BS ISO/IEC 23360-2-2:2021

BSI Standards Publication

Linux Standard Base (LSB)

Part 2-2: Core specification for X86-32 architecture

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23360-2-2:2021

National foreword

This British Standard is the UK implementation of ISO/IEC 23360-2-2:2021.

The UK participation in its preparation was entrusted to Technical Committee IST/5, Programming languages, their environments and system software interfaces.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2021
Published by BSI Standards Limited 2021

ISBN 978 0 539 05254 1

ICS 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2021.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23360-2-2:2021

INTERNATIONAL STANDARD

ISO/IEC 23360-2-2

First edition
2021-10

Linux Standard Base (LSB) —

Part 2-2:
Core specification for X86-32 architecture

Reference number
ISO/IEC 23360-2-2:2021(E)

Documents
Document Sharing Hub

© ISO/IEC 2021

Full document details: BS ISO/IEC 23360-2-2:2021

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Linux Foundation as Linux Standard Base (LSB): Core specification for X86-32 architecture and drafted in accordance with its editorial rules. It was assigned to Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*, and adopted by National Bodies.

This first edition of ISO/IEC 23360-2-2 cancels and replaces ISO/IEC 23360-2:2006, which has been technically revised.

This document is based on “The GNU Free Documentation License, version 1.1”. The license is available at https://www.gnu.org/licenses/old-licenses/fdl-1.1.html.

A list of all parts in the ISO/IEC 23660 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

Contents

Foreword ... **iii**
Introduction ... **vi**
I Introductory Elements ... **1**
1 Scope ... 2
2 References ... 3
2.1 Normative References ... 3
2.2 Informative References/Bibliography ... 5
3 Requirements ... 8
3.1 Relevant Libraries ... 8
3.2 LSB Implementation Conformance ... 8
3.3 LSB Application Conformance ... 9
4 Terms and Definitions ... 11
5 Documentation Conventions ... 13
II Executable and Linking Format (ELF) ... **14**
6 Introduction ... 15
7 Low Level System Information ... 16
7.1 Machine Interface ... 16
7.2 Function Calling Sequence ... 17
7.3 Operating System Interface ... 18
7.4 Process Initialization ... 19
7.5 Coding Examples ... 20
7.6 C Stack Frame ... 21
7.7 Debug Information ... 21
8 Object Format ... 22
8.1 Introduction ... 22
8.2 ELF Header ... 22
8.3 Special Sections ... 22
8.4 Symbol Table ... 23
8.5 Relocation ... 23
9 Program Loading and Dynamic Linking ... 24
9.1 Introduction ... 24
9.2 Program Header ... 24
9.3 Program Loading ... 24
9.4 Dynamic Linking ... 24
III Base Libraries ... **26**
10 Libraries ... 27
10.1 Program Interpreter/Dynamic Linker ... 27
10.2 Interfaces for libc ... 27
10.3 Data Definitions for libc ... 47
10.4 Interface Definitions for libc ... 66
10.5 Interfaces for libm ... 68
10.6 Data Definitions for libm ... 73
10.7 Interface Definitions for libm ... 74
10.8 Interfaces for libpthread ... 75
10.9 Data Definitions for libpthread ... 80
10.10 Interfaces for libgcc_s ... 82
10.11 Data Definitions for libgcc_s ... 82
10.12 Interface Definitions for libgcc_s ... 83
10.13 Interfaces for libdl ... 84
10.14 Data Definitions for libdl ... 84
10.15 Interfaces for libcrypt ... 85
10.16 Data Definitions for libcrypt ... 85
IV Utility Libraries ... **87**
11 Libraries ... 88

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

11.1 Interfaces for libz 88
11.2 Data Definitions for libz 88
11.3 Interfaces for libncurses 89
11.4 Data Definitions for libncurses 89
11.5 Interfaces for libncursesw 89
11.6 Data Definitions for libncursesw 90
11.7 Interfaces for libutil 90
V Base Libraries 92
12 Libraries 93
12.1 Interfaces for libstdcxx 93
12.2 Interface Definitions for libstdcxx 203
VI Package Format and Installation 204
13 Software Installation 205
13.1 Package Dependencies 205
13.2 Package Architecture Considerations 205
Annex A Alphabetical Listing of Interfaces by Library 206
A.1 libc 206
A.2 libcrypt 221
A.3 libdl 221
A.4 libgcc_s 222
A.5 libm 222
A.6 libpthread 227
A.7 librt 230
A.8 libutil 231

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

Introduction

The LSB defines a binary interface for application programs that are compiled and packaged for LSB-conforming implementations on many different hardware architectures. A binary specification must include information specific to the computer processor architecture for which it is intended. To avoid the complexity of conditional descriptions, the specification has instead been divided into generic parts which are augmented by one of several architecture-specific parts, depending on the target processor architecture; the generic part will indicate when reference must be made to the architecture part, and vice versa.

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

The specification carries a version number of either the form `x.y` or `x.y.z`. This version number carries the following meaning:

1. The first number (`x`) is the major version number. Versions sharing the same major version number shall be compatible in a backwards direction; that is, a newer version shall be compatible with an older version. Any deletion of a library results in a new major version number. Interfaces marked as deprecated may be removed from the specification at a major version change.
2. The second number (`y`) is the minor version number. Libraries and individual interfaces may be added, but not removed. Interfaces may be marked as deprecated at a minor version change. Other minor changes may be permitted at the discretion of the LSB workgroup.
3. The third number (`z`), if present, is the editorial level. Only editorial changes should be included in such versions.

Since this specification is a descriptive Application Binary Interface, and not a source level API specification, it is not possible to make a guarantee of 100% backward compatibility between major releases. However, it is the intent that those parts of the binary interface that are visible in the source level API will remain backward compatible from version to version, except where a feature marked as "Deprecated" in one release may be removed from a future release. Implementors are strongly encouraged to make use of symbol versioning to permit simultaneous support of applications conforming to different releases of this specification.

LSB is a trademark of the Linux Foundation. Developers of applications or implementations interested in using the trademark should see the Linux Foundation Certification Policy for details.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

I Introductory Elements

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

1 Scope

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

These specifications are composed of two basic parts: a common part describing those parts of the interface that remain constant across all implementations of the LSB, and an architecture-specific part describing the parts of the interface that vary by processor architecture. Together, the common part and the relevant architecture-specific part for a single hardware architecture provide a complete interface specification for compiled application programs on systems that share a common hardware architecture.

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

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

This is the X86 architecture specific part of the Core module of the Linux Standard Base (LSB). This part supplements the common part of the LSB Core module with those interfaces that differ between architectures.

This part should be used in conjunction with LSB Core - Generic, the common part. Whenever a section of the common part is supplemented by architecture-specific information, the common part includes a reference to the architecture-specific part. This part may also contain additional information that is not referenced in the common part.

Interfaces described in this part of the LSB Core Specification are mandatory except where explicitly listed otherwise. Interfaces described in the LSB Core module are supplemented by other LSB modules. All other modules depend on the presence of LSB Core.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

2 References

2.1 Normative References

The following specifications are incorporated by reference into this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced specification (including any amendments) applies.

Note: Where copies of a referenced specification are available on the World Wide Web, a Uniform Resource Locator (URL) is given, for informative purposes only. Such URL might at any given time resolve to a more recent copy of the specification, or be out of date (not resolve). Reference copies of specifications at the revision level indicated may be found at the Linux Foundation's Reference Specifications (http://refspecs.linuxbase.org) site.

Table 2-1 Normative References

Name	Title	URL
LSB Core - Generic	Linux Standard Base - Core Specification - Generic	http://www.linuxbase.org/spec/
Filesystem Hierarchy Standard	Filesystem Hierarchy Standard (FHS) 3.0	http://refspecs.linuxbase.org/fhs
Intel® Architecture Software Developer's Manual Volume 1	The IA-32 Intel® Architecture Software Developer's Manual Volume 1: Basic Architecture	http://developer.intel.com/design/pentium4/manuals/245470.htm
Intel® Architecture Software Developer's Manual Volume 2	The IA-32 Intel® Architecture Software Developer's Manual Volume 2: Instruction Set Reference	http://developer.intel.com/design/pentium4/manuals/245471.htm
Intel® Architecture Software Developer's Manual Volume 3	The IA-32 Intel® Architecture Software Developer's Manual Volume 3: System Programming Guide	http://developer.intel.com/design/pentium4/manuals/245472.htm
ISO C (1999)	ISO/IEC 9899:1999 - Programming Languages -- C	
ISO/IEC 14882: 2003 C++ Language	ISO/IEC 14882: 2003 Programming languages --C++	
Itanium™ C++ ABI	Itanium™ C++ ABI (Revision 1.86)	http://refspecs.linuxfoundation.org/cxxabi-1.86.html
Large File Support	Large File Support	http://www.UNIX-systems.org/version2/

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

Name	Title	URL
		whatsnew/lfs20mar.html
Libncursesw API	Libncursesw API	http://invisible-island.net/ncurses/man/ncurses.3x.html
Libncursesw Placeholder	Libncursesw Specification Placeholder	http://refspecs.linux-foundation.org/libncursesw/libncurses.html
POSIX 1003.1-2001 (ISO/IEC 9945-2003)	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 Including Technical Cor. 1: 2004	http://www.unix.org/version3/
POSIX 1003.1-2008 (ISO/IEC 9945-2009)	Portable Operating System Interface (POSIX®) 2008 Edition / The Open Group Technical Standard Base Specifications, Issue 7	http://www.unix.org/version4/
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

AVA Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Name	Title	URL
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	http://refspecs.linuxfoundation.org/svid4/
System V ABI	System V Application Binary Interface, Edition 4.1	http://www.sco.com/developers/devspecs/gabi41.pdf
System V ABI Update	System V Application Binary Interface - DRAFT - 17 December 2003	http://www.sco.com/developers/gabi/2003-12-17/contents.html
System V ABI, IA32 Supplement	System V Application Binary Interface - Intel386 Architecture Processor Supplement, Fourth Edition	http://www.sco.com/developers/devspecs/abi386-4.pdf
X/Open Curses, Issue 7	X/Open Curses, Issue 7 (ISBN: 1-931624-83-6, The Open Group, November 2009)	https://www2.opengroup.org/ogsys/catalog/C094

2.2 Informative References/Bibliography

The documents listed below provide essential background information to implementors of this specification. These references are included for information only, and do not represent normative parts of this specification.

Table 2-2 Other References

Name	Title	URL
DWARF Debugging Information Format, Version 4	DWARF Debugging Information Format, Version 4 (June 10, 2010)	http://www.dwarfstd.org/doc/DWARF4.pdf
IEC 60559/IEEE 754 Floating Point	IEC 60559:1989 Binary floating-point arithmetic for microprocessor systems	http://www.ieee.org/
ISO/IEC TR14652	ISO/IEC Technical Report 14652:2002 Specification method for cultural conventions	

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Name	Title	URL
ITU-T V.42	International Telecommunication Union Recommendation V.42 (2002): Error-correcting procedures for DCEs using asynchronous-to-synchronous conversionITUV	http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-V.42
Li18nux Globalization Specification	LI18NUX 2000 Globalization Specification, Version 1.0 with Amendment 4	http://www.openi18n.org/docs/html/LI18NUX-2000-amd4.htm
Linux Allocated Device Registry	LINUX ALLOCATED DEVICES	http://www.lanana.org/docs/device-list/devices-2.6+.txt
Linux Assigned Names And Numbers Authority	Linux Assigned Names And Numbers Authority	http://www.lanana.org/
Mozilla's NSS SSL Reference	Mozilla's NSS SSL Reference	http://www.mozilla.org/projects/security/pki/nss/ref/ssl/
NSPR Reference	Mozilla's NSPR Reference	http://refspecs.linuxfoundation.org/NSPR_API_Reference/NSPR_API.html
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 1950: ZLIB Compressed Data Format Specication	IETF RFC 1950: ZLIB Compressed Data Format Specification	http://www.ietf.org/rfc/rfc1950.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

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Name	Title	URL
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
RFC 2821:Simple Mail Transfer Protocol	IETF RFC 2821: Simple Mail Transfer Protocol	http://www.ietf.org/rfc/rfc2821.txt
RFC 2822:Internet Message Format	IETF RFC 2822: Internet Message Format	http://www.ietf.org/rfc/rfc2822.txt
RFC 5531/4506 RPC & XDR	IETF RFC 5531 & 4506	http://www.ietf.org/
RFC 791:Internet Protocol	IETF RFC 791: Internet Protocol Specification	http://www.ietf.org/rfc/rfc791.txt
RPM Package Format	RPM Package Format V3.0	http://www.rpm.org/max-rpm/s1-rpm-file-format-rpm-file-format.html
zlib Manual	zlib 1.2 Manual	http://www.gzip.org/zlib/

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

3 Requirements

3.1 Relevant Libraries

The libraries listed in Table 3-1 shall be available on IA32 Linux Standard Base systems, with the specified runtime names. These names override or supplement the names specified in the generic LSB (LSB Core - Generic) specification. The specified program interpreter, referred to as proginterp in this table, shall be used to load the shared libraries specified by `DT_NEEDED` entries at run time.

Table 3-1 Standard Library Names

Library	Runtime Name
libc	libc.so.6
libcrypt	libcrypt.so.1
libdl	libdl.so.2
libgcc_s	libgcc_s.so.1
libm	libm.so.6
libncurses	libncurses.so.5
libncursesw	libncursesw.so.5
libpthread	libpthread.so.0
libstdcxx	libstdc++.so.6
libutil	libutil.so.1
libz	libz.so.1
proginterp	/lib/ld-lsb.so.3

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

3.2 LSB Implementation Conformance

A conforming implementation is necessarily architecture specific, and must provide the interfaces specified by both the generic LSB Core specification (LSB Core - Generic) and the relevant architecture specific part of the LSB Core Specification.

> **Rationale:** An implementation must provide *at least* the interfaces specified in these specifications. It may also provide additional interfaces.

A conforming implementation shall satisfy the following requirements:

- A processor architecture represents a family of related processors which may not have identical feature sets. The architecture specific parts of the LSB Core Specification that supplement this specification for a given target processor architecture describe a minimum acceptable processor. The implementation shall provide all features of this processor, whether in hardware or through emulation transparent to the application.

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

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

3.3 LSB Application Conformance

A conforming application containing object files is necessarily architecture specific, and must conform to both the generic LSB Core specification (LSB Core - Generic) and the relevant architecture specific part of the LSB Core Specification. A conforming application which contains no object files may be architecture neutral. Architecture neutral applications shall conform only to the requirements of the generic LSB Core specification (LSB Core - Generic).

A conforming application shall satisfy the following requirements:

- Executable files shall be either object files in the format defined in the Object Format section of this specification, or script files in a scripting language where the interpreter is required by this specification.
- Object files shall participate in dynamic linking as defined in the Program Loading and Linking section of this specification.
- Object files shall employ only the instructions, traps, and other low-level facilities defined as being for use by applications in the Low-Level System Information section of this specification

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

- If the application requires any optional interface defined in this specification in order to be installed or to execute successfully, the requirement for that optional interface shall be stated in the application's documentation.
- The application shall not use any interface or data format that is not required to be provided by a conforming implementation, unless such an interface or data format is supplied by another application through direct invocation of that application during execution. The other application must also be a conforming application, and the use of such interface or data format, as well as its source (in other words, the other conforming application), shall be identified in the documentation of the application.
- The application shall not use any values for a named interface that are reserved for vendor extensions.

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

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

4 Terms and Definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382, ISO 80000–2, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

4.1

archLSB

Some LSB specification documents have both a generic, architecture-neutral part and an architecture-specific part. The latter describes elements whose definitions may be unique to a particular processor architecture. The term archLSB may be used in the generic part to refer to the corresponding section of the architecture-specific part.

4.2

Binary Standard, ABI

The total set of interfaces that are available to be used in the compiled binary code of a conforming application, including the run-time details such as calling conventions, binary format, C++ name mangling, etc.

4.3

Implementation-defined

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

4.4

Shell Script

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

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

4.5

Source Standard, API

The total set of interfaces that are available to be used in the source code of a conforming application. Due to translations, the Binary Standard and the Source Standard may contain some different interfaces.

4.6

Undefined

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

4.7

Unspecified

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

In addition, for the portions of this specification which build on IEEE Std 1003.1-2001, the definitions given in *IEEE Std 1003.1-2001, Base Definitions, Chapter 3* apply.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

5 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) [SUSv4]

refers to the interface named `forkpty()` with symbol version `GLIBC_2.0` that is defined in the reference indicated by the tag `SUSv4`.

Note: For symbols with versions which differ between architectures, the symbol versions are defined in the architecture specific parts of of this module specification only. In the generic part, they will appear without symbol versions.

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

II Executable and Linking Format (ELF)

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

6 Introduction

Executable and Linking Format (ELF) defines the object format for compiled applications. This specification supplements the information found in System V ABI Update and System V ABI, IA32 Supplement, and is intended to document additions made since the publication of that document.

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

7 Low Level System Information

7.1 Machine Interface

7.1.1 Processor Architecture

The IA32 Architecture is specified by the following documents

- Intel® Architecture Software Developer's Manual Volume 1
- Intel® Architecture Software Developer's Manual Volume 2
- Intel® Architecture Software Developer's Manual Volume 3

Only the features of the Intel486 processor instruction set may be assumed to be present. An application should determine if any additional instruction set features are available before using those additional features. If a feature is not present, then a conforming application shall not use it.

Conforming applications may use only instructions which do not require elevated privileges.

Conforming applications shall not invoke the implementations underlying system call interface directly. The interfaces in the implementation base libraries shall be used instead.

> **Rationale:** Implementation-supplied base libraries may use the system call interface but applications must not assume any particular operating system or kernel version is present.

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

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

7.1.2 Data Representation

LSB-conforming applications shall use the data representation as defined in Chapter 3 of the System V ABI, IA32 Supplement.

7.1.2.1 Byte Ordering

LSB-conforming systems and applications shall use the bit and byte ordering rules specified in Section 1.3.1 of the Intel® Architecture Software Developer's Manual Volume 1.

7.1.2.2 Fundamental Types

In addition to the fundamental types specified in Chapter 3 of the System V ABI, IA32 Supplement, a 64 bit data type is defined here.

Table 7-1 Scalar Types

Type	C	`sizeof`	Alignment (bytes)	Intel386 Architecture
Integral	long long	8	4	signed double word

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

Type	C	sizeof	Alignment (bytes)	Intel386 Architecture
	signed long long			
	unsigned long long	8	4	unsigned double word

7.1.2.3 Aggregates and Unions

LSB-conforming implementations shall support aggregates and unions with alignment and padding as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.1.2.4 Bit Fields

LSB-conforming implementations shall support structure and union definitions that include bit-fields as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2 Function Calling Sequence

LSB-conforming applications shall use the function calling sequence as defined in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.1 Registers

LSB-conforming applications shall use the general registers provided by the architecture in the manner described in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.2 Floating Point Registers

LSB-conforming applications shall use the floating point registers provided by the architecture in the manner described in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.3 Stack Frame

LSB-conforming applications shall use the stack frame in the manner specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.4 Arguments

7.2.4.1 Integral/Pointer

Integral and pointer arguments to functions shall be passed as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.4.2 Floating Point

Floating point arguments to functions shall be passed as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.4.3 Struct and Union Arguments

Structure and union arguments to functions shall be passed as specified in Chapter 3 of the System V ABI, IA32 Supplement.

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

7.2.4.4 Variable Arguments

As described in Chapter 3 of the System V ABI, IA32 Supplement, LSB-conforming applications using variable argument lists shall use the facilities defined in the header file `<stdarg.h>` to deal with variable argument lists.

> **Note:** This is a requirement of ISO C (1999) and POSIX 1003.1-2008 (ISO/IEC 9945-2009) as well as System V ABI, IA32 Supplement.

7.2.5 Return Values

7.2.5.1 Void

As described in chapter 3 of System V ABI, IA32 Supplement, functions returning no value need not set any register to any particular value.

7.2.5.2 Integral/Pointer

Functions return scalar values (integer or pointer), shall do so as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.5.3 Floating Point

Functions return floating point values shall do so as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.2.5.4 Struct and Union

Functions that return a structure or union shall do so as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.3 Operating System Interface

LSB-conforming applications shall use the following aspects of the Operating System Interfaces as defined in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.1 Virtual Address Space

LSB-conforming implementations shall support the virtual address space described in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.1.1 Page Size

LSB-conforming applications should call `sysconf()` to determine the current page size. See also Chapter 3 of the System V ABI, IA32 Supplement.

7.3.1.2 Virtual Address Assignments

LSB-conforming systems shall provide the virtual address space configuration as described in Chapter 3 of the System V ABI, IA32 Supplement (Virtual Address Assignments).

7.3.1.3 Managing the Process Stack

LSB-conforming systems shall manage the process stack as specified in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.1.4 Coding Guidlines

LSB-conforming applications should follow the coding guidleines provided in Chapter 3 of the System V ABI, IA32 Supplement.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

7.3.2 Processor Execution Mode

LSB-conforming applications shall run in the user-mode ring as described in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.3 Exception Interface

7.3.3.1 Introduction

LSB-conforming system shall provide the exception interface described in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.3.2 Hardware Exception Types

LSB-conforming systems shall map hardware exceptions to signals as described in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.3.3 Software Trap Types

Software generated traps are subject to the limitations described in Chapter 3 of the System V ABI, IA32 Supplement.

7.3.4 Signal Delivery

There are no architecture specific requirements for signal delivery.

7.3.4.1 Signal Handler Interface

There are no architecture specific requirements for the signal handler interface.

7.4 Process Initialization

An LSB-conforming implementation shall cause an application to be initialized as described in the Process Initialization section of Chapter 3 of the System V ABI, IA32 Supplement, and as described below.

7.4.1 Special Registers

The special registers shall be initialized as described in Chapter 3 of the System V ABI, IA32 Supplement.

7.4.2 Process Stack (on entry)

The process stack shall be initialized as described in Chapter 3 of the System V ABI, IA32 Supplement.

7.4.3 Auxilliary Vector

The auxilliary vector shall be initialized as described in Chapter 3 of the System V ABI, IA32 Supplement.

7.4.4 Environment

There are no architecture specific requirements for environment initialization.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

7.5 Coding Examples

7.5.1 Introduction

LSB-conforming applications may follow the coding examples provdied in chapter 3 of the System V ABI, IA32 Supplement in order to implement certain fundamental operations.

7.5.2 Code Model Overview/Architecture Constraints

Chapter 3 of the System V ABI, IA32 Supplement provides an overview of the code model.

7.5.3 Position-Independent Function Prologue

LSB-conforming applications using position independent functions may use the techniques described in Chapter 3 of the System V ABI, IA32 Supplement.

7.5.4 Data Objects

LSB-conforming applications accessing non-stack resident data objects may do so as described in Chapter 3 of the System V ABI, IA32 Supplement, including both absolute and position independent data access techniques.

7.5.5 Function Calls

7.5.5.1 Absolute Direct Function Call

LSB-conforming applications using direct function calls with absolute addressing may follow the examples given in Chapter 3 of the System V ABI, IA32 Supplement.

7.5.5.2 Absolute Indirect Function Call

LSB-conforming applications using indirect function calls with absolute addressing may follow the examples given in Chapter 3 of the System V ABI, IA32 Supplement.

7.5.5.3 Position-Independent Direct Function Call

LSB-conforming applications using direct function calls with position independent addressing may follow the examples given in Chapter 3 of the System V ABI, IA32 Supplement.

7.5.5.4 Position-Independent Indirect Function Call

LSB-conforming applications using indirect function calls with position independent addressing may follow the examples given in Chapter 3 of the System V ABI, IA32 Supplement.

7.5.6 Branching

LSB-conforming applications may follow the branching examples given in Chapter 3 of the System V ABI, IA32 Supplement.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

7.6 C Stack Frame

7.6.1 Variable Argument List

As described in Chapter 3 of the System V ABI, IA32 Supplement, LSB-conforming applications using variable argument lists shall use the facilities defined in the header file `<stdarg.h>` to deal with variable argument lists.

Note: This is a requirement of ISO C (1999) and POSIX 1003.1-2008 (ISO/IEC 9945-2009) as well as System V ABI, IA32 Supplement.

7.6.2 Dynamic Allocation of Stack Space

LSB-conforming applications may allocate space using the stack following the examples given in Chapter 3 of the System V ABI, IA32 Supplement.

7.7 Debug Information

There are no architecture specific requirements for debugging information for this architecture. LSB-conforming applications may utilize DWARF sections as described in the generic specification.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

8 Object Format

8.1 Introduction

LSB-conforming implementations shall support the Executable and Linking Format (ELF) object file format, as defined by the following documents:

- System V ABI
- System V ABI Update
- System V ABI, IA32 Supplement
- LSB Core - Generic
- this document

8.2 ELF Header

8.2.1 Machine Information

LSB-conforming applications shall use the Machine Information as defined in Chapter 4 of the System V ABI, IA32 Supplement, including the `e_ident` array members for `EI_CLASS` and `EI_DATA`, the processor identification in `e_machine` and flags in `e_flags`. The operating system identification field, in `e_ident[EI_OSABI]` shall be `ELFOSABI_NONE` (0).

8.3 Special Sections

8.3.1 Special Sections

Various sections hold program and control information. Sections in the lists below are used by the system and have the indicated types and attributes.

8.3.1.1 ELF Special Sections

The following sections are defined in Chapter 4 of the System V ABI, IA32 Supplement.

Table 8-1 ELF Special Sections

Name	Type	Attributes
.got	SHT_PROGBITS	SHF_AL-LOC+SHF_WRITE
.plt	SHT_PROGBITS	SHF_ALLOC+SHF_EX-ECINSTR

.got

This section holds the global offset table. See `Coding Examples' in Chapter 3, `Special Sections' in Chapter 4, and `Global Offset Table' in Chapter 5 of the processor supplement for more information.

.plt

This section holds the procedure linkage table.

8.3.1.2 Additional Special Sections

The following additional sections are defined here.

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

Table 8-2 Additional Special Sections

Name	Type	Attributes
.rel.dyn	SHT_REL	SHF_ALLOC

.rel.dyn

This section holds relocation information, as described in `Relocation' section in Chapter 4 of System V ABI Update. These relocations are applied to the .dyn section.

8.4 Symbol Table

LSB-conforming applications shall use the Symbol Table section as defined in Chapter 4 of the System V ABI, IA32 Supplement.

8.5 Relocation

8.5.1 Introduction

LSB-conforming implementations shall support Relocation as defined in Chapter 4 of the System V ABI, IA32 Supplement and as described below.

8.5.2 Relocation Types

The relocation types described in Chapter 4 of the System V ABI, IA32 Supplement shall be supported.

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

9 Program Loading and Dynamic Linking

9.1 Introduction

LSB-conforming implementations shall support the object file information and system actions that create running programs as specified in the System V ABI , System V ABI Update , System V ABI, IA32 Supplement and as supplemented by LSB Core - Generic and the generic LSB specification.

9.2 Program Header

9.2.1 Introduction

As described in System V ABI Update, the program header is an array of structures, each describing a segment or other information the system needs to prepare the program for execution.

9.2.2 Types

The IA32 architecture does not define any additional program header types beyond those required in the generic LSB Core specification.

9.2.3 Flags

The IA32 architecture does not define any additional program header flags beyond those required in the generic LSB Core specification.

9.3 Program Loading

LSB-conforming systems shall support program loading as defined in Chapter 5 of the System V ABI, IA32 Supplement.

9.4 Dynamic Linking

LSB-conforming systems shall support dynamic linking as defined in Chapter 5 of the System V ABI, IA32 Supplement.

9.4.1 Dynamic Section

The following dynamic entries are defined in the System V ABI, IA32 Supplement.

DT_PLTGOT

On the Intel386 architecture, this entrys d_ptr member gives the address of the first entry in the global offset table.

9.4.2 Global Offset Table

LSB-conforming implementations shall support use of the global offset table as described in Chapter 5 of the System V ABI, IA32 Supplement.

9.4.3 Shared Object Dependencies

There are no architecture specific requirements for shared object dependencies; see the generic LSB-Core specification.

9.4.4 Function Addresses

Function addresses shall behave as specified in Chapter 5 of the System V ABI, IA32 Supplement.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

9.4.5 Procedure Linkage Table

LSB-conforming implementations shall support a Procedure Linkage Table as described in Chapter 5 of the System V ABI, IA32 Supplement.

9.4.6 Initialization and Termination Functions

There are no architecture specific requirements for initialization and termination functions; see the generic LSB-Core specification.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

III Base Libraries

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

10 Libraries

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

Only interfaces and interface details which are unique to the X86 platform are defined here. This section should be used in conjunction with the corresponding section of LSB Core - Generic.

10.1 Program Interpreter/Dynamic Linker

The Program Interpreter shall be `/lib/ld-lsb.so.3`.

10.2 Interfaces for libc

Table 10-1 defines the library name and shared object name for the libc library

Table 10-1 libc Definition

Library:	libc
SONAME:	libc.so.6

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

[LFS] Large File Support
[LSB] LSB Core - Generic
[RPC + XDR] RFC 5531/4506 RPC & XDR
[SUSv2] SUSv2
[SUSv3] POSIX 1003.1-2001 (ISO/IEC 9945-2003)
[SUSv4] POSIX 1003.1-2008 (ISO/IEC 9945-2009)
[SVID.4] SVID Issue 4

10.2.1 RPC

10.2.1.1 Interfaces for RPC

An LSB conforming implementation shall provide the architecture specific functions for RPC specified in Table 10-2, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-2 libc - RPC Function Interfaces

authnone_create(GLIBC_2.0) [SVID.4]	callrpc(GLIBC_2.0) [RPC + XDR]	clnt_create(GLIBC_2.0) [SVID.4]	clnt_pcreateerror(GLIBC_2.0) [SVID.4]
clnt_perrno(GLIBC_2.0) [SVID.4]	clnt_perror(GLIBC_2.0) [SVID.4]	clnt_spcreateerror(GLIBC_2.0) [SVID.4]	clnt_sperrno(GLIBC_2.0) [SVID.4]
clnt_sperror(GLIBC_2.0) [SVID.4]	clntraw_create(GLIBC_2.0) [RPC + XDR]	clnttcp_create(GLIBC_2.0) [RPC + XDR]	clntudp_bufcreate(GLIBC_2.0) [RPC + XDR]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

clntudp_create(GLIBC_2.0) [RPC + XDR]	key_decryptsession(GLIBC_2.1) [SVID.4]	pmap_getport(GLIBC_2.0) [LSB]	pmap_set(GLIBC_2.0) [LSB]
pmap_unset(GLIBC_2.0) [LSB]	svc_getreqset(GLIBC_2.0) [SVID.4]	svc_register(GLIBC_2.0) [LSB]	svc_run(GLIBC_2.0) [LSB]
svc_sendreply(GLIBC_2.0) [LSB]	svcerr_auth(GLIBC_2.0) [SVID.4]	svcerr_decode(GLIBC_2.0) [SVID.4]	svcerr_noproc(GLIBC_2.0) [SVID.4]
svcerr_noprog(GLIBC_2.0) [SVID.4]	svcerr_progvers(GLIBC_2.0) [SVID.4]	svcerr_systemerr(GLIBC_2.0) [SVID.4]	svcerr_weakauth(GLIBC_2.0) [SVID.4]
svcfd_create(GLIBC_2.0) [RPC + XDR]	svcraw_create(GLIBC_2.0) [RPC + XDR]	svctcp_create(GLIBC_2.0) [LSB]	svcudp_create(GLIBC_2.0) [LSB]
xdr_accepted_reply(GLIBC_2.0) [SVID.4]	xdr_array(GLIBC_2.0) [SVID.4]	xdr_bool(GLIBC_2.0) [SVID.4]	xdr_bytes(GLIBC_2.0) [SVID.4]
xdr_callhdr(GLIBC_2.0) [SVID.4]	xdr_callmsg(GLIBC_2.0) [SVID.4]	xdr_char(GLIBC_2.0) [SVID.4]	xdr_double(GLIBC_2.0) [SVID.4]
xdr_enum(GLIBC_2.0) [SVID.4]	xdr_float(GLIBC_2.0) [SVID.4]	xdr_free(GLIBC_2.0) [SVID.4]	xdr_int(GLIBC_2.0) [SVID.4]
xdr_long(GLIBC_2.0) [SVID.4]	xdr_opaque(GLIBC_2.0) [SVID.4]	xdr_opaque_auth(GLIBC_2.0) [SVID.4]	xdr_pointer(GLIBC_2.0) [SVID.4]
xdr_reference(GLIBC_2.0) [SVID.4]	xdr_rejected_reply(GLIBC_2.0) [SVID.4]	xdr_replymsg(GLIBC_2.0) [SVID.4]	xdr_short(GLIBC_2.0) [SVID.4]
xdr_string(GLIBC_2.0) [SVID.4]	xdr_u_char(GLIBC_2.0) [SVID.4]	xdr_u_int(GLIBC_2.0) [LSB]	xdr_u_long(GLIBC_2.0) [SVID.4]
xdr_u_short(GLIBC_2.0) [SVID.4]	xdr_union(GLIBC_2.0) [SVID.4]	xdr_vector(GLIBC_2.0) [SVID.4]	xdr_void(GLIBC_2.0) [SVID.4]
xdr_wrapstring(GLIBC_2.0) [SVID.4]	xdrmem_create(GLIBC_2.0) [SVID.4]	xdrrec_create(GLIBC_2.0) [SVID.4]	xdrrec_endofrecord(GLIBC_2.0) [RPC + XDR]
xdrrec_eof(GLIBC_2.0) [SVID.4]	xdrrec_skiprecord(GLIBC_2.0) [RPC + XDR]	xdrstdio_create(GLIBC_2.0) [LSB]	

An LSB conforming implementation shall provide the architecture specific deprecated functions for RPC specified in Table 10-3, with the full mandatory functionality as described in the referenced underlying specification.

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

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Table 10-3 libc - RPC Deprecated Function Interfaces

key_decryptsession(GLIBC_2.1) [SVID.4]			

10.2.2 Epoll

10.2.2.1 Interfaces for Epoll

No external functions are defined for libc - Epoll in this part of the specification. See also the generic specification.

10.2.3 System Calls

10.2.3.1 Interfaces for System Calls

An LSB conforming implementation shall provide the architecture specific functions for System Calls specified in Table 10-4, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-4 libc - System Calls Function Interfaces

__fxstat(GLIBC_2.0) [LSB]	__getpgid(GLIBC_2.0) [LSB]	__lxstat(GLIBC_2.0) [LSB]	__xmknod(GLIBC_2.0) [LSB]
__xstat(GLIBC_2.0) [LSB]	access(GLIBC_2.0) [SUSv4]	acct(GLIBC_2.0) [LSB]	alarm(GLIBC_2.0) [SUSv4]
backtrace(GLIBC_2.1) [LSB]	backtrace_symbols(GLIBC_2.1) [LSB]	backtrace_symbols_fd(GLIBC_2.1) [LSB]	brk(GLIBC_2.0) [SUSv2]
chdir(GLIBC_2.0) [SUSv4]	chmod(GLIBC_2.0) [SUSv4]	chown(GLIBC_2.1) [SUSv4]	chroot(GLIBC_2.0) [SUSv2]
clock(GLIBC_2.0) [SUSv4]	close(GLIBC_2.0) [SUSv4]	closedir(GLIBC_2.0) [SUSv4]	creat(GLIBC_2.0) [SUSv4]
dup(GLIBC_2.0) [SUSv4]	dup2(GLIBC_2.0) [SUSv4]	execl(GLIBC_2.0) [SUSv4]	execle(GLIBC_2.0) [SUSv4]
execlp(GLIBC_2.0) [SUSv4]	execv(GLIBC_2.0) [SUSv4]	execve(GLIBC_2.0) [SUSv4]	execvp(GLIBC_2.0) [SUSv4]
exit(GLIBC_2.0) [SUSv4]	fchdir(GLIBC_2.0) [SUSv4]	fchmod(GLIBC_2.0) [SUSv4]	fchown(GLIBC_2.0) [SUSv4]
fcntl(GLIBC_2.0) [LSB]	fdatasync(GLIBC_2.0) [SUSv4]	fexecve(GLIBC_2.0) [SUSv4]	flock(GLIBC_2.0) [LSB]
fork(GLIBC_2.0) [SUSv4]	fstatfs(GLIBC_2.0) [LSB]	fstatvfs(GLIBC_2.1) [SUSv4]	fsync(GLIBC_2.0) [SUSv4]
ftime(GLIBC_2.0) [SUSv3]	ftruncate(GLIBC_2.0) [SUSv4]	getcontext(GLIBC_2.1) [SUSv3]	getdtablesize(GLIBC_2.0) [LSB]
getegid(GLIBC_2.0) [SUSv4]	geteuid(GLIBC_2.0) [SUSv4]	getgid(GLIBC_2.0) [SUSv4]	getgroups(GLIBC_2.0) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

getitimer(GLIBC_2.0) [SUSv4]	getloadavg(GLIBC_2.2) [LSB]	getpagesize(GLIBC_2.0) [LSB]	getpgid(GLIBC_2.0) [SUSv4]
getpgrp(GLIBC_2.0) [SUSv4]	getpid(GLIBC_2.0) [SUSv4]	getppid(GLIBC_2.0) [SUSv4]	getpriority(GLIBC_2.0) [SUSv4]
getrlimit(GLIBC_2.2) [LSB]	getrusage(GLIBC_2.0) [SUSv4]	getsid(GLIBC_2.0) [SUSv4]	getuid(GLIBC_2.0) [SUSv4]
getwd(GLIBC_2.0) [SUSv3]	initgroups(GLIBC_2.0) [LSB]	ioctl(GLIBC_2.0) [LSB]	ioperm(GLIBC_2.0) [LSB]
iopl(GLIBC_2.0) [LSB]	kill(GLIBC_2.0) [LSB]	killpg(GLIBC_2.0) [SUSv4]	lchown(GLIBC_2.0) [SUSv4]
link(GLIBC_2.0) [LSB]	lockf(GLIBC_2.0) [SUSv4]	lseek(GLIBC_2.0) [SUSv4]	mkdir(GLIBC_2.0) [SUSv4]
mkfifo(GLIBC_2.0) [SUSv4]	mlock(GLIBC_2.0) [SUSv4]	mlockall(GLIBC_2.0) [SUSv4]	mmap(GLIBC_2.0) [SUSv4]
mprotect(GLIBC_2.0) [SUSv4]	mremap(GLIBC_2.0) [LSB]	msync(GLIBC_2.0) [SUSv4]	munlock(GLIBC_2.0) [SUSv4]
munlockall(GLIBC_2.0) [SUSv4]	munmap(GLIBC_2.0) [SUSv4]	nanosleep(GLIBC_2.0) [SUSv4]	nice(GLIBC_2.0) [SUSv4]
open(GLIBC_2.0) [SUSv4]	opendir(GLIBC_2.0) [SUSv4]	pathconf(GLIBC_2.0) [SUSv4]	pause(GLIBC_2.0) [SUSv4]
pipe(GLIBC_2.0) [SUSv4]	poll(GLIBC_2.0) [SUSv4]	pread(GLIBC_2.1) [SUSv4]	pselect(GLIBC_2.0) [SUSv4]
ptrace(GLIBC_2.0) [LSB]	pwrite(GLIBC_2.1) [SUSv4]	read(GLIBC_2.0) [SUSv4]	readdir(GLIBC_2.0) [SUSv4]
readdir_r(GLIBC_2.0) [SUSv4]	readlink(GLIBC_2.0) [SUSv4]	readv(GLIBC_2.0) [SUSv4]	rename(GLIBC_2.0) [SUSv4]
rmdir(GLIBC_2.0) [SUSv4]	sbrk(GLIBC_2.0) [SUSv2]	sched_get_priority_max(GLIBC_2.0) [SUSv4]	sched_get_priority_min(GLIBC_2.0) [SUSv4]
sched_getparam(GLIBC_2.0) [SUSv4]	sched_getscheduler(GLIBC_2.0) [SUSv4]	sched_rr_get_interval(GLIBC_2.0) [SUSv4]	sched_setparam(GLIBC_2.0) [SUSv4]
sched_setscheduler(GLIBC_2.0) [LSB]	sched_yield(GLIBC_2.0) [SUSv4]	select(GLIBC_2.0) [SUSv4]	setcontext(GLIBC_2.0) [SUSv3]
setegid(GLIBC_2.0) [SUSv4]	seteuid(GLIBC_2.0) [SUSv4]	setgid(GLIBC_2.0) [SUSv4]	setitimer(GLIBC_2.0) [SUSv4]
setpgid(GLIBC_2.0) [SUSv4]	setpgrp(GLIBC_2.0) [SUSv4]	setpriority(GLIBC_2.0) [SUSv4]	setregid(GLIBC_2.0) [SUSv4]
setreuid(GLIBC_2.0) [SUSv4]	setrlimit(GLIBC_2.2) [LSB]	setrlimit64(GLIBC_2.1) [LFS]	setsid(GLIBC_2.0) [SUSv4]
setuid(GLIBC_2.0) [SUSv4]	sleep(GLIBC_2.0) [SUSv4]	statfs(GLIBC_2.0) [LSB]	statvfs(GLIBC_2.1) [SUSv4]

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

stime(GLIBC_2.0) [LSB]	symlink(GLIBC_2.0) [SUSv4]	sync(GLIBC_2.0) [SUSv4]	sysconf(GLIBC_2.0) [LSB]
sysinfo(GLIBC_2.0) [LSB]	time(GLIBC_2.0) [SUSv4]	times(GLIBC_2.0) [SUSv4]	truncate(GLIBC_2.0) [SUSv4]
ulimit(GLIBC_2.0) [SUSv4]	umask(GLIBC_2.0) [SUSv4]	uname(GLIBC_2.0) [SUSv4]	unlink(GLIBC_2.0) [LSB]
utime(GLIBC_2.0) [SUSv4]	utimes(GLIBC_2.0) [SUSv4]	vfork(GLIBC_2.0) [SUSv3]	wait(GLIBC_2.0) [SUSv4]
wait4(GLIBC_2.0) [LSB]	waitid(GLIBC_2.1) [SUSv4]	waitpid(GLIBC_2.0) [SUSv4]	write(GLIBC_2.0) [SUSv4]
writev(GLIBC_2.0) [SUSv4]			

An LSB conforming implementation shall provide the architecture specific deprecated functions for System Calls specified in Table 10-5, with the full mandatory functionality as described in the referenced underlying specification.

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

Table 10-5 libc - System Calls Deprecated Function Interfaces

fstatfs(GLIBC_2.0) [LSB]	getdtablesize(GLIBC_2.0) [LSB]	getpagesize(GLIBC_2.0) [LSB]	getwd(GLIBC_2.0) [SUSv3]
statfs(GLIBC_2.0) [LSB]			

10.2.4 Standard I/O

10.2.4.1 Interfaces for Standard I/O

An LSB conforming implementation shall provide the architecture specific functions for Standard I/O specified in Table 10-6, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-6 libc - Standard I/O Function Interfaces

_IO_feof(GLIBC_2.0) [LSB]	_IO_getc(GLIBC_2.0) [LSB]	_IO_putc(GLIBC_2.0) [LSB]	_IO_puts(GLIBC_2.0) [LSB]
__fprintf_chk(GLIBC_2.3.4) [LSB]	__printf_chk(GLIBC_2.3.4) [LSB]	__snprintf_chk(GLIBC_2.3.4) [LSB]	__sprintf_chk(GLIBC_2.3.4) [LSB]
__vfprintf_chk(GLIBC_2.3.4) [LSB]	__vprintf_chk(GLIBC_2.3.4) [LSB]	__vsnprintf_chk(GLIBC_2.3.4) [LSB]	__vsprintf_chk(GLIBC_2.3.4) [LSB]
asprintf(GLIBC_2.0) [LSB]	clearerr(GLIBC_2.0) [SUSv4]	clearerr_unlocked(GLIBC_2.0) [LSB]	ctermid(GLIBC_2.0) [SUSv4]
dprintf(GLIBC_2.0) [SUSv4]	fclose(GLIBC_2.1) [SUSv4]	fdopen(GLIBC_2.1) [SUSv4]	feof(GLIBC_2.0) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

feof_unlocked(GLIBC_2.0) [LSB]	ferror(GLIBC_2.0) [SUSv4]	ferror_unlocked(GLIBC_2.0) [LSB]	fflush(GLIBC_2.0) [SUSv4]
fflush_unlocked(GLIBC_2.0) [LSB]	fgetc(GLIBC_2.0) [SUSv4]	fgetc_unlocked(GLIBC_2.1) [LSB]	fgetpos(GLIBC_2.2) [SUSv4]
fgets(GLIBC_2.0) [SUSv4]	fgets_unlocked(GLIBC_2.1) [LSB]	fgetwc_unlocked (GLIBC_2.2) [LSB]	fgetws_unlocked (GLIBC_2.2) [LSB]
fileno(GLIBC_2.0) [SUSv4]	fileno_unlocked(GLIBC_2.0) [LSB]	flockfile(GLIBC_2.0) [SUSv4]	fopen(GLIBC_2.1) [SUSv4]
fprintf(GLIBC_2.0) [SUSv4]	fputc(GLIBC_2.0) [SUSv4]	fputc_unlocked(GLIBC_2.0) [LSB]	fputs(GLIBC_2.0) [SUSv4]
fputs_unlocked(GLIBC_2.1) [LSB]	fputwc_unlocked (GLIBC_2.2) [LSB]	fputws_unlocked (GLIBC_2.2) [LSB]	fread(GLIBC_2.0) [SUSv4]
fread_unlocked(GLIBC_2.1) [LSB]	freopen(GLIBC_2.0) [SUSv4]	fscanf(GLIBC_2.0) [LSB]	fseek(GLIBC_2.0) [SUSv4]
fseeko(GLIBC_2.1) [SUSv4]	fsetpos(GLIBC_2.2) [SUSv4]	ftell(GLIBC_2.0) [SUSv4]	ftello(GLIBC_2.1) [SUSv4]
fwrite(GLIBC_2.0) [SUSv4]	fwrite_unlocked(GLIBC_2.1) [LSB]	getc(GLIBC_2.0) [SUSv4]	getc_unlocked(GLIBC_2.0) [SUSv4]
getchar(GLIBC_2.0) [SUSv4]	getchar_unlocked(GLIBC_2.0) [SUSv4]	getdelim(GLIBC_2.0) [SUSv4]	getline(GLIBC_2.0) [SUSv4]
getw(GLIBC_2.0) [SUSv2]	getwc_unlocked(GLIBC_2.2) [LSB]	getwchar_unlocked(GLIBC_2.2) [LSB]	pclose(GLIBC_2.1) [SUSv4]
popen(GLIBC_2.1) [SUSv4]	printf(GLIBC_2.0) [SUSv4]	putc(GLIBC_2.0) [SUSv4]	putc_unlocked(GLIBC_2.0) [SUSv4]
putchar(GLIBC_2.0) [SUSv4]	putchar_unlocked(GLIBC_2.0) [SUSv4]	puts(GLIBC_2.0) [SUSv4]	putw(GLIBC_2.0) [SUSv2]
putwc_unlocked(GLIBC_2.2) [LSB]	putwchar_unlocked(GLIBC_2.2) [LSB]	remove(GLIBC_2.0) [SUSv4]	rewind(GLIBC_2.0) [SUSv4]
rewinddir(GLIBC_2.0) [SUSv4]	scanf(GLIBC_2.0) [LSB]	seekdir(GLIBC_2.0) [SUSv4]	setbuf(GLIBC_2.0) [SUSv4]
setbuffer(GLIBC_2.0) [LSB]	setvbuf(GLIBC_2.0) [SUSv4]	snprintf(GLIBC_2.0) [SUSv4]	sprintf(GLIBC_2.0) [SUSv4]
sscanf(GLIBC_2.0) [LSB]	telldir(GLIBC_2.0) [SUSv4]	tempnam(GLIBC_2.0) [SUSv4]	ungetc(GLIBC_2.0) [SUSv4]
vasprintf(GLIBC_2.0) [LSB]	vdprintf(GLIBC_2.0) [SUSv4]	vfprintf(GLIBC_2.0) [SUSv4]	vprintf(GLIBC_2.0) [SUSv4]

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

vsnprintf(GLIBC_2.0) [SUSv4]	vsprintf(GLIBC_2.0) [SUSv4]		

An LSB conforming implementation shall provide the architecture specific deprecated functions for Standard I/O specified in Table 10-7, with the full mandatory functionality as described in the referenced underlying specification.

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

Table 10-7 libc - Standard I/O Deprecated Function Interfaces

tempnam(GLIBC_2.0) [SUSv4]			

An LSB conforming implementation shall provide the architecture specific data interfaces for Standard I/O specified in Table 10-8, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-8 libc - Standard I/O Data Interfaces

stderr(GLIBC_2.0) [SUSv4]	stdin(GLIBC_2.0) [SUSv4]	stdout(GLIBC_2.0) [SUSv4]	

10.2.5 Signal Handling

10.2.5.1 Interfaces for Signal Handling

An LSB conforming implementation shall provide the architecture specific functions for Signal Handling specified in Table 10-9, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-9 libc - Signal Handling Function Interfaces

__libc_current_sigrtmax(GLIBC_2.1) [LSB]	__libc_current_sigrtmin(GLIBC_2.1) [LSB]	__sigsetjmp(GLIBC_2.0) [LSB]	__sysv_signal(GLIBC_2.0) [LSB]
__xpg_sigpause(GLIBC_2.2) [LSB]	bsd_signal(GLIBC_2.0) [SUSv3]	psignal(GLIBC_2.0) [SUSv4]	raise(GLIBC_2.0) [SUSv4]
sigaction(GLIBC_2.0) [SUSv4]	sigaddset(GLIBC_2.0) [SUSv4]	sigaltstack(GLIBC_2.0) [SUSv4]	sigandset(GLIBC_2.0) [LSB]
sigdelset(GLIBC_2.0) [SUSv4]	sigemptyset(GLIBC_2.0) [SUSv4]	sigfillset(GLIBC_2.0) [SUSv4]	sighold(GLIBC_2.1) [SUSv4]
sigignore(GLIBC_2.1) [SUSv4]	siginterrupt(GLIBC_2.0) [SUSv4]	sigisemptyset(GLIBC_2.0) [LSB]	sigismember(GLIBC_2.0) [SUSv4]
siglongjmp(GLIBC_2.0) [SUSv4]	signal(GLIBC_2.0) [SUSv4]	sigorset(GLIBC_2.0) [LSB]	sigpause(GLIBC_2.0) [LSB]
sigpending(GLIBC_2.0) [SUSv4]	sigprocmask(GLIBC_2.0) [SUSv4]	sigqueue(GLIBC_2.1) [SUSv4]	sigrelse(GLIBC_2.1) [SUSv4]
sigreturn(GLIBC_2.0) [LSB]	sigset(GLIBC_2.1) [SUSv4]	sigsuspend(GLIBC_2.0) [SUSv4]	sigtimedwait(GLIBC_2.1) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

sigwait(GLIBC_2.0) [SUSv4]	sigwaitinfo(GLIBC_2.1) [SUSv4]		

An LSB conforming implementation shall provide the architecture specific deprecated functions for Signal Handling specified in Table 10-10, with the full mandatory functionality as described in the referenced underlying specification.

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

Table 10-10 libc - Signal Handling Deprecated Function Interfaces

sigpause(GLIBC_2.0) [LSB]			

An LSB conforming implementation shall provide the architecture specific data interfaces for Signal Handling specified in Table 10-11, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-11 libc - Signal Handling Data Interfaces

_sys_siglist(GLIBC_2.3.3) [LSB]			

10.2.6 Localization Functions

10.2.6.1 Interfaces for Localization Functions

An LSB conforming implementation shall provide the architecture specific functions for Localization Functions specified in Table 10-12, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-12 libc - Localization Functions Function Interfaces

bind_textdomain_codeset(GLIBC_2.2) [LSB]	bindtextdomain(GLIBC_2.0) [LSB]	catclose(GLIBC_2.0) [SUSv4]	catgets(GLIBC_2.0) [SUSv4]
catopen(GLIBC_2.0) [SUSv4]	dcgettext(GLIBC_2.0) [LSB]	dcngettext(GLIBC_2.2) [LSB]	dgettext(GLIBC_2.0) [LSB]
dngettext(GLIBC_2.2) [LSB]	gettext(GLIBC_2.0) [LSB]	iconv(GLIBC_2.1) [SUSv4]	iconv_close(GLIBC_2.1) [SUSv4]
iconv_open(GLIBC_2.1) [SUSv4]	localeconv(GLIBC_2.2) [SUSv4]	ngettext(GLIBC_2.2) [LSB]	nl_langinfo(GLIBC_2.0) [SUSv4]
setlocale(GLIBC_2.0) [SUSv4]	textdomain(GLIBC_2.0) [LSB]		

An LSB conforming implementation shall provide the architecture specific data interfaces for Localization Functions specified in Table 10-13, with the full mandatory functionality as described in the referenced underlying specification.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Table 10-13 libc - Localization Functions Data Interfaces

_nl_msg_cat_cntr (GLIBC_2.0) [LSB]			

10.2.7 Posix Spawn Option

10.2.7.1 Interfaces for Posix Spawn Option

An LSB conforming implementation shall provide the architecture specific functions for Posix Spawn Option specified in Table 10-14, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-14 libc - Posix Spawn Option Function Interfaces

posix_spawn(GLIBC_2.15) [SUSv4]	posix_spawn_file_actions_addclose(GLIBC_2.2) [SUSv4]	posix_spawn_file_actions_adddup2(GLIBC_2.2) [SUSv4]	posix_spawn_file_actions_addopen(GLIBC_2.2) [SUSv4]
posix_spawn_file_actions_destroy (GLIBC_2.2) [SUSv4]	posix_spawn_file_actions_init(GLIBC_2.2) [SUSv4]	posix_spawnattr_destroy(GLIBC_2.2) [SUSv4]	posix_spawnattr_getflags(GLIBC_2.2) [SUSv4]
posix_spawnattr_getpgroup(GLIBC_2.2) [SUSv4]	posix_spawnattr_getschedparam(GLIBC_2.2) [SUSv4]	posix_spawnattr_getschedpolicy(GLIBC_2.2) [SUSv4]	posix_spawnattr_getsigdefault(GLIBC_2.2) [SUSv4]
posix_spawnattr_getsigmask(GLIBC_2.2) [SUSv4]	posix_spawnattr_init(GLIBC_2.2) [SUSv4]	posix_spawnattr_setflags(GLIBC_2.2) [SUSv4]	posix_spawnattr_setpgroup(GLIBC_2.2) [SUSv4]
posix_spawnattr_setschedparam(GLIBC_2.2) [SUSv4]	posix_spawnattr_setschedpolicy(GLIBC_2.2) [SUSv4]	posix_spawnattr_setsigdefault(GLIBC_2.2) [SUSv4]	posix_spawnattr_setsigmask(GLIBC_2.2) [SUSv4]
posix_spawnp(GLIBC_2.15) [SUSv4]			

10.2.8 Posix Advisory Option

10.2.8.1 Interfaces for Posix Advisory Option

An LSB conforming implementation shall provide the architecture specific functions for Posix Advisory Option specified in Table 10-15, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-15 libc - Posix Advisory Option Function Interfaces

posix_fadvise(GLIBC_2.2) [SUSv4]	posix_fallocate(GLIBC_2.2) [SUSv4]	posix_madvise(GLIBC_2.2) [SUSv4]	posix_memalign(GLIBC_2.2) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

10.2.9 Socket Interface

10.2.9.1 Interfaces for Socket Interface

An LSB conforming implementation shall provide the architecture specific functions for Socket Interface specified in Table 10-16, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-16 libc - Socket Interface Function Interfaces

__h_errno_location(GLIBC_2.0) [LSB]	accept(GLIBC_2.0) [SUSv4]	bind(GLIBC_2.0) [SUSv4]	bindresvport(GLIBC_2.0) [LSB]
connect(GLIBC_2.0) [SUSv4]	gethostid(GLIBC_2.0) [SUSv4]	gethostname(GLIBC_2.0) [SUSv4]	getpeername(GLIBC_2.0) [SUSv4]
getsockname(GLIBC_2.0) [SUSv4]	getsockopt(GLIBC_2.0) [LSB]	if_freenameindex(GLIBC_2.1) [SUSv4]	if_indextoname(GLIBC_2.1) [SUSv4]
if_nameindex(GLIBC_2.1) [SUSv4]	if_nametoindex(GLIBC_2.1) [SUSv4]	listen(GLIBC_2.0) [SUSv4]	recv(GLIBC_2.0) [SUSv4]
recvfrom(GLIBC_2.0) [SUSv4]	recvmsg(GLIBC_2.0) [SUSv4]	send(GLIBC_2.0) [SUSv4]	sendmsg(GLIBC_2.0) [SUSv4]
sendto(GLIBC_2.0) [SUSv4]	setsockopt(GLIBC_2.0) [LSB]	shutdown(GLIBC_2.0) [SUSv4]	sockatmark(GLIBC_2.2.4) [SUSv4]
socket(GLIBC_2.0) [SUSv4]	socketpair(GLIBC_2.0) [SUSv4]		

An LSB conforming implementation shall provide the architecture specific data interfaces for Socket Interface specified in Table 10-17, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-17 libc - Socket Interface Data Interfaces

in6addr_any(GLIBC_2.1) [SUSv3]	in6addr_loopback(GLIBC_2.1) [SUSv3]		

10.2.10 Wide Characters

10.2.10.1 Interfaces for Wide Characters

An LSB conforming implementation shall provide the architecture specific functions for Wide Characters specified in Table 10-18, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-18 libc - Wide Characters Function Interfaces

__wcstod_internal(GLIBC_2.0) [LSB]	__wcstof_internal(GLIBC_2.0) [LSB]	__wcstol_internal(GLIBC_2.0) [LSB]	__wcstold_internal(GLIBC_2.0) [LSB]

© 2021 ISO/IEC – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

__wcstoul_internal(GLIBC_2.0) [LSB]	btowc(GLIBC_2.0) [SUSv4]	fgetwc(GLIBC_2.2) [SUSv4]	fgetws(GLIBC_2.2) [SUSv4]
fputwc(GLIBC_2.2) [SUSv4]	fputws(GLIBC_2.2) [SUSv4]	fwide(GLIBC_2.2) [SUSv4]	fwprintf(GLIBC_2.2) [SUSv4]
fwscanf(GLIBC_2.2) [LSB]	getwc(GLIBC_2.2) [SUSv4]	getwchar(GLIBC_2.2) [SUSv4]	mblen(GLIBC_2.0) [SUSv4]
mbrlen(GLIBC_2.0) [SUSv4]	mbrtowc(GLIBC_2.0) [SUSv4]	mbsinit(GLIBC_2.0) [SUSv4]	mbsnrtowcs(GLIBC_2.0) [SUSv4]
mbsrtowcs(GLIBC_2.0) [SUSv4]	mbstowcs(GLIBC_2.0) [SUSv4]	mbtowc(GLIBC_2.0) [SUSv4]	putwc(GLIBC_2.2) [SUSv4]
putwchar(GLIBC_2.2) [SUSv4]	swprintf(GLIBC_2.2) [SUSv4]	swscanf(GLIBC_2.2) [LSB]	towctrans(GLIBC_2.0) [SUSv4]
towlower(GLIBC_2.0) [SUSv4]	towupper(GLIBC_2.0) [SUSv4]	ungetwc(GLIBC_2.2) [SUSv4]	vfwprintf(GLIBC_2.2) [SUSv4]
vfwscanf(GLIBC_2.2) [LSB]	vswprintf(GLIBC_2.2) [SUSv4]	vswscanf(GLIBC_2.2) [LSB]	vwprintf(GLIBC_2.2) [SUSv4]
vwscanf(GLIBC_2.2) [LSB]	wcpcpy(GLIBC_2.0) [SUSv4]	wcpncpy(GLIBC_2.0) [SUSv4]	wcrtomb(GLIBC_2.0) [SUSv4]
wcscasecmp(GLIBC_2.1) [SUSv4]	wcscat(GLIBC_2.0) [SUSv4]	wcschr(GLIBC_2.0) [SUSv4]	wcscmp(GLIBC_2.0) [SUSv4]
wcscoll(GLIBC_2.0) [SUSv4]	wcscpy(GLIBC_2.0) [SUSv4]	wcscspn(GLIBC_2.0) [SUSv4]	wcsdup(GLIBC_2.0) [SUSv4]
wcsftime(GLIBC_2.2) [SUSv4]	wcslen(GLIBC_2.0) [SUSv4]	wcsncasecmp(GLIBC_2.1) [SUSv4]	wcsncat(GLIBC_2.0) [SUSv4]
wcsncmp(GLIBC_2.0) [SUSv4]	wcsncpy(GLIBC_2.0) [SUSv4]	wcsnlen(GLIBC_2.1) [SUSv4]	wcsnrtombs(GLIBC_2.0) [SUSv4]
wcspbrk(GLIBC_2.0) [SUSv4]	wcsrchr(GLIBC_2.0) [SUSv4]	wcsrtombs(GLIBC_2.0) [SUSv4]	wcsspn(GLIBC_2.0) [SUSv4]
wcsstr(GLIBC_2.0) [SUSv4]	wcstod(GLIBC_2.0) [SUSv4]	wcstof(GLIBC_2.0) [SUSv4]	wcstoimax(GLIBC_2.1) [SUSv4]
wcstok(GLIBC_2.0) [SUSv4]	wcstol(GLIBC_2.0) [SUSv4]	wcstold(GLIBC_2.0) [SUSv4]	wcstoll(GLIBC_2.1) [SUSv4]
wcstombs(GLIBC_2.0) [SUSv4]	wcstoq(GLIBC_2.0) [LSB]	wcstoul(GLIBC_2.0) [SUSv4]	wcstoull(GLIBC_2.1) [SUSv4]
wcstoumax(GLIBC_2.1) [SUSv4]	wcstouq(GLIBC_2.0) [LSB]	wcswcs(GLIBC_2.1) [SUSv3]	wcswidth(GLIBC_2.0) [SUSv4]
wcsxfrm(GLIBC_2.0) [SUSv4]	wctob(GLIBC_2.0) [SUSv4]	wctomb(GLIBC_2.0) [SUSv4]	wctrans(GLIBC_2.0) [SUSv4]
wctype(GLIBC_2.0) [SUSv4]	wcwidth(GLIBC_2.0) [SUSv4]	wmemchr(GLIBC_2.0) [SUSv4]	wmemcmp(GLIBC_2.0) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

wmemcpy(GLIBC_2.0) [SUSv4]	wmemmove(GLIBC_2.0) [SUSv4]	wmemset(GLIBC_2.0) [SUSv4]	wprintf(GLIBC_2.2) [SUSv4]
wscanf(GLIBC_2.2) [LSB]			

10.2.11 String Functions

10.2.11.1 Interfaces for String Functions

An LSB conforming implementation shall provide the architecture specific functions for String Functions specified in Table 10-19, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-19 libc - String Functions Function Interfaces

__mempcpy(GLIBC_2.0) [LSB]	__rawmemchr(GLIBC_2.1) [LSB]	__stpcpy(GLIBC_2.0) [LSB]	__strdup(GLIBC_2.0) [LSB]
__strtod_internal(GLIBC_2.0) [LSB]	__strtof_internal(GLIBC_2.0) [LSB]	__strtok_r(GLIBC_2.0) [LSB]	__strtol_internal(GLIBC_2.0) [LSB]
__strtold_internal(GLIBC_2.0) [LSB]	__strtoll_internal(GLIBC_2.0) [LSB]	__strtoul_internal(GLIBC_2.0) [LSB]	__strtoull_internal(GLIBC_2.0) [LSB]
__xpg_strerror_r(GLIBC_2.3.4) [LSB]	bcmp(GLIBC_2.0) [SUSv3]	bcopy(GLIBC_2.0) [SUSv3]	bzero(GLIBC_2.0) [SUSv3]
ffs(GLIBC_2.0) [SUSv4]	index(GLIBC_2.0) [SUSv3]	memccpy(GLIBC_2.0) [SUSv4]	memchr(GLIBC_2.0) [SUSv4]
memcmp(GLIBC_2.0) [SUSv4]	memcpy(GLIBC_2.0) [SUSv4]	memmove(GLIBC_2.0) [SUSv4]	memrchr(GLIBC_2.2) [LSB]
memset(GLIBC_2.0) [SUSv4]	rindex(GLIBC_2.0) [SUSv3]	stpcpy(GLIBC_2.0) [SUSv4]	stpncpy(GLIBC_2.0) [SUSv4]
strcasecmp(GLIBC_2.0) [SUSv4]	strcasestr(GLIBC_2.1) [LSB]	strcat(GLIBC_2.0) [SUSv4]	strchr(GLIBC_2.0) [SUSv4]
strcmp(GLIBC_2.0) [SUSv4]	strcoll(GLIBC_2.0) [SUSv4]	strcpy(GLIBC_2.0) [SUSv4]	strcspn(GLIBC_2.0) [SUSv4]
strdup(GLIBC_2.0) [SUSv4]	strerror(GLIBC_2.0) [SUSv4]	strerror_r(GLIBC_2.0) [LSB]	strfmon(GLIBC_2.0) [SUSv4]
strftime(GLIBC_2.0) [SUSv4]	strlen(GLIBC_2.0) [SUSv4]	strncasecmp(GLIBC_2.0) [SUSv4]	strncat(GLIBC_2.0) [SUSv4]
strncmp(GLIBC_2.0) [SUSv4]	strncpy(GLIBC_2.0) [SUSv4]	strndup(GLIBC_2.0) [SUSv4]	strnlen(GLIBC_2.0) [SUSv4]
strpbrk(GLIBC_2.0) [SUSv4]	strptime(GLIBC_2.0) [LSB]	strrchr(GLIBC_2.0) [SUSv4]	strsep(GLIBC_2.0) [LSB]
strsignal(GLIBC_2.0) [SUSv4]	strspn(GLIBC_2.0) [SUSv4]	strstr(GLIBC_2.0) [SUSv4]	strtof(GLIBC_2.0) [SUSv4]

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

strtoimax(GLIBC_2.1) [SUSv4]	strtok(GLIBC_2.0) [SUSv4]	strtok_r(GLIBC_2.0) [SUSv4]	strtold(GLIBC_2.0) [SUSv4]
strtoll(GLIBC_2.0) [SUSv4]	strtoq(GLIBC_2.0) [LSB]	strtoull(GLIBC_2.0) [SUSv4]	strtoumax(GLIBC_2.1) [SUSv4]
strtouq(GLIBC_2.0) [LSB]	strxfrm(GLIBC_2.0) [SUSv4]	swab(GLIBC_2.0) [SUSv4]	

An LSB conforming implementation shall provide the architecture specific deprecated functions for String Functions specified in Table 10-20, with the full mandatory functionality as described in the referenced underlying specification.

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

Table 10-20 libc - String Functions Deprecated Function Interfaces

strerror_r(GLIBC_2.0) [LSB]			

10.2.12 IPC Functions

10.2.12.1 Interfaces for IPC Functions

An LSB conforming implementation shall provide the architecture specific functions for IPC Functions specified in Table 10-21, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-21 libc - IPC Functions Function Interfaces

ftok(GLIBC_2.0) [SUSv4]	msgctl(GLIBC_2.2) [SUSv4]	msgget(GLIBC_2.0) [SUSv4]	msgrcv(GLIBC_2.0) [SUSv4]
msgsnd(GLIBC_2.0) [SUSv4]	semctl(GLIBC_2.2) [SUSv4]	semget(GLIBC_2.0) [SUSv4]	semop(GLIBC_2.0) [SUSv4]
shmat(GLIBC_2.0) [SUSv4]	shmctl(GLIBC_2.2) [SUSv4]	shmdt(GLIBC_2.0) [SUSv4]	shmget(GLIBC_2.0) [SUSv4]

10.2.13 Regular Expressions

10.2.13.1 Interfaces for Regular Expressions

An LSB conforming implementation shall provide the architecture specific functions for Regular Expressions specified in Table 10-22, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-22 libc - Regular Expressions Function Interfaces

regcomp(GLIBC_2.0) [SUSv4]	regerror(GLIBC_2.0) [SUSv4]	regexec(GLIBC_2.3.4) [LSB]	regfree(GLIBC_2.0) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

10.2.14 Character Type Functions

10.2.14.1 Interfaces for Character Type Functions

An LSB conforming implementation shall provide the architecture specific functions for Character Type Functions specified in Table 10-23, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-23 libc - Character Type Functions Function Interfaces

__ctype_get_mb_cur_max(GLIBC_2.0) [LSB]	_tolower(GLIBC_2.0) [SUSv4]	_toupper(GLIBC_2.0) [SUSv4]	isalnum(GLIBC_2.0) [SUSv4]
isalpha(GLIBC_2.0) [SUSv4]	isascii(GLIBC_2.0) [SUSv4]	iscntrl(GLIBC_2.0) [SUSv4]	isdigit(GLIBC_2.0) [SUSv4]
isgraph(GLIBC_2.0) [SUSv4]	islower(GLIBC_2.0) [SUSv4]	isprint(GLIBC_2.0) [SUSv4]	ispunct(GLIBC_2.0) [SUSv4]
isspace(GLIBC_2.0) [SUSv4]	isupper(GLIBC_2.0) [SUSv4]	iswalnum(GLIBC_2.0) [SUSv4]	iswalpha(GLIBC_2.0) [SUSv4]
iswblank(GLIBC_2.1) [SUSv4]	iswcntrl(GLIBC_2.0) [SUSv4]	iswctype(GLIBC_2.0) [SUSv4]	iswdigit(GLIBC_2.0) [SUSv4]
iswgraph(GLIBC_2.0) [SUSv4]	iswlower(GLIBC_2.0) [SUSv4]	iswprint(GLIBC_2.0) [SUSv4]	iswpunct(GLIBC_2.0) [SUSv4]
iswspace(GLIBC_2.0) [SUSv4]	iswupper(GLIBC_2.0) [SUSv4]	iswxdigit(GLIBC_2.0) [SUSv4]	isxdigit(GLIBC_2.0) [SUSv4]
toascii(GLIBC_2.0) [SUSv4]	tolower(GLIBC_2.0) [SUSv4]	toupper(GLIBC_2.0) [SUSv4]	

10.2.15 Time Manipulation

10.2.15.1 Interfaces for Time Manipulation

An LSB conforming implementation shall provide the architecture specific functions for Time Manipulation specified in Table 10-24, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-24 libc - Time Manipulation Function Interfaces

adjtime(GLIBC_2.0) [LSB]	asctime(GLIBC_2.0) [SUSv4]	asctime_r(GLIBC_2.0) [SUSv4]	ctime(GLIBC_2.0) [SUSv4]
ctime_r(GLIBC_2.0) [SUSv4]	difftime(GLIBC_2.0) [SUSv4]	gmtime(GLIBC_2.0) [SUSv4]	gmtime_r(GLIBC_2.0) [SUSv4]
localtime(GLIBC_2.0) [SUSv4]	localtime_r(GLIBC_2.0) [SUSv4]	mktime(GLIBC_2.0) [SUSv4]	tzset(GLIBC_2.0) [SUSv4]
ualarm(GLIBC_2.0) [SUSv3]			

An LSB conforming implementation shall provide the architecture specific data interfaces for Time Manipulation specified in Table 10-25, with the full mandatory functionality as described in the referenced underlying specification.

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

Table 10-25 libc - Time Manipulation Data Interfaces

__daylight(GLIBC_2.0) [LSB]	__timezone(GLIBC_2.0) [LSB]	__tzname(GLIBC_2.0) [LSB]	daylight(GLIBC_2.0) [SUSv4]
timezone(GLIBC_2.0) [SUSv4]	tzname(GLIBC_2.0) [SUSv4]		

10.2.16 Terminal Interface Functions

10.2.16.1 Interfaces for Terminal Interface Functions

An LSB conforming implementation shall provide the architecture specific functions for Terminal Interface Functions specified in Table 10-26, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-26 libc - Terminal Interface Functions Function Interfaces

cfgetispeed(GLIBC_2.0) [SUSv4]	cfgetospeed(GLIBC_2.0) [SUSv4]	cfmakeraw(GLIBC_2.0) [LSB]	cfsetispeed(GLIBC_2.0) [SUSv4]
cfsetospeed(GLIBC_2.0) [SUSv4]	cfsetspeed(GLIBC_2.0) [LSB]	tcdrain(GLIBC_2.0) [SUSv4]	tcflow(GLIBC_2.0) [SUSv4]
tcflush(GLIBC_2.0) [SUSv4]	tcgetattr(GLIBC_2.0) [SUSv4]	tcgetpgrp(GLIBC_2.0) [SUSv4]	tcgetsid(GLIBC_2.1) [SUSv4]
tcsendbreak(GLIBC_2.0) [SUSv4]	tcsetattr(GLIBC_2.0) [SUSv4]	tcsetpgrp(GLIBC_2.0) [SUSv4]	

10.2.17 System Database Interface

10.2.17.1 Interfaces for System Database Interface

An LSB conforming implementation shall provide the architecture specific functions for System Database Interface specified in Table 10-27, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-27 libc - System Database Interface Function Interfaces

endgrent(GLIBC_2.0) [SUSv4]	endprotoent(GLIBC_2.0) [SUSv4]	endpwent(GLIBC_2.0) [SUSv4]	endservent(GLIBC_2.0) [SUSv4]
endutent(GLIBC_2.0) [LSB]	endutxent(GLIBC_2.1) [SUSv4]	getgrent(GLIBC_2.0) [SUSv4]	getgrent_r(GLIBC_2.1.2) [LSB]
getgrgid(GLIBC_2.0) [SUSv4]	getgrgid_r(GLIBC_2.1.2) [SUSv4]	getgrnam(GLIBC_2.0) [SUSv4]	getgrnam_r(GLIBC_2.1.2) [SUSv4]
getgrouplist(GLIBC_2.2.4) [LSB]	gethostbyaddr(GLIBC_2.0) [SUSv3]	gethostbyaddr_r(GLIBC_2.1.2) [LSB]	gethostbyname(GLIBC_2.0) [SUSv3]
gethostbyname2(GLIBC_2.0) [LSB]	gethostbyname2_r(GLIBC_2.1.2) [LSB]	gethostbyname_r(GLIBC_2.1.2) [LSB]	getprotobyname(GLIBC_2.0) [SUSv4]

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021

getprotobyname_r(GLIBC_2.1.2) [LSB]	getprotobynumber(GLIBC_2.0) [SUSv4]	getprotobynumber_r(GLIBC_2.1.2) [LSB]	getprotoent(GLIBC_2.0) [SUSv4]
getprotoent_r(GLIBC_2.1.2) [LSB]	getpwent(GLIBC_2.0) [SUSv4]	getpwent_r(GLIBC_2.1.2) [LSB]	getpwnam(GLIBC_2.0) [SUSv4]
getpwnam_r(GLIBC_2.1.2) [SUSv4]	getpwuid(GLIBC_2.0) [SUSv4]	getpwuid_r(GLIBC_2.1.2) [SUSv4]	getservbyname(GLIBC_2.0) [SUSv4]
getservbyname_r(GLIBC_2.1.2) [LSB]	getservbyport(GLIBC_2.0) [SUSv4]	getservbyport_r(GLIBC_2.1.2) [LSB]	getservent(GLIBC_2.0) [SUSv4]
getservent_r(GLIBC_2.1.2) [LSB]	getutent(GLIBC_2.0) [LSB]	getutent_r(GLIBC_2.0) [LSB]	getutxent(GLIBC_2.1) [SUSv4]
getutxid(GLIBC_2.1) [SUSv4]	getutxline(GLIBC_2.1) [SUSv4]	pututxline(GLIBC_2.1) [SUSv4]	setgrent(GLIBC_2.0) [SUSv4]
setgroups(GLIBC_2.0) [LSB]	setprotoent(GLIBC_2.0) [SUSv4]	setpwent(GLIBC_2.0) [SUSv4]	setservent(GLIBC_2.0) [SUSv4]
setutent(GLIBC_2.0) [LSB]	setutxent(GLIBC_2.1) [SUSv4]	utmpname(GLIBC_2.0) [LSB]	

An LSB conforming implementation shall provide the architecture specific deprecated functions for System Database Interface specified in Table 10-28, with the full mandatory functionality as described in the referenced underlying specification.

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

Table 10-28 libc - System Database Interface Deprecated Function Interfaces

gethostbyaddr(GLIBC_2.0) [SUSv3]	gethostbyaddr_r(GLIBC_2.1.2) [LSB]	gethostbyname(GLIBC_2.0) [SUSv3]	gethostbyname2(GLIBC_2.0) [LSB]
gethostbyname2_r(GLIBC_2.1.2) [LSB]	gethostbyname_r(GLIBC_2.1.2) [LSB]		

10.2.18 Language Support

10.2.18.1 Interfaces for Language Support

An LSB conforming implementation shall provide the architecture specific functions for Language Support specified in Table 10-29, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-29 libc - Language Support Function Interfaces

__libc_start_main(GLIBC_2.0) [LSB]			

Documents
Document Sharing Hub

© 2021 ISO/IEC – All rights reserved

Full document details: BS ISO/IEC 23360-2-2:2021

10.2.19 Large File Support

10.2.19.1 Interfaces for Large File Support

An LSB conforming implementation shall provide the architecture specific functions for Large File Support specified in Table 10-30, with the full mandatory functionality as described in the referenced underlying specification.

Table 10-30 libc - Large File Support Function Interfaces

__fxstat64(GLIBC_2.2) [LSB]	__lxstat64(GLIBC_2.2) [LSB]	__xstat64(GLIBC_2.2) [LSB]	creat64(GLIBC_2.1) [LFS]
fgetpos64(GLIBC_2.2) [LFS]	fopen64(GLIBC_2.1) [LFS]	freopen64(GLIBC_2.1) [LFS]	fseeko64(GLIBC_2.1) [LFS]
fsetpos64(GLIBC_2.2) [LFS]	fstatfs64(GLIBC_2.1) [LSB]	fstatvfs64(GLIBC_2.1) [LFS]	ftello64(GLIBC_2.1) [LFS]
ftruncate64(GLIBC_2.1) [LFS]	ftw64(GLIBC_2.1) [LFS]	getrlimit64(GLIBC_2.2) [LFS]	lockf64(GLIBC_2.1) [LFS]
lseek64(GLIBC_2.1) [LFS]	mkstemp64(GLIBC_2.2) [LSB]	mmap64(GLIBC_2.1) [LFS]	nftw64(GLIBC_2.3.3) [LFS]
open64(GLIBC_2.1) [LFS]	posix_fadvise64(GLIBC_2.3.3) [LSB]	posix_fallocate64(GLIBC_2.3.3) [LSB]	pread64(GLIBC_2.1) [LSB]
pwrite64(GLIBC_2.1) [LSB]	readdir64(GLIBC_2.2) [LFS]	readdir64_r(GLIBC_2.2) [LSB]	statfs64(GLIBC_2.1) [LSB]
statvfs64(GLIBC_2.1) [LFS]	tmpfile64(GLIBC_2.1) [LFS]	truncate64(GLIBC_2.1) [LFS]	

An LSB conforming implementation shall provide the architecture specific deprecated functions for Large File Support specified in Table 10-31, with the full mandatory functionality as described in the referenced underlying specification.

> **Note:** These interfaces are deprecated, and applications should avoid using them. These interfaces may be withdrawn in future releases of this specification.

Table 10-31 libc - Large File Support Deprecated Function Interfaces

fstatfs64(GLIBC_2.1) [LSB]	statfs64(GLIBC_2.1) [LSB]		

10.2.20 Inotify

10.2.20.1 Interfaces for Inotify

No external functions are defined for libc - Inotify in this part of the specification. See also the generic specification.

© 2021 ISO/IEC – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23360-2-2:2021