

**Linux Standard Base Core Module
Specification for S390X 2.0.1**

Linux Standard Base Core Module Specification for S390X 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

Foreword	i
Introduction	ii
I. Introductory Elements	3
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 S390X. 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 S390X 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	http://www.caldera.com/developers/gabi/2003-12-17/contents.html
DWARF Debugging Information Format, Revision 2.0.0 (July 27, 1993)	http://www.eagereon.com/dwarf/dwarf-2.0.0.pdf
Filesystem Hierarchy Standard (FHS) 2.3	http://www.pathname.com/fhs/
IEEE Standard 754 for Binary Floating-Point Arithmetic	http://www.ieee.org/
System V Application Binary Interface, Edition 4.1	http://www.caldera.com/developers/devspeecs/gabi41.pdf
ISO/IEC 9899: 1999, Programming Languages—C	
Linux Assigned Names And Numbers Authority	http://www.lanana.org/
Large File Support	http://www.UNIX-systems.org/version2/whatsnew/lfs20mar.html
LI18NUNIX 2000 Globalization Specification, Version 1.0 with Amendment 4	http://www.li18nux.org/docs/html/LI18NUNIX-2000-amd4.htm
Linux Standard Base	http://www.linuxbase.org/spec/
OSF RFC 86.0	http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt
RFC 1833: Binding Protocols for ONC RPC Version 2	http://www.ietf.org/rfc/rfc1833.txt
RFC 1952: GZIP file format specification version 4.3	http://www.ietf.org/rfc/rfc1952.txt
RFC 2440: OpenPGP Message Format	http://www.ietf.org/rfc/rfc2440.txt
LINUX for zSeries Application Binary Interface Supplement	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
z/Architecture Principles of Operation	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), plus Corrigendum U018	http://www.opengroup.org/publications/catalog/un.htm
The Single UNIX® Specification(SUS) Version 2,	http://www.opengroup.org/publications/catalog/un.htm

Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)		
CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)		http://www.opengroup.org/publications/catalog/un.htm
ISO/IEC 9945:2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3		http://www.unix.org/version3/
System V Interface Definition, Issue 3 (ISBN 0201566524)		
System V Interface Definition, Fourth Edition		
zlib 1.2 Manual		http://www.gzip.org/zlib/
Name	Title	URL
DWARF Debugging Information Format	DWARF Debugging Information Format, Revision 2.0.0 (July 27, 1993)	http://www.eagercon.com/dwarf/dwarf-2.0.0.pdf
Filesystem Hierarchy Standard	Filesystem Hierarchy Standard (FHS) 2.3	http://www.pathname.com/fhs/
IEEE Std 754-1985	IEEE Standard 754 for Binary Floating-Point Arithmetic	http://www.ieee.org/
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 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	http://www.unix.org/version3/
Large File Support	Large File Support	http://www.UNIX-systems.org/vers

		ion2/whatsnew/lfs20mar.html
Li18nux Globalization Specification	LI18NUNIX 2000 Globalization Specification, Version 1.0 with Amendment 4	http://www.li18nux.org/docs/html/LI18NUNIX-2000-amd4.htm
Linux Allocated Device Registry	LINUX ALLOCATED DEVICES	http://www.lanana.org/docs/device-list/devices.txt
LINUX for zSeries Application Binary Interface Supplement	LINUX for zSeries Application Binary Interface Supplement	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
PAM	Open Software Foundation, Request For Comments: 86.0 , October 1995, V. Samar & R.Schemers (SunSoft)	http://www.opengroup.org/tech/rfc/mirror-rfc/rfc86.0.txt
RFC 1321: The MD5 Message-Digest Algorithm	IETF RFC 1321: The MD5 Message-Digest Algorithm	http://www.ietf.org/rfc/rfc1321.txt
RFC 1833: Binding Protocols for ONC RPC Version 2	IETF RFC 1833: Binding Protocols for ONC RPC Version 2	http://www.ietf.org/rfc/rfc1833.txt
RFC 1951: DEFLATE Compressed Data Format Specification	IETF RFC 1951: DEFLATE Compressed Data Format Specification version 1.3	http://www.ietf.org/rfc/rfc1951.txt
RFC 1952: GZIP File Format Specification	IETF RFC 1952: GZIP file format specification version 4.3	http://www.ietf.org/rfc/rfc1952.txt
RFC 2440: OpenPGP Message Format	IETF RFC 2440: OpenPGP Message Format	http://www.ietf.org/rfc/rfc2440.txt
SUSv2	CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)	http://www.opengroup.org/publications/catalog/un.htm
SUSv2 Command and Utilities	The Single UNIX® Specification(SUS) Version 2, Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)	http://www.opengroup.org/publications/catalog/un.htm
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	http://www.caldera.com/developers/devspecs/gabi41.pdf

System V ABI Update	System V Application Binary Interface - DRAFT - 17 December 2003	http://www.caldera.com/developers/gabi/2003-12-17/contents.html
this specification	Linux Standard Base	http://www.linuxbase.org/spec/
X/Open Curses	CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), plus Corrigendum U018	http://www.opengroup.org/publications/catalog/un.htm
z/Architecture Principles of Operation	z/Architecture Principles of Operation	http://oss.software.ibm.com/linux390/documentation-2.2.shtml
zlib Manual	zlib 1.2 Manual	http://www.gzip.org/zlib/

Chapter 3. Requirements

3.1. Relevant Libraries

1 The libraries listed in Table 3-1 shall be available on S390X 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
libz	libz.so.1
libncurses	libncurses.so.5
libutil	libutil.so.1
libc	libc.so.6
libpthread	libpthread.so.0
proginterp	/lib64/ld-lsb-s390x.so.2
libgcc_s	libgcc_s.so.1
libz	libz.so.1
libncurses	libncurses.so.5
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

I. Low Level System Information	16
1. Machine Interface.....	1
1.1. Processor Architecture.....	1
1.2. Data Representation.....	1
1.2.1. Byte Ordering.....	1
1.2.2. Fundamental Types.....	1
1.2.3. Aggregates and Unions.....	1
1.2.4. Bit Fields.....	1
2. Function Calling Sequence.....	2
2.1. Registers.....	2
2.2. Stack Frame.....	2
2.3. Parameter Passing.....	2
2.4. Variable Argument Lists.....	2
2.5. Return Values.....	2
3. Operating System Interface.....	3
3.1. Virtual Address Space.....	3
3.2. Page Size.....	3
3.3. Virtual Address Assignments.....	3
3.4. Managing the Process Stack.....	3
3.5. Coding Guidelines.....	3
3.6. Processor Execution Mode.....	3
3.7. Exception Interface.....	3
3.8. Signal Delivery.....	3
3.8.1. Signal Handler Interface.....	3
4. Process Initialization.....	4
4.1. Registers.....	4
4.2. Process Stack.....	4
5. Coding Examples.....	5
5.1. Code Model Overview.....	5
5.2. Function Prolog and Epilog.....	5
5.3. Profiling.....	5
5.4. Data Objects.....	5
5.5. Function Calls.....	5
5.6. Dynamic Stack Space Allocation.....	5
6. Debug Information.....	6
II. Object Format	7
7. ELF Header.....	8
7.1. Machine Information.....	8
8. Sections.....	9
8.1. Special Sections.....	9
8.2. Linux Special Sections.....	9
9. Symbol Table.....	10
10. Relocation.....	11

10.1. Relocation Types	11
III. Program Loading and Dynamic Linking	12
11. Program Loading.....	13
12. Dynamic Linking.....	14
12.1. Dynamic Section.....	14
12.2. Global Offset Table	14
12.3. Function Addresses.....	14
12.4. Procedure Linkage Table	14

List of Tables

8-1. ELF Special Sections.....	9
8-2. Additional Special Sections.....	9

I. Low Level System Information

Chapter 1. Machine Interface

1.1. Processor Architecture

- 1 The z/Architecture is specified by the following documents
- 2 • LINUX for zSeries Application Binary Interface Supplement
 - 3 • z/Architecture Principles of Operation
- 4 Only the non optional features of z/Architecture processor instruction set may be assumed to be present. An
5 application is responsible for determining if any additional instruction set features are available before using those
6 additional features. If a feature is not present, then the application may not use it.
- 7 Applications may not make system calls directly. The interfaces in the C library must be used instead.
- 8 Applications conforming to this specification must provide feedback to the user if a feature that is required for correct
9 execution of the application is not present. Applications conforming to this specification should attempt to execute in
10 a diminished capacity if a required instruction set feature is not present.
- 11 This specification does not provide any performance guarantees of a conforming system. A system conforming to this
12 specification may be implemented in either hardware or software.

1.2. Data Representation

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

1.2.1. Byte Ordering

- 15 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

1.2.2. Fundamental Types

- 16 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

1.2.3. Aggregates and Unions

- 17 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

1.2.4. Bit Fields

- 18 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 2. Function Calling Sequence

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

2.1. Registers

3 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

2.2. Stack Frame

4 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

2.3. Parameter Passing

5 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

2.4. Variable Argument Lists

6 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

2.5. Return Values

7 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 3. Operating System Interface

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

3.1. Virtual Address Space

3 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.2. Page Size

4 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.3. Virtual Address Assignments

5 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.4. Managing the Process Stack

6 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.5. Coding Guidelines

7 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.6. Processor Execution Mode

8 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.7. Exception Interface

9 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.8. Signal Delivery

10 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

3.8.1. Signal Handler Interface

11 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 4. Process Initialization

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

4.1. Registers

- 3 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

4.2. Process Stack

- 4 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 5. Coding Examples

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

5.1. Code Model Overview

3 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.2. Function Prolog and Epilog

4 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.3. Profiling

5 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.4. Data Objects

6 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.5. Function Calls

7 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

5.6. Dynamic Stack Space Allocation

8 | See chapter 1 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 6. Debug Information

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

II. Object Format

2 LSB-conforming implementations shall support an object file , called Executable and Linking Format (ELF) as
3 defined by the System V ~~Application Binary Interface, Edition 4.~~ABI , System V ~~Application Binary Interface~~
4 ~~DRAFT—17 December 2003~~ABI Update , LINUX for zSeries Application Binary Interface Supplement and as
5 supplemented by the ~~Linux Standard Base~~this specification and this document.

Chapter 7. ELF Header

7.1. Machine Information

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

Chapter 8. Sections

1 See chapter 2 of the LINUX for zSeries Application Binary Interface Supplement.

8.1. Special Sections

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

3 **Table 8-1. ELF Special Sections**

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

5 .got

6 This section holds the global offset table

7 .plt

8 This section holds the procedure linkage table

8.2. Linux Special Sections

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

10 **Table 8-2. Additional Special Sections**

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

12 .rela.dyn

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

14 .rela.plt

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

17 .sbss

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

Chapter 9. Symbol Table

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

Chapter 10. Relocation

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

10.1. Relocation Types

- 3 | See chapter 2 of the LINUX for zSeries Application Binary Interface 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 , System V ~~Application Binary~~
4 ~~Interface DRAFT 17 December 2003~~ABI Update , LINUX for zSeries Application Binary Interface Supplement
5 and as supplemented by the ~~Linux Standard Base~~this specification and this document.

Chapter 11. Program Loading

- 1 See Chapter 3 of the LINUX for zSeries Application Binary Interface Supplement.

Chapter 12. Dynamic Linking

1 See Chapter 3 of the LINUX for zSeries Application Binary Interface Supplement.

12.1. Dynamic Section

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

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

12.2. Global Offset Table

8 See Chapter 3 of the LINUX for zSeries Application Binary Interface Supplement.

12.3. Function Addresses

9 | See chapter 3 of the LINUX for zSeries Application Binary Interface Supplement.

12.4. Procedure Linkage Table

10 | See chapter 3 of the LINUX for zSeries Application Binary Interface Supplement.

Linux Standard Base Specification

2

3 **Linux Standard Base Specification**

Table of Contents

I. Base Libraries	21
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	25
1.3.8. sys/ipc.h	25
1.3.9. sys/mman.h	25
1.3.10. sys/msg.h	25
1.3.11. sys/sem.h	26
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. utmp.h	30
1.3.20. 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	37
1.5.2.1. Interfaces for Advanced Realtime Threads	37
1.5.3. Posix Threads	37
1.5.3.1. Interfaces for Posix Threads	37
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	40
_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
II. Utility Libraries	48
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
A. Alphabetical Listing of Interfaces	51
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 z/Architecture 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-s390x.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**

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

BC_2.2) [1]	name(GLIBC_2.2) [2]	(GLIBC_2.2) [2]	2.2) [3]	C_2.2) [2]
clnt_pcreateerror(GLIBC_2.2)clnt_pcreateerror(GLIBC_2.2) [1]	svc_getreqset(GLIBC_2.2)svc_getreqset(GLIBC_2.2) [3]	sveudp_create(GLIBC_2.2)svcudp_create(GLIBC_2.2) [2]	xdr_int(GLIBC_2.2)xdr_int(GLIBC_2.2) [3]	xdr_u_long(GLIBC_2.2)xdr_u_long(GLIBC_2.2) [3]
clnt_perrno(GLIBC_2.2)clnt_perrno(GLIBC_2.2) [1]	svc_register(GLIBC_2.2)svc_register(GLIBC_2.2) [2]	xdr_accepted_reply(GLIBC_2.2)xdr_accepted_reply(GLIBC_2.2) [3]	xdr_long(GLIBC_2.2)xdr_long(GLIBC_2.2) [3]	xdr_u_short(GLIBC_2.2)xdr_u_short(GLIBC_2.2) [3]
clnt_perror(GLIBC_2.2)clnt_perror(GLIBC_2.2) [1]	svc_run(GLIBC_2.2)svc_run(GLIBC_2.2) [2]	xdr_array(GLIBC_2.2)xdr_array(GLIBC_2.2) [3]	xdr_opaque(GLIBC_2.2)xdr_opaque(GLIBC_2.2) [3]	xdr_union(GLIBC_2.2)xdr_union(GLIBC_2.2) [3]
clnt_screateerror(GLIBC_2.2)clnt_screateerror(GLIBC_2.2) [1]	svc_sendreply(GLIBC_2.2)svc_sendreply(GLIBC_2.2) [2]	xdr_bool(GLIBC_2.2)xdr_bool(GLIBC_2.2) [3]	xdr_opaque_auth(GLIBC_2.2)xdr_opaque_auth(GLIBC_2.2) [3]	xdr_vector(GLIBC_2.2)xdr_vector(GLIBC_2.2) [3]
clnt_serrno(GLIBC_2.2)clnt_serrno(GLIBC_2.2) [1]	svcerr_auth(GLIBC_2.2)svcerr_auth(GLIBC_2.2) [3]	xdr_bytes(GLIBC_2.2)xdr_bytes(GLIBC_2.2) [3]	xdr_pointer(GLIBC_2.2)xdr_pointer(GLIBC_2.2) [3]	xdr_void(GLIBC_2.2)xdr_void(GLIBC_2.2) [3]
clnt_serror(GLIBC_2.2)clnt_serror(GLIBC_2.2) [1]	svcerr_decode(GLIBC_2.2)svcerr_decode(GLIBC_2.2) [3]	xdr_callhdr(GLIBC_2.2)xdr_callhdr(GLIBC_2.2) [3]	xdr_reference(GLIBC_2.2)xdr_reference(GLIBC_2.2) [3]	xdr_wrapstring(GLIBC_2.2)xdr_wrapstring(GLIBC_2.2) [3]
getdomainname(GLIBC_2.2)getdomainname(GLIBC_2.2) [2]	svcerr_noproc(GLIBC_2.2)svcerr_noproc(GLIBC_2.2) [3]	xdr_callmsg(GLIBC_2.2)xdr_callmsg(GLIBC_2.2) [3]	xdr_rejected_reply(GLIBC_2.2)xdr_rejected_reply(GLIBC_2.2) [3]	xdrmem_create(GLIBC_2.2)xdrmem_create(GLIBC_2.2) [3]
key_decryptsession(GLIBC_2.2)key_decryptsession(GLIBC_2.2) [3]	svcerr_noprog(GLIBC_2.2)svcerr_noprog(GLIBC_2.2) [3]	xdr_char(GLIBC_2.2)xdr_char(GLIBC_2.2) [3]	xdr_replymsg(GLIBC_2.2)xdr_replymsg(GLIBC_2.2) [3]	xdrrec_create(GLIBC_2.2)xdrrec_create(GLIBC_2.2) [3]
pmap_getport(GLIBC_2.2)pmap_getport(GLIBC_2.2) [2]	svcerr_progvers(GLIBC_2.2)svcerr_progvers(GLIBC_2.2) [3]	xdr_double(GLIBC_2.2)xdr_double(GLIBC_2.2) [3]	xdr_short(GLIBC_2.2)xdr_short(GLIBC_2.2) [3]	xdrrec_eof(GLIBC_2.2)xdrrec_eof(GLIBC_2.2) [3]
pmap_set(GLIBC_2.2)pmap_set(GLIBC_2.2) [2]	svcerr_systemerr(GLIBC_2.2)svcerr_systemerr(GLIBC_2.2) [3]	xdr_enum(GLIBC_2.2)xdr_enum(GLIBC_2.2) [3]	xdr_string(GLIBC_2.2)xdr_string(GLIBC_2.2) [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_2.2)</code> <code>__fxstat(GLIBC_2.2)</code> [1]	<code>fchmod(GLIBC_2.2)</code> <code>fchmod(GLIBC_2.2)</code> [2]	<code>getwd(GLIBC_2.2)</code> <code>getwd(GLIBC_2.2)</code> [2]	<code>read(GLIBC_2.2)</code> <code>read(GLIBC_2.2)</code> [2]	<code>setrlimit(GLIBC_2.2)</code> <code>setrlimit(GLIBC_2.2)</code> [2]
<code>__getpgid(GLIBC_2.2)</code> <code>__getpgid(GLIBC_2.2)</code> [1]	<code>fchown(GLIBC_2.2)</code> <code>fchown(GLIBC_2.2)</code> [2]	<code>initgroups(GLIBC_2.2)</code> <code>initgroups(GLIBC_2.2)</code> [1]	<code>readdir(GLIBC_2.2)</code> <code>readdir(GLIBC_2.2)</code> [2]	<code>setrlimit64(GLIBC_2.2)</code> <code>setrlimit64(GLIBC_2.2)</code> [3]
<code>__lxstat(GLIBC_2.2)</code> <code>__lxstat(GLIBC_2.2)</code> [1]	<code>fentl(GLIBC_2.2)</code> <code>fentl(GLIBC_2.2)</code> [1]	<code>ioctl(GLIBC_2.2)</code> <code>ioctl(GLIBC_2.2)</code> [1]	<code>readdir_r(GLIBC_2.2)</code> <code>readdir_r(GLIBC_2.2)</code> [2]	<code>setsid(GLIBC_2.2)</code> <code>setsid(GLIBC_2.2)</code> [2]
<code>__xmknod(GLIBC_2.2)</code> <code>__xmknod(GLIBC_2.2)</code> [1]	<code>fdatasync(GLIBC_2.2)</code> <code>fdatasync(GLIBC_2.2)</code> [2]	<code>kill(GLIBC_2.2)</code> <code>kill(GLIBC_2.2)</code> [1]	<code>readlink(GLIBC_2.2)</code> <code>readlink(GLIBC_2.2)</code> [2]	<code>setuid(GLIBC_2.2)</code> <code>setuid(GLIBC_2.2)</code> [2]
<code>__xstat(GLIBC_2.2)</code> <code>__xstat(GLIBC_2.2)</code> [1]	<code>flock(GLIBC_2.2)</code> <code>flock(GLIBC_2.2)</code> [1]	<code>killpg(GLIBC_2.2)</code> <code>killpg(GLIBC_2.2)</code> [2]	<code>readv(GLIBC_2.2)</code> <code>readv(GLIBC_2.2)</code> [2]	<code>sleep(GLIBC_2.2)</code> <code>sleep(GLIBC_2.2)</code> [2]
<code>access(GLIBC_2.2)</code> <code>access(GLIBC_2.2)</code> [2]	<code>fork(GLIBC_2.2)</code> <code>fork(GLIBC_2.2)</code> [2]	<code>lchown(GLIBC_2.2)</code> <code>lchown(GLIBC_2.2)</code> [2]	<code>rename(GLIBC_2.2)</code> <code>rename(GLIBC_2.2)</code> [2]	<code>statvfs(GLIBC_2.2)</code> <code>statvfs(GLIBC_2.2)</code> [2]
<code>aect(GLIBC_2.2)</code> <code>aect(GLIBC_2.2)</code> [1]	<code>fstatvfs(GLIBC_2.2)</code> <code>fstatvfs(GLIBC_2.2)</code> [2]	<code>link(GLIBC_2.2)</code> <code>link(GLIBC_2.2)</code> [2]	<code>rmdir(GLIBC_2.2)</code> <code>rmdir(GLIBC_2.2)</code> [2]	<code>stime(GLIBC_2.2)</code> <code>stime(GLIBC_2.2)</code> [1]
<code>alarm(GLIBC_2.2)</code> <code>alarm(GLIBC_2.2)</code> [2]	<code>fsync(GLIBC_2.2)</code> <code>fsync(GLIBC_2.2)</code> [2]	<code>lockf(GLIBC_2.2)</code> <code>lockf(GLIBC_2.2)</code> [2]	<code>sbrk(GLIBC_2.2)</code> <code>sbrk(GLIBC_2.2)</code> [4]	<code>symlink(GLIBC_2.2)</code> <code>symlink(GLIBC_2.2)</code> [2]
<code>brk(GLIBC_2.2)</code> <code>brk(GLIBC_2.2)</code> [4]	<code>ftime(GLIBC_2.2)</code> <code>ftime(GLIBC_2.2)</code> [2]	<code>lseek(GLIBC_2.2)</code> <code>lseek(GLIBC_2.2)</code> [2]	<code>sched_get_priority_max(GLIBC_2.2)</code> <code>sched_get_priority_max(GLIBC_2.2)</code> [2]	<code>sync(GLIBC_2.2)</code> <code>sync(GLIBC_2.2)</code> [2]
<code>chdir(GLIBC_2.2)</code> <code>chdir(GLIBC_2.2)</code> [2]	<code>ftruncate(GLIBC_2.2)</code> <code>ftruncate(GLIBC_2.2)</code> [2]	<code>mkdir(GLIBC_2.2)</code> <code>mkdir(GLIBC_2.2)</code> [2]	<code>sched_get_priority_min(GLIBC_2.2)</code> <code>sched_get_priority_min(GLIBC_2.2)</code> [2]	<code>sysconf(GLIBC_2.2)</code> <code>sysconf(GLIBC_2.2)</code> [2]

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

[2]	2) [2]		2.2) [2]	[1]
<code>execv(GLIBC_2.2)</code> <code>execv(GLIBC_2.2)</code> [2]	<code>getpriority(GLIBC_2.2)</code> <code>getpriority(GLIBC_2.2)</code> [2]	<code>opendir(GLIBC_2.2)</code> <code>opendir(GLIBC_2.2)</code> [2]	<code>setpgid(GLIBC_2.2)</code> <code>setpgid(GLIBC_2.2)</code> [2]	<code>wait4(GLIBC_2.2)</code> <code>wait4(GLIBC_2.2)</code> [1]
<code>execve(GLIBC_2.2)</code> <code>execve(GLIBC_2.2)</code> [2]	<code>getrlimit(GLIBC_2.2)</code> <code>getrlimit(GLIBC_2.2)</code> [2]	<code>pathconf(GLIBC_2.2)</code> <code>pathconf(GLIBC_2.2)</code> [2]	<code>setpgrp(GLIBC_2.2)</code> <code>setpgrp(GLIBC_2.2)</code> [2]	<code>waitpid(GLIBC_2.2)</code> <code>waitpid(GLIBC_2.2)</code> [1]
<code>execvp(GLIBC_2.2)</code> <code>execvp(GLIBC_2.2)</code> [2]	<code>getrusage(GLIBC_2.2)</code> <code>getrusage(GLIBC_2.2)</code> [2]	<code>pause(GLIBC_2.2)</code> <code>pause(GLIBC_2.2)</code> [2]	<code>setpriority(GLIBC_2.2)</code> <code>setpriority(GLIBC_2.2)</code> [2]	<code>write(GLIBC_2.2)</code> <code>write(GLIBC_2.2)</code> [2]
<code>exit(GLIBC_2.2)</code> <code>exit(GLIBC_2.2)</code> [2]	<code>getsid(GLIBC_2.2)</code> <code>getsid(GLIBC_2.2)</code> [2]	<code>pipe(GLIBC_2.2)</code> <code>pipe(GLIBC_2.2)</code> [2]	<code>setregid(GLIBC_2.2)</code> <code>setregid(GLIBC_2.2)</code> [2]	<code>writew(GLIBC_2.2)</code> <code>writew(GLIBC_2.2)</code> [2]
<code>fcntl(GLIBC_2.2)</code> <code>fcntl(GLIBC_2.2)</code> [2]	<code>getuid(GLIBC_2.2)</code> <code>getuid(GLIBC_2.2)</code> [2]	<code>poll(GLIBC_2.2)</code> <code>poll(GLIBC_2.2)</code> [2]	<code>setreuid(GLIBC_2.2)</code> <code>setreuid(GLIBC_2.2)</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_2.2)</code> <code>_IO_feof(GLIBC_2.2)</code> [1]	<code>fgetpos(GLIBC_2.2)</code> <code>fgetpos(GLIBC_2.2)</code> [2]	<code>fsetpos(GLIBC_2.2)</code> <code>fsetpos(GLIBC_2.2)</code> [2]	<code>putchar(GLIBC_2.2)</code> <code>putchar(GLIBC_2.2)</code> [2]	<code>sscanf(GLIBC_2.2)</code> <code>sscanf(GLIBC_2.2)</code> [2]
<code>_IO_getc(GLIBC_2.2)</code> <code>_IO_getc(GLIBC_2.2)</code> [1]	<code>fgets(GLIBC_2.2)</code> <code>fgets(GLIBC_2.2)</code> [2]	<code>ftell(GLIBC_2.2)</code> <code>ftell(GLIBC_2.2)</code> [2]	<code>putchar_unlocked(GLIBC_2.2)</code> <code>putchar_unlocked(GLIBC_2.2)</code> [2]	<code>telldir(GLIBC_2.2)</code> <code>telldir(GLIBC_2.2)</code> [2]
<code>_IO_putc(GLIBC_2.2)</code> <code>_IO_putc(GLIBC_2.2)</code> [1]	<code>fgetwc_unlocked(GLIBC_2.2)</code> <code>fgetwc_unlocked(GLIBC_2.2)</code> [2]	<code>ftello(GLIBC_2.2)</code> <code>ftello(GLIBC_2.2)</code> [2]	<code>puts(GLIBC_2.2)</code> <code>puts(GLIBC_2.2)</code> [2]	<code>tempnam(GLIBC_2.2)</code> <code>tempnam(GLIBC_2.2)</code> [2]

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

36

37 *Referenced Specification(s)*

38 [1]. Linux Standard Basethis 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_2.2)s tderr(GLIBC_2.2) [1]	stdin(GLIBC_2.2)st din(GLIBC_2.2) [1]	stdout(GLIBC_2.2)s tdout(GLIBC_2.2) [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_2.2) __ libc_current_sigrtm ax(GLIBC_2.2) [1]	sigaddset(GLIBC_2 _2) sigaddset(GLIBC _2.2) [2]	sighold(GLIBC_2.2) sighold(GLIBC_2. 2) [2]	sigpause(GLIBC_2. 2) sigpause(GLIBC_ 2.2) [2]	sigsuspend(GLIBC_ 2.2) sigsuspend(GLI BC_2.2) [2]
__libc_current_sigrt min(GLIBC_2.2) __l ibc_current_sigrtmi n(GLIBC_2.2) [1]	sigaltstack(GLIBC_ 2.2) sigaltstack(GLI BC_2.2) [2]	sigignore(GLIBC_2 _2) sigignore(GLIBC _2.2) [2]	sigpending(GLIBC_ 2.2) sigpending(GLI BC_2.2) [2]	sigtimedwait(GLIB C_2.2) sigtimedwait(GLIBC_2.2) [2]
__sigsetjmp(GLIBC _2.2) __sigsetjmp(G LIBC_2.2) [1]	sigandset(GLIBC_2 _2) sigandset(GLIBC _2.2) [1]	siginterrupt(GLIBC _2.2) siginterrupt(G LIBC_2.2) [2]	sigprocmask(GLIB C_2.2) sigprocmask(GLIBC_2.2) [2]	sigwait(GLIBC_2.2) sigwait(GLIBC_2. 2) [2]
__sysv_signal(GLI BC_2.2) __sysv_sig nal(GLIBC_2.2) [1]	sigblock(GLIBC_2. 2) sigblock(GLIBC_ 2.2) [1]	sigisemptyset(GLIB C_2.2) sigisemptyset (GLIBC_2.2) [1]	sigqueue(GLIBC_2. 2) sigqueue(GLIBC_ 2.2) [2]	sigwaitinfo(GLIBC _2.2) sigwaitinfo(GL IBC_2.2) [2]
bsd_signal(GLIBC_ 2.2) bsd_signal(GLI BC_2.2) [2]	sigdelset(GLIBC_2. 2) sigdelset(GLIBC_ 2.2) [2]	sigismember(GLIB C_2.2) sigismember(GLIBC_2.2) [2]	sigrelse(GLIBC_2.2) sigrelse(GLIBC_2. 2) [2]	
psignal(GLIBC_2.2) psignal(GLIBC_2.	sigemptyset(GLIBC _2.2) sigemptyset(G	siglongjmp(GLIBC _2.2) siglongjmp(GL	sigreturn(GLIBC_2. 2) sigreturn(GLIBC_	

2) [1]	LIBC_2.2) [2]	IBC_2.2) [2]	2.2) [1]	
raise(GLIBC_2.2)raise(GLIBC_2.2) [2]	sigfillset(GLIBC_2.2)sigfillset(GLIBC_2.2) [2]	signal(GLIBC_2.2)signal(GLIBC_2.2) [2]	sigset(GLIBC_2.2)sigset(GLIBC_2.2) [2]	
sigaction(GLIBC_2.2)sigaction(GLIBC_2.2) [2]	siggetmask(GLIBC_2.2)siggetmask(GLIBC_2.2) [1]	sigorset(GLIBC_2.2)sigorset(GLIBC_2.2) [1]	sigstack(GLIBC_2.2)sigstack(GLIBC_2.2) [3]	

54

55 *Referenced Specification(s)*

56 [1]. Linux Standard Basethis specification

57 [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) SUSv261 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**

_sys_siglist(GLIBC_2.2)_sys_siglist(GLIBC_2.2) [1]				
---------------------------------------------------------------	--	--	--	--

64

65 *Referenced Specification(s)*

66 [1]. Linux Standard Basethis 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**

bind_textdomain_codeset(GLIBC_2.2)bind_textdomain_codeset(GLIBC_2.2) [1]	eatopen(GLIBC_2.2)catopen(GLIBC_2.2) [2]	dngettext(GLIBC_2.2)dngettext(GLIBC_2.2) [1]	iconv_open(GLIBC_2.2)iconv_open(GLIBC_2.2) [2]	setlocale(GLIBC_2.2)setlocale(GLIBC_2.2) [2]
bindtextdomain(GLIBC_2.2)bindtextdomain(GLIBC_2.2) [1]	dcgettext(GLIBC_2.2)dcgettext(GLIBC_2.2) [1]	gettext(GLIBC_2.2)gettext(GLIBC_2.2) [1]	localeconv(GLIBC_2.2)localeconv(GLIBC_2.2) [2]	textdomain(GLIBC_2.2)textdomain(GLIBC_2.2) [1]

<code>eatclose(GLIBC_2.2)</code> <code>catclose(GLIBC_2.2)</code> [2]	<code>dgettext(GLIBC_2.2)</code> <code>dcgettext(GLIBC_2.2)</code> [1]	<code>iconv(GLIBC_2.2)</code> <code>iconv(GLIBC_2.2)</code> [2]	<code>ngettext(GLIBC_2.2)</code> <code>ngettext(GLIBC_2.2)</code> [1]	
<code>catgets(GLIBC_2.2)</code> <code>catgets(GLIBC_2.2)</code> [2]	<code>dgettext(GLIBC_2.2)</code> <code>dgettext(GLIBC_2.2)</code> [1]	<code>iconv_close(GLIBC_2.2)</code> <code>iconv_close(GLIBC_2.2)</code> [2]	<code>nl_langinfo(GLIBC_2.2)</code> <code>nl_langinfo(GLIBC_2.2)</code> [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 ~~⋮~~)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**

<code>_nl_msg_cat_cntr(GLIBC_2.2)</code> <code>_nl_msg_cat_cntr(GLIBC_2.2)</code> [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**

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

GLIBC_2.2) [1]	(GLIBC_2.2) [2]	2.2) [2]	C_2.2) [2]	
econnect(GLIBC_2.2) connect(GLIBC_2.2) [2]	getsockopt(GLIBC_2.2) getsockopt(GLIBC_2.2) [2]	send(GLIBC_2.2) send(GLIBC_2.2) [2]	socket(GLIBC_2.2) socket(GLIBC_2.2) [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**

gethostbyname_r(GLIBC_2.2) gethostbyname_r(GLIBC_2.2) [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
100 An LSB conforming implementation shall provide the architecture specific functions for Wide Characters specified in
101 Table 1-12, with the full functionality as described in the referenced underlying specification.102 **Table 1-12. libc - Wide Characters Function Interfaces**

__westod_internal(GLIBC_2.2) __westod_internal(GLIBC_2.2) [1]	mbsinit(GLIBC_2.2) mbsinit(GLIBC_2.2) [2]	vwscanf(GLIBC_2.2) vwscanf(GLIBC_2.2) [2]	wesnlen(GLIBC_2.2) wcsnlen(GLIBC_2.2) [1]	wstoumax(GLIBC_2.2) wcstoumax(GLIBC_2.2) [2]
__westof_internal(GLIBC_2.2) __westof_internal(GLIBC_2.2) [1]	mbsnrtowcs(GLIBC_2.2) mbsnrtowcs(GLIBC_2.2) [1]	wepcpy(GLIBC_2.2) wcpcpy(GLIBC_2.2) [1]	wesnrtombs(GLIBC_2.2) wcsnrtombs(GLIBC_2.2) [1]	wstouq(GLIBC_2.2) wcstouq(GLIBC_2.2) [1]
__westol_internal(GLIBC_2.2) __westol_internal(GLIBC_2.2) [1]	mbsrtowcs(GLIBC_2.2) mbsrtowcs(GLIBC_2.2) [2]	wepncpy(GLIBC_2.2) wcpcpy(GLIBC_2.2) [1]	wesprk(GLIBC_2.2) wcspbrk(GLIBC_2.2) [2]	wswcs(GLIBC_2.2) wcswcs(GLIBC_2.2) [2]

<code>__westold_internal(GLIBC_2.2) __wst old_internal(GLIBC _2.2) [1]</code>	<code>mbstowes(GLIBC_ 2.2)mbstowcs(GLIB C_2.2) [2]</code>	<code>wertomb(GLIBC_2. 2)wrtomb(GLIBC_ 2.2) [2]</code>	<code>wesrehr(GLIBC_2.2)wscrchr(GLIBC_2. 2) [2]</code>	<code>weswidth(GLIBC_2 2)wcswidth(GLIBC _2.2) [2]</code>
<code>__westoul_internal(GLIBC_2.2) __wst oul_internal(GLIBC _2.2) [1]</code>	<code>mbtowe(GLIBC_2. 2)mbtowc(GLIBC_ 2.2) [2]</code>	<code>wescasecmp(GLIB C_2.2)wscasecmp(GLIBC_2.2) [1]</code>	<code>wesrtombs(GLIBC_ 2.2)wscrombs(GLI BC_2.2) [2]</code>	<code>wesxfrm(GLIBC_2. 2)wscxfrm(GLIBC_ 2.2) [2]</code>
<code>btowe(GLIBC_2.2) btowc(GLIBC_2.2) [2]</code>	<code>putwe(GLIBC_2.2) putwc(GLIBC_2.2) [2]</code>	<code>wesecat(GLIBC_2.2) wscat(GLIBC_2.2) [2]</code>	<code>wesspn(GLIBC_2.2)wcsspn(GLIBC_2. 2) [2]</code>	<code>wetob(GLIBC_2.2) wctob(GLIBC_2.2) [2]</code>
<code>fgetwe(GLIBC_2.2) fgetwc(GLIBC_2.2) [2]</code>	<code>putwehar(GLIBC_2 2)putwchar(GLIBC _2.2) [2]</code>	<code>weschr(GLIBC_2.2) wchr(GLIBC_2.2) [2]</code>	<code>wesstr(GLIBC_2.2) wcsstr(GLIBC_2.2) [2]</code>	<code>wetomb(GLIBC_2. 2)wctomb(GLIBC_ 2.2) [2]</code>
<code>fgetws(GLIBC_2.2) fgetws(GLIBC_2.2) [2]</code>	<code>swprintf(GLIBC_2. 2)swprintf(GLIBC_ 2.2) [2]</code>	<code>wesemp(GLIBC_2. 2)wscmp(GLIBC_ 2.2) [2]</code>	<code>westod(GLIBC_2.2) westod(GLIBC_2.2) [2]</code>	<code>wetrans(GLIBC_2.2)wctrans(GLIBC_2. 2) [2]</code>
<code>fputwe(GLIBC_2.2) fputwc(GLIBC_2.2) [2]</code>	<code>swscanf(GLIBC_2. 2)swscanf(GLIBC_ 2.2) [2]</code>	<code>wescoll(GLIBC_2.2)wscoll(GLIBC_2. 2) [2]</code>	<code>westof(GLIBC_2.2) westof(GLIBC_2.2) [2]</code>	<code>wetype(GLIBC_2.2)wctype(GLIBC_2. 2) [2]</code>
<code>fputws(GLIBC_2.2) fputws(GLIBC_2.2) [2]</code>	<code>towetrans(GLIBC_2 2)towctrans(GLIB C_2.2) [2]</code>	<code>wesepy(GLIBC_2.2)wscpy(GLIBC_2. 2) [2]</code>	<code>westoimax(GLIBC_ 2.2)wctoimax(GLI BC_2.2) [2]</code>	<code>wewidth(GLIBC_2. 2)wcwidth(GLIBC_ 2.2) [2]</code>
<code>fwide(GLIBC_2.2)f wide(GLIBC_2.2) [2]</code>	<code>towlower(GLIBC_2 2)towlower(GLIBC _2.2) [2]</code>	<code>wesespn(GLIBC_2. 2)wscspn(GLIBC_ 2.2) [2]</code>	<code>westok(GLIBC_2.2) westok(GLIBC_2.2) [2]</code>	<code>wmemchr(GLIBC_ 2.2)wmemchr(GLIB C_2.2) [2]</code>
<code>fwprintf(GLIBC_2. 2)fwprintf(GLIBC_ 2.2) [2]</code>	<code>towupper(GLIBC_2 2)towupper(GLIBC _2.2) [2]</code>	<code>wesdup(GLIBC_2.2)wcdup(GLIBC_2. 2) [1]</code>	<code>westol(GLIBC_2.2) westol(GLIBC_2.2) [2]</code>	<code>wmemcmp(GLIBC_ _2.2)wmemcmp(GL IBC_2.2) [2]</code>
<code>fwscanf(GLIBC_2.2)fwscanf(GLIBC_2. 2) [2]</code>	<code>ungetwe(GLIBC_2. 2)ungetwc(GLIBC_ 2.2) [2]</code>	<code>wesftime(GLIBC_2. 2)wscftime(GLIBC _2.2) [2]</code>	<code>westold(GLIBC_2.2)wcstold(GLIBC_2. 2) [2]</code>	<code>wmemcpy(GLIBC_ 2.2)wmemcpy(GLI BC_2.2) [2]</code>
<code>getwe(GLIBC_2.2) getwc(GLIBC_2.2) [2]</code>	<code>vfwprintf(GLIBC_2 2)vfwprintf(GLIBC _2.2) [2]</code>	<code>weslen(GLIBC_2.2) wscnlen(GLIBC_2.2) [2]</code>	<code>westoll(GLIBC_2.2)wcstoll(GLIBC_2. 2) [2]</code>	<code>wmemmove(GLIB C_2.2)wmemmove(GLIBC_2.2) [2]</code>
<code>getwehar(GLIBC_2. 2)getwchar(GLIBC _2.2) [2]</code>	<code>vfwscanf(GLIBC_2. 2)vfwscanf(GLIBC _2.2) [2]</code>	<code>wescasecmp(GLIB C_2.2)wscasecmp (GLIBC_2.2) [1]</code>	<code>westombs(GLIBC_ 2.2)wcstombs(GLIB C_2.2) [2]</code>	<code>wmemset(GLIBC_2 2)wmemset(GLIBC _2.2) [2]</code>
<code>mblen(GLIBC_2.2) mblen(GLIBC_2.2)</code>	<code>vswprintf(GLIBC_2 2)vswprintf(GLIBC</code>	<code>wesnecat(GLIBC_2. 2)wscncat(GLIBC_</code>	<code>westoq(GLIBC_2.2) westoq(GLIBC_2.2)</code>	<code>wprintf(GLIBC_2.2)wprintf(GLIBC_2.</code>

[2]	_2.2) [2]	2.2) [2]	[1]	2) [2]
mbrlen(GLIBC_2.2) mbrlen(GLIBC_2.2) [2]	vswscanf(GLIBC_2.2) vswscanf(GLIBC_2.2) [2]	wesnemp(GLIBC_2.2))wscncmp(GLIBC_2.2) [2]	westoul(GLIBC_2.2))wcstoul(GLIBC_2.2) [2]	wscanf(GLIBC_2.2) wscanf(GLIBC_2.2) [2]
mbrtowc(GLIBC_2.2) mbrtowc(GLIBC_2.2) [2]	vwprintf(GLIBC_2.2) vwprintf(GLIBC_2.2) [2]	wesnecpy(GLIBC_2.2))wscncpy(GLIBC_2.2) [2]	westoull(GLIBC_2.2))wcstoull(GLIBC_2.2) [2]	

103

104 *Referenced Specification(s)*

105 [1]. Linux Standard Base this specification

106 [2]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)

107 V3)

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**

__mempepy(GLIBC_2.2) __mempcpy(GLIBC_2.2) [1]	bzero(GLIBC_2.2) bzero(GLIBC_2.2) [2]	streasestr(GLIBC_2.2))strcasestr(GLIBC_2.2) [1]	strncasecmp(GLIBC_2.2))strncasecmp(GLIBC_2.2) [2]	strtoimax(GLIBC_2.2))strtoimax(GLIBC_2.2) [2]
__rawmemchr(GLIBC_2.2) __rawmemchr(GLIBC_2.2) [1]	ffs(GLIBC_2.2) ffs(GLIBC_2.2) [2]	streat(GLIBC_2.2))strcat(GLIBC_2.2) [2]	strncat(GLIBC_2.2))strncat(GLIBC_2.2) [2]	strtok(GLIBC_2.2))strtok(GLIBC_2.2) [2]
__stpepy(GLIBC_2.2) __stpcpy(GLIBC_2.2) [1]	index(GLIBC_2.2) index(GLIBC_2.2) [2]	strchr(GLIBC_2.2))strchr(GLIBC_2.2) [2]	strnemp(GLIBC_2.2))strncmp(GLIBC_2.2) [2]	strtok_r(GLIBC_2.2))strtok_r(GLIBC_2.2) [4+2]
__strdup(GLIBC_2.2) __strdup(GLIBC_2.2) [1]	memecpy(GLIBC_2.2))mempcpy(GLIBC_2.2) [2]	strempp(GLIBC_2.2))strcmp(GLIBC_2.2) [2]	strncpy(GLIBC_2.2))strncpy(GLIBC_2.2) [2]	strtold(GLIBC_2.2))strtold(GLIBC_2.2) [2]
__strtod_internal(GLIBC_2.2))__strtod_internal(GLIBC_2.2) [1]	memchr(GLIBC_2.2))memchr(GLIBC_2.2) [2]	stroll(GLIBC_2.2))strcoll(GLIBC_2.2) [2]	strndup(GLIBC_2.2))strndup(GLIBC_2.2) [1]	strtoll(GLIBC_2.2))strtoll(GLIBC_2.2) [2]
__strtof_internal(GLIBC_2.2))__strtof_internal(GLIBC_2.2) [1]	mememp(GLIBC_2.2))mempcpy(GLIBC_2.2) [2]	strepypy(GLIBC_2.2))strcpy(GLIBC_2.2) [2]	strnlen(GLIBC_2.2))strnlen(GLIBC_2.2) [1]	strtoq(GLIBC_2.2))strtoq(GLIBC_2.2) [1]
__strtok_r(GLIBC_2.2)	mempypy(GLIBC_2.2)	strespn(GLIBC_2.2)	strpbrk(GLIBC_2.2)	strtoull(GLIBC_2.2)

<code>__strtok_r(GLIBC_2.2)</code> [1]	<code>memcpy(GLIBC_2.2)</code> [2]	<code>strcspn(GLIBC_2.2)</code> [2]	<code>strpbrk(GLIBC_2.2)</code> [2]	<code>strtoull(GLIBC_2.2)</code> [2]
<code>__strtol_internal(GLIBC_2.2)</code> [1]	<code>memmove(GLIBC_2.2)</code> [2]	<code>strdup(GLIBC_2.2)</code> [2]	<code>strptime(GLIBC_2.2)</code> [1]	<code>strtoumax(GLIBC_2.2)</code> [2]
<code>__strtold_internal(GLIBC_2.2)</code> [1]	<code>memrchr(GLIBC_2.2)</code> [1]	<code>strerror(GLIBC_2.2)</code> [2]	<code>strchr(GLIBC_2.2)</code> [2]	<code>strtouq(GLIBC_2.2)</code> [1]
<code>__strtoll_internal(GLIBC_2.2)</code> [1]	<code>memset(GLIBC_2.2)</code> [2]	<code>strerror_r(GLIBC_2.2)</code> [1]	<code>strsep(GLIBC_2.2)</code> [1]	<code>strverscmp(GLIBC_2.2)</code> [1]
<code>__strtoul_internal(GLIBC_2.2)</code> [1]	<code>rindex(GLIBC_2.2)</code> [2]	<code>strfmon(GLIBC_2.2)</code> [2]	<code>strsignal(GLIBC_2.2)</code> [1]	<code>strxfrm(GLIBC_2.2)</code> [2]
<code>__strtoull_internal(GLIBC_2.2)</code> [1]	<code>stpcpy(GLIBC_2.2)</code> [1]	<code>strfry(GLIBC_2.2)</code> [1]	<code>strspn(GLIBC_2.2)</code> [2]	<code>swab(GLIBC_2.2)</code> [2]
<code>bcmp(GLIBC_2.2)</code> [2]	<code>stpncpy(GLIBC_2.2)</code> [1]	<code>strftime(GLIBC_2.2)</code> [2]	<code>strstr(GLIBC_2.2)</code> [2]	
<code>bcopy(GLIBC_2.2)</code> [2]	<code>strcasecmp(GLIBC_2.2)</code> [2]	<code>strlen(GLIBC_2.2)</code> [2]	<code>strtof(GLIBC_2.2)</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_2.2)</code>	<code>msgrev(GLIBC_2.2)</code>	<code>semget(GLIBC_2.2)</code>	<code>shmetl(GLIBC_2.2)</code>	
------------------------------	--------------------------------	--------------------------------	--------------------------------	--

k(GLIBC_2.2) [1]	̀msgrcv(GLIBC_2.2) [1]	semget(GLIBC_2.2) [1]	shmctl(GLIBC_2.2) [1]	
msgctl(GLIBC_2.2) [1]	msgsnd(GLIBC_2.2) [1]	semop(GLIBC_2.2) [1]	shmdt(GLIBC_2.2) [1]	
msgget(GLIBC_2.2) [1]	semctl(GLIBC_2.2) [1]	shmat(GLIBC_2.2) [1]	shmget(GLIBC_2.2) [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_2.2) [1]	regerror(GLIBC_2.2) [1]	regexec(GLIBC_2.2) [1]	regfree(GLIBC_2.2) [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_2.2) [1]	re_comp(GLIBC_2.2) [1]	re_exec(GLIBC_2.2) [1]	step(GLIBC_2.2) [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	<code>loel(GLIBC_2.2)lo c1(GLIBC_2.2) [1]</code>	<code>loe2(GLIBC_2.2)lo c2(GLIBC_2.2) [1]</code>	<code>loes(GLIBC_2.2)loc s(GLIBC_2.2) [1]</code>		
-----	------------------------------------------------------	------------------------------------------------------	------------------------------------------------------	--	--

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	<code>__ctype_get_mb_cu r_max(GLIBC_2.2) __ctype_get_mb_cu r_max(GLIBC_2.2) [1]</code>	<code>isdigit(GLIBC_2.2)i sdigit(GLIBC_2.2) [2]</code>	<code>iswalnum(GLIBC_2. 2)iswalnum(GLIB C_2.2) [2]</code>	<code>iswlower(GLIBC_2. 2)iswlower(GLIBC _2.2) [2]</code>	<code>toascii(GLIBC_2.2) toascii(GLIBC_2.2) [2]</code>
	<code>_tolower(GLIBC_2. 2)_tolower(GLIBC_ 2.2) [2]</code>	<code>isgraph(GLIBC_2.2) isgraph(GLIBC_2. 2) [2]</code>	<code>iswalpha(GLIBC_2. 2)iswalpha(GLIBC_ 2.2) [2]</code>	<code>iswprint(GLIBC_2. 2)iswprint(GLIBC_ 2.2) [2]</code>	<code>tolower(GLIBC_2.2) tolower(GLIBC_2. 2) [2]</code>
	<code>_toupper(GLIBC_2. 2)_toupper(GLIBC_ 2.2) [2]</code>	<code>islower(GLIBC_2.2) islower(GLIBC_2. 2) [2]</code>	<code>iswblank(GLIBC_2. 2)iswblank(GLIBC _2.2) [2]</code>	<code>iswpunct(GLIBC_2. 2)iswpunct(GLIBC _2.2) [2]</code>	<code>toupper(GLIBC_2.2) toupper(GLIBC_2. 2) [2]</code>
	<code>isalnum(GLIBC_2.2) isalnum(GLIBC_2. 2) [2]</code>	<code>isprint(GLIBC_2.2)i sprint(GLIBC_2.2) [2]</code>	<code>iswcntrl(GLIBC_2. 2)iswcntrl(GLIBC_ 2.2) [2]</code>	<code>iswspace(GLIBC_2. 2)iswspace(GLIBC _2.2) [2]</code>	
	<code>isalpha(GLIBC_2.2) isalpha(GLIBC_2.2) [2]</code>	<code>ispunct(GLIBC_2.2) ispunct(GLIBC_2. 2) [2]</code>	<code>iswctype(GLIBC_2. 2)iswctype(GLIBC_ 2.2) [1+2]</code>	<code>iswupper(GLIBC_2. 2)iswupper(GLIBC _2.2) [2]</code>	
	<code>isascii(GLIBC_2.2)i sascii(GLIBC_2.2) [2]</code>	<code>isspace(GLIBC_2.2) isspace(GLIBC_2. 2) [2]</code>	<code>iswdigit(GLIBC_2. 2)iswdigit(GLIBC_ 2.2) [2]</code>	<code>iswxdigit(GLIBC_2. 2)iswxdigit(GLIBC _2.2) [2]</code>	
	<code>isctrl(GLIBC_2.2)i sctrl(GLIBC_2.2) [2]</code>	<code>isupper(GLIBC_2.2) isupper(GLIBC_2. 2) [2]</code>	<code>iswgraph(GLIBC_2. 2)iswgraph(GLIBC _2.2) [2]</code>	<code>isxdigit(GLIBC_2.2) isxdigit(GLIBC_2. 2) [2]</code>	

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**

adjtime(GLIBC_2.2)adjtime(GLIBC_2. 2) [1]	etime(GLIBC_2.2)c time(GLIBC_2.2) [2]	gmtime(GLIBC_2.2)gmtime(GLIBC_2. 2) [2]	localtime_r(GLIBC _2.2)localtime_r(G LIBC_2.2) [2]	ualarm(GLIBC_2.2) ualarm(GLIBC_2.2) [2]
asctime(GLIBC_2.2)asctime(GLIBC_2. 2) [2]	etime_r(GLIBC_2.2)ctime_r(GLIBC_2. 2) [2]	gmtime_r(GLIBC_2 .2)gmtime_r(GLIB C_2.2) [2]	mktime(GLIBC_2.2)mktime(GLIBC_2. 2) [2]	
asctime_r(GLIBC_2 .2)asctime_r(GLIB C_2.2) [2]	difftime(GLIBC_2. 2)difftime(GLIBC_ 2.2) [2]	localtime(GLIBC_2)localtime(GLIBC _2.2) [2]	tzset(GLIBC_2.2)tz set(GLIBC_2.2) [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**

adjtimex(GLIBC_2. 2)adjtimex(GLIBC_ 2.2) [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**

__daylight(GLIBC_2.2) __daylight(GLIBC_2.2) [1]	__tzname(GLIBC_2.2) __tzname(GLIBC_2.2) [1]	timezone(GLIBC_2.2) timezone(GLIBC_2.2) [2]		
__timezone(GLIBC_2.2) __timezone(GLIBC_2.2) [1]	daylight(GLIBC_2.2) daylight(GLIBC_2.2) [2]	tzname(GLIBC_2.2) tzname(GLIBC_2.2) [2]		

181

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)~~185 ~~∇3)~~

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**

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

190

191 *Referenced Specification(s)*192 [1]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS)~~193 ~~∇3)~~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**

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

210	<code>__libc_start_main(GLIBC_2.2) __libc_ start_main(GLIBC_ 2.2) [1]</code>	<code>__obstack_begin(GLI IBC_2.2) __obstack_ begin(GLIBC_2.2) [1]</code>	<code>__obstack_newchunk (GLIBC_2.2) __obsta ck_newchunk(GLIB C_2.2) [1]</code>	<code>obstack_free(GLIB C_2.2) obstack_free(GLIBC_2.2) [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**

217	<code>__fxstat64(GLIBC_ 2.2) __fxstat64(GLI BC_2.2) [1]</code>	<code>fopen64(GLIBC_2. 2)fopen64(GLIBC_ 2.2) [2]</code>	<code>ftello64(GLIBC_2.2)ftello64(GLIBC_2. 2) [2]</code>	<code>lseek64(GLIBC_2.2)lseek64(GLIBC_2. 2) [2]</code>	<code>readdir64(GLIBC_2)readdir64(GLIBC _2.2) [2]</code>
	<code>__lxstat64(GLIBC_ 2.2) __lxstat64(GLI BC_2.2) [1]</code>	<code>freopen64(GLIBC_ 2.2)freopen64(GLI BC_2.2) [2]</code>	<code>ftruncate64(GLIBC _2.2)ftruncate64(G LIBC_2.2) [2]</code>	<code>mkstemp64(GLIBC _2.2)mkstemp64(G LIBC_2.2) [2]</code>	<code>statvfs64(GLIBC_2. 2)statvfs64(GLIBC _2.2) [2]</code>
	<code>__xstat64(GLIBC_ 2.2) __xstat64(GLIB C_2.2) [1]</code>	<code>fseeko64(GLIBC_2. 2)fseeko64(GLIBC _2.2) [2]</code>	<code>ftw64(GLIBC_2.2)f tw64(GLIBC_2.2) [2]</code>	<code>mmap64(GLIBC_2. 2)mmap64(GLIBC_ 2.2) [2]</code>	<code>tmpfile64(GLIBC_2)tmpfile64(GLIB C_2.2) [2]</code>
	<code>creat64(GLIBC_2.2)creat64(GLIBC_2. 2) [2]</code>	<code>fsetpos64(GLIBC_2)fsetpos64(GLIBC _2.2) [2]</code>	<code>getrlimit64(GLIBC _2.2)getrlimit64(GL IBC_2.2) [2]</code>	<code>nftw64(GLIBC_2.2) nftw64(GLIBC_2.2) [2]</code>	<code>truncate64(GLIBC_ 2.2)truncate64(GLI BC_2.2) [2]</code>
	<code>fgetpos64(GLIBC_ 2.2)fgetpos64(GLIB C_2.2) [2]</code>	<code>fstatvfs64(GLIBC_ 2.2)fstatvfs64(GLIB C_2.2) [2]</code>	<code>lockf64(GLIBC_2.2)lockf64(GLIBC_2. 2) [2]</code>	<code>open64(GLIBC_2.2)open64(GLIBC_2. 2) [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_2.2)</code> <code>Exit(GLIBC_2.2)</code> [1]	<code>dirname(GLIBC_2.2)</code> <code>dirname(GLIBC_2.2)</code> [1]	<code>glob(GLIBC_2.2)</code> <code>glob(GLIBC_2.2)</code> [1]	<code>lsearch(GLIBC_2.2)</code> <code>lsearch(GLIBC_2.2)</code> [1]	<code>srand(GLIBC_2.2)</code> <code>rand(GLIBC_2.2)</code> [1]
<code>__assert_fail(GLIBC_2.2)</code> <code>__assert_fail(GLIBC_2.2)</code> [2]	<code>div(GLIBC_2.2)</code> <code>div(GLIBC_2.2)</code> [1]	<code>glob64(GLIBC_2.2)</code> <code>glob64(GLIBC_2.2)</code> [2]	<code>makecontext(GLIBC_2.2)</code> <code>makecontext(GLIBC_2.2)</code> [1]	<code>srand48(GLIBC_2.2)</code> <code>rand48(GLIBC_2.2)</code> [1]
<code>__cxa_atexit(GLIBC_2.2)</code> <code>__cxa_atexit(GLIBC_2.2)</code> [2]	<code>drand48(GLIBC_2.2)</code> <code>drand48(GLIBC_2.2)</code> [1]	<code>globfree(GLIBC_2.2)</code> <code>globfree(GLIBC_2.2)</code> [1]	<code>malloc(GLIBC_2.2)</code> <code>malloc(GLIBC_2.2)</code> [1]	<code>srandom(GLIBC_2.2)</code> <code>random(GLIBC_2.2)</code> [1]
<code>__errno_location(GLIBC_2.2)</code> <code>__errno_location(GLIBC_2.2)</code> [2]	<code>ecvt(GLIBC_2.2)</code> <code>ecvt(GLIBC_2.2)</code> [1]	<code>globfree64(GLIBC_2.2)</code> <code>globfree64(GLIBC_2.2)</code> [2]	<code>memmem(GLIBC_2.2)</code> <code>memmem(GLIBC_2.2)</code> [2]	<code>strtod(GLIBC_2.2)</code> <code>strtod(GLIBC_2.2)</code> [1]
<code>__fpending(GLIBC_2.2)</code> <code>__fpending(GLIBC_2.2)</code> [2]	<code>erand48(GLIBC_2.2)</code> <code>erand48(GLIBC_2.2)</code> [1]	<code>grantpt(GLIBC_2.2)</code> <code>grantpt(GLIBC_2.2)</code> [1]	<code>mkstemp(GLIBC_2.2)</code> <code>mkstemp(GLIBC_2.2)</code> [1]	<code>strtol(GLIBC_2.2)</code> <code>strtol(GLIBC_2.2)</code> [1]
<code>__getpagesize(GLIBC_2.2)</code> <code>__getpagesize(GLIBC_2.2)</code> [2]	<code>err(GLIBC_2.2)</code> <code>err(GLIBC_2.2)</code> [2]	<code>hcreate(GLIBC_2.2)</code> <code>hcreate(GLIBC_2.2)</code> [1]	<code>mktemp(GLIBC_2.2)</code> <code>mktemp(GLIBC_2.2)</code> [1]	<code>strtoul(GLIBC_2.2)</code> <code>strtoul(GLIBC_2.2)</code> [1]
<code>__isinf(GLIBC_2.2)</code> <code>__isinf(GLIBC_2.2)</code> [2]	<code>error(GLIBC_2.2)</code> <code>error(GLIBC_2.2)</code> [2]	<code>hdestroy(GLIBC_2.2)</code> <code>hdestroy(GLIBC_2.2)</code> [1]	<code>mrand48(GLIBC_2.2)</code> <code>mrand48(GLIBC_2.2)</code> [1]	<code>swapecontext(GLIBC_2.2)</code> <code>swapcontext(GLIBC_2.2)</code> [1]
<code>__isinff(GLIBC_2.2)</code> <code>__isinff(GLIBC_2.2)</code> [2]	<code>errx(GLIBC_2.2)</code> <code>errx(GLIBC_2.2)</code> [2]	<code>hsearch(GLIBC_2.2)</code> <code>hsearch(GLIBC_2.2)</code> [1]	<code>nftw(GLIBC_2.2)</code> <code>nftw(GLIBC_2.2)</code> [1]	<code>syslog(GLIBC_2.2)</code> <code>syslog(GLIBC_2.2)</code> [1]
<code>__isinfl(GLIBC_2.2)</code> <code>__isinfl(GLIBC_2.2)</code> [2]	<code>fevt(GLIBC_2.2)</code> <code>fevt(GLIBC_2.2)</code> [1]	<code>htonl(GLIBC_2.2)</code> <code>htonl(GLIBC_2.2)</code> [1]	<code>nrand48(GLIBC_2.2)</code> <code>nrand48(GLIBC_2.2)</code> [1]	<code>system(GLIBC_2.2)</code> <code>system(GLIBC_2.2)</code> [2]
<code>__isnan(GLIBC_2.2)</code> <code>__isnan(GLIBC_2.2)</code> [2]	<code>fmtmsg(GLIBC_2.2)</code> <code>fmtmsg(GLIBC_2.2)</code> [1]	<code>htons(GLIBC_2.2)</code> <code>htons(GLIBC_2.2)</code> [1]	<code>ntohl(GLIBC_2.2)</code> <code>ntohl(GLIBC_2.2)</code> [1]	<code>tdelete(GLIBC_2.2)</code> <code>tdelete(GLIBC_2.2)</code> [1]
<code>__isnanf(GLIBC_2.2)</code> <code>__isnanf(GLIBC_2.2)</code> [2]	<code>fnmatch(GLIBC_2.2)</code> <code>fnmatch(GLIBC_2.2)</code> [1]	<code>imaxabs(GLIBC_2.2)</code> <code>imaxabs(GLIBC_2.2)</code> [1]	<code>ntohs(GLIBC_2.2)</code> <code>ntohs(GLIBC_2.2)</code> [1]	<code>tfind(GLIBC_2.2)</code> <code>tfind(GLIBC_2.2)</code> [1]
<code>__isnank(GLIBC_2.2)</code> <code>__isnank(GLIBC_2.2)</code> [2]	<code>fpathconf(GLIBC_2.2)</code> <code>fpathconf(GLIBC_2.2)</code> [1]	<code>imaxdiv(GLIBC_2.2)</code> <code>imaxdiv(GLIBC_2.2)</code> [1]	<code>openlog(GLIBC_2.2)</code> <code>openlog(GLIBC_2.2)</code> [1]	<code>tmpfile(GLIBC_2.2)</code> <code>tmpfile(GLIBC_2.2)</code> [1]
<code>__sysconf(GLIBC_2.2)</code> <code>__sysconf(GLIBC_2.2)</code> [1]	<code>free(GLIBC_2.2)</code> <code>free(GLIBC_2.2)</code> [1]	<code>inet_addr(GLIBC_2.2)</code> <code>inet_addr(GLIBC_2.2)</code> [1]	<code>perror(GLIBC_2.2)</code> <code>perror(GLIBC_2.2)</code> [1]	<code>tmpnam(GLIBC_2.2)</code> <code>tmpnam(GLIBC_2.2)</code> [1]

BC_2.2) [2]		_2.2) [1]	[1]	2.2) [1]
_exit(GLIBC_2.2) exit(GLIBC_2.2) [1]	freeaddrinfo(GLIBC_2.2) freeaddrinfo(GLIBC_2.2) [1]	inet_ntoa(GLIBC_2.2) inet_ntoa(GLIBC_2.2) [1]	posix_memalign(GLIBC_2.2) posix_memalign(GLIBC_2.2) [1]	tsearch(GLIBC_2.2) tsearch(GLIBC_2.2) [1]
_longjmp(GLIBC_2.2) _longjmp(GLIBC_2.2) [1]	ftw(GLIBC_2.2) ftw(GLIBC_2.2) [1]	inet_ntop(GLIBC_2.2) inet_ntop(GLIBC_2.2) [1]	ptsname(GLIBC_2.2) ptsname(GLIBC_2.2) [1]	ttynamename(GLIBC_2.2) ttynamename(GLIBC_2.2) [1]
_setjmp(GLIBC_2.2) _setjmp(GLIBC_2.2) [1]	ftw(GLIBC_2.2) ftw(GLIBC_2.2) [1]	inet_pton(GLIBC_2.2) inet_pton(GLIBC_2.2) [1]	putenv(GLIBC_2.2) putenv(GLIBC_2.2) [1]	ttynamename_r(GLIBC_2.2) ttynamename_r(GLIBC_2.2) [1]
a64(GLIBC_2.2) a64(GLIBC_2.2) [1]	funlockfile(GLIBC_2.2) funlockfile(GLIBC_2.2) [1]	initstate(GLIBC_2.2) initstate(GLIBC_2.2) [1]	qsort(GLIBC_2.2) qsort(GLIBC_2.2) [1]	twalk(GLIBC_2.2) twalk(GLIBC_2.2) [1]
abort(GLIBC_2.2) abort(GLIBC_2.2) [1]	gai_strerror(GLIBC_2.2) gai_strerror(GLIBC_2.2) [1]	insque(GLIBC_2.2) insque(GLIBC_2.2) [1]	rand(GLIBC_2.2) rand(GLIBC_2.2) [1]	unlockpt(GLIBC_2.2) unlockpt(GLIBC_2.2) [1]
abs(GLIBC_2.2) abs(GLIBC_2.2) [1]	gevt(GLIBC_2.2) gevt(GLIBC_2.2) [1]	isatty(GLIBC_2.2) isatty(GLIBC_2.2) [1]	rand_r(GLIBC_2.2) rand_r(GLIBC_2.2) [1]	unsetenv(GLIBC_2.2) unsetenv(GLIBC_2.2) [1]
atof(GLIBC_2.2) atof(GLIBC_2.2) [1]	getaddrinfo(GLIBC_2.2) getaddrinfo(GLIBC_2.2) [1]	isblank(GLIBC_2.2) isblank(GLIBC_2.2) [1]	random(GLIBC_2.2) random(GLIBC_2.2) [1]	usleep(GLIBC_2.2) usleep(GLIBC_2.2) [1]
atoi(GLIBC_2.2) atoi(GLIBC_2.2) [1]	getcwd(GLIBC_2.2) getcwd(GLIBC_2.2) [1]	jrand48(GLIBC_2.2) jrand48(GLIBC_2.2) [1]	random_r(GLIBC_2.2) random_r(GLIBC_2.2) [2]	verrx(GLIBC_2.2) verrx(GLIBC_2.2) [2]
atol(GLIBC_2.2) atol(GLIBC_2.2) [1]	getdate(GLIBC_2.2) getdate(GLIBC_2.2) [1]	l64a(GLIBC_2.2) l64a(GLIBC_2.2) [1]	realloc(GLIBC_2.2) realloc(GLIBC_2.2) [1]	vfscanf(GLIBC_2.2) vfscanf(GLIBC_2.2) [1]
atoll(GLIBC_2.2) atoll(GLIBC_2.2) [1]	getenv(GLIBC_2.2) getenv(GLIBC_2.2) [1]	labs(GLIBC_2.2) labs(GLIBC_2.2) [1]	realpath(GLIBC_2.3) realpath(GLIBC_2.3) [1]	vscanf(GLIBC_2.2) vscanf(GLIBC_2.2) [1]
basename(GLIBC_2.2) basename(GLIBC_2.2) [1]	getlogin(GLIBC_2.2) getlogin(GLIBC_2.2) [1]	leong48(GLIBC_2.2) leong48(GLIBC_2.2) [1]	remque(GLIBC_2.2) remque(GLIBC_2.2) [1]	vsscanf(GLIBC_2.2) vsscanf(GLIBC_2.2) [1]
bsearch(GLIBC_2.2) bsearch(GLIBC_2.2) [1]	getnameinfo(GLIBC_2.2) getnameinfo(GLIBC_2.2) [1]	ldiv(GLIBC_2.2) ldiv(GLIBC_2.2) [1]	seed48(GLIBC_2.2) seed48(GLIBC_2.2) [1]	vsyslog(GLIBC_2.2) vsyslog(GLIBC_2.2) [2]
calloc(GLIBC_2.2) calloc(GLIBC_2.2)	getopt(GLIBC_2.2) getopt(GLIBC_2.2)	lfind(GLIBC_2.2) lfind(GLIBC_2.2) [1]	setenv(GLIBC_2.2) setenv(GLIBC_2.2)	warn(GLIBC_2.2) warn(GLIBC_2.2) [2]

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

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**

__environ(GLIBC_2.2)__environ(GLIBC_2.2) [1]	__sys_errlist(GLIBC_2.2)__sys_errlist(GLIBC_2.2) [1]	getdate_err(GLIBC_2.2)getdate_err(GLIBC_2.2) [2]	opterr(GLIBC_2.2)opterr(GLIBC_2.2) [1]	optopt(GLIBC_2.2)optopt(GLIBC_2.2) [1]
__environ(GLIBC_2.2)__environ(GLIBC_2.2) [1]	environ(GLIBC_2.2)environ(GLIBC_2.2) [2]	optarg(GLIBC_2.2)optarg(GLIBC_2.2) [2]	optind(GLIBC_2.2)optind(GLIBC_2.2) [1]	

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          35
```

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[18];
```

1.3.5. signal.h

```
262
263 #define __NUM_ACRS          16
264 #define __NUM_FPRS          16
265 #define __NUM_GPRS          16
266
267 typedef struct
268 {
269     unsigned long mask;
270     unsigned long addr;
271 }
```

```

272 __attribute__((aligned (8))) _psw_t;
273 typedef struct
274 {
275     _psw_t psw;
276     unsigned long gprs[16];
277     unsigned int acrs[16];
278 }
279 _s390_regs_common;
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     unsigned long sa_flags;
290     void (*sa_restorer) (void);
291     sigset_t sa_mask;
292 }
293 ;
294 #define MINSIGSTKSZ      2048
295 #define SIGSTKSZ        8192
296
297 typedef struct
298 {
299     unsigned int fpc;
300     double fprs[__NUM_FPRS];
301 }
302 _s390_fp_regs;
303 typedef struct
304 {
305     _s390_regs_common regs;
306     _s390_fp_regs fpregs;
307 }
308 _sigregs;
309
310 struct sigcontext
311 {
312     unsigned long oldmask;
313     _sigregs *sregs;
314 }
315 ;

```

1.3.6. stddef.h

```

316
317 typedef unsigned long size_t;
318 typedef long ptrdiff_t;

```


1.3.7. sys/ioctl.h

```

319
320 #define FIONREAD          21531
321 #define TIOCNOTTY       21538

```

1.3.8. sys/ipc.h

```

322
323 struct ipc_perm
324 {
325     key_t __key;
326     uid_t uid;
327     gid_t gid;
328     uid_t cuid;
329     gid_t cgid;
330     mode_t mode;
331     unsigned short __seq;
332     unsigned short __pad2;
333     unsigned long __unused1;
334     unsigned long __unused2;
335 }
336 ;

```

1.3.9. sys/mman.h

```

337
338 #define MCL_CURRENT      1
339 #define MCL_FUTURE      2

```

1.3.10. sys/msg.h

```

340
341 typedef unsigned long msgqnum_t;
342 typedef unsigned long msglen_t;
343
344 struct msqid_ds
345 {
346     struct ipc_perm msg_perm;
347     time_t msg_stime;
348     time_t msg_rtime;
349     time_t msg_ctime;
350     unsigned long __msg_cbytes;
351     msgqnum_t msg_qnum;
352     msglen_t msg_qbytes;
353     pid_t msg_lspid;
354     pid_t msg_lrpid;
355     unsigned long __unused4;
356     unsigned long __unused5;
357 }
358 ;

```

1.3.11. sys/sem.h

```

359
360 struct semid_ds
361 {
362     struct ipc_perm sem_perm;
363     time_t sem_otime;
364     time_t sem_ctime;
365     unsigned long sem_nsems;
366     unsigned long __unused3;
367     unsigned long __unused4;
368 }
369 ;

```

1.3.12. sys/shm.h

```

370
371 #define SHMLBA 4096
372
373 typedef unsigned long shmatt_t;
374
375 struct shmid_ds
376 {
377     struct ipc_perm shm_perm;
378     size_t shm_segsz;
379     time_t shm_atime;
380     time_t shm_dtime;
381     time_t shm_ctime;
382     pid_t shm_cpid;
383     pid_t shm_lpid;
384     shmatt_t shm_nattch;
385     unsigned long __unused4;
386     unsigned long __unused5;
387 }
388 ;

```

1.3.13. sys/socket.h

```

389
390 typedef uint64_t __ss_aligntype;

```

1.3.14. sys/stat.h

```

391
392 #define _STAT_VER 1
393
394 struct stat
395 {
396     dev_t st_dev;
397     ino_t st_ino;
398     nlink_t st_nlink;

```

```

399     mode_t st_mode;
400     uid_t st_uid;
401     gid_t st_gid;
402     int pad0;
403     dev_t st_rdev;
404     off_t st_size;
405     struct timespec st_atim;
406     struct timespec st_mtim;
407     struct timespec st_ctim;
408     blksize_t st_blksize;
409     blkcnt_t st_blocks;
410     long __unused[3];
411 }
412 ;
413 struct stat64
414 {
415     dev_t st_dev;
416     ino64_t st_ino;
417     nlink_t st_nlink;
418     mode_t st_mode;
419     uid_t st_uid;
420     gid_t st_gid;
421     int pad0;
422     dev_t st_rdev;
423     off_t st_size;
424     struct timespec st_atim;
425     struct timespec st_mtim;
426     struct timespec st_ctim;
427     blksize_t st_blksize;
428     blkcnt64_t st_blocks;
429     long __unused[3];
430 }
431 ;

```

1.3.15. sys/statvfs.h

```

432
433 struct statvfs
434 {
435     unsigned long f_bsize;
436     unsigned long f_frsize;
437     fsblkcnt64_t f_blocks;
438     fsblkcnt64_t f_bfree;
439     fsblkcnt64_t f_bavail;
440     fsfilcnt64_t f_files;
441     fsfilcnt64_t f_ffree;
442     fsfilcnt64_t f_favail;
443     unsigned long f_fsid;
444     unsigned long f_flag;
445     unsigned long f_namemax;
446     int __f_spare[6];
447 }

```

```

448     ;
449     struct statvfs64
450     {
451         unsigned long f_bsize;
452         unsigned long f_frsize;
453         fsblkcnt64_t f_blocks;
454         fsblkcnt64_t f_bfree;
455         fsblkcnt64_t f_bavail;
456         fsfilcnt64_t f_files;
457         fsfilcnt64_t f_ffree;
458         fsfilcnt64_t f_favail;
459         unsigned long f_fsid;
460         unsigned long f_flag;
461         unsigned long f_namemax;
462         int __f_spare[6];
463     }
464     ;

```

1.3.16. sys/types.h

```

465
466     typedef long int64_t;
467
468     typedef int64_t ssize_t;

```

1.3.17. termios.h

```

469
470     #define CR2      1024
471     #define CR3      1536
472     #define CRDLY    1536
473     #define VT1      16384
474     #define VTDLY    16384
475     #define OLCUC    2
476     #define TAB1     2048
477     #define NLDLY    256
478     #define FF1      32768
479     #define FFDLY    32768
480     #define ONLCR    4
481     #define XCASE    4
482     #define TAB2     4096
483     #define CR1      512
484     #define IUCLC    512
485     #define TAB3     6144
486     #define TABDLY   6144
487     #define BS1      8192
488     #define BSDLY    8192
489
490     #define VSUSP    10
491     #define VEOL     11
492     #define VREPRINT 12
493     #define VDISCARD 13

```

```

494 #define VWERASE 14
495 #define VEOL2 16
496 #define VMIN 6
497 #define VSWTC 7
498 #define VSTART 8
499 #define VSTOP 9
500
501 #define IXON 1024
502 #define IXOFF 4096
503
504 #define HUPCL 1024
505 #define CREAD 128
506 #define CS6 16
507 #define CLOCAL 2048
508 #define PARENB 256
509 #define CS7 32
510 #define CS8 48
511 #define CSIZE 48
512 #define VTIME 5
513 #define PARODD 512
514 #define CSTOPB 64
515
516 #define ISIG 1
517 #define ECHOPRT 1024
518 #define NOFLSH 128
519 #define ECHOE 16
520 #define PENDIN 16384
521 #define ICANON 2
522 #define ECHOKE 2048
523 #define TOSTOP 256
524 #define ECHOK 32
525 #define IEXTEN 32768
526 #define FLUSHO 4096
527 #define ECHOCTL 512
528 #define ECHONL 64

```

1.3.18. ucontext.h

```

529
530 #define NGREG 27
531
532 typedef union
533 {
534     double d;
535     float f;
536 }
537 fpreg_t;
538
539 typedef struct
540 {
541     unsigned int fpc;
542     fpreg_t fprs[16];

```

```

543 }
544 fpregset_t;
545
546 typedef struct
547 {
548     _psw_t psw;
549     unsigned long gregs[16];
550     unsigned int aregs[16];
551     fpregset_t fpregs;
552 }
553 mcontext_t;
554
555 typedef struct ucontext
556 {
557     unsigned long uc_flags;
558     struct ucontext *uc_link;
559     stack_t uc_stack;
560     mcontext_t uc_mcontext;
561     sigset_t uc_sigmask;
562 }
563 ucontext_t;

```

1.3.19. utmp.h

```

564
565 struct lastlog
566 {
567     int32_ttime_t ll_time;
568     char ll_line[UT_LINESIZE];
569     char ll_host[UT_HOSTSIZE];
570 }
571 ;
572
573 struct utmp
574 {
575     short ut_type;
576     pid_t ut_pid;
577     char ut_line[UT_LINESIZE];
578     char ut_id[4];
579     char ut_user[UT_NAMESIZE];
580     char ut_host[UT_HOSTSIZE];
581     struct exit_status ut_exit;
582     long ut_session;
583     struct timeval ut_tv;
584     int32_t ut_addr_v6[4];
585     char __unused[20];
586 }
587 ;

```

1.3.20. utmpx.h

```

588

```

```

589 struct utmpx
590 {
591     short ut_type;
592     pid_t ut_pid;
593     char ut_line[UT_LINESIZE];
594     char ut_id[4];
595     char ut_user[UT_NAMESIZE];
596     char ut_host[UT_HOSTSIZE];
597     struct exit_status ut_exit;
598     long ut_session;
599     struct timeval ut_tv;
600     int32_t ut_addr_v6[4];
601     char __unused[20];
602 }
603 ;

```

1.4. Interfaces for libm

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

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

Library:	libm
SONAME:	libm.so.6

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

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

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

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

1.4.1. Math

609 1.4.1.1. Interfaces for Math

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

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

<code>acos(GLIBC_2.2)</code> [1]	<code>exp(GLIBC_2.2)</code> [1]	<code>expf(GLIBC_2.2)</code> [1]	<code>jnf(GLIBC_2.2)</code> [2]	<code>remquof(GLIBC_2.2)</code> [1]
<code>acosf(GLIBC_2.2)</code> [1]	<code>expf(GLIBC_2.2)</code> [1]	<code>expl(GLIBC_2.2)</code> [1]	<code>jnl(GLIBC_2.2)</code> [2]	<code>remquol(GLIBC_2.2)</code> [1]
<code>acosh(GLIBC_2.2)</code>	<code>expl(GLIBC_2.2)</code>	<code>expm1(GLIBC_2.2)</code>	<code>ldexp(GLIBC_2.2)</code>	<code>rint(GLIBC_2.2)</code> [1]

[1]	[1]	[1]	[1]	
acoshf(GLIBC_2.2) acoshf(GLIBC_2.2) [1]	eimag(GLIBC_2.2) cimag(GLIBC_2.2) [1]	fabs(GLIBC_2.2)fa bs(GLIBC_2.2) [1]	ldexpf(GLIBC_2.2)l dexpf(GLIBC_2.2) [1]	rintf(GLIBC_2.2)rin tf(GLIBC_2.2) [1]
acoshl(GLIBC_2.2) acoshl(GLIBC_2.2) [1]	eimagf(GLIBC_2.2) cimagf(GLIBC_2.2) [1]	fabsf(GLIBC_2.2)fa bsf(GLIBC_2.2) [1]	ldexpl(GLIBC_2.2)l dexpl(GLIBC_2.2) [1]	rintl(GLIBC_2.2)rin tl(GLIBC_2.2) [1]
acosl(GLIBC_2.2)a cosl(GLIBC_2.2) [1]	eimagl(GLIBC_2.2) cimagl(GLIBC_2.2) [1]	fbsl(GLIBC_2.2)fa bsl(GLIBC_2.2) [1]	lgamma(GLIBC_2. 2)lgamma(GLIBC_ 2.2) [1]	round(GLIBC_2.2)r ound(GLIBC_2.2) [1]
asin(GLIBC_2.2)asi n(GLIBC_2.2) [1]	elog(GLIBC_2.2)cl og(GLIBC_2.2) [1]	fdim(GLIBC_2.2)fd im(GLIBC_2.2) [1]	lgamma_r(GLIBC_ 2.2)lgamma_r(GLI BC_2.2) [2]	roundf(GLIBC_2.2) roundf(GLIBC_2.2) [1]
asinf(GLIBC_2.2)as inf(GLIBC_2.2) [1]	elog10(GLIBC_2.2) clog10(GLIBC_2.2) [2]	fdimf(GLIBC_2.2)f dimf(GLIBC_2.2) [1]	lgammaf(GLIBC_2. 2)lgammaf(GLIBC_ 2.2) [1]	roundl(GLIBC_2.2) roundl(GLIBC_2.2) [1]
asinh(GLIBC_2.2)a sinh(GLIBC_2.2) [1]	elog10f(GLIBC_2.2) clog10f(GLIBC_2. 2) [2]	fdiml(GLIBC_2.2)f diml(GLIBC_2.2) [1]	lgammaf_r(GLIBC_ 2.2)lgammaf_r(GLI BC_2.2) [2]	scalb(GLIBC_2.2)s calb(GLIBC_2.2) [1]
asinhf(GLIBC_2.2) asinhf(GLIBC_2.2) [1]	elog10l(GLIBC_2.2) clog10l(GLIBC_2. 2) [2]	feclearexcept(GLIB C_2.2)feclearexcept (GLIBC_2.2) [1]	lgammal(GLIBC_2. 2)lgammal(GLIBC_ 2.2) [1]	scalbf(GLIBC_2.2)s calbf(GLIBC_2.2) [2]
asinhl(GLIBC_2.2)a sinhl(GLIBC_2.2) [1]	elogf(GLIBC_2.2)cl ogf(GLIBC_2.2) [1]	fegetenv(GLIBC_2. 2)fegetenv(GLIBC_ 2.2) [1]	lgammal_r(GLIBC_ 2.2)lgammal_r(GLI BC_2.2) [2]	scalbl(GLIBC_2.2)s calbl(GLIBC_2.2) [2]
asinl(GLIBC_2.2)as inl(GLIBC_2.2) [1]	elogl(GLIBC_2.2)cl ogf(GLIBC_2.2) [1]	fegetexceptflag(GLI BC_2.2)fegetexcept flag(GLIBC_2.2) [1]	llrint(GLIBC_2.2)llr int(GLIBC_2.2) [1]	scalbln(GLIBC_2.2) scalbln(GLIBC_2.2) [1]
atan(GLIBC_2.2)ata n(GLIBC_2.2) [1]	eonj(GLIBC_2.2)co nj(GLIBC_2.2) [1]	fegetround(GLIBC_ 2.2)fegetround(GLI BC_2.2) [1]	llrintf(GLIBC_2.2)ll rintf(GLIBC_2.2) [1]	scalblnf(GLIBC_2.2) scalblnf(GLIBC_2. 2) [1]
atan2(GLIBC_2.2)a tan2(GLIBC_2.2) [1]	eonjf(GLIBC_2.2)c onjf(GLIBC_2.2) [1]	feholdexcept(GLIB C_2.2)feholdexcept(GLIBC_2.2) [1]	llrintl(GLIBC_2.2)ll rintl(GLIBC_2.2) [1]	scalblnl(GLIBC_2.2) scalblnl(GLIBC_2. 2) [1]
atan2f(GLIBC_2.2) atan2f(GLIBC_2.2) [1]	eonjl(GLIBC_2.2)c onjl(GLIBC_2.2) [1]	feraiseexcept(GLIB C_2.2)feraiseexcept (GLIBC_2.2) [1]	llround(GLIBC_2.2) llround(GLIBC_2. 2) [1]	scalbn(GLIBC_2.2) scalbn(GLIBC_2.2) [1]
atan2l(GLIBC_2.2) atan2l(GLIBC_2.2)	eopysign(GLIBC_2. 2)copysign(GLIBC	fesetenv(GLIBC_2. 2)fesetenv(GLIBC_	llroundf(GLIBC_2. 2)llroundf(GLIBC_	scalbnf(GLIBC_2.2) scalbnf(GLIBC_2.

[1]	_2.2) [1]	2.2) [1]	2.2) [1]	2) [1]
atanf(GLIBC_2.2)atanf(GLIBC_2.2) [1]	eopysignf(GLIBC_2.2)copysignf(GLIBC_2.2) [1]	fesetexceptflag(GLIBC_2.2)fesetexceptflag(GLIBC_2.2) [1]	llroundl(GLIBC_2.2)llroundl(GLIBC_2.2) [1]	scalbnl(GLIBC_2.2)scalbnl(GLIBC_2.2) [1]
atanh(GLIBC_2.2)atanh(GLIBC_2.2) [1]	eopysignl(GLIBC_2.2)copysignl(GLIBC_2.2) [1]	fesetround(GLIBC_2.2)fesetround(GLIBC_2.2) [1]	log(GLIBC_2.2)log(GLIBC_2.2) [1]	significand(GLIBC_2.2)significand(GLIBC_2.2) [2]
atanhf(GLIBC_2.2)atanhf(GLIBC_2.2) [1]	eosc(GLIBC_2.2)cos(GLIBC_2.2) [1]	fetestexcept(GLIBC_2.2)fetestexcept(GLIBC_2.2) [1]	log10(GLIBC_2.2)log10(GLIBC_2.2) [1]	significandf(GLIBC_2.2)significandf(GLIBC_2.2) [2]
atanhl(GLIBC_2.2)atanhl(GLIBC_2.2) [1]	eosf(GLIBC_2.2)cosf(GLIBC_2.2) [1]	feupdateenv(GLIBC_2.2)feupdateenv(GLIBC_2.2) [1]	log10f(GLIBC_2.2)log10f(GLIBC_2.2) [1]	significandl(GLIBC_2.2)significandl(GLIBC_2.2) [2]
atanl(GLIBC_2.2)atanl(GLIBC_2.2) [1]	eosh(GLIBC_2.2)cosh(GLIBC_2.2) [1]	finite(GLIBC_2.2)finite(GLIBC_2.2) [3]	log10l(GLIBC_2.2)log10l(GLIBC_2.2) [1]	sin(GLIBC_2.2)sin(GLIBC_2.2) [1]
eabs(GLIBC_2.2)fabs(GLIBC_2.2) [1]	eoshf(GLIBC_2.2)coshf(GLIBC_2.2) [1]	finitef(GLIBC_2.2)finitef(GLIBC_2.2) [2]	log1p(GLIBC_2.2)log1p(GLIBC_2.2) [1]	sineos(GLIBC_2.2)sincos(GLIBC_2.2) [2]
eabsf(GLIBC_2.2)fabsf(GLIBC_2.2) [1]	eoshl(GLIBC_2.2)coshl(GLIBC_2.2) [1]	finitel(GLIBC_2.2)finitel(GLIBC_2.2) [2]	logb(GLIBC_2.2)logb(GLIBC_2.2) [1]	sineosf(GLIBC_2.2)sincosf(GLIBC_2.2) [2]
eabsl(GLIBC_2.2)fabsl(GLIBC_2.2) [1]	eosl(GLIBC_2.2)cosl(GLIBC_2.2) [1]	floor(GLIBC_2.2)floor(GLIBC_2.2) [1]	logf(GLIBC_2.2)logf(GLIBC_2.2) [1]	sineosl(GLIBC_2.2)sincosl(GLIBC_2.2) [2]
eacos(GLIBC_2.2)acos(GLIBC_2.2) [1]	epow(GLIBC_2.2)cpow(GLIBC_2.2) [1]	floorf(GLIBC_2.2)floorf(GLIBC_2.2) [1]	logl(GLIBC_2.2)logl(GLIBC_2.2) [1]	sinf(GLIBC_2.2)sinf(GLIBC_2.2) [1]
eacosf(GLIBC_2.2)acosf(GLIBC_2.2) [1]	epowf(GLIBC_2.2)cpowf(GLIBC_2.2) [1]	floorl(GLIBC_2.2)floorl(GLIBC_2.2) [1]	lrint(GLIBC_2.2)lrint(GLIBC_2.2) [1]	sinh(GLIBC_2.2)sinh(GLIBC_2.2) [1]
eacosh(GLIBC_2.2)acosh(GLIBC_2.2) [1]	epowl(GLIBC_2.2)cpowl(GLIBC_2.2) [1]	fma(GLIBC_2.2)fma(GLIBC_2.2) [1]	lrintf(GLIBC_2.2)lrintf(GLIBC_2.2) [1]	sinhf(GLIBC_2.2)sinhf(GLIBC_2.2) [1]
eacoshf(GLIBC_2.2)acoshf(GLIBC_2.2) [1]	eprojl(GLIBC_2.2)cproj(GLIBC_2.2) [1]	fmaf(GLIBC_2.2)fmaf(GLIBC_2.2) [1]	lrintl(GLIBC_2.2)lrintl(GLIBC_2.2) [1]	sinhl(GLIBC_2.2)sinhl(GLIBC_2.2) [1]
eacoshl(GLIBC_2.2)acoshl(GLIBC_2.2) [1]	eprojfl(GLIBC_2.2)cprojfl(GLIBC_2.2) [1]	fmal(GLIBC_2.2)fmal(GLIBC_2.2) [1]	lround(GLIBC_2.2)lround(GLIBC_2.2) [1]	sinl(GLIBC_2.2)sinl(GLIBC_2.2) [1]

<code>eaacosl(GLIBC_2.2)</code> <code>caacosl(GLIBC_2.2)</code> [1]	<code>eprojl(GLIBC_2.2)</code> <code>cprojl(GLIBC_2.2)</code> [1]	<code>fmax(GLIBC_2.2)</code> <code>fmax(GLIBC_2.2)</code> [1]	<code>lroundf(GLIBC_2.2)</code> <code>lroundf(GLIBC_2.2)</code> [1]	<code>sqrt(GLIBC_2.2)</code> <code>sqrt(GLIBC_2.2)</code> [1]
<code>earg(GLIBC_2.2)</code> <code>carg(GLIBC_2.2)</code> [1]	<code>erealf(GLIBC_2.2)</code> <code>crealf(GLIBC_2.2)</code> [1]	<code>fmaxf(GLIBC_2.2)</code> <code>fmaxf(GLIBC_2.2)</code> [1]	<code>lroundl(GLIBC_2.2)</code> <code>lroundl(GLIBC_2.2)</code> [1]	<code>sqrtf(GLIBC_2.2)</code> <code>sqrtf(GLIBC_2.2)</code> [1]
<code>eargf(GLIBC_2.2)</code> <code>cargf(GLIBC_2.2)</code> [1]	<code>erealf(GLIBC_2.2)</code> <code>crealf(GLIBC_2.2)</code> [1]	<code>fmaxl(GLIBC_2.2)</code> <code>fmaxl(GLIBC_2.2)</code> [1]	<code>matherr(GLIBC_2.2)</code> <code>matherr(GLIBC_2.2)</code> [2]	<code>sqrtl(GLIBC_2.2)</code> <code>sqrtl(GLIBC_2.2)</code> [1]
<code>eargl(GLIBC_2.2)</code> <code>cargl(GLIBC_2.2)</code> [1]	<code>ereall(GLIBC_2.2)</code> <code>creall(GLIBC_2.2)</code> [1]	<code>fmin(GLIBC_2.2)</code> <code>fmin(GLIBC_2.2)</code> [1]	<code>modf(GLIBC_2.2)</code> <code>modf(GLIBC_2.2)</code> [1]	<code>tan(GLIBC_2.2)</code> <code>tan(GLIBC_2.2)</code> [1]
<code>easin(GLIBC_2.2)</code> <code>csin(GLIBC_2.2)</code> [1]	<code>esin(GLIBC_2.2)</code> <code>csin(GLIBC_2.2)</code> [1]	<code>fminf(GLIBC_2.2)</code> <code>fminf(GLIBC_2.2)</code> [1]	<code>modff(GLIBC_2.2)</code> <code>modff(GLIBC_2.2)</code> [1]	<code>tanf(GLIBC_2.2)</code> <code>tanf(GLIBC_2.2)</code> [1]
<code>easinf(GLIBC_2.2)</code> <code>csinf(GLIBC_2.2)</code> [1]	<code>esinf(GLIBC_2.2)</code> <code>csinf(GLIBC_2.2)</code> [1]	<code>fminl(GLIBC_2.2)</code> <code>fminl(GLIBC_2.2)</code> [1]	<code>modfl(GLIBC_2.2)</code> <code>modfl(GLIBC_2.2)</code> [1]	<code>tanh(GLIBC_2.2)</code> <code>tanh(GLIBC_2.2)</code> [1]
<code>easinh(GLIBC_2.2)</code> <code>casinh(GLIBC_2.2)</code> [1]	<code>esinh(GLIBC_2.2)</code> <code>csinh(GLIBC_2.2)</code> [1]	<code>fmod(GLIBC_2.2)</code> <code>fmod(GLIBC_2.2)</code> [1]	<code>nan(GLIBC_2.2)</code> <code>nan(GLIBC_2.2)</code> [1]	<code>tanhf(GLIBC_2.2)</code> <code>tanhf(GLIBC_2.2)</code> [1]
<code>easinhf(GLIBC_2.2)</code> <code>casinhf(GLIBC_2.2)</code> [1]	<code>esinhf(GLIBC_2.2)</code> <code>csinhf(GLIBC_2.2)</code> [1]	<code>fmodf(GLIBC_2.2)</code> <code>fmodf(GLIBC_2.2)</code> [1]	<code>nanf(GLIBC_2.2)</code> <code>nanf(GLIBC_2.2)</code> [1]	<code>tanhf(GLIBC_2.2)</code> <code>tanhf(GLIBC_2.2)</code> [1]
<code>easinhf(GLIBC_2.2)</code> <code>casinhf(GLIBC_2.2)</code> [1]	<code>esinhf(GLIBC_2.2)</code> <code>csinhf(GLIBC_2.2)</code> [1]	<code>fmodl(GLIBC_2.2)</code> <code>fmodl(GLIBC_2.2)</code> [1]	<code>nanl(GLIBC_2.2)</code> <code>nanl(GLIBC_2.2)</code> [1]	<code>tanl(GLIBC_2.2)</code> <code>tanl(GLIBC_2.2)</code> [1]
<code>easinl(GLIBC_2.2)</code> <code>csinl(GLIBC_2.2)</code> [1]	<code>esinl(GLIBC_2.2)</code> <code>csinl(GLIBC_2.2)</code> [1]	<code>frexp(GLIBC_2.2)</code> <code>frexp(GLIBC_2.2)</code> [1]	<code>nearbyint(GLIBC_2.2)</code> <code>nearbyint(GLIBC_2.2)</code> [1]	<code>tgamma(GLIBC_2.2)</code> <code>tgamma(GLIBC_2.2)</code> [1]
<code>eatanf(GLIBC_2.2)</code> <code>catanf(GLIBC_2.2)</code> [1]	<code>esqrtf(GLIBC_2.2)</code> <code>csqrtf(GLIBC_2.2)</code> [1]	<code>frexpf(GLIBC_2.2)</code> <code>frexpf(GLIBC_2.2)</code> [1]	<code>nearbyintf(GLIBC_2.2)</code> <code>nearbyintf(GLIBC_2.2)</code> [1]	<code>tgammaf(GLIBC_2.2)</code> <code>tgammaf(GLIBC_2.2)</code> [1]
<code>eatanf(GLIBC_2.2)</code> <code>catanf(GLIBC_2.2)</code> [1]	<code>esqrtf(GLIBC_2.2)</code> <code>csqrtf(GLIBC_2.2)</code> [1]	<code>frexpl(GLIBC_2.2)</code> <code>frexpl(GLIBC_2.2)</code> [1]	<code>nearbyintl(GLIBC_2.2)</code> <code>nearbyintl(GLIBC_2.2)</code> [1]	<code>tgammal(GLIBC_2.2)</code> <code>tgammal(GLIBC_2.2)</code> [1]
<code>eatanh(GLIBC_2.2)</code> <code>catanh(GLIBC_2.2)</code> [1]	<code>esqrtl(GLIBC_2.2)</code> <code>csqrtl(GLIBC_2.2)</code> [1]	<code>gamma(GLIBC_2.2)</code> <code>gamma(GLIBC_2.2)</code> [3]	<code>nextafter(GLIBC_2.2)</code> <code>nextafter(GLIBC_2.2)</code> [1]	<code>trunc(GLIBC_2.2)</code> <code>trunc(GLIBC_2.2)</code> [1]
<code>eatanhf(GLIBC_2.2)</code>	<code>etan(GLIBC_2.2)</code> <code>ctan</code>	<code>gammaf(GLIBC_2.2)</code>	<code>nextafterf(GLIBC_2.2)</code>	<code>truncf(GLIBC_2.2)</code>

<code>γcatanhf(GLIBC_2.2)</code> [1]	<code>n(GLIBC_2.2)</code> [1]	<code>γgammalf(GLIBC_2.2)</code> [2]	<code>γnextafterf(GLIBC_2.2)</code> [1]	<code>runcf(GLIBC_2.2)</code> [1]
<code>γatanhl(GLIBC_2.2)</code> <code>γcatanh(GLIBC_2.2)</code> [1]	<code>γtanf(GLIBC_2.2)</code> <code>ctanf(GLIBC_2.2)</code> [1]	<code>γgammaf(GLIBC_2.2)</code> <code>γgammal(GLIBC_2.2)</code> [2]	<code>nextafterl(GLIBC_2.2)</code> <code>γnextafterl(GLIBC_2.2)</code> [1]	<code>γtruncl(GLIBC_2.2)</code> <code>ttruncl(GLIBC_2.2)</code> [1]
<code>γatanf(GLIBC_2.2)</code> <code>γatanl(GLIBC_2.2)</code> [1]	<code>γtanhf(GLIBC_2.2)</code> <code>ctanhf(GLIBC_2.2)</code> [1]	<code>hypotf(GLIBC_2.2)</code> <code>hypotl(GLIBC_2.2)</code> [1]	<code>nexttowardf(GLIBC_2.2)</code> <code>nexttowardl(GLIBC_2.2)</code> [1]	<code>γy0f(GLIBC_2.2)</code> <code>y0f(GLIBC_2.2)</code> [1]
<code>γbrtf(GLIBC_2.2)</code> <code>γbrtl(GLIBC_2.2)</code> [1]	<code>γctanhf(GLIBC_2.2)</code> <code>ctanhf(GLIBC_2.2)</code> [1]	<code>hypotff(GLIBC_2.2)</code> <code>hypotfl(GLIBC_2.2)</code> [1]	<code>nexttowardff(GLIBC_2.2)</code> <code>nexttowardfl(GLIBC_2.2)</code> [1]	<code>γy0ff(GLIBC_2.2)</code> <code>y0ff(GLIBC_2.2)</code> [2]
<code>γbrtff(GLIBC_2.2)</code> <code>γbrtfl(GLIBC_2.2)</code> [1]	<code>γctanhf(GLIBC_2.2)</code> <code>ctanhf(GLIBC_2.2)</code> [1]	<code>hypotfl(GLIBC_2.2)</code> <code>hypotfl(GLIBC_2.2)</code> [1]	<code>nexttowardfl(GLIBC_2.2)</code> <code>nexttowardfl(GLIBC_2.2)</code> [1]	<code>γy0fl(GLIBC_2.2)</code> <code>y0fl(GLIBC_2.2)</code> [2]
<code>γbrtl(GLIBC_2.2)</code> <code>γbrtl(GLIBC_2.2)</code> [1]	<code>γctanf(GLIBC_2.2)</code> <code>ctanf(GLIBC_2.2)</code> [1]	<code>ilogbf(GLIBC_2.2)</code> <code>ilogbl(GLIBC_2.2)</code> [1]	<code>powf(GLIBC_2.2)</code> <code>powl(GLIBC_2.2)</code> [1]	<code>γy1f(GLIBC_2.2)</code> <code>y1f(GLIBC_2.2)</code> [1]
<code>γeosf(GLIBC_2.2)</code> <code>γeosl(GLIBC_2.2)</code> [1]	<code>γdremff(GLIBC_2.2)</code> <code>dremff(GLIBC_2.2)</code> [2]	<code>ilogbff(GLIBC_2.2)</code> <code>ilogbfl(GLIBC_2.2)</code> [1]	<code>pow10f(GLIBC_2.2)</code> <code>pow10l(GLIBC_2.2)</code> [2]	<code>γy1ff(GLIBC_2.2)</code> <code>y1ff(GLIBC_2.2)</code> [2]
<code>γeosff(GLIBC_2.2)</code> <code>γeosfl(GLIBC_2.2)</code> [1]	<code>γdremfl(GLIBC_2.2)</code> <code>dremfl(GLIBC_2.2)</code> [2]	<code>ilogblf(GLIBC_2.2)</code> <code>ilogblf(GLIBC_2.2)</code> [1]	<code>pow10ff(GLIBC_2.2)</code> <code>pow10fl(GLIBC_2.2)</code> [2]	<code>γy1fl(GLIBC_2.2)</code> <code>y1fl(GLIBC_2.2)</code> [2]
<code>γeoshf(GLIBC_2.2)</code> <code>γcoshf(GLIBC_2.2)</code> [1]	<code>γerff(GLIBC_2.2)</code> <code>erfl(GLIBC_2.2)</code> [1]	<code>γj0f(GLIBC_2.2)</code> <code>j0f(GLIBC_2.2)</code> [1]	<code>γpow10f(GLIBC_2.2)</code> <code>pow10f(GLIBC_2.2)</code> [2]	<code>γynf(GLIBC_2.2)</code> <code>ynf(GLIBC_2.2)</code> [1]
<code>γeoshff(GLIBC_2.2)</code> <code>γcoshff(GLIBC_2.2)</code> [1]	<code>γerff(GLIBC_2.2)</code> <code>erff(GLIBC_2.2)</code> [1]	<code>γj0ff(GLIBC_2.2)</code> <code>j0ff(GLIBC_2.2)</code> [2]	<code>γpowff(GLIBC_2.2)</code> <code>powff(GLIBC_2.2)</code> [1]	<code>γynff(GLIBC_2.2)</code> <code>ynff(GLIBC_2.2)</code> [2]
<code>γeoshl(GLIBC_2.2)</code> <code>γcoshl(GLIBC_2.2)</code> [1]	<code>γerff(GLIBC_2.2)</code> <code>erfl(GLIBC_2.2)</code> [1]	<code>γj0l(GLIBC_2.2)</code> <code>j0l(GLIBC_2.2)</code> [2]	<code>γpowl(GLIBC_2.2)</code> <code>powl(GLIBC_2.2)</code> [1]	<code>γynl(GLIBC_2.2)</code> <code>ynl(GLIBC_2.2)</code> [2]
<code>γeosl(GLIBC_2.2)</code> <code>γcosl(GLIBC_2.2)</code> [1]	<code>γerflf(GLIBC_2.2)</code> <code>erflf(GLIBC_2.2)</code> [1]	<code>γj1f(GLIBC_2.2)</code> <code>j1f(GLIBC_2.2)</code> [1]	<code>remainderf(GLIBC_2.2)</code> <code>remainderf(GLIBC_2.2)</code> [1]	
<code>γeilf(GLIBC_2.2)</code> <code>γeilf(GLIBC_2.2)</code> [1]	<code>γerff(GLIBC_2.2)</code> <code>erff(GLIBC_2.2)</code> [1]	<code>γj1ff(GLIBC_2.2)</code> <code>j1ff(GLIBC_2.2)</code> [2]	<code>remainderff(GLIBC_2.2)</code> <code>remainderff(GLIBC_2.2)</code> [1]	
<code>γeifl(GLIBC_2.2)</code> <code>γeifl(GLIBC_2.2)</code> [1]	<code>γerflf(GLIBC_2.2)</code> <code>erflf(GLIBC_2.2)</code> [1]	<code>γj1l(GLIBC_2.2)</code> <code>j1l(GLIBC_2.2)</code> [2]	<code>remainderfl(GLIBC_2.2)</code> <code>remainderfl(GLIBC_2.2)</code> [1]	

			BC_2.2) [1]	
ceil(GLIBC_2.2)ceil(GLIBC_2.2) [1]	exp(GLIBC_2.2)exp(GLIBC_2.2) [1]	jn(GLIBC_2.2)jn(GLIBC_2.2) [1]	remquo(GLIBC_2.2)remquo(GLIBC_2.2) [1]	

613

614 *Referenced Specification(s)*

615 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)

616

617 [2]. ISO/IEC 9899: C (1999, Programming Language—C)

618 [3]. CAE Specification, January 1997, System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0,

619 C606) SUSv2

620 An LSB conforming implementation shall provide the architecture specific data interfaces for Math specified in Table
621 1-30, with the full functionality as described in the referenced underlying specification.622 **Table 1-30. libm - Math Data Interfaces**

siggam(GLIBC_2.2)siggam(GLIBC_2.2) [1]				
----------------------------------------	--	--	--	--

623

624 *Referenced Specification(s)*

625 [1]. ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)

626

1.5. Interfaces for libpthread

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

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

Library:	libpthread
SONAME:	libpthread.so.0

629

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

Large File Support

Linux Standard Base this specification

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

1.5.1. Realtime Threads

632 **1.5.1.1. Interfaces for Realtime Threads**

633 No external functions are defined for libpthread - Realtime Threads

1.5.2. Advanced Realtime Threads

1.5.2.1. Interfaces for Advanced Realtime Threads

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

1.5.3. Posix Threads

1.5.3.1. Interfaces for Posix Threads

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

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

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

<code>pthread_attr_getguardsize(GLIBC_2.2)</code> [2]	<code>ad_cond_wait(GLIBC_2.3.2)</code> [2]	<code>d_mutex_lock(GLIBC_2.2)</code> [2]	<code>pthread_rwlock_trywrlock(GLIBC_2.2)</code> [2]	BC_2.2) [2]
<code>pthread_attr_getschedparam(GLIBC_2.2)</code> <code>pthread_attr_getschedparam(GLIBC_2.2)</code> [2]	<code>pthread_condattr_destroy(GLIBC_2.2)</code> <code>pthread_condattr_destroy(GLIBC_2.2)</code> [2]	<code>pthread_mutex_trylock(GLIBC_2.2)</code> <code>pthread_mutex_trylock(GLIBC_2.2)</code> [2]	<code>pthread_rwlock_unlink(GLIBC_2.2)</code> <code>pthread_rwlock_unlock(GLIBC_2.2)</code> [2]	<code>sem_destroy(GLIBC_2.2)</code> <code>sem_destroy(GLIBC_2.2)</code> [2]
<code>pthread_attr_getstackaddr(GLIBC_2.2)</code> <code>pthread_attr_getstackaddr(GLIBC_2.2)</code> [2]	<code>pthread_condattr_getpshared(GLIBC_2.2)</code> <code>pthread_condattr_getpshared(GLIBC_2.2)</code> [2]	<code>pthread_mutex_unlink(GLIBC_2.2)</code> <code>pthread_mutex_unlock(GLIBC_2.2)</code> [2]	<code>pthread_rwlock_writelock(GLIBC_2.2)</code> <code>pthread_rwlock_writelock(GLIBC_2.2)</code> [2]	<code>sem_getvalue(GLIBC_2.2)</code> <code>sem_getvalue(GLIBC_2.2)</code> [2]
<code>pthread_attr_getstacksize(GLIBC_2.2)</code> <code>pthread_attr_getstacksize(GLIBC_2.2)</code> [2]	<code>pthread_condattr_init(GLIBC_2.2)</code> <code>pthread_condattr_init(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_destroy(GLIBC_2.2)</code> <code>pthread_mutexattr_destroy(GLIBC_2.2)</code> [2]	<code>pthread_rwlockattr_destroy(GLIBC_2.2)</code> <code>pthread_rwlockattr_destroy(GLIBC_2.2)</code> [2]	<code>sem_init(GLIBC_2.2)</code> <code>sem_init(GLIBC_2.2)</code> [2]
<code>pthread_attr_init(GLIBC_2.2)</code> <code>pthread_attr_init(GLIBC_2.2)</code> [2]	<code>pthread_condattr_setpshared(GLIBC_2.2)</code> <code>pthread_condattr_setpshared(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_getpshared(GLIBC_2.2)</code> <code>pthread_mutexattr_getpshared(GLIBC_2.2)</code> [2]	<code>pthread_rwlockattr_getpshared(GLIBC_2.2)</code> <code>pthread_rwlockattr_getpshared(GLIBC_2.2)</code> [2]	<code>sem_open(GLIBC_2.2)</code> <code>sem_open(GLIBC_2.2)</code> [2]
<code>pthread_attr_setdetachstate(GLIBC_2.2)</code> <code>pthread_attr_setdetachstate(GLIBC_2.2)</code> [2]	<code>pthread_create(GLIBC_2.2)</code> <code>pthread_create(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_gettype(GLIBC_2.2)</code> <code>pthread_mutexattr_gettype(GLIBC_2.2)</code> [2]	<code>pthread_rwlockattr_init(GLIBC_2.2)</code> <code>pthread_rwlockattr_init(GLIBC_2.2)</code> [2]	<code>sem_post(GLIBC_2.2)</code> <code>sem_post(GLIBC_2.2)</code> [2]
<code>pthread_attr_setguardsize(GLIBC_2.2)</code> <code>pthread_attr_setguardsize(GLIBC_2.2)</code> [2]	<code>pthread_detach(GLIBC_2.2)</code> <code>pthread_detach(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_init(GLIBC_2.2)</code> <code>pthread_mutexattr_init(GLIBC_2.2)</code> [2]	<code>pthread_rwlockattr_setpshared(GLIBC_2.2)</code> <code>pthread_rwlockattr_setpshared(GLIBC_2.2)</code> [2]	<code>sem_timedwait(GLIBC_2.2)</code> <code>sem_timedwait(GLIBC_2.2)</code> [2]
<code>pthread_attr_setschedparam(GLIBC_2.2)</code> <code>pthread_attr_setschedparam(GLIBC_2.2)</code> [2]	<code>pthread_equal(GLIBC_2.2)</code> <code>pthread_equal(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_setpshared(GLIBC_2.2)</code> <code>pthread_mutexattr_setpshared(GLIBC_2.2)</code> [2]	<code>pthread_self(GLIBC_2.2)</code> <code>pthread_self(GLIBC_2.2)</code> [2]	<code>sem_trywait(GLIBC_2.2)</code> <code>sem_trywait(GLIBC_2.2)</code> [2]
<code>pthread_attr_setstackaddr(GLIBC_2.2)</code> <code>pthread_attr_setstackaddr(GLIBC_2.2)</code> [2]	<code>pthread_exit(GLIBC_2.2)</code> <code>pthread_exit(GLIBC_2.2)</code> [2]	<code>pthread_mutexattr_settype(GLIBC_2.2)</code> <code>pthread_mutexattr_settype(GLIBC_2.2)</code> [2]	<code>pthread_setcancelstate(GLIBC_2.2)</code> <code>pthread_setcancelstate(GLIBC_2.2)</code> [2]	<code>sem_unlink(GLIBC_2.2)</code> <code>sem_unlink(GLIBC_2.2)</code> [2]

pthread_attr_setstacksize(GLIBC_2.2)pthread_attr_setstacksize(GLIBC_2.2) [2]	pthread_getspecific(GLIBC_2.2)pthread_getspecific(GLIBC_2.2) [2]	pthread_once(GLIBC_2.2)pthread_once(GLIBC_2.2) [2]	pthread_setcanceltype(GLIBC_2.2)pthread_setcanceltype(GLIBC_2.2) [2]	sem_wait(GLIBC_2.2)sem_wait(GLIBC_2.2) [2]
------------------------------------------------------------------------------	------------------------------------------------------------------	----------------------------------------------------	----------------------------------------------------------------------	--------------------------------------------

641 *Referenced Specification(s)*

642 [1]. ~~Linux Standard Base~~this specification

643 [2]. ~~ISO/IEC 9945: POSIX (2003 Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3)~~

645 [3]. Large File Support

1.6. Interfaces for libgcc_s

646 Table 1-33 defines the library name and shared object name for the libgcc_s library

647 **Table 1-33. libgcc_s Definition**

Library:	libgcc_s
SONAME:	libgcc_s.so.1

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

650 ~~Linux Standard Base~~this specification

1.6.1. Unwind Library

1.6.1.1. Interfaces for Unwind Library

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

654 **Table 1-34. libgcc_s - Unwind Library Function Interfaces**

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

656 *Referenced Specification(s)*
 657 ~~[1]. Linux Standard Base~~this specification

1.7. Interface Definitions for libgcc_s

658 The following interfaces are included in libgcc_s and are defined by this specification. Unless otherwise noted, these
 659 interfaces shall be included in the source standard.

660 Other interfaces listed above for libgcc_s shall behave as described in the referenced base document.

`_Unwind_DeleteException`

Name

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

Synopsis

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

Description

663 `_Unwind_DeleteException` deletes the given exception *object*. If a given runtime resumes normal execution
 664 after catching a foreign exception, it will not know how to delete that exception. Such an exception shall be deleted by
 665 calling `_Unwind_DeleteException`. This is a convenience function that calls the function pointed to by the
 666 *exception_cleanup* field of the exception header.

`_Unwind_Find_FDE`

Name

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

Synopsis

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

Description

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

`_Unwind_ForcedUnwind`

Name

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

Synopsis

```
671 _Unwind_Reason_Code _Unwind_ForcedUnwind((struct _Unwind_Exception *object),  
672 _Unwind_Stop_Fn stop, void *stop_parameter);
```

Description

673 `_Unwind_ForcedUnwind` raises an exception for forced unwinding, passing along the given exception *object*,
674 which should have its *exception_class* and *exception_cleanup* fields set. The exception *object* has been allocated by
675 the language-specific runtime, and has a language-specific format, except that it shall contain an `_Unwind_Exception`
676 struct.

677 Forced unwinding is a single-phase process. *stop* and *stop_parameter* control the termination of the unwind
678 process instead of the usual personality routine query. *stop* is called for each unwind frame, with the parameters
679 described for the usual personality routine below, plus an additional *stop_parameter*.

Return Value

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

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

685 `_URC_NO_REASON`

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

689 `_URC_END_OF_STACK`

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

693 `_URC_FATAL_PHASE2_ERROR`

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

`_Unwind_GetDataRelBase`

Name

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

Synopsis

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

Description

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

`_Unwind_GetGR`

Name

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

Synopsis

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

Description

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

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

`_Unwind_GetIP`

Name

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

Synopsis

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

Description

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

`_Unwind_GetLanguageSpecificData`

Name

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

Synopsis

```
708 _Unwind_Ptr _Unwind_GetLanguageSpecificData((struct _Unwind_Context *context), uint  
709 value);
```

Description

710 `_Unwind_GetLanguageSpecificData` returns the address of the language specific data area for the current stack
711 frame.

`_Unwind_GetRegionStart`

Name

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

Synopsis

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

Description

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

`_Unwind_GetTextRelBase`

Name

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

Synopsis

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

Description

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

`_Unwind_RaiseException`

Name

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

Synopsis

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

Description

721 `_Unwind_RaiseException` raises an exception, passing along the given exception *object*, which should have its
 722 *exception_class* and *exception_cleanup* fields set. The exception object has been allocated by the
 723 language-specific runtime, and has a language-specific format, exception that it shall contain an
 724 `_Unwind_Exception`.

Return Value

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

727 `_URC_END_OF_STACK`

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

730 `_URC_FATAL_PHASE1_ERROR`

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

734 `_URC_FATAL_PHASE2_ERROR`

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

`_Unwind_Resume`

Name

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

Synopsis

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

Description

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

`_Unwind_SetGR`

Name

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

Synopsis

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

Description

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

`_Unwind_SetIP`

Name

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

Synopsis

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

Description

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

1.8. Interfaces for libdl

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

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

Library:	libdl
SONAME:	libdl.so.2

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

751 ~~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

752 1.8.1.1. Interfaces for Dynamic Loader

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

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

dldaddr(GLIBC_2.2) dldaddr(GLIBC_2.2) [1]	dldclose(GLIBC_2.2) dldclose(GLIBC_2.2) [2]	dlderror(GLIBC_2.2) dlderror(GLIBC_2.2) [2]	dldopen(GLIBC_2.2) dldopen(GLIBC_2.2) [1]	dldsym(GLIBC_2.2) dldsym(GLIBC_2.2) [1]
------------------------------------------------------------	--------------------------------------------------------------	--------------------------------------------------------------	------------------------------------------------------------	----------------------------------------------------------

757 *Referenced Specification(s)*

758 [1]. ~~Linux Standard Base~~this specification

759 [2]. ISO/IEC 9945: POSIX (2003 ~~Portable Operating System (POSIX) and The Single UNIX® Specification (SUS) V3~~)
760

1.9. Interfaces for libcrypt

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

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

Library:	libcrypt
SONAME:	libcrypt.so.1

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

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

1.9.1. Encryption

766 1.9.1.1. Interfaces for Encryption

767 An LSB conforming implementation shall provide the architecture specific functions for Encryption specified in Table
768 1-38, with the full functionality as described in the referenced underlying specification.

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

crypt(GLIBC_2.2)cr ypt(GLIBC_2.2) [1]	encrypt(GLIBC_2.2)encrypt(GLIBC_2. 2) [1]	setkey(GLIBC_2.2) setkey(GLIBC_2.2) [1]		
------------------------------------------	--------------------------------------------------	-----------------------------------------------	--	--

771 *Referenced Specification(s)*

772 **[1]. ISO/IEC 9945: POSIX (2003-Portable Operating System(POSIX)and The Single UNIX® Specification(SUS)**
 773 **↯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 Base¹

2.5.1. Utility Functions

2.5.1.1. Interfaces for Utility Functions

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

forkpty(GLIBC_2.2) ¹	login_tty(GLIBC_2.2) ¹	logwtmp(GLIBC_2.2) ¹		
login(GLIBC_2.2) ¹	logout(GLIBC_2.2) ¹	openpty(GLIBC_2.2) ¹		

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**

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

4

Linux Packaging Specification

2

3 **Linux Packaging Specification**

Table of Contents

I. Package Format and Installation	55
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-s390x`
 - 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-s390x`.

1.2. Package Architecture Considerations

- 6 All packages must specify an architecture of `s390x`. A LSB runtime environment must accept an architecture of `s390`
- 7 even if the native architecture is different.
- 8 The `archnum` value in the Lead Section shall be `0x000E`.

Free Documentation License

2 |
3 | **Free Documentation License**

Table of Contents

A. GNU Free Documentation License	1
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.