NAME

setproctitle setproctitle_fast - set process title

SYNOPSIS

#include <unistd.h>

void
setproctitle(*const char *fmt, ...*);

void
setproctitle_fast(*const char *fmt, ...*);

DESCRIPTION

The **setproctitle**() library routine sets the process title that appears on the ps(1) command. The **setproctitle_fast**() variant is optimized for high frequency updates, but may make the ps(1) command slightly slower by not updating the kernel cache of the program arguments.

The title is set from the executable's name, followed by the result of a printf(3) style expansion of the arguments as specified by the *fmt* argument. If the *fmt* argument begins with a "-" character, the executable's name is skipped.

If *fmt* is NULL, the process title is restored.

EXAMPLES

To set the title on a daemon to indicate its activity:

```
setproctitle("talking to %s", inet_ntoa(addr));
```

SEE ALSO

ps(1), w(1), setprogname(3), kvm(3), kvm_getargv(3), printf(3)

STANDARDS

The **setproctitle**() function is implicitly non-standard. Other methods of causing the ps(1) command line to change, including copying over the argv[0] string are also implicitly non-portable. It is preferable to use an operating system supplied **setproctitle**() if present.

Unfortunately, it is possible that there are other calling conventions to other versions of **setproctitle**(), although none have been found by the author as yet. This is believed to be the predominant convention.

It is thought that the implementation is compatible with other systems, including NetBSD and BSD/OS.

HISTORY

The **setproctitle**() function first appeared in FreeBSD 2.2. The **setproctitle_fast**() function first appeared in FreeBSD 12. Other operating systems have similar functions.

AUTHORS

Peter Wemm <*peter@FreeBSD.org*> stole the idea from the **Sendmail 8.7.3** source code by Eric Allman <*eric@sendmail.org*>.

BUGS

Never pass a string with user-supplied data as a format without using ‘%s’. An attacker can put format specifiers in the string to mangle your stack, leading to a possible security hole. This holds true even if the string was built using a function like **snprintf**(), as the resulting string may still contain user-supplied conversion specifiers for later interpolation by **setproctitle**().

Always use the proper secure idiom:

```
setproctitle("%s", string);
```