

NAME

`getpid`, `getppid` - get process identification

SYNOPSIS

```
#include <sys/types.h>
#include <unistd.h>

pid_t getpid(void);
pid_t getppid(void);
```

DESCRIPTION

`getpid()` returns the process ID of the calling process. (This is often used by routines that generate unique temporary filenames.)

`getppid()` returns the process ID of the parent of the calling process.

ERRORS

These functions are always successful.

CONFORMING TO

POSIX.1-2001, 4.3BSD, SVr4.

NOTES

Since glibc version 2.3.4, the glibc wrapper function for `getpid()` caches PIDs, so as to avoid additional system calls when a process calls `getpid()` repeatedly. Normally this caching is invisible, but its correct operation relies on support in the wrapper functions for `fork(2)`, `vfork(2)`, and `clone(2)`: if an application bypasses the glibc wrappers for these system calls by using `syscall(2)`, then a call to `getpid()` in the child will return the wrong value (to be precise: it will return the PID of the parent process). See also `clone(2)` for discussion of a case where `getpid()` may return the wrong value even when invoking `clone(2)` via the glibc wrapper function.

SEE ALSO

[clone\(2\)](#), [fork\(2\)](#), [kill\(2\)](#), [exec\(3\)](#), [mkstemp\(3\)](#), [tempnam\(3\)](#), [tmpfile\(3\)](#), [tmpnam\(3\)](#), [credentials\(7\)](#), [pid_namespaces\(7\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.