NAME

login, logout − write utmp and wtmp entries

LIBRARY

System utilities library (*libutil*, *−lutil*)

SYNOPSIS

#include <utmp.h>

void login(const struct utmp **ut);**
int logout(const char **ut_line);**

DESCRIPTION

The utmp file records who is currently using the system. The wtmp file records all logins and logouts. See **utmp**(5).

The function **login**() takes the supplied *struct utmp*, *ut*, and writes it to both the utmp and the wtmp file.

The function **logout**() clears the entry in the utmp file again.

GNU details

More precisely, **login**() takes the argument *ut* struct, fills the field *ut−>ut_type* (if there is such a field) with the value **USER_PROCESS**, and fills the field *ut−>ut_pid* (if there is such a field) with the process ID of the calling process. Then it tries to fill the field *ut−>ut_line*. It takes the first of *stdin*, *stdout*, *stderr* that is a terminal, and stores the corresponding pathname minus a possible leading */dev/* into this field, and then writes the struct to the utmp file. On the other hand, if no terminal name was found, this field is filled with "???" and the struct is not written to the utmp file. After this, the struct is written to the wtmp file.

The **logout**() function searches the utmp file for an entry matching the *ut_line* argument. If a record is found, it is updated by zeroing out the *ut_name* and *ut_host* fields, updating the *ut_tv* timestamp field and setting *ut_type* (if there is such a field) to **DEAD_PROCESS**.

RETURN VALUE

The **logout**() function returns 1 if the entry was successfully written to the database, or 0 if an error occurred.

FILES

/var/run/utmp
: user accounting database, configured through **_PATH_UTMP** in <*paths.h*>

/var/log/wtmp
: user accounting log file, configured through **_PATH_WTMP** in <*paths.h*>

ATTRIBUTES

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

Interface	Attribute	Value
login(), **logout**()	Thread safety	MT-Unsafe race:utent sig:ALRM timer

In the above table, *utent* in *race:utent* signifies that if any of the functions **setutent**(3), **getutent**(3), or **endutent**(3) are used in parallel in different threads of a program, then data races could occur. **login**() and **logout**() calls those functions, so we use race:utent to remind users.

STANDARDS

Not in POSIX.1. Present on the BSDs.

NOTES

Note that the member *ut_user* of *struct utmp* is called *ut_name* in BSD. Therefore, *ut_name* is defined as an alias for *ut_user* in <*utmp.h*>.

SEE ALSO

getutent(3), **utmp**(5)