

NAME

ttypslot – find the slot of the current user’s terminal in some file

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

```
#include <unistd.h>    /* See NOTES */
```

```
int ttypslot(void);
```

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

ttypslot():

Since glibc 2.24:

```
_DEFAULT_SOURCE
```

From glibc 2.20 to glibc 2.23:

```
_DEFAULT_SOURCE || (_XOPEN_SOURCE && _XOPEN_SOURCE < 500)
```

glibc 2.19 and earlier:

```
_BSD_SOURCE || (_XOPEN_SOURCE && _XOPEN_SOURCE < 500)
```

DESCRIPTION

The legacy function **ttypslot()** returns the index of the current user’s entry in some file.

Now "What file?" you ask. Well, let’s first look at some history.

Ancient history

There used to be a file */etc/ttyps* in UNIX V6, that was read by the **init(1)** program to find out what to do with each terminal line. Each line consisted of three characters. The first character was either '0' or '1', where '0' meant "ignore". The second character denoted the terminal: '8' stood for "/dev/tty8". The third character was an argument to **getty(8)** indicating the sequence of line speeds to try ('-' was: start trying 110 baud). Thus a typical line was "18-". A hang on some line was solved by changing the '1' to a '0', signaling **init**, changing back again, and signaling **init** again.

In UNIX V7 the format was changed: here the second character was the argument to **getty(8)** indicating the sequence of line speeds to try ('0' was: cycle through 300-1200-150-110 baud; '4' was for the on-line console DECwriter) while the rest of the line contained the name of the tty. Thus a typical line was "14console".

Later systems have more elaborate syntax. System V-like systems have */etc/inittab* instead.

Ancient history (2)

On the other hand, there is the file */etc/utmp* listing the people currently logged in. It is maintained by **login(1)**. It has a fixed size, and the appropriate index in the file was determined by **login(1)** using the **ttypslot()** call to find the number of the line in */etc/ttyps* (counting from 1).

The semantics of ttypslot

Thus, the function **ttypslot()** returns the index of the controlling terminal of the calling process in the file */etc/ttyps*, and that is (usually) the same as the index of the entry for the current user in the file */etc/utmp*. BSD still has the */etc/ttyps* file, but System V-like systems do not, and hence cannot refer to it. Thus, on such systems the documentation says that **ttypslot()** returns the current user’s index in the user accounting data base.

RETURN VALUE

If successful, this function returns the slot number. On error (e.g., if none of the file descriptors 0, 1, or 2 is associated with a terminal that occurs in this data base) it returns 0 on UNIX V6 and V7 and BSD-like systems, but -1 on System V-like systems.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

| Interface | Attribute | Value |
|------------------|---------------|-----------|
| ttyslot() | Thread safety | MT-Unsafe |

STANDARDS

SUSv1; marked as LEGACY in SUSv2; removed in POSIX.1-2001. SUSv2 requires `-1` on error.

NOTES

The `utmp` file is found in various places on various systems, such as `/etc/utmp`, `/var/adm/utmp`, `/var/run/utmp`.

The glibc2 implementation of this function reads the file `_PATH_TTY`, defined in `<ttyent.h>` as `"/etc/ttys"`. It returns 0 on error. Since Linux systems do not usually have `"/etc/ttys"`, it will always return 0.

On BSD-like systems and Linux, the declaration of `ttyslot()` is provided by `<unistd.h>`. On System V-like systems, the declaration is provided by `<stdlib.h>`. Since glibc 2.24, `<stdlib.h>` also provides the declaration with the following feature test macro definitions:

```
(_XOPEN_SOURCE >= 500 ||
  (_XOPEN_SOURCE && _XOPEN_SOURCE_EXTENDED) )
&& ! (_POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE >= 600)
```

Minix also has `fttyslot(fd)`.

SEE ALSO

`getttyent(3)`, `ttynam(3)`, `utmp(5)`