



## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'unistd.h.0p' command***

**\$ man unistd.h.0p**

unistd.h(0P)          POSIX Programmer's Manual          unistd.h(0P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

unistd.h ? standard symbolic constants and types

### SYNOPSIS

```
#include <unistd.h>
```

### DESCRIPTION

The <unistd.h> header defines miscellaneous symbolic constants and types, and declares miscellaneous functions. The actual values of the constants are unspecified except as shown. The contents of this header are shown below.

#### Version Test Macros

The <unistd.h> header shall define the following symbolic constants.

The values shall be suitable for use in #if preprocessing directives.

#### `_POSIX_VERSION`

Integer value indicating version of this standard (C-language binding) to which the implementation conforms. For implementations conforming to POSIX.1?2008, the value shall be 200809L.

#### `_POSIX2_VERSION`

Integer value indicating version of the Shell and Utilities volume of POSIX.1 to which the implementation conforms. For implementations conforming to POSIX.1-2008, the value shall be 200809L. For profile implementations that define `_POSIX_SUBPROFILE` (see Section 2.1.5.1, Subprofiling Considerations) in `<unistd.h>`, `_POSIX2_VERSION` may be left undefined or be defined with the value -1 to indicate that the Shell and Utilities volume of POSIX.1 is not supported. In this case, a call to `sysconf(_SC_2_VERSION)` shall return either 200809L or -1 indicating that the Shell and Utilities volume of POSIX.1 is or is not, respectively, supported at runtime.

The `<unistd.h>` header shall define the following symbolic constant only if the implementation supports the XSI option; see Section 2.1.4, XSI Conformance. If defined, its value shall be suitable for use in `#if` preprocessing directives.

#### `_XOPEN_VERSION`

Integer value indicating version of the X/Open Portability Guide to which the implementation conforms. The value shall be 700.

#### Constants for Options and Option Groups

The following symbolic constants, if defined in `<unistd.h>`, shall have a value of -1, 0, or greater, unless otherwise specified below. For profile implementations that define `_POSIX_SUBPROFILE` (see Section 2.1.5.1, Subprofiling Considerations) in `<unistd.h>`, constants described below as always having a value greater than zero need not be defined and, if defined, may have a value of -1, 0, or greater. The values shall be suitable for use in `#if` preprocessing directives.

If a symbolic constant is not defined or is defined with the value -1, the option is not supported for compilation. If it is defined with a value greater than zero, the option shall always be supported when the application is executed. If it is defined with the value zero, the option shall be supported for compilation and might or might not be supported at runtime. See Section 2.1.6, Options for further information about the conformance requirements of these three categories of sup-

port.

#### `_POSIX_ADVISORY_INFO`

The implementation supports the Advisory Information option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_ASYNCHRONOUS_IO`

The implementation supports asynchronous input and output. This symbol shall always be set to the value 200809L.

#### `_POSIX_BARRIERS`

The implementation supports barriers. This symbol shall always be set to the value 200809L.

#### `_POSIX_CHOWN_RESTRICTED`

The use of `chown()` and `fchown()` is restricted to a process with appropriate privileges, and to changing the group ID of a file only to the effective group ID of the process or to one of its supplementary group IDs. This symbol shall be defined with a value other than -1.

#### `_POSIX_CLOCK_SELECTION`

The implementation supports clock selection. This symbol shall always be set to the value 200809L.

#### `_POSIX_CPUTIME`

The implementation supports the Process CPU-Time Clocks option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_FSYNC`

The implementation supports the File Synchronization option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_IPV6`

The implementation supports the IPv6 option. If this symbol is

defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_JOB_CONTROL`

The implementation supports job control. This symbol shall always be set to a value greater than zero.

#### `_POSIX_MAPPED_FILES`

The implementation supports memory mapped Files. This symbol shall always be set to the value 200809L.

#### `_POSIX_MEMLOCK`

The implementation supports the Process Memory Locking option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_MEMLOCK_RANGE`

The implementation supports the Range Memory Locking option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_MEMORY_PROTECTION`

The implementation supports memory protection. This symbol shall always be set to the value 200809L.

#### `_POSIX_MESSAGE_PASSING`

The implementation supports the Message Passing option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_MONOTONIC_CLOCK`

The implementation supports the Monotonic Clock option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_NO_TRUNC`

Pathname components longer than {NAME\_MAX} generate an error.

This symbol shall be defined with a value other than -1.

#### `_POSIX_PRIORITIZED_IO`

The implementation supports the Prioritized Input and Output option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_PRIORITY_SCHEDULING`

The implementation supports the Process Scheduling option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_RAW_SOCKETS`

The implementation supports the Raw Sockets option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

#### `_POSIX_READER_WRITER_LOCKS`

The implementation supports read-write locks. This symbol shall always be set to the value 200809L.

#### `_POSIX_REALTIME_SIGNALS`

The implementation supports realtime signals. This symbol shall always be set to the value 200809L.

#### `_POSIX_REGEX`

The implementation supports the Regular Expression Handling option. This symbol shall always be set to a value greater than zero.

#### `_POSIX_SAVED_IDS`

Each process has a saved set-user-ID and a saved set-group-ID. This symbol shall always be set to a value greater than zero.

#### `_POSIX_SEMAPHORES`

The implementation supports semaphores. This symbol shall always be set to the value 200809L.

## `_POSIX_SHARED_MEMORY_OBJECTS`

The implementation supports the Shared Memory Objects option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_SHELL`

The implementation supports the POSIX shell. This symbol shall always be set to a value greater than zero.

## `_POSIX_SPAWN`

The implementation supports the Spawn option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_SPIN_LOCKS`

The implementation supports spin locks. This symbol shall always be set to the value 200809L.

## `_POSIX_SPORADIC_SERVER`

The implementation supports the Process Sporadic Server option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_SYNCHRONIZED_IO`

The implementation supports the Synchronized Input and Output option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_THREAD_ATTR_STACKADDR`

The implementation supports the Thread Stack Address Attribute option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_THREAD_ATTR_STACKSIZE`

The implementation supports the Thread Stack Size Attribute option.

tion. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_CPUTIME`

The implementation supports the Thread CPU-Time Clocks option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_PRIO_INHERIT`

The implementation supports the Non-Robust Mutex Priority Inheritance option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_PRIO_PROTECT`

The implementation supports the Non-Robust Mutex Priority Protection option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_PRIORITY_SCHEDULING`

The implementation supports the Thread Execution Scheduling option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_PROCESS_SHARED`

The implementation supports the Thread Process-Shared Synchronization option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_THREAD_ROBUST_PRIO_INHERIT`

The implementation supports the Robust Mutex Priority Inheritance option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

## `_POSIX_THREAD_ROBUST_PRIO_PROTECT`

The implementation supports the Robust Mutex Priority Protection option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_THREAD_SAFE_FUNCTIONS`

The implementation supports thread-safe functions. This symbol shall always be set to the value 200809L.

## `_POSIX_THREAD_SPORADIC_SERVER`

The implementation supports the Thread Sporadic Server option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_THREADS`

The implementation supports threads. This symbol shall always be set to the value 200809L.

## `_POSIX_TIMEOUTS`

The implementation supports timeouts. This symbol shall always be set to the value 200809L.

## `_POSIX_TIMERS`

The implementation supports timers. This symbol shall always be set to the value 200809L.

## `_POSIX_TRACE`

The implementation supports the Trace option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_TRACE_EVENT_FILTER`

The implementation supports the Trace Event Filter option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX_TRACE_INHERIT`

The implementation supports the Trace Inherit option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_TRACE_LOG`

The implementation supports the Trace Log option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_TYPED_MEMORY_OBJECTS`

The implementation supports the Typed Memory Objects option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX_V6_ILP32_OFF32`

The implementation provides a C-language compilation environment with 32-bit int, long, pointer, and off\_t types.

#### `_POSIX_V6_ILP32_OFFBIG`

The implementation provides a C-language compilation environment with 32-bit int, long, and pointer types and an off\_t type using at least 64 bits.

#### `_POSIX_V6_LP64_OFF64`

The implementation provides a C-language compilation environment with 32-bit int and 64-bit long, pointer, and off\_t types.

#### `_POSIX_V6_LPBIG_OFFBIG`

The implementation provides a C-language compilation environment with an int type using at least 32 bits and long, pointer, and off\_t types using at least 64 bits.

#### `_POSIX_V7_ILP32_OFF32`

The implementation provides a C-language compilation environment with 32-bit int, long, pointer, and off\_t types.

#### `_POSIX_V7_ILP32_OFFBIG`

The implementation provides a C-language compilation environment

with 32-bit int, long, and pointer types and an off\_t type using at least 64 bits.

#### `_POSIX_V7_LP64_OFF64`

The implementation provides a C-language compilation environment with 32-bit int and 64-bit long, pointer, and off\_t types.

#### `_POSIX_V7_LPBIG_OFFBIG`

The implementation provides a C-language compilation environment with an int type using at least 32 bits and long, pointer, and off\_t types using at least 64 bits.

#### `_POSIX2_C_BIND`

The implementation supports the C-Language Binding option. This symbol shall always have the value 200809L.

#### `_POSIX2_C_DEV`

The implementation supports the C-Language Development Utilities option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_CHAR_TERM`

The implementation supports the Terminal Characteristics option. The value of this symbol reported by sysconf() shall either be -1 or a value greater than zero.

#### `_POSIX2_FORT_DEV`

The implementation supports the FORTRAN Development Utilities option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_FORT_RUN`

The implementation supports the FORTRAN Runtime Utilities option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_LOCALEDEF`

The implementation supports the creation of locales by the lo?

caledef utility. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS`

The implementation supports the Batch Environment Services and Utilities option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS_ACCOUNTING`

The implementation supports the Batch Accounting option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS_CHECKPOINT`

The implementation supports the Batch Checkpoint/Restart option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS_LOCATE`

The implementation supports the Locate Batch Job Request option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS_MESSAGE`

The implementation supports the Batch Job Message Request option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

#### `_POSIX2_PBS_TRACK`

The implementation supports the Track Batch Job Request option. If this symbol is defined in <unistd.h>, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by sysconf() shall either be -1 or 200809L.

## `_POSIX2_SW_DEV`

The implementation supports the Software Development Utilities option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_POSIX2_UPE`

The implementation supports the User Portability Utilities option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall either be -1 or 200809L.

## `_XOPEN_CRYPT`

The implementation supports the X/Open Encryption Option Group.

## `_XOPEN_ENH_I18N`

The implementation supports the Issue 4, Version 2 Enhanced Internationalization Option Group. This symbol shall always be set to a value other than -1.

## `_XOPEN_REALTIME`

The implementation supports the X/Open Realtime Option Group.

## `_XOPEN_REALTIME_THREADS`

The implementation supports the X/Open Realtime Threads Option Group.

## `_XOPEN_SHM`

The implementation supports the Issue 4, Version 2 Shared Memory Option Group. This symbol shall always be set to a value other than -1.

## `_XOPEN_STREAMS`

The implementation supports the XSI STREAMS Option Group.

## `_XOPEN_UNIX`

The implementation supports the XSI option.

## `_XOPEN_UUCP`

The implementation supports the UUCP Utilities option. If this symbol is defined in `<unistd.h>`, it shall be defined to be -1, 0, or 200809L. The value of this symbol reported by `sysconf()` shall

be either -1 or 200809L.

## Execution-Time Symbolic Constants

If any of the following symbolic constants are not defined in the `<unistd.h>` header, the value shall vary depending on the file to which it is applied. If defined, they shall have values suitable for use in `#if` preprocessing directives.

If any of the following symbolic constants are defined to have value -1 in the `<unistd.h>` header, the implementation shall not provide the option on any file; if any are defined to have a value other than -1 in the `<unistd.h>` header, the implementation shall provide the option on all applicable files.

All of the following values, whether defined as symbolic constants in `<unistd.h>` or not, may be queried with respect to a specific file using the `pathconf()` or `fpathconf()` functions:

### `_POSIX_ASYNC_IO`

Asynchronous input or output operations may be performed for the associated file.

### `_POSIX_PRIO_IO`

Prioritized input or output operations may be performed for the associated file.

### `_POSIX_SYNC_IO`

Synchronized input or output operations may be performed for the associated file.

If the following symbolic constants are defined in the `<unistd.h>` header, they apply to files and all paths in all file systems on the implementation:

### `_POSIX_TIMESTAMP_RESOLUTION`

The resolution in nanoseconds for all file timestamps.

### `_POSIX2_SYMLINKS`

Symbolic links can be created.

## Constants for Functions

The `<unistd.h>` header shall define `NULL` as described in `<stddef.h>`.

The `<unistd.h>` header shall define the following symbolic constants for

use with the `access()` function. The values shall be suitable for use in `#if` preprocessing directives.

`F_OK` Test for existence of file.

`R_OK` Test for read permission.

`W_OK` Test for write permission.

`X_OK` Test for execute (search) permission.

The constants `F_OK`, `R_OK`, `W_OK`, and `X_OK` and the expressions `R_OK|W_OK`, `R_OK|X_OK`, and `R_OK|W_OK|X_OK` shall all have distinct values.

The `<unistd.h>` header shall define the following symbolic constants for the `confstr()` function:

#### `_CS_PATH`

This is the value for the `PATH` environment variable that finds all of the standard utilities that are provided in a manner accessible via the `exec` family of functions.

#### `_CS_POSIX_V7_ILP32_OFF32_CFLAGS`

If `sysconf(_SC_V7_ILP32_OFF32)` returns `-1`, the meaning of this value is unspecified. Otherwise, this value is the set of initial options to be given to the `c99` utility to build an application using a programming model with 32-bit `int`, `long`, `pointer`, and `off_t` types.

#### `_CS_POSIX_V7_ILP32_OFF32_LDFLAGS`

If `sysconf(_SC_V7_ILP32_OFF32)` returns `-1`, the meaning of this value is unspecified. Otherwise, this value is the set of final options to be given to the `c99` utility to build an application using a programming model with 32-bit `int`, `long`, `pointer`, and `off_t` types.

#### `_CS_POSIX_V7_ILP32_OFF32_LIBS`

If `sysconf(_SC_V7_ILP32_OFF32)` returns `-1`, the meaning of this value is unspecified. Otherwise, this value is the set of libraries to be given to the `c99` utility to build an application using a programming model with 32-bit `int`, `long`, `pointer`, and `off_t` types.

#### `_CS_POSIX_V7_ILP32_OFFBIG_CFLAGS`

If `sysconf(_SC_V7_ILP32_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of initial options to be given to the c99 utility to build an application using a programming model with 32-bit int, long, and pointer types, and an `off_t` type using at least 64 bits.

#### `_CS_POSIX_V7_ILP32_OFFBIG_LDFLAGS`

If `sysconf(_SC_V7_ILP32_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of final options to be given to the c99 utility to build an application using a programming model with 32-bit int, long, and pointer types, and an `off_t` type using at least 64 bits.

#### `_CS_POSIX_V7_ILP32_OFFBIG_LIBS`

If `sysconf(_SC_V7_ILP32_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of libraries to be given to the c99 utility to build an application using a programming model with 32-bit int, long, and pointer types, and an `off_t` type using at least 64 bits.

#### `_CS_POSIX_V7_LP64_OFF64_CFLAGS`

If `sysconf(_SC_V7_LP64_OFF64)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of initial options to be given to the c99 utility to build an application using a programming model with 32-bit int and 64-bit long, pointer, and `off_t` types.

#### `_CS_POSIX_V7_LP64_OFF64_LDFLAGS`

If `sysconf(_SC_V7_LP64_OFF64)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of final options to be given to the c99 utility to build an application using a programming model with 32-bit int and 64-bit long, pointer, and `off_t` types.

#### `_CS_POSIX_V7_LP64_OFF64_LIBS`

If `sysconf(_SC_V7_LP64_OFF64)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of libraries to be given to the c99 utility to build an application

using a programming model with 32-bit int and 64-bit long, pointer, and off\_t types.

#### `_CS_POSIX_V7_LPBIG_OFFBIG_CFLAGS`

If `sysconf(_SC_V7_LPBIG_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of initial options to be given to the c99 utility to build an application using a programming model with an int type using at least 32 bits and long, pointer, and off\_t types using at least 64 bits.

#### `_CS_POSIX_V7_LPBIG_OFFBIG_LDFLAGS`

If `sysconf(_SC_V7_LPBIG_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of final options to be given to the c99 utility to build an application using a programming model with an int type using at least 32 bits and long, pointer, and off\_t types using at least 64 bits.

#### `_CS_POSIX_V7_LPBIG_OFFBIG_LIBS`

If `sysconf(_SC_V7_LPBIG_OFFBIG)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of libraries to be given to the c99 utility to build an application using a programming model with an int type using at least 32 bits and long, pointer, and off\_t types using at least 64 bits.

#### `_CS_POSIX_V7_THREADS_CFLAGS`

If `sysconf(_SC_POSIX_THREADS)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of initial options to be given to the c99 utility to build a multi-threaded application. These flags are in addition to those associated with any of the other `_CS_POSIX_V7_*_CFLAGS` values used to specify particular type size programming environments.

#### `_CS_POSIX_V7_THREADS_LDFLAGS`

If `sysconf(_SC_POSIX_THREADS)` returns -1, the meaning of this value is unspecified. Otherwise, this value is the set of final options to be given to the c99 utility to build a multi-threaded application. These flags are in addition to those associated with any of the other `_CS_POSIX_V7_*_LDFLAGS` values used to specify

particular type size programming environments.

#### `_CS_POSIX_V7_WIDTH_RESTRICTED_ENVS`

This value is a <newline>-separated list of names of programming environments supported by the implementation in which the widths of the `blksize_t`, `cc_t`, `mode_t`, `nfds_t`, `pid_t`, `ptrdiff_t`, `size_t`, `speed_t`, `ssize_t`, `suseconds_t`, `tcflag_t`, `wchar_t`, and `wint_t` types are no greater than the width of type `long`. The format of each name shall be suitable for use with the `getconf -v` option.

#### `_CS_V7_ENV`

This is the value that provides the environment variable information (other than that provided by `_CS_PATH`) that is required by the implementation to create a conforming environment, as described in the implementation's conformance documentation.

The following symbolic constants are reserved for compatibility with Issue 6:

`_CS_POSIX_V6_ILP32_OFF32_CFLAGS`  
`_CS_POSIX_V6_ILP32_OFF32_LDFLAGS`  
`_CS_POSIX_V6_ILP32_OFF32_LIBS`  
`_CS_POSIX_V6_ILP32_OFFBIG_CFLAGS`  
`_CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS`  
`_CS_POSIX_V6_ILP32_OFFBIG_LIBS`  
`_CS_POSIX_V6_LP64_OFF64_CFLAGS`  
`_CS_POSIX_V6_LP64_OFF64_LDFLAGS`  
`_CS_POSIX_V6_LP64_OFF64_LIBS`  
`_CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS`  
`_CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS`  
`_CS_POSIX_V6_LPBIG_OFFBIG_LIBS`  
`_CS_POSIX_V6_WIDTH_RESTRICTED_ENVS`  
`_CS_V6_ENV`

The `<unistd.h>` header shall define `SEEK_CUR`, `SEEK_END`, and `SEEK_SET` as described in `<stdio.h>`.

The `<unistd.h>` header shall define the following symbolic constants as possible values for the function argument to the `lockf()` function:

F\_LOCK Lock a section for exclusive use.  
F\_TEST Test section for locks by other processes.  
F\_TLOCK Test and lock a section for exclusive use.  
F\_ULOCK Unlock locked sections.

The <unistd.h> header shall define the following symbolic constants for pathconf():

\_PC\_2\_SYMLINKS  
\_PC\_ALLOC\_SIZE\_MIN  
\_PC\_ASYNC\_IO  
\_PC\_CHOWN\_RESTRICTED  
\_PC\_FILESIZEBITS  
\_PC\_LINK\_MAX  
\_PC\_MAX\_CANON  
\_PC\_MAX\_INPUT  
\_PC\_NAME\_MAX  
\_PC\_NO\_TRUNC  
\_PC\_PATH\_MAX  
\_PC\_PIPE\_BUF  
\_PC\_PRIO\_IO  
\_PC\_REC\_INCR\_XFER\_SIZE  
\_PC\_REC\_MAX\_XFER\_SIZE  
\_PC\_REC\_MIN\_XFER\_SIZE  
\_PC\_REC\_XFER\_ALIGN  
\_PC\_SYMLINK\_MAX  
\_PC\_SYNC\_IO  
\_PC\_TIMESTAMP\_RESOLUTION  
\_PC\_VDISABLE

The <unistd.h> header shall define the following symbolic constants for sysconf():

\_SC\_2\_C\_BIND  
\_SC\_2\_C\_DEV  
\_SC\_2\_CHAR\_TERM  
\_SC\_2\_FORT\_DEV

\_SC\_2\_FORT\_RUN  
\_SC\_2\_LOCALEDEF  
\_SC\_2\_PBS  
\_SC\_2\_PBS\_ACCOUNTING  
\_SC\_2\_PBS\_CHECKPOINT  
\_SC\_2\_PBS\_LOCATE  
\_SC\_2\_PBS\_MESSAGE  
\_SC\_2\_PBS\_TRACK  
\_SC\_2\_SW\_DEV  
\_SC\_2\_UPE  
\_SC\_2\_VERSION  
\_SC\_ADVISORY\_INFO  
\_SC\_AIO\_LISTIO\_MAX  
\_SC\_AIO\_MAX  
\_SC\_AIO\_PRIO\_DELTA\_MAX  
\_SC\_ARG\_MAX  
\_SC\_ASYNCHRONOUS\_IO  
\_SC\_ATEXIT\_MAX  
\_SC\_BARRIERS  
\_SC\_BC\_BASE\_MAX  
\_SC\_BC\_DIM\_MAX  
\_SC\_BC\_SCALE\_MAX  
\_SC\_BC\_STRING\_MAX  
\_SC\_CHILD\_MAX  
\_SC\_CLK\_TCK  
\_SC\_CLOCK\_SELECTION  
\_SC\_COLL\_WEIGHTS\_MAX  
\_SC\_CPUTIME  
\_SC\_DELAYTIMER\_MAX  
\_SC\_EXPR\_NEST\_MAX  
\_SC\_FSYNC  
\_SC\_GETGR\_R\_SIZE\_MAX  
\_SC\_GETPW\_R\_SIZE\_MAX

\_SC\_HOST\_NAME\_MAX  
\_SC\_IOV\_MAX  
\_SC\_IPV6  
\_SC\_JOB\_CONTROL  
\_SC\_LINE\_MAX  
\_SC\_LOGIN\_NAME\_MAX  
\_SC\_MAPPED\_FILES  
\_SC\_MEMLOCK  
\_SC\_MEMLOCK\_RANGE  
\_SC\_MEMORY\_PROTECTION  
\_SC\_MESSAGE\_PASSING  
\_SC\_MONOTONIC\_CLOCK  
\_SC\_MQ\_OPEN\_MAX  
\_SC\_MQ\_PRIO\_MAX  
\_SC\_NGROUPS\_MAX  
\_SC\_OPEN\_MAX  
\_SC\_PAGE\_SIZE  
\_SC\_PAGESIZE  
\_SC\_PRIORITIZED\_IO  
\_SC\_PRIORITY\_SCHEDULING  
\_SC\_RAW\_SOCKETS  
\_SC\_RE\_DUP\_MAX  
\_SC\_READER\_WRITER\_LOCKS  
\_SC\_REALTIME\_SIGNALS  
\_SC\_REGEX  
\_SC\_RTSIG\_MAX  
\_SC\_SAVED\_IDS  
\_SC\_SEM\_NSEMS\_MAX  
\_SC\_SEM\_VALUE\_MAX  
\_SC\_SEMAPHORES  
\_SC\_SHARED\_MEMORY\_OBJECTS  
\_SC\_SHELL  
\_SC\_SIGQUEUE\_MAX

\_SC\_SPAWN  
\_SC\_SPIN\_LOCKS  
\_SC\_SPORADIC\_SERVER  
\_SC\_SS\_REPL\_MAX  
\_SC\_STREAM\_MAX  
\_SC\_SYMLoop\_MAX  
\_SC\_SYNCHRONIZED\_IO  
\_SC\_THREAD\_ATTR\_STACKADDR  
\_SC\_THREAD\_ATTR\_STACKSIZE  
\_SC\_THREAD\_CPUTIME  
\_SC\_THREAD\_DESTRUCTOR\_ITERATIONS  
\_SC\_THREAD\_KEYS\_MAX  
\_SC\_THREAD\_PRIO\_INHERIT  
\_SC\_THREAD\_PRIO\_PROTECT  
\_SC\_THREAD\_PRIORITY\_SCHEDULING  
\_SC\_THREAD\_PROCESS\_SHARED  
\_SC\_THREAD\_ROBUST\_PRIO\_INHERIT  
\_SC\_THREAD\_ROBUST\_PRIO\_PROTECT  
\_SC\_THREAD\_SAFE\_FUNCTIONS  
\_SC\_THREAD\_SPORADIC\_SERVER  
\_SC\_THREAD\_STACK\_MIN  
\_SC\_THREAD\_THREADS\_MAX  
\_SC\_THREADS  
\_SC\_TIMEOUTS  
\_SC\_TIMER\_MAX  
\_SC\_TIMERS  
\_SC\_TRACE  
\_SC\_TRACE\_EVENT\_FILTER  
\_SC\_TRACE\_EVENT\_NAME\_MAX  
\_SC\_TRACE\_INHERIT  
\_SC\_TRACE\_LOG  
\_SC\_TRACE\_NAME\_MAX  
\_SC\_TRACE\_SYS\_MAX

`_SC_TRACE_USER_EVENT_MAX`  
`_SC_TTY_NAME_MAX`  
`_SC_TYPED_MEMORY_OBJECTS`  
`_SC_TZNAME_MAX`  
`_SC_V7_ILP32_OFF32`  
`_SC_V7_ILP32_OFFBIG`  
`_SC_V7_LP64_OFF64`  
`_SC_V7_LPBIG_OFFBIG`  
`_SC_V6_ILP32_OFF32`  
`_SC_V6_ILP32_OFFBIG`  
`_SC_V6_LP64_OFF64`  
`_SC_V6_LPBIG_OFFBIG`  
`_SC_VERSION`  
`_SC_XOPEN_CRYPT`  
`_SC_XOPEN_ENH_I18N`  
`_SC_XOPEN_REALTIME`  
`_SC_XOPEN_REALTIME_THREADS`  
`_SC_XOPEN_SHM`  
`_SC_XOPEN_STREAMS`  
`_SC_XOPEN_UNIX`  
`_SC_XOPEN_UUCP`  
`_SC_XOPEN_VERSION`

The two constants `_SC_PAGESIZE` and `_SC_PAGE_SIZE` may be defined to have the same value.

The `<unistd.h>` header shall define the following symbolic constants for file streams:

`STDERR_FILENO` File number of `stderr`; 2.

`STDIN_FILENO` File number of `stdin`; 0.

`STDOUT_FILENO` File number of `stdout`; 1.

The `<unistd.h>` header shall define the following symbolic constant for terminal special character handling:

`_POSIX_VDISABLE`

This symbol shall be defined to be the value of a charac?

ter that shall disable terminal special character handling as described in Section 11.2.6, Special Control Characters. This symbol shall always be set to a value other than -1.

## Type Definitions

The `<unistd.h>` header shall define the `size_t`, `ssize_t`, `uid_t`, `gid_t`, `off_t`, and `pid_t` types as described in `<sys/types.h>`.

The `<unistd.h>` header shall define the `intptr_t` type as described in `<stdint.h>`.

## Declarations

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int      access(const char *, int);
unsigned alarm(unsigned);
int      chdir(const char *);
int      chown(const char *, uid_t, gid_t);
int      close(int);
size_t   confstr(int, char *, size_t);
char     *crypt(const char *, const char *);
int      dup(int);
int      dup2(int, int);
void     _exit(int);
void     encrypt(char [64], int);
int      execl(const char *, const char *, ...);
int      execlp(const char *, const char *, ...);
int      execlp(const char *, const char *, ...);
int      execv(const char *, char *const []);
int      execve(const char *, char *const [], char *const []);
int      execvp(const char *, char *const []);
int      faccessat(int, const char *, int, int);
int      fchdir(int);
int      fchown(int, uid_t, gid_t);
int      fchownat(int, const char *, uid_t, gid_t, int);
```

```
int    fdatsync(int);
int    fexecve(int, char *const [], char *const []);
pid_t  fork(void);
long   fpathconf(int, int);
int    fsync(int);
int    ftruncate(int, off_t);
char   *getcwd(char *, size_t);
gid_t  getegid(void);
uid_t  geteuid(void);
gid_t  getgid(void);
int    getgroups(int, gid_t []);
long   gethostid(void);
int    gethostname(char *, size_t);
char   *getlogin(void);
int    getlogin_r(char *, size_t);
int    getopt(int, char * const [], const char *);
pid_t  getpgid(pid_t);
pid_t  getpgrp(void);
pid_t  getpid(void);
pid_t  getppid(void);
pid_t  getsid(pid_t);
uid_t  getuid(void);
int    isatty(int);
int    lchown(const char *, uid_t, gid_t);
int    link(const char *, const char *);
int    linkat(int, const char *, int, const char *, int);
int    lockf(int, int, off_t);
off_t  lseek(int, off_t, int);
int    nice(int);
long   pathconf(const char *, int);
int    pause(void);
int    pipe(int [2]);
ssize_t pread(int, void *, size_t, off_t);
```

```

ssize_t  pwrite(int, const void *, size_t, off_t);
ssize_t  read(int, void *, size_t);
ssize_t  readlink(const char *restrict, char *restrict, size_t);
ssize_t  readlinkat(int, const char *restrict, char *restrict, size_t);
int      rmdir(const char *);
int      setegid(gid_t);
int      seteuid(uid_t);
int      setgid(gid_t);
int      setpgid(pid_t, pid_t);
pid_t    setpgrp(void);
int      setregid(gid_t, gid_t);
int      setreuid(uid_t, uid_t);
pid_t    setsid(void);
int      setuid(uid_t);
unsigned sleep(unsigned);
void     swab(const void *restrict, void *restrict, ssize_t);
int      symlink(const char *, const char *);
int      symlinkat(const char *, int, const char *);
void     sync(void);
long     sysconf(int);
pid_t    tcgetpgrp(int);
int      tcsetpgrp(int, pid_t);
int      truncate(const char *, off_t);
char     *ttyname(int);
int      ttyname_r(int, char *, size_t);
int      unlink(const char *);
int      unlinkat(int, const char *, int);
ssize_t  write(int, const void *, size_t);

```

Implementations may also include the `pthread_atfork()` prototype as defined in `<pthread.h>`. Implementations may also include the `ctermid()` prototype as defined in `<stdio.h>`.

The `<unistd.h>` header shall declare the following external variables:

```
extern char *optarg;
```

```
extern int opterr, optind, optopt;
```

Inclusion of the `<unistd.h>` header may make visible all symbols from the headers `<stddef.h>`, `<stdint.h>`, and `<stdio.h>`.

The following sections are informative.

## APPLICATION USAGE

POSIX.1?2008 only describes the behavior of systems that claim conformance to it. However, application developers who want to write applications that adapt to other versions of this standard (or to systems that do not conform to any POSIX standard) may find it useful to code them so as to conditionally compile different code depending on the value of `_POSIX_VERSION`, for example:

```
#if _POSIX_VERSION >= 200112L
/* Use the newer function that copes with large files. */
off_t pos=ftello(fp);
#else
/* Either this is an old version of POSIX, or _POSIX_VERSION is
not even defined, so use the traditional function. */
long pos=ftell(fp);
#endif
```

Earlier versions of POSIX.1?2008 and of the Single UNIX Specification can be identified by the following macros:

POSIX.1?1988 standard

```
_POSIX_VERSION==198808L
```

POSIX.1?1990 standard

```
_POSIX_VERSION==199009L
```

ISO POSIX?1:1996 standard

```
_POSIX_VERSION==199506L
```

Single UNIX Specification, Version 1

```
_XOPEN_UNIX and _XOPEN_VERSION==4
```

Single UNIX Specification, Version 2

```
_XOPEN_UNIX and _XOPEN_VERSION==500
```

ISO POSIX?1:2001 and Single UNIX Specification, Version 3

```
_POSIX_VERSION==200112L, plus (if the XSI option is supported)
```

`_XOPEN_UNIX` and `_XOPEN_VERSION==600`

POSIX.1-2008 does not make any attempt to define application binary interaction with the underlying operating system. However, application developers may find it useful to query `_SC_VERSION` at runtime via `sysconf()` to determine whether the current version of the operating system supports the necessary functionality as in the following program fragment:

```
if (sysconf(_SC_VERSION) < 200809L) {  
    fprintf(stderr, "POSIX.1-2008 system required, terminating \n");  
    exit(1);  
}
```

New applications should not use `_XOPEN_SHM` or `_XOPEN_ENH_I18N`.

## RATIONALE

As POSIX.1-2008 evolved, certain options became sufficiently standardized that it was concluded that simply requiring one of the option choices was simpler than retaining the option. However, for backwards-compatibility, the option flags (with required constant values) are retained.

## Version Test Macros

The standard developers considered altering the definition of `_POSIX_VERSION` and removing `_SC_VERSION` from the specification of `sysconf()` since the utility to an application was deemed by some to be minimal, and since the implementation of the functionality is potentially problematic. However, they recognized that support for existing application binaries is a concern to manufacturers, application developers, and the users of implementations conforming to POSIX.1-2008. While the example using `_SC_VERSION` in the APPLICATION USAGE section does not provide the greatest degree of imaginable utility to the application developer or user, it is arguably better than a core file or some other equally obscure result. (It is also possible for implementations to encode and recognize application binaries compiled in various POSIX.1-conforming environments, and modify the semantics of the underlying system to conform to the expectations of the application.) For

the reasons outlined in the preceding paragraphs and in the APPLICATION USAGE section, the standard developers elected to retain the `_POSIX_VERSION` and `_SC_VERSION` functionality.

#### Compile-Time Symbolic Constants for System-Wide Options

POSIX.1?2008 includes support in certain areas for the newly adopted policy governing options and stubs.

This policy provides flexibility for implementations in how they support options. It also specifies how conforming applications can adapt to different implementations that support different sets of options. It allows the following:

1. If an implementation has no interest in supporting an option, it does not have to provide anything associated with that option beyond the announcement that it does not support it.
2. An implementation can support a partial or incompatible version of an option (as a non-standard extension) as long as it does not claim to support the option.
3. An application can determine whether the option is supported. A strictly conforming application must check this announcement mechanism before first using anything associated with the option.

There is an important implication of this policy. POSIX.1?2008 cannot dictate the behavior of interfaces associated with an option when the implementation does not claim to support the option. In particular, it cannot require that a function associated with an unsupported option will fail if it does not perform as specified. However, this policy does not prevent a standard from requiring certain functions to always be present, but that they shall always fail on some implementations. The `setpgid()` function in the POSIX.1?1990 standard, for example, is considered appropriate.

The POSIX standards include various options, and the C-language binding support for an option implies that the implementation must supply data types and function interfaces. An application must be able to discover whether the implementation supports each option.

Any application must consider the following three cases for each op?

tion:

1. Option never supported.

The implementation advertises at compile time that the option will never be supported. In this case, it is not necessary for the implementation to supply any of the data types or function interfaces that are provided only as part of the option. The implementation might provide data types and functions that are similar to those defined by POSIX.1?2008, but there is no guarantee for any particular behavior.

2. Option always supported.

The implementation advertises at compile time that the option will always be supported. In this case, all data types and function interfaces shall be available and shall operate as specified.

3. Option might or might not be supported.

Some implementations might not provide a mechanism to specify support of options at compile time. In addition, the implementation might be unable or unwilling to specify support or non-support at compile time. In either case, any application that might use the option at runtime must be able to compile and execute. The implementation must provide, at compile time, all data types and function interfaces that are necessary to allow this. In this situation, there must be a mechanism that allows the application to query, at runtime, whether the option is supported. If the application attempts to use the option when it is not supported, the result is unspecified unless explicitly specified otherwise in POSIX.1?2008.

## FUTURE DIRECTIONS

None.

## SEE ALSO

<limits.h>, <stddef.h>, <stdint.h>, <stdio.h>, <sys\_socket.h>, <sys\_types.h>, <termios.h>, <wctype.h>

The System Interfaces volume of POSIX.1?2017, access(), alarm(), chown(), close(), confstr(), crypt(), ctermid(), dup(), \_Exit(), en?

crypt(), exec, fchdir(), fchown(), fdatsync(), fork(), fpathconf(),  
fsync(), ftruncate(), getcwd(), getegid(), geteuid(), getgid(), get?  
groups(), gethostid(), gethostname(), getlogin(), getopt(), getpgid(),  
getpgrp(), getpid(), getppid(), getsid(), getuid(), isatty(), lchown(),  
link(), lockf(), lseek(), nice(), pause(), pipe(), read(), readlink(),  
rmdir(), setegid(), seteuid(), setgid(), setpgid(), setpgrp(), se?  
tregid(), seteuid(), setsid(), setuid(), sleep(), swab(), symlink(),  
sync(), sysconf(), tcgetpgrp(), tcsetpgrp(), truncate(), ttyname(), un?  
link(), write()

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form  
from IEEE Std 1003.1-2017, Standard for Information Technology -- Por?  
table Operating System Interface (POSIX), The Open Group Base Specifi?  
cations Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of  
Electrical and Electronics Engineers, Inc and The Open Group. In the  
event of any discrepancy between this version and the original IEEE and  
The Open Group Standard, the original IEEE and The Open Group Standard  
is the referee document. The original Standard can be obtained online  
at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are  
most likely to have been introduced during the conversion of the source  
files to man page format. To report such errors, see [https://www.ker?  
nel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .