NAME

getpagesize − get memory page size

SYNOPSIS

#include <unistd.h>

int getpagesize(void);

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

getpagesize(): _BSD_SOURCE || _XOPEN_SOURCE >= 500

DESCRIPTION

The function **getpagesize**() returns the number of bytes in a page, where a "page" is the thing used where it says in the description of **mmap**(2) that files are mapped in page-sized units.

The size of the kind of pages that **mmap**(2) uses, is found using

```
#include <unistd.h>
long sz = sysconf(_SC_PAGESIZE);
```

(most systems allow the synonym **_SC_PAGE_SIZE** for **_SC_PAGESIZE**), or

```
#include <unistd.h>
int sz = getpagesize();
```

CONFORMING TO

SVr4, 4.4BSD, SUSv2. In SUSv2 the **getpagesize**() call is labeled LEGACY, and in POSIX.1-2001 it has been dropped; HP-UX does not have this call. Portable applications should employ *sysconf(_SC_PAGESIZE)* instead of this call.

NOTES

Whether **getpagesize**() is present as a Linux system call depends on the architecture. If it is, it returns the kernel symbol **PAGE_SIZE**, whose value depends on the architecture and machine model. Generally, one uses binaries that are dependent on the architecture but not on the machine model, in order to have a single binary distribution per architecture. This means that a user program should not find **PAGE_SIZE** at compile time from a header file, but use an actual system call, at least for those architectures (like sun4) where this dependency exists. Here libc4, libc5, glibc 2.0 fail because their **getpagesize**() returns a statically derived value, and does not use a system call. Things are OK in glibc 2.1.

SEE ALSO

mmap(2), **sysconf**(3)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.