



Linux Ubuntu 26.04 Manual Pages on command 'procps_misc.3'

\$ man procps_misc.3

PROCPS_MISC(3)

Library Functions Manual

PROCPS_MISC(3)

NAME

procps_misc - API for miscellaneous information in the /proc filesystem

SYNOPSIS

```
#include <libproc2/misc.h>
```

Platform Particulars

```
long    procps_cpu_count (void);
long    procps_hertz_get (void);
unsigned int procps_pid_length (void);
int      procps_linux_version (void);
```

Runtime Particulars

```
int  procps_loadavg (double *av1, double *av5, double *av15);
int  procps_uptime (double *uptime_secs, double *idle_secs);
char *procps_uptime_sprint (void);
char *procps_uptime_sprint_short (void);
```

Namespace Particulars

```
int      procps_ns_get_id (const char *name);
const char *procps_ns_get_name (int id);
int      procps_ns_read_pid (int pid, struct procps_ns *nsp);
```

Link with -lproc2.

DESCRIPTION

procps_cpu_count() returns the number of CPUs that are currently online as sysconf(_SC_NPROCESSORS_ONLY) or an assumed 1.

procps_hertz_get() returns the number of clock ticks per second as sysconf(_SC_CLK_TCK) or an assumed 100. Dividing ticks by this value yields seconds.

procps_pid_length() returns the maximum string length for a PID on the system. For example, if the largest possible PID value on was 123, then the length would be 3. If the file /proc/sys/kernel/pid_max is unreadable, the value is assumed to be 5.

procps_linux_version() returns the current Linux version as an encoded integer. On non-Linux systems that have an emulated proc filesystem this function returns the version of the Linux emulation instead. The version consists of three positive integers representing the major, minor and patch levels. The following macros are provided for encoding a given Linux version or separating out the components of the current version.

```
LINUX_VERSION( major , minor , patch )
```

```
LINUX_VERSION_MAJOR( ver )
```

```
LINUX_VERSION_MINOR( ver )
```

```
LINUX_VERSION_PATCH( ver )
```

procps_loadavg() fetches the system load average and puts the 1, 5 and 15 minute averages into location(s) specified by any pointer which is not NULL.

procps_uptime() returns uptime and/or idle seconds into location(s) specified by any pointer which is not NULL. The sprint varieties return a human-readable string in one of two forms.

```
HH:MM:SS up HH:MM, # users, load average: 1, 5, 15 MM averages
```

```
up HH, MM
```

procps_ns_get_id() returns the integer id (enum namespace_type) of the namespace for the given namespace name.

procps_ns_get_name() returns the name of the namespace for the given id (enum namespace_type).

procps_ns_read_pid() returns the inodes for the namespaces of the given process in the procps_ns structure pointed to by nsp. Those inodes will appear in the order proscribed by enum namespace_type.

```
enum namespace_type {  
    PROCPS_NS_CGROUP,  
    PROCPS_NS_IPC,  
    PROCPS_NS_MNT,  
    PROCPS_NS_NET,  
    PROCPS_NS_PID,
```

```
PROCPS_NS_TIME,  
PROCPS_NS_USER,  
PROCPS_NS_UTS
```

```
};
```

RETURN VALUE

Functions Returning an `int` or `long`

An error will be indicated by a negative number that is always the inverse of some well known `errno.h` value.

Functions Returning an `address`

An error will be indicated by a NULL return pointer with the reason found in the formal `errno` value.

FILES

`/proc/loadavg`

The raw values for load average.

`/proc/sys/kernel/osrelease`

Contains the release version of the Linux kernel or `proc` filesystem.

`/proc/sys/kernel/pid_max`

Contains the value at which PIDs wrap around, one greater than the maximum PID value.

`/proc/uptime`

The raw values for uptime and idle time.

`/proc/<PID>/ns`

contains the set of namespaces for a particular PID.

SEE ALSO

`procps(3)`, `procps_pids(3)`, `proc(5)`.

`libproc2`

August 2022

`PROCPS_MISC(3)`