

Package ‘unix’

April 21, 2017

Title Unix System Utilities

Version 1.3

Author Jeroen Ooms

Maintainer Jeroen Ooms <jeroen@berkeley.edu>

Description Bindings to system utilities found in most Unix systems such as POSIX functions which are not part of the Standard C Library.

License MIT + file LICENSE

URL <https://github.com/jeroen/unix>

BugReports <https://github.com/jeroen/unix/issues>

OS_type unix

SystemRequirements POSIX.1-2001

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-04-21 06:11:54 UTC

R topics documented:

chroot	2
getuid	2
rlimit	4
userinfo	6

Index	7
--------------	----------

chroot

Change Root Dir

Description

Changes the root directory of the calling process to that specified in path. This directory will be used for pathnames beginning with /. **Only a privileged process (i.e. sudo) may call chroot().**

Usage

```
chroot(path = getwd())
```

Arguments

path directory of the new root

Details

This call changes an ingredient in the pathname resolution process and does nothing else. In particular, it is not intended to be used for any kind of security purpose, neither to fully sandbox a process nor to restrict filesystem system calls.

References

[CHROOT\(2\)](#)

getuid

Process Info

Description

Get or set attributes of the current process.

Usage

```
getuid()
```

```
getgid()
```

```
geteuid()
```

```
getegid()
```

```
getpid()
```

```
getppid()
```

```
getpgid()  
getpriority()  
setuid(uid)  
seteuid(uid)  
setgid(gid)  
setegid(gid)  
setpgid(pgid = 0)  
setpriority(prio)  
kill(pid, signal = SIGTERM)
```

Arguments

uid	User ID from /etc/passwd.
gid	Group ID from /etc/group.
pgid	Process Group ID. Default 0 sets pgid to the current pid.
prio	Priority level
pid	process ID (integer)
signal	a signal number (integer), defaults to tools::SIGTERM .

Details

Acronyms stand for:

- pid Process ID
- ppid Parent-Process ID
- pgid Process-Group ID
- uid User ID
- euid Effective User ID
- gid Group ID
- egid Effective Group ID
- prio Priority level

An unprivileged (non-root) process cannot change its uid and only lower process priority (higher value).

References

[GETUID\(2\)](#) [GETPID\(2\)](#) [GETPGID\(2\)](#) [GETPRIORITY\(2\)](#)

Examples

```
# Current User:
getuid()
# Current UserGroup:
getgid()
# Current UserGroup:
geteuid()
# Current UserGroup:
getegid()
# Process ID
getpid()
# parent PID:
getppid()
# Process group id:
getpgid()

# Detach process group
setpgid(0)
getpgid()
# Process priority:
getpriority()
# Decrease priority
setpriority(getpriority() + 1)
```

rlimit

Resource Limits

Description

Get and set process resource limits. Each function returns the current limits, and can optionally update the limit by passing argument values. The `rlimit_all()` function is a convenience wrapper which prints all current hard and soft limits.

Usage

```
rlimit_all()

rlimit_as(cur = NULL, max = NULL)

rlimit_core(cur = NULL, max = NULL)

rlimit_cpu(cur = NULL, max = NULL)

rlimit_data(cur = NULL, max = NULL)

rlimit_fsize(cur = NULL, max = NULL)

rlimit_memlock(cur = NULL, max = NULL)
```

```
rlimit_nofile(cur = NULL, max = NULL)
```

```
rlimit_nproc(cur = NULL, max = NULL)
```

```
rlimit_stack(cur = NULL, max = NULL)
```

Arguments

cur	set the current (soft) limit for this resource. See details.
max	set the max (hard) limit for this resource. See details.

Details

Each resource has an associated soft and hard limit. The soft limit is the value that the kernel enforces for the corresponding resource. The hard limit acts as a ceiling for the soft limit: an unprivileged process may set only its soft limit to a value in the range from 0 up to the hard limit, and (irreversibly) lower its hard limit.

Definitons from the [Linux manual page](#) are as follows:

- RLIMIT_AS : the maximum size of the process's virtual memory (address space) in bytes.
- RLIMIT_CORE : the maximum size of a core file that the process may dump.
- RLIMIT_CPU : a limit in seconds on the amount of CPU time (**not** elapsed time) that the process may consume. When the process reaches the soft limit, it is sent a SIGXCPU signal.
- RLIMIT_DATA : the maximum size of the process's data segment (initialized data, uninitialized data, and heap).
- RLIMIT_FSIZE : the maximum size of files that the process may create. Attempts to extend a file beyond this limit result in delivery of a SIGXFSZ signal.
- RLIMIT_MEMLOCK : the maximum number of bytes of memory that may be locked into RAM.
- RLIMIT_NOFILE : a value one greater than the maximum file descriptor number that can be opened by this process.
- RLIMIT_NPROC : the maximum number of processes that can be created for the real user ID of the calling process. Upon encountering this limit, fork fails with the error EAGAIN. Not enforced for root user.
- RLIMIT_STACK : the maximum size of the process stack, in bytes.

Note that the support for enforcing limits vary widely by system. In particular RLIMIT_AS has a different meaning depending on how memory allocation is managed by the operating system (and doesn't work at all on MacOS).

References

[GETRLIMIT\(2\)](#)

Examples

```
# Print all limits
rlimit_all()

# Get one limit
rlimit_as()

# Set a soft limit
lim <- rlimit_as(1e9)
print(lim)

# Reset the limit to max
rlimit_as(cur = lim$max)

## Not run:
# Set a hard limit (irreversible)
rlimit_as(max = 1e10)

## End(Not run)
```

userinfo

User / Group Info

Description

Lookup a user or group info via user uid/name or group gid/name.

Usage

```
user_info(uid = getuid())

group_info(gid = getgid())
```

Arguments

uid	user ID (integer) or name (string)
gid	group ID (integer) or name (string)

References

[GETPWNAM\(3\)](#) [GETGRNAM\(3\)](#)

Examples

```
# Get info current user
user_info()
group_info()
```

Index

[chroot](#), 2

[getegid \(getuid\)](#), 2
[geteuid \(getuid\)](#), 2
[getgid \(getuid\)](#), 2
[getpgid \(getuid\)](#), 2
[getpid \(getuid\)](#), 2
[getppid \(getuid\)](#), 2
[getpriority \(getuid\)](#), 2
[getuid](#), 2
[group_info \(userinfo\)](#), 6

[kill \(getuid\)](#), 2

[rlimit](#), 4
[rlimit_all \(rlimit\)](#), 4
[rlimit_as \(rlimit\)](#), 4
[rlimit_core \(rlimit\)](#), 4
[rlimit_cpu \(rlimit\)](#), 4
[rlimit_data \(rlimit\)](#), 4
[rlimit_fsize \(rlimit\)](#), 4
[rlimit_memlock \(rlimit\)](#), 4
[rlimit_nofile \(rlimit\)](#), 4
[rlimit_nproc \(rlimit\)](#), 4
[rlimit_stack \(rlimit\)](#), 4

[setegid \(getuid\)](#), 2
[seteuid \(getuid\)](#), 2
[setgid \(getuid\)](#), 2
[setpgid \(getuid\)](#), 2
[setpriority \(getuid\)](#), 2
[setuid \(getuid\)](#), 2

[tools::SIGTERM](#), 3

[user_info \(userinfo\)](#), 6
[userinfo](#), 6