Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'sysconf.3p' command

$ man sysconf.3p

SYSCONF(3P) POSIX Programmer's Manual SYSCONF(3P)

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

sysconf ? get configurable system variables

SYNOPSIS

#include <unistd.h>

long sysconf(int name);

DESCRIPTION

The sysconf() function provides a method for the application to deter? mine the current value of a configurable system limit or option (vari? able). The implementation shall support all of the variables listed in the following table and may support others.

The name argument represents the system variable to be queried. The following table lists the minimal set of system variables from <lim? its.h> or <unistd.h> that can be returned by sysconf(), and the sym? bolic constants defined in <unistd.h> that are the corresponding values used for name.

???

? Variable ? Value of Name ?

??

?{AIO_LISTIO_MAX} ?_SC_AIO_LISTIO_MAX ?

?{AIO_MAX} ?_SC_AIO_MAX ?

?{AIO_PRIO_DELTA_MAX} ?_SC_AIO_PRIO_DELTA_MAX ?

?{ARG_MAX} ?_SC_ARG_MAX ?

?{ATEXIT_MAX} ?_SC_ATEXIT_MAX ?

?{BC_BASE_MAX} ?_SC_BC_BASE_MAX ?

?{BC_DIM_MAX} ?_SC_BC_DIM_MAX ?

?{BC_SCALE_MAX} ?_SC_BC_SCALE_MAX ?

?{BC_STRING_MAX} ?_SC_BC_STRING_MAX ?

?{CHILD_MAX} ?_SC_CHILD_MAX ?

?Clock ticks/second ?_SC_CLK_TCK ?

?{COLL_WEIGHTS_MAX} ?_SC_COLL_WEIGHTS_MAX ?

?{DELAYTIMER_MAX} ?_SC_DELAYTIMER_MAX ?

?{EXPR_NEST_MAX} ?_SC_EXPR_NEST_MAX ?

?{HOST_NAME_MAX} ?_SC_HOST_NAME_MAX ?

?{IOV_MAX} ?_SC_IOV_MAX ?

?{LINE_MAX} ?_SC_LINE_MAX ?

?{LOGIN_NAME_MAX} ?_SC_LOGIN_NAME_MAX ?

?{NGROUPS_MAX} ?_SC_NGROUPS_MAX ?

?Initial size of getgrgid_r() and ?_SC_GETGR_R_SIZE_MAX ?

?getgrnam_r() data buffers ? ?

?Initial size of getpwuid_r() and ?_SC_GETPW_R_SIZE_MAX ?

?getpwnam_r() data buffers ? ?

?{MQ_OPEN_MAX} ?_SC_MQ_OPEN_MAX ?

?{MQ_PRIO_MAX} ?_SC_MQ_PRIO_MAX ?

?{OPEN_MAX} ?_SC_OPEN_MAX ?

?{PAGE_SIZE} ?_SC_PAGE_SIZE ?

?{PAGESIZE} ?_SC_PAGESIZE ?

?{PTHREAD_DESTRUCTOR_ITERATIONS} ?_SC_THREAD_DESTRUCTOR_ITERATIONS ?

?{PTHREAD_KEYS_MAX} ?_SC_THREAD_KEYS_MAX ?

?{PTHREAD_STACK_MIN} ?_SC_THREAD_STACK_MIN ?

?{PTHREAD_THREADS_MAX} ?_SC_THREAD_THREADS_MAX ?

?{RE_DUP_MAX} ?_SC_RE_DUP_MAX ?

?{RTSIG_MAX} ?_SC_RTSIG_MAX ?

?{SEM_NSEMS_MAX} ?_SC_SEM_NSEMS_MAX ?

?{SEM_VALUE_MAX} ?_SC_SEM_VALUE_MAX ?

?{SIGQUEUE_MAX} ?_SC_SIGQUEUE_MAX ?

?{STREAM_MAX} ?_SC_STREAM_MAX ?

?{SYMLOOP_MAX} ?_SC_SYMLOOP_MAX ?

?{TIMER_MAX} ?_SC_TIMER_MAX ?

?{TTY_NAME_MAX} ?_SC_TTY_NAME_MAX ?

?{TZNAME_MAX} ?_SC_TZNAME_MAX ?

?_POSIX_ADVISORY_INFO ?_SC_ADVISORY_INFO ?

?_POSIX_BARRIERS ?_SC_BARRIERS ?

?_POSIX_ASYNCHRONOUS_IO ?_SC_ASYNCHRONOUS_IO ?

?_POSIX_CLOCK_SELECTION ?_SC_CLOCK_SELECTION ?

?_POSIX_CPUTIME ?_SC_CPUTIME ?

?_POSIX_FSYNC ?_SC_FSYNC ?

?_POSIX_IPV6 ?_SC_IPV6 ?

?_POSIX_JOB_CONTROL ?_SC_JOB_CONTROL ?

?_POSIX_MAPPED_FILES ?_SC_MAPPED_FILES ?

?_POSIX_MEMLOCK ?_SC_MEMLOCK ?

?_POSIX_MEMLOCK_RANGE ?_SC_MEMLOCK_RANGE ?

?_POSIX_MEMORY_PROTECTION ?_SC_MEMORY_PROTECTION ?

?_POSIX_MESSAGE_PASSING ?_SC_MESSAGE_PASSING ?

?_POSIX_MONOTONIC_CLOCK ?_SC_MONOTONIC_CLOCK ?

?_POSIX_PRIORITIZED_IO ?_SC_PRIORITIZED_IO ?

?_POSIX_PRIORITY_SCHEDULING ?_SC_PRIORITY_SCHEDULING ?

?_POSIX_RAW_SOCKETS ?_SC_RAW_SOCKETS ?

?_POSIX_READER_WRITER_LOCKS ?_SC_READER_WRITER_LOCKS ?

?_POSIX_REALTIME_SIGNALS ?_SC_REALTIME_SIGNALS ?

?_POSIX_REGEXP ?_SC_REGEXP ?

?_POSIX_SAVED_IDS ?_SC_SAVED_IDS ?

?_POSIX_SEMAPHORES ?_SC_SEMAPHORES ?

?_POSIX_SHARED_MEMORY_OBJECTS ?_SC_SHARED_MEMORY_OBJECTS ?

?_POSIX_SHELL ?_SC_SHELL ?

?_POSIX_SPAWN ?_SC_SPAWN ?

?_POSIX_SPIN_LOCKS ?_SC_SPIN_LOCKS ?

?_POSIX_SPORADIC_SERVER ?_SC_SPORADIC_SERVER ?

?_POSIX_SS_REPL_MAX ?_SC_SS_REPL_MAX ?

?_POSIX_SYNCHRONIZED_IO ?_SC_SYNCHRONIZED_IO ?

?_POSIX_THREAD_ATTR_STACKADDR ?_SC_THREAD_ATTR_STACKADDR ?

?_POSIX_THREAD_ATTR_STACKSIZE ?_SC_THREAD_ATTR_STACKSIZE ?

?_POSIX_THREAD_CPUTIME ?_SC_THREAD_CPUTIME ?

?_POSIX_THREAD_PRIO_INHERIT ?_SC_THREAD_PRIO_INHERIT ?

?_POSIX_THREAD_PRIO_PROTECT ?_SC_THREAD_PRIO_PROTECT ?

?_POSIX_THREAD_PRIORITY_SCHEDULING?_SC_THREAD_PRIORITY_SCHEDULING ?

?_POSIX_THREAD_PROCESS_SHARED ?_SC_THREAD_PROCESS_SHARED ?

?_POSIX_THREAD_ROBUST_PRIO_INHERIT?_SC_THREAD_ROBUST_PRIO_INHERIT ?

?_POSIX_THREAD_ROBUST_PRIO_PROTECT?_SC_THREAD_ROBUST_PRIO_PROTECT ?

?_POSIX_THREAD_SAFE_FUNCTIONS ?_SC_THREAD_SAFE_FUNCTIONS ?

?_POSIX_THREAD_SPORADIC_SERVER ?_SC_THREAD_SPORADIC_SERVER ?

?_POSIX_THREADS ?_SC_THREADS ?

?_POSIX_TIMEOUTS ?_SC_TIMEOUTS ?

??

??

? Variable ? Value of Name ?

??

?_POSIX_TIMERS _SC_TIMERS ?

?_POSIX_TRACE _SC_TRACE ?

?_POSIX_TRACE_EVENT_FILTER _SC_TRACE_EVENT_FILTER ?

?_POSIX_TRACE_EVENT_NAME_MAX _SC_TRACE_EVENT_NAME_MAX ?

?_POSIX_TRACE_INHERIT _SC_TRACE_INHERIT ?

?_POSIX_TRACE_LOG _SC_TRACE_LOG ?

?_POSIX_TRACE_NAME_MAX _SC_TRACE_NAME_MAX ?

?_POSIX_TRACE_SYS_MAX _SC_TRACE_SYS_MAX ?

?_POSIX_TRACE_USER_EVENT_MAX _SC_TRACE_USER_EVENT_MAX ?

?_POSIX_TYPED_MEMORY_OBJECTS _SC_TYPED_MEMORY_OBJECTS ?

?_POSIX_VERSION _SC_VERSION ?

?_POSIX_V7_ILP32_OFF32 _SC_V7_ILP32_OFF32 ?

?_POSIX_V7_ILP32_OFFBIG _SC_V7_ILP32_OFFBIG ?

?_POSIX_V7_LP64_OFF64 _SC_V7_LP64_OFF64 ?

?_POSIX_V7_LPBIG_OFFBIG _SC_V7_LPBIG_OFFBIG ?

?_POSIX_V6_ILP32_OFF32 _SC_V6_ILP32_OFF32 ?

?_POSIX_V6_ILP32_OFFBIG _SC_V6_ILP32_OFFBIG ?

?_POSIX_V6_LP64_OFF64 _SC_V6_LP64_OFF64 ?

?_POSIX_V6_LPBIG_OFFBIG _SC_V6_LPBIG_OFFBIG ?

?_POSIX2_C_BIND _SC_2_C_BIND ?

?_POSIX2_C_DEV _SC_2_C_DEV ?

?_POSIX2_CHAR_TERM _SC_2_CHAR_TERM ?

?_POSIX2_FORT_DEV _SC_2_FORT_DEV ?

?_POSIX2_FORT_RUN _SC_2_FORT_RUN ?

?_POSIX2_LOCALEDEF _SC_2_LOCALEDEF ?

?_POSIX2_PBS _SC_2_PBS ?

?_POSIX2_PBS_ACCOUNTING _SC_2_PBS_ACCOUNTING ?

?_POSIX2_PBS_CHECKPOINT _SC_2_PBS_CHECKPOINT ?

?_POSIX2_PBS_LOCATE _SC_2_PBS_LOCATE ?

?_POSIX2_PBS_MESSAGE _SC_2_PBS_MESSAGE ?

?_POSIX2_PBS_TRACK _SC_2_PBS_TRACK ?

?_POSIX2_SW_DEV _SC_2_SW_DEV ?

?_POSIX2_UPE _SC_2_UPE ?

?_POSIX2_VERSION _SC_2_VERSION ?

?_XOPEN_CRYPT _SC_XOPEN_CRYPT ?

?_XOPEN_ENH_I18N _SC_XOPEN_ENH_I18N ?

?_XOPEN_REALTIME _SC_XOPEN_REALTIME ?

?_XOPEN_REALTIME_THREADS _SC_XOPEN_REALTIME_THREADS ?

?_XOPEN_SHM _SC_XOPEN_SHM ?

?_XOPEN_STREAMS _SC_XOPEN_STREAMS ?

?_XOPEN_UNIX _SC_XOPEN_UNIX ?

?_XOPEN_UUCP _SC_XOPEN_UUCP ?

?_XOPEN_VERSION _SC_XOPEN_VERSION ?

??

RETURN VALUE

If name is an invalid value, sysconf() shall return -1 and set errno to indicate the error. If the variable corresponding to name is described in <limits.h> as a maximum or minimum value and the variable has no limit, sysconf() shall return -1 without changing the value of errno.

Note that indefinite limits do not imply infinite limits; see <lim? its.h>.

Otherwise, sysconf() shall return the current variable value on the system. The value returned shall not be more restrictive than the cor? responding value described to the application when it was compiled with the implementation's <limits.h> or <unistd.h>. The value shall not change during the lifetime of the calling process, except that sysconf(_SC_OPEN_MAX) may return different values before and after a call to setrlimit() which changes the RLIMIT_NOFILE soft limit.

If the variable corresponding to name is dependent on an unsupported option, the results are unspecified.

ERRORS

The sysconf() function shall fail if:

EINVAL The value of the name argument is invalid.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

As -1 is a permissible return value in a successful situation, an ap? plication wishing to check for error situations should set errno to 0, then call sysconf(), and, if it returns -1, check to see if errno is non-zero.

Application developers should check whether an option, such as _POSIX_TRACE, is supported prior to obtaining and using values for re? lated variables, such as _POSIX_TRACE_NAME_MAX.

RATIONALE

This functionality was added in response to requirements of application

developers and of system vendors who deal with many international sys? tem configurations. It is closely related to pathconf() and fpath? conf().

Although a conforming application can run on all systems by never de? manding more resources than the minimum values published in this volume of POSIX.1?2017, it is useful for that application to be able to use the actual value for the quantity of a resource available on any given system. To do this, the application makes use of the value of a sym? bolic constant in <limits.h> or <unistd.h>.

However, once compiled, the application must still be able to cope if the amount of resource available is increased. To that end, an applica? tion may need a means of determining the quantity of a resource, or the presence of an option, at execution time.

Two examples are offered:

1. Applications may wish to act differently on systems with or without job control. Applications vendors who wish to distribute only a single binary package to all instances of a computer architecture would be forced to assume job control is never available if it were to rely solely on the <unistd.h> value published in this volume of POSIX.1?2017.
2. International applications vendors occasionally require knowledge of the number of clock ticks per second. Without these facilities, they would be required to either distribute their applications par? tially in source form or to have 50 Hz and 60 Hz versions for the various countries in which they operate.

It is the knowledge that many applications are actually distributed widely in executable form that leads to this facility. If limited to the most restrictive values in the headers, such applications would have to be prepared to accept the most limited environments offered by the smallest microcomputers. Although this is entirely portable, there was a consensus that they should be able to take advantage of the fa? cilities offered by large systems, without the restrictions associated with source and object distributions.

During the discussions of this feature, it was pointed out that it is almost always possible for an application to discern what a value might be at runtime by suitably testing the various functions themselves. And, in any event, it could always be written to adequately deal with error returns from the various functions. In the end, it was felt that this imposed an unreasonable level of complication and sophistication on the application developer.

This runtime facility is not meant to provide ever-changing values that applications have to check multiple times. The values are seen as changing no more frequently than once per system initialization, such as by a system administrator or operator with an automatic configura? tion program. This volume of POSIX.1?2017 specifies that they shall not change within the lifetime of the process.

Some values apply to the system overall and others vary at the file system or directory level. The latter are described in fpathconf().

Note that all values returned must be expressible as integers. String values were considered, but the additional flexibility of this approach was rejected due to its added complexity of implementation and use.

Some values, such as {PATH_MAX}, are sometimes so large that they must not be used to, say, allocate arrays. The sysconf() function returns a negative value to show that this symbolic constant is not even defined in this case.

Similar to pathconf(), this permits the implementation not to have a limit. When one resource is infinite, returning an error indicating that some other resource limit has been reached is conforming behavior.

FUTURE DIRECTIONS

None.

SEE ALSO

confstr(), fpathconf()

The Base Definitions volume of POSIX.1?2017, <limits.h>, <unistd.h>

The Shell and Utilities volume of POSIX.1?2017, getconf

COPYRIGHT

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 .