(GLIBC_2.3) [1]	ttynamename(GLIBC_2.3) ttynamename(GLIBC_2.3) [1]
_setjmp(GLIBC_2.3) _setjmp(GLIBC_2.3) [1]	ftw(GLIBC_2.3) ftw(GLIBC_2.3) [1]	inet_pton(GLIBC_2.3) inet_pton(GLIBC_2.3) [1]	putenv(GLIBC_2.3) putenv(GLIBC_2.3) [1]	ttynamename_r(GLIBC_2.3) ttynamename_r(GLIBC_2.3) [1]
a64(GLIBC_2.3) a64(GLIBC_2.3) [1]	funlockfile(GLIBC_2.3) funlockfile(GLIBC_2.3) [1]	initstate(GLIBC_2.3) initstate(GLIBC_2.3) [1]	qsort(GLIBC_2.3) qsort(GLIBC_2.3) [1]	twalk(GLIBC_2.3) twalk(GLIBC_2.3) [1]
abort(GLIBC_2.3) abort(GLIBC_2.3) [1]	gai_strerror(GLIBC_2.3) gai_strerror(GLIBC_2.3) [1]	insque(GLIBC_2.3) insque(GLIBC_2.3) [1]	rand(GLIBC_2.3) rand(GLIBC_2.3) [1]	unlockpt(GLIBC_2.3) unlockpt(GLIBC_2.3) [1]
abs(GLIBC_2.3) abs(GLIBC_2.3) [1]	gevt(GLIBC_2.3) gevt(GLIBC_2.3) [1]	isatty(GLIBC_2.3) isatty(GLIBC_2.3) [1]	rand_r(GLIBC_2.3) rand_r(GLIBC_2.3) [1]	unsetenv(GLIBC_2.3) unsetenv(GLIBC_2.3) [1]
atof(GLIBC_2.3) atof(GLIBC_2.3) [1]	getaddrinfo(GLIBC_2.3) getaddrinfo(GLIBC_2.3) [1]	isblank(GLIBC_2.3) isblank(GLIBC_2.3) [1]	random(GLIBC_2.3) random(GLIBC_2.3) [1]	usleep(GLIBC_2.3) usleep(GLIBC_2.3) [1]
atoi(GLIBC_2.3) atoi(GLIBC_2.3) [1]	getcwd(GLIBC_2.3) getcwd(GLIBC_2.3) [1]	jrand48(GLIBC_2.3) jrand48(GLIBC_2.3) [1]	random_r(GLIBC_2.3) random_r(GLIBC_2.3) [2]	verrx(GLIBC_2.3) verrx(GLIBC_2.3) [2]
atol(GLIBC_2.3) atol(GLIBC_2.3) [1]	getdate(GLIBC_2.3) getdate(GLIBC_2.3) [1]	l64a(GLIBC_2.3) l64a(GLIBC_2.3) [1]	realloc(GLIBC_2.3) realloc(GLIBC_2.3) [1]	vfprintf(GLIBC_2.3) vfprintf(GLIBC_2.3) [1]
atoll(GLIBC_2.3) atoll(GLIBC_2.3) [1]	getenv(GLIBC_2.3) getenv(GLIBC_2.3) [1]	labs(GLIBC_2.3) labs(GLIBC_2.3) [1]	realpath(GLIBC_2.3) realpath(GLIBC_2.3) [1]	vscanf(GLIBC_2.3) vscanf(GLIBC_2.3) [1]
basename(GLIBC_2.3) basename(GLIBC_2.3) [1]	getlogin(GLIBC_2.3) getlogin(GLIBC_2.3) [1]	leong48(GLIBC_2.3) leong48(GLIBC_2.3) [1]	remque(GLIBC_2.3) remque(GLIBC_2.3) [1]	vsscanf(GLIBC_2.3) vsscanf(GLIBC_2.3) [1]
bsearch(GLIBC_2.3) bsearch(GLIBC_2.3) [1]	getnameinfo(GLIBC_2.3) getnameinfo(GLIBC_2.3) [1]	ldiv(GLIBC_2.3) ldiv(GLIBC_2.3) [1]	seed48(GLIBC_2.3) seed48(GLIBC_2.3) [1]	vsyslog(GLIBC_2.3) vsyslog(GLIBC_2.3) [2]
calloc(GLIBC_2.3) calloc(GLIBC_2.3)	getopt(GLIBC_2.3) getopt(GLIBC_2.3)	lfind(GLIBC_2.3) lfind(GLIBC_2.3) [1]	setenv(GLIBC_2.3) setenv(GLIBC_2.3)	warn(GLIBC_2.3) warn(GLIBC_2.3) [2]

[1]	[2]		[1]	
<del>closelog(GLIBC_2.3)closelog(GLIBC_2.3) [1]</del>	<del>getopt_long(GLIBC_2.3)getopt_long(GLIBC_2.3) [2]</del>	<del>llabs(GLIBC_2.3)labs(GLIBC_2.3) [1]</del>	<del>sethostid(GLIBC_2.3)sethostid(GLIBC_2.3) [2]</del>	<del>warnx(GLIBC_2.3)warnx(GLIBC_2.3) [2]</del>
<del>confstr(GLIBC_2.3)confstr(GLIBC_2.3) [1]</del>	<del>getopt_long_only(GLIBC_2.3)getopt_long_only(GLIBC_2.3) [2]</del>	<del>lldiv(GLIBC_2.3)lldiv(GLIBC_2.3) [1]</del>	<del>sethostname(GLIBC_2.3)sethostname(GLIBC_2.3) [2]</del>	<del>wordexp(GLIBC_2.3)wordexp(GLIBC_2.3) [1]</del>
<del>euserid(GLIBC_2.3)cuserid(GLIBC_2.3) [3]</del>	<del>getsubopt(GLIBC_2.3)getsubopt(GLIBC_2.3) [1]</del>	<del>longjmp(GLIBC_2.3)longjmp(GLIBC_2.3) [1]</del>	<del>setlogmask(GLIBC_2.3)setlogmask(GLIBC_2.3) [1]</del>	<del>wordfree(GLIBC_2.3)wordfree(GLIBC_2.3) [1]</del>
<del>daemon(GLIBC_2.3)daemon(GLIBC_2.3) [2]</del>	<del>gettimeofday(GLIBC_2.3)gettimeofday(GLIBC_2.3) [1]</del>	<del>lrand48(GLIBC_2.3)lrand48(GLIBC_2.3) [1]</del>	<del>setstate(GLIBC_2.3)setstate(GLIBC_2.3) [1]</del>	

225

226 *Referenced Specification(s)*227 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))  
228 V3)

229 [2]. Linux Standard Base this specification

230 [3]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,  
231 C606) SUSv2232 An LSB conforming implementation shall provide the architecture specific data interfaces for Standard Library  
233 specified in Table 1-27, with the full functionality as described in the referenced underlying specification.234 **Table 1-27. libc - Standard Library Data Interfaces**

<del>__environ(GLIBC_2.3)__environ(GLIBC_2.3) [1]</del>	<del>__sys_errlist(GLIBC_2.3)__sys_errlist(GLIBC_2.3) [1]</del>	<del>getdate_err(GLIBC_2.3)getdate_err(GLIBC_2.3) [2]</del>	<del>opterr(GLIBC_2.3)opterr(GLIBC_2.3) [1]</del>	<del>optopt(GLIBC_2.3)optopt(GLIBC_2.3) [1]</del>
<del>__environ(GLIBC_2.3)__environ(GLIBC_2.3) [1]</del>	<del>environ(GLIBC_2.3)environ(GLIBC_2.3) [2]</del>	<del>optarg(GLIBC_2.3)optarg(GLIBC_2.3) [2]</del>	<del>optind(GLIBC_2.3)optind(GLIBC_2.3) [1]</del>	

235

236 *Referenced Specification(s)*

237 [1]. Linux Standard Base this specification

238 [2]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS))  
239 V3)



## 1.3. Data Definitions for libc

240 This section defines global identifiers and their values that are associated with interfaces contained in libc. These  
 241 definitions are organized into groups that correspond to system headers. This convention is used as a convenience for  
 242 the reader, and does not imply the existence of these headers, or their content.

243 These definitions are intended to supplement those provided in the referenced underlying specifications.

244 This specification uses ISO/IEC 9899 C Language as the reference programming language, and data definitions are  
 245 specified in ISO C format. The C language is used here as a convenient notation. Using a C language description of  
 246 these data objects does not preclude their use by other programming languages.

### 1.3.1. errno.h

```
247
248 #define EDEADLOCK          58
```

### 1.3.2. inttypes.h

```
249
250 typedef long intmax_t;
251 typedef unsigned long uintmax_t;
252 typedef unsigned long uintptr_t;
253 typedef unsigned long uint64_t;
```

### 1.3.3. limits.h

```
254
255 #define ULONG_MAX          0xFFFFFFFFFFFFFFFFUL
256 #define LONG_MAX          9223372036854775807L
257
258 #define CHAR_MIN          0
259 #define CHAR_MAX          255
```

### 1.3.4. setjmp.h

```
260
261 typedef long __jmp_buf[40];
```

### 1.3.5. signal.h

```
262
263 struct pt_regs
264 {
265     unsigned long gpr[32];
266     unsigned long nip;
267     unsigned long msr;
268     unsigned long orig_gpr3;
269     unsigned long ctr;
270     unsigned long link;
271     unsigned long xer;
```

```

272     unsigned long ccr;
273     unsigned long softe;
274     unsigned long trap;
275     unsigned long dar;
276     unsigned long dsisr;
277     unsigned long result;
278 }
279 ;
280
281 struct sigaction
282 {
283     union
284     {
285         sighandler_t _sa_handler;
286         void (*_sa_sigaction) (int, siginfo_t *, void *);
287     }
288     __sigaction_handler;
289     sigset_t sa_mask;
290     int sa_flags;
291     void (*sa_restorer) (void);
292 }
293 ;
294 #define MINSIGSTKSZ      2048
295 #define SIGSTKSZ        8192
296
297 struct sigcontext
298 {
299     unsigned long _unused[4];
300     int signal;
301     unsigned long handler;
302     unsigned long oldmask;
303     struct pt_regs *regs;
304     unsigned long gp_regs[48];
305     double fp_regs[33];
306 }
307 ;

```

### 1.3.6. stddef.h

```

308
309 typedef unsigned long size_t;
310 typedef long ptrdiff_t;

```

### 1.3.7. sys/ioctl.h

```

311
312 #define FIONREAD          1074030207
313 #define TIOCNOTTY        21538

```

### 1.3.8. sys/ipc.h

```

314

```

```

315 struct ipc_perm
316 {
317     key_t __key;
318     uid_t uid;
319     gid_t gid;
320     uid_t cuid;
321     gid_t cgid;
322     mode_t mode;
323     unsigned int __seq;
324     unsigned int __pad1;
325     unsigned long __unused1;
326     unsigned long __unused2;
327 }
328 ;

```

### 1.3.9. sys/mman.h

```

329
330 #define MCL_FUTURE      16384
331 #define MCL_CURRENT    8192

```

### 1.3.10. sys/msg.h

```

332
333 typedef unsigned long msglen_t;
334 typedef unsigned long msgqnum_t;
335
336 struct msqid_ds
337 {
338     struct ipc_perm msg_perm;
339     time_t msg_stime;
340     time_t msg_rtime;
341     time_t msg_ctime;
342     unsigned long __msg_cbytes;
343     msgqnum_t msg_qnum;
344     msglen_t msg_qbytes;
345     pid_t msg_lspid;
346     pid_t msg_lrpid;
347     unsigned long __unused4;
348     unsigned long __unused5;
349 }
350 ;

```

### 1.3.11. sys/sem.h

```

351
352 struct semid_ds
353 {
354     struct ipc_perm sem_perm;
355     time_t sem_otime;
356     time_t sem_ctime;
357     unsigned long sem_nsems;

```

```

358     unsigned long __unused3;
359     unsigned long __unused4;
360 }
361 ;

```

### 1.3.12. sys/shm.h

```

362
363 #define SHMLBA (__getpagesize())
364
365 typedef unsigned long shmatt_t;
366
367 struct shmid_ds
368 {
369     struct ipc_perm shm_perm;
370     time_t shm_atime;
371     time_t shm_dtime;
372     time_t shm_ctime;
373     size_t shm_segsz;
374     pid_t shm_cpid;
375     pid_t shm_lpid;
376     shmatt_t shm_nattch;
377     unsigned long __unused5;
378     unsigned long __unused6;
379 }
380 ;

```

### 1.3.13. sys/socket.h

```

381
382 typedef uint64_t __ss_aligntype;

```

### 1.3.14. sys/stat.h

```

383
384 #define _STAT_VER          1
385
386 struct stat
387 {
388     dev_t st_dev;
389     ino_t st_ino;
390     nlink_t st_nlink;
391     mode_t st_mode;
392     uid_t st_uid;
393     gid_t st_gid;
394     int __pad2;
395     dev_t st_rdev;
396     off_t st_size;
397     blksize_t st_blksize;
398     blkcnt_t st_blocks;
399     struct timespec st_atim;
400     struct timespec st_mtim;

```

```

401     struct timespec st_ctim;
402     unsigned long __unused4;
403     unsigned long __unused5;
404     unsigned long __unused6;
405 }
406 ;
407 struct stat64
408 {
409     dev_t st_dev;
410     ino64_t st_ino;
411     nlink_t st_nlink;
412     mode_t st_mode;
413     uid_t st_uid;
414     gid_t st_gid;
415     int __pad2;
416     dev_t st_rdev;
417     off64_t st_size;
418     blksize_t st_blksize;
419     blkcnt64_t st_blocks;
420     struct timespec st_atim;
421     struct timespec st_mtim;
422     struct timespec st_ctim;
423     unsigned long __unused4;
424     unsigned long __unused5;
425     unsigned long __unused6;
426 }
427 ;

```

### 1.3.15. sys/statvfs.h

```

428
429 struct statvfs
430 {
431     unsigned long f_bsize;
432     unsigned long f_frsize;
433     fsblkcnt_t f_blocks;
434     fsblkcnt_t f_bfree;
435     fsblkcnt_t f_bavail;
436     fsfilcnt_t f_files;
437     fsfilcnt_t f_ffree;
438     fsfilcnt_t f_favail;
439     unsigned long f_fsid;
440     unsigned long f_flag;
441     unsigned long f_namemax;
442     int __f_spare[6];
443 }
444 ;
445 struct statvfs64
446 {
447     unsigned long f_bsize;
448     unsigned long f_frsize;
449     fsblkcnt64_t f_blocks;

```

```

450     fsblkcnt64_t f_bfree;
451     fsblkcnt64_t f_bavail;
452     fsfilcnt64_t f_files;
453     fsfilcnt64_t f_ffree;
454     fsfilcnt64_t f_favail;
455     unsigned long f_fsid;
456     unsigned long f_flag;
457     unsigned long f_namemax;
458     int __f_spare[6];
459 }
460 ;

```

### 1.3.16. sys/types.h

```

461
462 typedef long int64_t;
463
464 typedef int64_t ssize_t;

```

### 1.3.17. termios.h

```

465
466 #define TAB1      1024
467 #define CR3      12288
468 #define CRDLY    12288
469 #define FF1      16384
470 #define FFDLY    16384
471 #define XCASE    16384
472 #define ONLCR    2
473 #define TAB2     2048
474 #define TAB3     3072
475 #define TABDLY   3072
476 #define BS1      32768
477 #define BSDLY    32768
478 #define OLCUC    4
479 #define CR1      4096
480 #define IUCLC    4096
481 #define VT1      65536
482 #define VTDLY    65536
483 #define NLDLY    768
484 #define CR2      8192
485
486 #define VWERASE  10
487 #define VREPRINT      11
488 #define VSUSP    12
489 #define VSTART   13
490 #define VSTOP    14
491 #define VDISCARD 16
492 #define VMIN     5
493 #define VEOL     6
494 #define VEOL2    8
495 #define VSWTC    9

```

```

496
497 #define IXOFF 1024
498 #define IXON 512
499
500 #define CSTOPB 1024
501 #define HUPCL 16384
502 #define CREAD 2048
503 #define CS6 256
504 #define CLOCAL 32768
505 #define PARENB 4096
506 #define CS7 512
507 #define VTIME 7
508 #define CS8 768
509 #define CSIZE 768
510 #define PARODD 8192
511
512 #define NOFLSH 0x80000000
513 #define ECHOKE 1
514 #define IEXTEN 1024
515 #define ISIG 128
516 #define ECHONL 16
517 #define ECHOE 2
518 #define ICANON 256
519 #define ECHOPRT 32
520 #define ECHOK 4
521 #define TOSTOP 4194304
522 #define PENDIN 536870912
523 #define ECHOCTL 64
524 #define FLUSHO 8388608

```

### 1.3.18. ucontext.h

```

525
526 #define NGREG 48
527
528 typedef struct sigcontext mcontext_t;
529
530 typedef struct ucontext
531 {
532     unsigned long uc_flags;
533     struct ucontext *uc_link;
534     stack_t uc_stack;
535     sigset_t uc_sigmask;
536     mcontext_t uc_mcontext;
537 }
538 ucontext_t;

```

### 1.3.19.unistd.h

```

539
540 typedef long intp_t;

```

### 1.3.20. utmp.h

```

541
542 struct lastlog
543 {
544     int32_t ll_time;
545     char ll_line[UT_LINESIZE];
546     char ll_host[UT_HOSTSIZE];
547 }
548 ;
549
550 struct utmp
551 {
552     short ut_type;
553     pid_t ut_pid;
554     char ut_line[UT_LINESIZE];
555     char ut_id[4];
556     char ut_user[UT_NAMESIZE];
557     char ut_host[UT_HOSTSIZE];
558     struct exit_status ut_exit;
559     int32_t ut_session;
560     struct
561     {
562         int32_t tv_sec;
563         int32_t tv_usec;
564     }
565     ut_tv;
566     int32_t ut_addr_v6[4];
567     char __unused[20];
568 }
569 ;

```

### 1.3.21. utmpx.h

```

570
571 struct utmpx
572 {
573     short ut_type;
574     pid_t ut_pid;
575     char ut_line[UT_LINESIZE];
576     char ut_id[4];
577     char ut_user[UT_NAMESIZE];
578     char ut_host[UT_HOSTSIZE];
579     struct exit_status ut_exit;
580     int32_t ut_session;
581     struct
582     {
583         int32_t tv_sec;
584         int32_t tv_usec;
585     }
586     ut_tv;
587     int32_t ut_addr_v6[4];

```



```

588     char __unused[20];
589 }
590 ;

```

## 1.4. Interfaces for libm

591 Table 1-28 defines the library name and shared object name for the libm library

592 **Table 1-28. libm Definition**

Library:	libm
SONAME:	libm.so.6

593  
594 The behavior of the interfaces in this library is specified by the following specifications:

ISO/IEC 9899: C (1999, Programming Languages—C)

CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606) SUSv2

595 ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)

### 1.4.1. Math

#### 1.4.1.1. Interfaces for Math

596  
597 An LSB conforming implementation shall provide the architecture specific functions for Math specified in Table 1-29,  
598 with the full functionality as described in the referenced underlying specification.

599 **Table 1-29. libm - Math Function Interfaces**

<code>aeos(GLIBC_2.3)</code> <code>acos(GLIBC_2.3)</code> [1]	<code>eeexp(GLIBC_2.3)</code> <code>exp(GLIBC_2.3)</code> [1]	<code>expf(GLIBC_2.3)</code> <code>expf(GLIBC_2.3)</code> [1]	<code>jnf(GLIBC_2.3)</code> <code>jnl(GLIBC_2.3)</code> [2]	<code>remquof(GLIBC_2.3)</code> <code>remquof(GLIBC_2.3)</code> [1]
<code>aeosf(GLIBC_2.3)</code> <code>acosf(GLIBC_2.3)</code> [1]	<code>eeexpf(GLIBC_2.3)</code> <code>expf(GLIBC_2.3)</code> [1]	<code>expl(GLIBC_2.3)</code> <code>expl(GLIBC_2.3)</code> [1]	<code>jnl(GLIBC_2.3)</code> <code>jnl(GLIBC_2.3)</code> [2]	<code>remquo(GLIBC_2.3)</code> <code>remquo(GLIBC_2.3)</code> [1]
<code>aeosh(GLIBC_2.3)</code> <code>cosh(GLIBC_2.3)</code> [1]	<code>eeexpl(GLIBC_2.3)</code> <code>expl(GLIBC_2.3)</code> [1]	<code>expm1(GLIBC_2.3)</code> <code>expm1(GLIBC_2.3)</code> [1]	<code>ldexp(GLIBC_2.3)</code> <code>dexp(GLIBC_2.3)</code> [1]	<code>rint(GLIBC_2.3)</code> <code>rint(GLIBC_2.3)</code> [1]
<code>aeoshf(GLIBC_2.3)</code> <code>acoshf(GLIBC_2.3)</code> [1]	<code>eimag(GLIBC_2.3)</code> <code>cimag(GLIBC_2.3)</code> [1]	<code>fabs(GLIBC_2.3)</code> <code>fabs(GLIBC_2.3)</code> [1]	<code>ldexpf(GLIBC_2.3)</code> <code>dexpf(GLIBC_2.3)</code> [1]	<code>rintf(GLIBC_2.3)</code> <code>rintf(GLIBC_2.3)</code> [1]
<code>aeoshl(GLIBC_2.3)</code> <code>acoshl(GLIBC_2.3)</code> [1]	<code>eimagf(GLIBC_2.3)</code> <code>cimagf(GLIBC_2.3)</code> [1]	<code>fabsf(GLIBC_2.3)</code> <code>fabsf(GLIBC_2.3)</code> [1]	<code>ldexpl(GLIBC_2.3)</code> <code>dexpl(GLIBC_2.3)</code> [1]	<code>rintl(GLIBC_2.3)</code> <code>rintl(GLIBC_2.3)</code> [1]
<code>aeosl(GLIBC_2.3)</code> <code>cosl(GLIBC_2.3)</code>	<code>eimagl(GLIBC_2.3)</code> <code>cimagl(GLIBC_2.3)</code>	<code>fabsl(GLIBC_2.3)</code> <code>fabsl(GLIBC_2.3)</code> [1]	<code>lgamma(GLIBC_2.3)</code> <code>lgamma(GLIBC_2.3)</code>	<code>round(GLIBC_2.3)</code> <code>round(GLIBC_2.3)</code>

[1]	[1]		2.3) [1]	[1]
asin(GLIBC_2.3)asin(GLIBC_2.3) [1]	elog(GLIBC_2.3)clog(GLIBC_2.3) [1]	fdim(GLIBC_2.3)fdim(GLIBC_2.3) [1]	lgamma_r(GLIBC_2.3)lgamma_r(GLIBC_2.3) [2]	roundf(GLIBC_2.3)roundf(GLIBC_2.3) [1]
asinf(GLIBC_2.3)asinf(GLIBC_2.3) [1]	elog10(GLIBC_2.3)clog10(GLIBC_2.3) [2]	fdimf(GLIBC_2.3)fdimf(GLIBC_2.3) [1]	lgammaf(GLIBC_2.3)lgammaf(GLIBC_2.3) [1]	roundl(GLIBC_2.3)roundl(GLIBC_2.3) [1]
asinh(GLIBC_2.3)asinh(GLIBC_2.3) [1]	elog10f(GLIBC_2.3)clog10f(GLIBC_2.3) [2]	fdiml(GLIBC_2.3)fdiml(GLIBC_2.3) [1]	lgammaf_r(GLIBC_2.3)lgammaf_r(GLIBC_2.3) [2]	scalb(GLIBC_2.3)scalb(GLIBC_2.3) [1]
asinhf(GLIBC_2.3)asinhf(GLIBC_2.3) [1]	elog10l(GLIBC_2.3)clog10l(GLIBC_2.3) [2]	feclearexcept(GLIBC_2.3)feclearexcept(GLIBC_2.3) [1]	lgammal(GLIBC_2.3)lgammal(GLIBC_2.3) [1]	scalbf(GLIBC_2.3)scalbf(GLIBC_2.3) [2]
asinhl(GLIBC_2.3)asinhl(GLIBC_2.3) [1]	elogf(GLIBC_2.3)clogf(GLIBC_2.3) [1]	fegetenv(GLIBC_2.3)fegetenv(GLIBC_2.3) [1]	lgammal_r(GLIBC_2.3)lgammal_r(GLIBC_2.3) [2]	scalbl(GLIBC_2.3)scalbl(GLIBC_2.3) [2]
asinl(GLIBC_2.3)asinl(GLIBC_2.3) [1]	elogl(GLIBC_2.3)clogl(GLIBC_2.3) [1]	fegetexceptflag(GLIBC_2.3)fegetexceptflag(GLIBC_2.3) [1]	llrint(GLIBC_2.3)llrint(GLIBC_2.3) [1]	scalbln(GLIBC_2.3)scalbln(GLIBC_2.3) [1]
atan(GLIBC_2.3)atan(GLIBC_2.3) [1]	eonj(GLIBC_2.3)conj(GLIBC_2.3) [1]	fegetround(GLIBC_2.3)fegetround(GLIBC_2.3) [1]	llrintf(GLIBC_2.3)llrintf(GLIBC_2.3) [1]	scalblnf(GLIBC_2.3)scalblnf(GLIBC_2.3) [1]
atan2(GLIBC_2.3)atan2(GLIBC_2.3) [1]	eonjf(GLIBC_2.3)conjf(GLIBC_2.3) [1]	feholdexcept(GLIBC_2.3)feholdexcept(GLIBC_2.3) [1]	llrintl(GLIBC_2.3)llrintl(GLIBC_2.3) [1]	scalblnl(GLIBC_2.3)scalblnl(GLIBC_2.3) [1]
atan2f(GLIBC_2.3)atan2f(GLIBC_2.3) [1]	eonjl(GLIBC_2.3)conjl(GLIBC_2.3) [1]	feraiseexcept(GLIBC_2.3)feraiseexcept(GLIBC_2.3) [1]	llround(GLIBC_2.3)llround(GLIBC_2.3) [1]	scalbn(GLIBC_2.3)scalbn(GLIBC_2.3) [1]
atan2l(GLIBC_2.3)atan2l(GLIBC_2.3) [1]	eopysign(GLIBC_2.3)copysign(GLIBC_2.3) [1]	fesetenv(GLIBC_2.3)fesetenv(GLIBC_2.3) [1]	llroundf(GLIBC_2.3)llroundf(GLIBC_2.3) [1]	scalbnf(GLIBC_2.3)scalbnf(GLIBC_2.3) [1]
atanf(GLIBC_2.3)atanf(GLIBC_2.3) [1]	eopysignf(GLIBC_2.3)copysignf(GLIBC_2.3) [1]	fesetexceptflag(GLIBC_2.3)fesetexceptflag(GLIBC_2.3) [1]	llroundl(GLIBC_2.3)llroundl(GLIBC_2.3) [1]	scalbnl(GLIBC_2.3)scalbnl(GLIBC_2.3) [1]
atanh(GLIBC_2.3)atanh(GLIBC_2.3) [1]	eopysignl(GLIBC_2.3)copysignl(GLIBC_2.3) [1]	fesetround(GLIBC_2.3)fesetround(GLIBC_2.3) [1]	log(GLIBC_2.3)log(GLIBC_2.3) [1]	significand(GLIBC_2.3)significand(GLIBC_2.3) [2]
atanhf(GLIBC_2.3)atanhf(GLIBC_2.3) [1]	eos(GLIBC_2.3)cos(GLIBC_2.3) [1]	fetestexcept(GLIBC_2.3)fetestexcept(G	log10(GLIBC_2.3)log10(GLIBC_2.3)	significandf(GLIBC_2.3)significandf(G

[1]		LIBC_2.3) [1]	[1]	LIBC_2.3) [2]
atanhl(GLIBC_2.3) atanhl(GLIBC_2.3) [1]	eosf(GLIBC_2.3)co sf(GLIBC_2.3) [1]	feupdateenv(GLIBC _2.3)feupdateenv(G LIBC_2.3) [1]	log10f(GLIBC_2.3) log10f(GLIBC_2.3) [1]	significandl(GLIBC _2.3)significandl(G LIBC_2.3) [2]
atanl(GLIBC_2.3)at anl(GLIBC_2.3) [1]	eosh(GLIBC_2.3)co sh(GLIBC_2.3) [1]	finite(GLIBC_2.3)fi nite(GLIBC_2.3) [3]	log10l(GLIBC_2.3)l og10l(GLIBC_2.3) [1]	sin(GLIBC_2.3)sin( GLIBC_2.3) [1]
eabs(GLIBC_2.3)ca bs(GLIBC_2.3) [1]	eoshf(GLIBC_2.3)c oshf(GLIBC_2.3) [1]	finitef(GLIBC_2.3)f initf(GLIBC_2.3) [2]	log1pf(GLIBC_2.3)l og1p(GLIBC_2.3) [1]	sineos(GLIBC_2.3) sincos(GLIBC_2.3) [2]
eabsf(GLIBC_2.3)c absf(GLIBC_2.3) [1]	eoshl(GLIBC_2.3)c oshl(GLIBC_2.3) [1]	finitel(GLIBC_2.3)f initel(GLIBC_2.3) [2]	logb(GLIBC_2.3)lo gb(GLIBC_2.3) [1]	sineosf(GLIBC_2.3) sincosf(GLIBC_2.3) [2]
eabsl(GLIBC_2.3)c absl(GLIBC_2.3) [1]	eosl(GLIBC_2.3)co sl(GLIBC_2.3) [1]	floor(GLIBC_2.3)fl oor(GLIBC_2.3) [1]	logf(GLIBC_2.3)lo gf(GLIBC_2.3) [1]	sineosl(GLIBC_2.3) sincosl(GLIBC_2.3) [2]
eacos(GLIBC_2.3)c acos(GLIBC_2.3) [1]	epow(GLIBC_2.3)c pow(GLIBC_2.3) [1]	floorf(GLIBC_2.3)f loorf(GLIBC_2.3) [1]	logl(GLIBC_2.3)log l(GLIBC_2.3) [1]	sinf(GLIBC_2.3)sin f(GLIBC_2.3) [1]
eacosf(GLIBC_2.3) cacosf(GLIBC_2.3) [1]	epowf(GLIBC_2.3) cpowf(GLIBC_2.3) [1]	floorl(GLIBC_2.3)fl oorl(GLIBC_2.3) [1]	rint(GLIBC_2.3)lri nt(GLIBC_2.3) [1]	sinh(GLIBC_2.3)sin h(GLIBC_2.3) [1]
eacosh(GLIBC_2.3) cacosh(GLIBC_2.3) [1]	epowl(GLIBC_2.3) cpowl(GLIBC_2.3) [1]	fma(GLIBC_2.3)fm a(GLIBC_2.3) [1]	rintf(GLIBC_2.3)lr intf(GLIBC_2.3) [1]	sinhf(GLIBC_2.3)si nhf(GLIBC_2.3) [1]
eacoshf(GLIBC_2.3) cacoshf(GLIBC_2. 3) [1]	eprojf(GLIBC_2.3)c projf(GLIBC_2.3) [1]	fmaf(GLIBC_2.3)f maf(GLIBC_2.3) [1]	rintl(GLIBC_2.3)lri ntl(GLIBC_2.3) [1]	sinhl(GLIBC_2.3)si nhl(GLIBC_2.3) [1]
eacoshl(GLIBC_2.3) cacoshl(GLIBC_2. 3) [1]	eprojf(GLIBC_2.3)c projf(GLIBC_2.3) [1]	fmal(GLIBC_2.3)f mal(GLIBC_2.3) [1]	lround(GLIBC_2.3) lround(GLIBC_2.3) [1]	sinl(GLIBC_2.3)sin l(GLIBC_2.3) [1]
eacosl(GLIBC_2.3) cacosl(GLIBC_2.3) [1]	eprojl(GLIBC_2.3)c projl(GLIBC_2.3) [1]	fmax(GLIBC_2.3)f max(GLIBC_2.3) [1]	lroundf(GLIBC_2.3) lroundf(GLIBC_2. 3) [1]	sqrt(GLIBC_2.3)sqr t(GLIBC_2.3) [1]
earg(GLIBC_2.3)ca rg(GLIBC_2.3) [1]	erealf(GLIBC_2.3)cr eal(GLIBC_2.3) [1]	fmaxf(GLIBC_2.3)f maxf(GLIBC_2.3) [1]	lroundl(GLIBC_2.3) lroundl(GLIBC_2. 3) [1]	sqrtf(GLIBC_2.3)sq rtf(GLIBC_2.3) [1]
eargf(GLIBC_2.3)c argf(GLIBC_2.3) [1]	erealf(GLIBC_2.3)c realf(GLIBC_2.3) [1]	fmaxl(GLIBC_2.3)f maxl(GLIBC_2.3) [1]	matherr(GLIBC_2.3) matherr(GLIBC_2. 3) [2]	sqrtl(GLIBC_2.3)sq rtl(GLIBC_2.3) [1]



t(GLIBC_2.3) [1]	ctanhf(GLIBC_2.3) [1]	hypotf(GLIBC_2.3) [1]	<del>_2.3</del> nexttowardf(GLIBC_2.3) [1]	(GLIBC_2.3) [2]
ebrtf(GLIBC_2.3)cb rtf(GLIBC_2.3) [1]	etanhf(GLIBC_2.3) ctanhf(GLIBC_2.3) [1]	hypotf(GLIBC_2.3) hypotf(GLIBC_2.3) [1]	nexttowardf(GLIBC_2.3)nexttowardf(GLIBC_2.3) [1]	y0f(GLIBC_2.3)y0f(GLIBC_2.3) [2]
erbf(GLIBC_2.3)cb rtf(GLIBC_2.3) [1]	etanhf(GLIBC_2.3)ct anf(GLIBC_2.3) [1]	ilogb(GLIBC_2.3)il ogb(GLIBC_2.3) [1]	pow(GLIBC_2.3)po w(GLIBC_2.3) [1]	y1f(GLIBC_2.3)y1f(GLIBC_2.3) [1]
eeosf(GLIBC_2.3)cc osf(GLIBC_2.3) [1]	dremf(GLIBC_2.3) dremf(GLIBC_2.3) [2]	ilogbf(GLIBC_2.3)il ogbf(GLIBC_2.3) [1]	pow10(GLIBC_2.3) pow10(GLIBC_2.3) [2]	y1f(GLIBC_2.3)y1f(GLIBC_2.3) [2]
eeosf(GLIBC_2.3)cc cosf(GLIBC_2.3) [1]	dremf(GLIBC_2.3)d remf(GLIBC_2.3) [2]	ilogbl(GLIBC_2.3)il ogbl(GLIBC_2.3) [1]	pow10f(GLIBC_2.3) pow10f(GLIBC_2.3) [2]	y1f(GLIBC_2.3)y1f(GLIBC_2.3) [2]
eeoshf(GLIBC_2.3)cc oshf(GLIBC_2.3) [1]	erf(GLIBC_2.3)erf(GLIBC_2.3) [1]	j0f(GLIBC_2.3)j0f(GLIBC_2.3) [1]	pow10l(GLIBC_2.3) pow10l(GLIBC_2.3) [2]	ynf(GLIBC_2.3)ynf(GLIBC_2.3) [1]
eeoshf(GLIBC_2.3)cc oshf(GLIBC_2.3) [1]	erfc(GLIBC_2.3)erfc(GLIBC_2.3) [1]	j0f(GLIBC_2.3)j0f(GLIBC_2.3) [2]	powf(GLIBC_2.3)powf(GLIBC_2.3) [1]	ynf(GLIBC_2.3)ynf(GLIBC_2.3) [2]
eeoshl(GLIBC_2.3)cc oshl(GLIBC_2.3) [1]	erfcf(GLIBC_2.3)erfcf(GLIBC_2.3) [1]	j0l(GLIBC_2.3)j0l(GLIBC_2.3) [2]	powl(GLIBC_2.3)powl(GLIBC_2.3) [1]	ynl(GLIBC_2.3)ynl(GLIBC_2.3) [2]
eeosl(GLIBC_2.3)cc osl(GLIBC_2.3) [1]	erfcf(GLIBC_2.3)erfcf(GLIBC_2.3) [1]	j1f(GLIBC_2.3)j1f(GLIBC_2.3) [1]	remainder(GLIBC_2.3)remainder(GLIBC_2.3) [1]	
eeilf(GLIBC_2.3)ce ilf(GLIBC_2.3) [1]	erff(GLIBC_2.3)erff(GLIBC_2.3) [1]	j1f(GLIBC_2.3)j1f(GLIBC_2.3) [2]	remainderf(GLIBC_2.3)remainderf(GLIBC_2.3) [1]	
eeilf(GLIBC_2.3)ce ilf(GLIBC_2.3) [1]	erfl(GLIBC_2.3)erfl(GLIBC_2.3) [1]	j1l(GLIBC_2.3)j1l(GLIBC_2.3) [2]	remainderl(GLIBC_2.3)remainderl(GLIBC_2.3) [1]	
eeill(GLIBC_2.3)ce ill(GLIBC_2.3) [1]	exp(GLIBC_2.3)exp(GLIBC_2.3) [1]	jnl(GLIBC_2.3)jnl(GLIBC_2.3) [1]	remquo(GLIBC_2.3)remquo(GLIBC_2.3) [1]	

600

601 *Referenced Specification(s)*602 **[1].** ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)  
603 V3)604 **[2].** ISO/IEC 9899: C (1999, Programming Language—C)

605 **[3].** ~~CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,~~  
 606 ~~€606)SUSv2~~

607 An LSB conforming implementation shall provide the architecture specific data interfaces for Math specified in Table  
 608 1-30, with the full functionality as described in the referenced underlying specification.

609 **Table 1-30. libm - Math Data Interfaces**

610	signgam(GLIBC_2.3) [1]				
-----	------------------------	--	--	--	--

611 *Referenced Specification(s)*

612 **[1].** ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~  
 613 ~~V3)~~

## 1.5. Interfaces for libpthread

614 Table 1-31 defines the library name and shared object name for the libpthread library

615 **Table 1-31. libpthread Definition**

616	Library:	libpthread
	SONAME:	libpthread.so.0

617 The behavior of the interfaces in this library is specified by the following specifications:

Large File Support

~~Linux Standard Base~~this specification

618 ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

### 1.5.1. Realtime Threads

#### 1.5.1.1. Interfaces for Realtime Threads

620 No external functions are defined for libpthread - Realtime Threads

### 1.5.2. Advanced Realtime Threads

#### 1.5.2.1. Interfaces for Advanced Realtime Threads

622 No external functions are defined for libpthread - Advanced Realtime Threads

### 1.5.3. Posix Threads

#### 1.5.3.1. Interfaces for Posix Threads

624 An LSB conforming implementation shall provide the architecture specific functions for Posix Threads specified in  
 625 Table 1-32, with the full functionality as described in the referenced underlying specification.

**Table 1-32. libpthread - Posix Threads Function Interfaces**

<code>_pthread_cleanup_pop(GLIBC_2.3)_pthread_cleanup_pop(GLIBC_2.3) [1]</code>	<code>pthread_cancel(GLIBC_2.3)pthread_cancel(GLIBC_2.3) [2]</code>	<code>pthread_join(GLIBC_2.3)pthread_join(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_destroy(GLIBC_2.3)pthread_rwlock_destroy(GLIBC_2.3) [2]</code>	<code>pthread_setconcurrency(GLIBC_2.3)pthread_setconcurrency(GLIBC_2.3) [2]</code>
<code>_pthread_cleanup_push(GLIBC_2.3)_pthread_cleanup_push(GLIBC_2.3) [1]</code>	<code>pthread_cond_broadcast(GLIBC_2.3.2)pthread_cond_broadcast(GLIBC_2.3.2) [2]</code>	<code>pthread_key_create(GLIBC_2.3)pthread_key_create(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_init(GLIBC_2.3)pthread_rwlock_init(GLIBC_2.3) [2]</code>	<code>pthread_setspecific(GLIBC_2.3)pthread_setspecific(GLIBC_2.3) [2]</code>
<code>pread(GLIBC_2.3)pread(GLIBC_2.3) [2]</code>	<code>pthread_cond_destroy(GLIBC_2.3.2)pthread_cond_destroy(GLIBC_2.3.2) [2]</code>	<code>pthread_key_delete(GLIBC_2.3)pthread_key_delete(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_rdlock(GLIBC_2.3)pthread_rwlock_rdlock(GLIBC_2.3) [2]</code>	<code>pthread_sigmask(GLIBC_2.3)pthread_sigmask(GLIBC_2.3) [2]</code>
<code>pread64(GLIBC_2.3)pread64(GLIBC_2.3) [3]</code>	<code>pthread_cond_init(GLIBC_2.3.2)pthread_cond_init(GLIBC_2.3.2) [2]</code>	<code>pthread_kill(GLIBC_2.3)pthread_kill(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_timedrdlock(GLIBC_2.3)pthread_rwlock_timedrdlock(GLIBC_2.3) [2]</code>	<code>pthread_testcancel(GLIBC_2.3)pthread_testcancel(GLIBC_2.3) [2]</code>
<code>pthread_attr_destroy(GLIBC_2.3)pthread_attr_destroy(GLIBC_2.3) [2]</code>	<code>pthread_cond_signal(GLIBC_2.3.2)pthread_cond_signal(GLIBC_2.3.2) [2]</code>	<code>pthread_mutex_destroy(GLIBC_2.3)pthread_mutex_destroy(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_timedwrlock(GLIBC_2.3)pthread_rwlock_timedwrlock(GLIBC_2.3) [2]</code>	<code>pwrite(GLIBC_2.3)pwrite(GLIBC_2.3) [2]</code>
<code>pthread_attr_getdetachstate(GLIBC_2.3)pthread_attr_getdetachstate(GLIBC_2.3) [2]</code>	<code>pthread_cond_timedwait(GLIBC_2.3.2)pthread_cond_timedwait(GLIBC_2.3.2) [2]</code>	<code>pthread_mutex_init(GLIBC_2.3)pthread_mutex_init(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_tryrdlock(GLIBC_2.3)pthread_rwlock_tryrdlock(GLIBC_2.3) [2]</code>	<code>pwrite64(GLIBC_2.3)pwrite64(GLIBC_2.3) [3]</code>
<code>pthread_attr_getguardsize(GLIBC_2.3)pthread_attr_getguardsize(GLIBC_2.3) [2]</code>	<code>pthread_cond_wait(GLIBC_2.3.2)pthread_cond_wait(GLIBC_2.3.2) [2]</code>	<code>pthread_mutex_lock(GLIBC_2.3)pthread_mutex_lock(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_trywrlock(GLIBC_2.3)pthread_rwlock_trywrlock(GLIBC_2.3) [2]</code>	<code>sem_close(GLIBC_2.3)sem_close(GLIBC_2.3) [2]</code>
<code>pthread_attr_getschedparam(GLIBC_2.3)pthread_attr_getschedparam(GLIBC_2.3) [2]</code>	<code>pthread_condattr_destroy(GLIBC_2.3)pthread_condattr_destroy(GLIBC_2.3) [2]</code>	<code>pthread_mutex_trylock(GLIBC_2.3)pthread_mutex_trylock(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_unlock(GLIBC_2.3)pthread_rwlock_unlock(GLIBC_2.3) [2]</code>	<code>sem_destroy(GLIBC_2.3)sem_destroy(GLIBC_2.3) [2]</code>
<code>pthread_attr_getstackaddr(GLIBC_2.3)pthread_attr_getstackaddr(GLIBC_2.3) [2]</code>	<code>pthread_condattr_getshared(GLIBC_2.3)pthread_condattr_getshared(GLIBC_2.3) [2]</code>	<code>pthread_mutex_unlock(GLIBC_2.3)pthread_mutex_unlock(GLIBC_2.3) [2]</code>	<code>pthread_rwlock_wrl</code>	<code>sem_getvalue(GLIBC_2.3)sem_getvalue(GLIBC_2.3) [2]</code>

addr(GLIBC_2.3) [2]	getpshared(GLIBC_2.3) [2]	(GLIBC_2.3) [2]	k(GLIBC_2.3) [2]	
pthread_attr_getstacksize(GLIBC_2.3)pthread_attr_getstacksize(GLIBC_2.3) [2]	pthread_condattr_init(GLIBC_2.3)pthread_condattr_init(GLIBC_2.3) [2]	pthread_mutexattr_destroy(GLIBC_2.3)pthread_mutexattr_destroy(GLIBC_2.3) [2]	pthread_rwlockattr_destroy(GLIBC_2.3)pthread_rwlockattr_destroy(GLIBC_2.3) [2]	sem_init(GLIBC_2.3)sem_init(GLIBC_2.3) [2]
pthread_attr_init(GLIBC_2.3)pthread_attr_init(GLIBC_2.3) [2]	pthread_condattr_setpshared(GLIBC_2.3)pthread_condattr_setpshared(GLIBC_2.3) [2]	pthread_mutexattr_getpshared(GLIBC_2.3)pthread_mutexattr_getpshared(GLIBC_2.3) [2]	pthread_rwlockattr_getpshared(GLIBC_2.3)pthread_rwlockattr_getpshared(GLIBC_2.3) [2]	sem_open(GLIBC_2.3)sem_open(GLIBC_2.3) [2]
pthread_attr_setdetachstate(GLIBC_2.3)pthread_attr_setdetachstate(GLIBC_2.3) [2]	pthread_create(GLIBC_2.3)pthread_create(GLIBC_2.3) [2]	pthread_mutexattr_gettype(GLIBC_2.3)pthread_mutexattr_gettype(GLIBC_2.3) [2]	pthread_rwlockattr_init(GLIBC_2.3)pthread_rwlockattr_init(GLIBC_2.3) [2]	sem_post(GLIBC_2.3)sem_post(GLIBC_2.3) [2]
pthread_attr_setguardsize(GLIBC_2.3)pthread_attr_setguardsize(GLIBC_2.3) [2]	pthread_detach(GLIBC_2.3)pthread_detach(GLIBC_2.3) [2]	pthread_mutexattr_init(GLIBC_2.3)pthread_mutexattr_init(GLIBC_2.3) [2]	pthread_rwlockattr_setpshared(GLIBC_2.3)pthread_rwlockattr_setpshared(GLIBC_2.3) [2]	sem_timedwait(GLIBC_2.3)sem_timedwait(GLIBC_2.3) [2]
pthread_attr_setschedparam(GLIBC_2.3)pthread_attr_setschedparam(GLIBC_2.3) [2]	pthread_equal(GLIBC_2.3)pthread_equal(GLIBC_2.3) [2]	pthread_mutexattr_setpshared(GLIBC_2.3)pthread_mutexattr_setpshared(GLIBC_2.3) [2]	pthread_self(GLIBC_2.3)pthread_self(GLIBC_2.3) [2]	sem_trywait(GLIBC_2.3)sem_trywait(GLIBC_2.3) [2]
pthread_attr_setstackaddr(GLIBC_2.3)pthread_attr_setstackaddr(GLIBC_2.3) [2]	pthread_exit(GLIBC_2.3)pthread_exit(GLIBC_2.3) [2]	pthread_mutexattr_settype(GLIBC_2.3)pthread_mutexattr_settype(GLIBC_2.3) [2]	pthread_setcancelstate(GLIBC_2.3)pthread_setcancelstate(GLIBC_2.3) [2]	sem_unlink(GLIBC_2.3)sem_unlink(GLIBC_2.3) [2]
pthread_attr_setstacksize(GLIBC_2.3)pthread_attr_setstacksize(GLIBC_2.3) [2]	pthread_getspecific(GLIBC_2.3)pthread_getspecific(GLIBC_2.3) [2]	pthread_once(GLIBC_2.3)pthread_once(GLIBC_2.3) [2]	pthread_setcanceltype(GLIBC_2.3)pthread_setcanceltype(GLIBC_2.3) [2]	sem_wait(GLIBC_2.3)sem_wait(GLIBC_2.3) [2]

627

628 *Referenced Specification(s)*629 [1]. ~~Linux Standard Base~~this specification630 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~631 ~~V3)~~

632 [3]. Large File Support



## 1.6. Interfaces for libgcc\_s

633 Table 1-33 defines the library name and shared object name for the libgcc\_s library

634 **Table 1-33. libgcc\_s Definition**

Library:	libgcc_s
SONAME:	libgcc_s.so.1

636 The behavior of the interfaces in this library is specified by the following specifications:

637 | ~~Linux Standard Base~~this specification

### 1.6.1. Unwind Library

#### 638 1.6.1.1. Interfaces for Unwind Library

639 An LSB conforming implementation shall provide the architecture specific functions for Unwind Library specified in  
640 Table 1-34, with the full functionality as described in the referenced underlying specification.

641 **Table 1-34. libgcc\_s - Unwind Library Function Interfaces**

<del>_Unwind_DeleteException(GCC_3.0)_Unwind_DeleteException(GCC_3.0)</del> [1]	<del>_Unwind_GetDataRelBase(GCC_3.0)_Unwind_GetDataRelBase(GCC_3.0)</del> [1]	<del>_Unwind_GetLanguageSpecificData(GCC_3.0)_Unwind_GetLanguageSpecificData(GCC_3.0)</del> [1]	<del>_Unwind_RaiseException(GCC_3.0)_Unwind_RaiseException(GCC_3.0)</del> [1]	<del>_Unwind_SetIP(GCC_3.0)_Unwind_SetIP(GCC_3.0)</del> [1]
<del>_Unwind_Find_FDE(GCC_3.0)_Unwind_Find_FDE(GCC_3.0)</del> [1]	<del>_Unwind_GetGR(GCC_3.0)_Unwind_GetGR(GCC_3.0)</del> [1]	<del>_Unwind_GetRegionStart(GCC_3.0)_Unwind_GetRegionStart(GCC_3.0)</del> [1]	<del>_Unwind_Resume(GCC_3.0)_Unwind_Resume(GCC_3.0)</del> [1]	
<del>_Unwind_ForcedUnwind(GCC_3.0)_Unwind_ForcedUnwind(GCC_3.0)</del> [1]	<del>_Unwind_GetIP(GCC_3.0)_Unwind_GetIP(GCC_3.0)</del> [1]	<del>_Unwind_GetTextRelBase(GCC_3.0)_Unwind_GetTextRelBase(GCC_3.0)</del> [1]	<del>_Unwind_SetGR(GCC_3.0)_Unwind_SetGR(GCC_3.0)</del> [1]	

643 *Referenced Specification(s)*

644 | [1]. ~~Linux Standard Base~~this specification

## 1.7. Interface Definitions for libgcc\_s

645 The following interfaces are included in libgcc\_s and are defined by this specification. Unless otherwise noted, these  
646 interfaces shall be included in the source standard.

647 Other interfaces listed above for libgcc\_s shall behave as described in the referenced base document.

## **`_Unwind_DeleteException`**

### **Name**

648 `_Unwind_DeleteException` — private C++ error handling method

### **Synopsis**

649 `void _Unwind_DeleteException((struct _Unwind_Exception *object));`

### **Description**

650 `_Unwind_DeleteException` deletes the given exception *object*. If a given runtime resumes normal execution  
 651 after catching a foreign exception, it will not know how to delete that exception. Such an exception shall be deleted by  
 652 calling `_Unwind_DeleteException`. This is a convenience function that calls the function pointed to by the  
 653 *exception\_cleanup* field of the exception header.

## **`_Unwind_Find_FDE`**

### **Name**

654 `_Unwind_Find_FDE` — private C++ error handling method

### **Synopsis**

655 `fde * _Unwind_Find_FDE(void *pc, (struct dwarf_eh_bases *bases));`

### **Description**

656 `_Unwind_Find_FDE` looks for the object containing *pc*, then inserts into *bases*.

## **`_Unwind_ForcedUnwind`**

### **Name**

657 `_Unwind_ForcedUnwind` — private C++ error handling method

### **Synopsis**

```
658 _Unwind_Reason_Code _Unwind_ForcedUnwind((struct _Unwind_Exception *object),  
659 _Unwind_Stop_Fn stop, void *stop_parameter);
```

### **Description**

660 `_Unwind_ForcedUnwind` raises an exception for forced unwinding, passing along the given exception *object*,  
661 which should have its *exception\_class* and *exception\_cleanup* fields set. The exception *object* has been allocated by  
662 the language-specific runtime, and has a language-specific format, except that it shall contain an `_Unwind_Exception`  
663 struct.

664 Forced unwinding is a single-phase process. *stop* and *stop\_parameter* control the termination of the unwind  
665 process instead of the usual personality routine query. *stop* is called for each unwind frame, with the parameters  
666 described for the usual personality routine below, plus an additional *stop\_parameter*.

### **Return Value**

667 When *stop* identifies the destination frame, it transfers control to the user code as appropriate without returning,  
668 normally after calling `_Unwind_DeleteException`. If not, then it should return an `_Unwind_Reason_Code` value.

669 If *stop* returns any reason code other than `_URC_NO_REASON`, then the stack state is indeterminate from the point  
670 of view of the caller of `_Unwind_ForcedUnwind`. Rather than attempt to return, therefore, the unwind library should  
671 use the *exception\_cleanup* entry in the exception, and then call `abort`.

672 `_URC_NO_REASON`

673 This is not the destination from. The unwind runtime will call frame's personality routine with the  
674 `_UA_FORCE_UNWIND` and `_UA_CLEANUP_PHASE` flag set in *actions*, and then unwind to the next frame and call  
675 the *stop* function again.

676 `_URC_END_OF_STACK`

677 In order to allow `_Unwind_ForcedUnwind` to perform special processing when it reaches the end of the stack,  
678 the unwind runtime will call it after the last frame is rejected, with a `NULL` stack pointer in the context, and the  
679 *stop* function shall catch this condition. It may return this code if it cannot handle end-of-stack.

680 `_URC_FATAL_PHASE2_ERROR`

681 The *stop* function may return this code for other fatal conditions like stack corruption.

## **`_Unwind_GetDataRelBase`**

### **Name**

682 `_Unwind_GetDataRelBase` — private IA64 C++ error handling method

### **Synopsis**

683 `_Unwind_Ptr _Unwind_GetDataRelBase((struct _Unwind_Context *context));`

### **Description**

684 `_Unwind_GetDataRelBase` returns the global pointer in register one for *context*.

## **`_Unwind_GetGR`**

### **Name**

685 `_Unwind_GetGR` — private C++ error handling method

### **Synopsis**

686 `_Unwind_Word _Unwind_GetGR((struct _Unwind_Context *context), int index);`

### **Description**

687 `_Unwind_GetGR` returns data at *index* found in *context*. The register is identified by its index: 0 to 31 are for the  
688 fixed registers, and 32 to 127 are for the stacked registers.

689 During the two phases of unwinding, only GR1 has a guaranteed value, which is the global pointer of the frame  
690 referenced by the unwind *context*. If the register has its NAT bit set, the behavior is unspecified.

## **`_Unwind_GetIP`**

### **Name**

691 `_Unwind_GetIP` — private C++ error handling method

### **Synopsis**

692 `_Unwind_Ptr _Unwind_GetIP((struct _Unwind_Context *context));`

### **Description**

693 `_Unwind_GetIP` returns the instruction pointer value for the routine identified by the unwind *context*.

## **`_Unwind_GetLanguageSpecificData`**

### **Name**

694 `_Unwind_GetLanguageSpecificData` — private C++ error handling method

### **Synopsis**

```
695 _Unwind_Ptr _Unwind_GetLanguageSpecificData((struct _Unwind_Context *context), uint  
696 value);
```

### **Description**

697 `_Unwind_GetLanguageSpecificData` returns the address of the language specific data area for the current stack  
698 frame.

## **`_Unwind_GetRegionStart`**

### **Name**

699 `_Unwind_GetRegionStart` — private C++ error handling method

### **Synopsis**

```
700 _Unwind_Ptr _Unwind_GetRegionStart((struct _Unwind_Context *context));
```

### **Description**

701 `_Unwind_GetRegionStart` routine returns the address (i.e., 0) of the beginning of the procedure or code fragment  
702 described by the current unwind descriptor block.

## **`_Unwind_GetTextRelBase`**

### **Name**

703 `_Unwind_GetTextRelBase` — private IA64 C++ error handling method

### **Synopsis**

```
704 _Unwind_Ptr _Unwind_GetTextRelBase((struct _Unwind_Context *context));
```

### **Description**

705 `_Unwind_GetTextRelBase` calls the abort method, then returns.

## **`_Unwind_RaiseException`**

### **Name**

706 `_Unwind_RaiseException` — private C++ error handling method

### **Synopsis**

707 `_Unwind_Reason_Code _Unwind_RaiseException((struct _Unwind_Exception *object));`

### **Description**

708 `_Unwind_RaiseException` raises an exception, passing along the given exception *object*, which should have its  
 709 *exception\_class* and *exception\_cleanup* fields set. The exception object has been allocated by the  
 710 language-specific runtime, and has a language-specific format, exception that it shall contain an  
 711 `_Unwind_Exception`.

### **Return Value**

712 `_Unwind_RaiseException` does not return unless an error condition is found. If an error condition occurs, an  
 713 `_Unwind_Reason_Code` is returned:

714 `_URC_END_OF_STACK`

715       The unwinder encountered the end of the stack during phase one without finding a handler. The unwind runtime  
 716       will not have modified the stack. The C++ runtime will normally call `uncaught_exception` in this case.

717 `_URC_FATAL_PHASE1_ERROR`

718       The unwinder encountered an unexpected error during phase one, because of something like stack corruption.  
 719       The unwind runtime will not have modified the stack. The C++ runtime will normally call `terminate` in this  
 720       case.

721 `_URC_FATAL_PHASE2_ERROR`

722       The unwinder encountered an unexpected error during phase two. This is usually a *throw*, which will call  
 723       `terminate`.

## **`_Unwind_Resume`**

### **Name**

724 `_Unwind_Resume` — private C++ error handling method

### **Synopsis**

725 `void _Unwind_Resume((struct _Unwind_Exception *object));`

### **Description**

726 `_Unwind_Resume` resumes propagation of an existing exception *object*. A call to this routine is inserted as the end  
727 of a landing pad that performs cleanup, but does not resume normal execution. It causes unwinding to proceed further.

## **`_Unwind_SetGR`**

### **Name**

728 `_Unwind_SetGR` — private C++ error handling method

### **Synopsis**

729 `void _Unwind_SetGR((struct _Unwind_Context *context), int index, uint value);`

### **Description**

730 `_Unwind_SetGR` sets the *value* of the register *indexed* for the routine identified by the unwind *context*.

## **`_Unwind_SetIP`**

### **Name**

731 `_Unwind_SetIP` — private C++ error handling method

### **Synopsis**

732 `void _Unwind_SetIP((struct _Unwind_Context *context), uint value);`

### **Description**

733 `_Unwind_SetIP` sets the *value* of the instruction pointer for the routine identified by the unwind *context*

## **1.8. Interfaces for libdl**

734 Table 1-35 defines the library name and shared object name for the libdl library

735 **Table 1-35. libdl Definition**

Library:	libdl
SONAME:	libdl.so.2

737 The behavior of the interfaces in this library is specified by the following specifications:

738 | ~~Linux Standard Base~~this specification  
 | ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

## 1.8.1. Dynamic Loader

### 739 1.8.1.1. Interfaces for Dynamic Loader

740 An LSB conforming implementation shall provide the architecture specific functions for Dynamic Loader specified in  
 741 Table 1-36, with the full functionality as described in the referenced underlying specification.

742 **Table 1-36. libdl - Dynamic Loader Function Interfaces**

<del>dladdr(GLIBC_2.3)</del> dladdr(GLIBC_2.3) [1]	<del>dleclose(GLIBC_2.3)</del> dlclose(GLIBC_2.3) [2]	<del>dLError(GLIBC_2.3)</del> dlerror(GLIBC_2.3) [2]	<del>dlopen(GLIBC_2.3)</del> dlopen(GLIBC_2.3) [1]	<del>dlsym(GLIBC_2.3)</del> dlsym(GLIBC_2.3) [1]
--	---	--	--	--

744 *Referenced Specification(s)*

745 | [1]. ~~Linux Standard Base~~this specification  
 746 | [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~  
 747 | ~~V3)~~

## 1.9. Interfaces for libcrypt

748 Table 1-37 defines the library name and shared object name for the libcrypt library

749 **Table 1-37. libcrypt Definition**

Library:	libcrypt
SONAME:	libcrypt.so.1

751 The behavior of the interfaces in this library is specified by the following specifications:

752 | ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

### 1.9.1. Encryption

#### 753 1.9.1.1. Interfaces for Encryption

754 An LSB conforming implementation shall provide the architecture specific functions for Encryption specified in Table  
 755 1-38, with the full functionality as described in the referenced underlying specification.



756 **Table 1-38. libcrypt - Encryption Function Interfaces**

757	crypt(GLIBC_2.3)cr ypt(GLIBC_2.3) [1]	enrypt(GLIBC_2.3 )encrypt(GLIBC_2. 3) [1]	setkey(GLIBC_2.3) setkey(GLIBC_2.3) [1]		
-----	--	---	---	--	--

758 *Referenced Specification(s)*

759 **[1]. ISO/IEC 9945: POSIX (2003-Portable Operating System(POSIX)and The Single UNIX® Specification(SUS)**  
760 **↯3)**

## **II. Utility Libraries**

# Chapter 2. Libraries

1 The Utility libraries are those that are commonly used, but not part of the Single Unix Specification.

## 2.1. Interfaces for libz

2 **Table 2-1. libz Definition**

Library:	libz
SONAME:	libz.so.1

### 2.1.1. Compression Library

#### 4 2.1.1.1. Interfaces for Compression Library

## 2.2. Data Definitions for libz

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

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

## 2.3. Interfaces for libncurses

11 **Table 2-2. libncurses Definition**

Library:	libncurses
SONAME:	libncurses.so.5

### 2.3.1. Curses

#### 13 2.3.1.1. Interfaces for Curses

## 2.4. Data Definitions for libncurses

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

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

## 2.4.1. curses.h

20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35 `typedef int bool;`

## 2.5. Interfaces for libutil

36 **Table 2-3. libutil Definition**

Library:	libutil
SONAME:	libutil.so.1

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

39 Linux Standard Base<sup>1</sup>

### 2.5.1. Utility Functions

#### 2.5.1.1. Interfaces for Utility Functions

41 **Table 2-4. libutil - Utility Functions Function Interfaces**

forkpty(GLIBC_2.3) <sup>1</sup>	login_tty(GLIBC_2.3) <sup>1</sup>	logwtmp(GLIBC_2.3) <sup>1</sup>		
login(GLIBC_2.3) <sup>1</sup>	logout(GLIBC_2.3) <sup>1</sup>	openpty(GLIBC_2.3) <sup>1</sup>		

## 43 Notes

44 1. Linux Standard Base

# Appendix A. Alphabetical Listing of Interfaces

## A.1. libgcc\_s

1 The behaviour of the interfaces in this library is specified by the following Standards.

2 | ~~Linux Standard Base~~this specification

3 **Table A-1. libgcc\_s Function Interfaces**

<del>_Unwind_DeleteException[1]</del>	<del>_Unwind_GetIP_Unwind_GetIP[1]</del>	<del>_Unwind_Resume_Unwind_Resume[1]</del>
<del>_Unwind_Find_FDE_Unwind_Find_FDE[1]</del>	<del>_Unwind_GetLanguageSpecificData[1]</del>	<del>_Unwind_SetGR_Unwind_SetGR[1]</del>
<del>_Unwind_ForcedUnwind_Unwind_ForcedUnwind[1]</del>	<del>_Unwind_GetRegionStart[1]</del>	<del>_Unwind_SetIP_Unwind_SetIP[1]</del>
<del>_Unwind_GetDataRelBase[1]</del>	<del>_Unwind_GetTextRelBase[1]</del>	
<del>_Unwind_GetGR_Unwind_GetGR[1]</del>	<del>_Unwind_RaiseException[1]</del>	

4

# **Linux Packaging Specification**

2

3 **Linux Packaging Specification**

# Table of Contents

<b>I. Package Format and Installation</b> .....	<b>55</b>
1. Software Installation .....	1
1.1. Package Dependencies.....	1
1.2. Package Architecture Considerations .....	1



# **I. Package Format and Installation**



# Chapter 1. Software Installation

## 1.1. Package Dependencies

- 1 The LSB runtime environment shall provide the following dependencies.
- 2 `lsb-core-ppc64`
  - 3 This dependency is used to indicate that the application is dependent on features contained in the LSB-Core
  - 4 specification.
- 5 Other LSB modules may add additional dependencies; such dependencies shall have the format `lsb-module-ppc64`.

## 1.2. Package Architecture Considerations

- 6 All packages must specify an architecture of `ppc64`. A LSB runtime environment must accept an architecture of
- 7 `ppc64` even if the native architecture is different.
- 8 The `archnum` value in the Lead Section shall be `0x0010`.

# Free Documentation License

2 |  
3 | **Free Documentation License**

# Table of Contents

<b>A. GNU Free Documentation License</b> .....	<b>1</b>
A.1. PREAMBLE.....	1
A.2. APPLICABILITY AND DEFINITIONS.....	1
A.3. VERBATIM COPYING.....	2
A.4. COPYING IN QUANTITY.....	2
A.5. MODIFICATIONS.....	3
A.6. COMBINING DOCUMENTS.....	4
A.7. COLLECTIONS OF DOCUMENTS.....	4
A.8. AGGREGATION WITH INDEPENDENT WORKS.....	4
A.9. TRANSLATION.....	5
A.10. TERMINATION.....	5
A.11. FUTURE REVISIONS OF THIS LICENSE.....	5
A.12. How to use this License for your documents.....	5

# Appendix A. 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.

## A.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. Secondly, 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.

## A.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

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

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

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

### A.3. VERBATIM COPYING

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

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

### A.4. COPYING IN QUANTITY

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

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

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

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



## A.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.

111 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  
112 Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of  
113 Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already  
114 includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are  
115 acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the  
116 previous publisher that added the old one.

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

## A.6. COMBINING DOCUMENTS

119 You may combine the Document with other documents released under this License, under the terms defined in section  
120 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the  
121 original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

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

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

## A.7. COLLECTIONS OF DOCUMENTS

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

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

## A.8. AGGREGATION WITH INDEPENDENT WORKS

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

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

## A.9. TRANSLATION

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

## A.10. TERMINATION

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

## A.11. FUTURE REVISIONS OF THIS LICENSE

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

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

## A.12. How to use this License for your documents

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

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

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

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