NAME

login − begin session on the system

SYNOPSIS

login [−p] [−h *host*] [*username*] [*ENV=VAR...*]

login [−p] [−h *host*] −f *username*

login [−p] −r *host*

DESCRIPTION

The **login** program is used to establish a new session with the system. It is normally invoked automatically by responding to the *login:* prompt on the user's terminal. **login** may be special to the shell and may not be invoked as a sub−process. When called from a shell, **login** should be executed as **exec login** which will cause the user to exit from the current shell (and thus will prevent the new logged in user to return to the session of the caller). Attempting to execute **login** from any shell but the login shell will produce an error message.

The user is then prompted for a password, where appropriate. Echoing is disabled to prevent revealing the password. Only a small number of password failures are permitted before **login** exits and the communications link is severed.

If password aging has been enabled for your account, you may be prompted for a new password before proceeding. You will be forced to provide your old password and the new password before continuing. Please refer to **passwd**(1) for more information.

Your user and group ID will be set according to their values in the /etc/passwd file. The value for **$HOME**, **$SHELL**, **$PATH**, **$LOGNAME**, and **$MAIL** are set according to the appropriate fields in the password entry. Ulimit, umask and nice values may also be set according to entries in the GECOS field.

On some installations, the environmental variable **$TERM** will be initialized to the terminal type on your tty line, as specified in /etc/ttytype.

An initialization script for your command interpreter may also be executed. Please see the appropriate manual section for more information on this function.

A subsystem login is indicated by the presence of a "*" as the first character of the login shell. The given home directory will be used as the root of a new file system which the user is actually logged into.

The **login** program is NOT responsible for removing users from the utmp file. It is the responsibility of **getty**(8) and **init**(8) to clean up apparent ownership of a terminal session. If you use **login** from the shell prompt without **exec**, the user you use will continue to appear to be logged in even after you log out of the "subsession".

OPTIONS

−f

Do not perform authentication, user is preauthenticated.

Note: In that case, *username* is mandatory.

−h

Name of the remote host for this login.

−p

Preserve environment.

−r

Perform autologin protocol for rlogin.

The **−r**, **−h** and **−f** options are only used when **login** is invoked by root.

CAVEATS

This version of **login** has many compilation options, only some of which may be in use at any particular site.

The location of files is subject to differences in system configuration.

The **login** program is NOT responsible for removing users from the utmp file. It is the responsibility of **getty**(8) and **init**(8) to clean up apparent ownership of a terminal session. If you use **login** from the shell prompt without **exec**, the user you use will continue to appear to be logged in even after you log out of the "subsession".

As with any program, **login**'s appearance can be faked. If non−trusted users have physical access to a machine, an attacker could use this to obtain the password of the next person coming to sit in front of the machine. Under Linux, the SAK mechanism can be used by users to initiate a trusted path and prevent this kind of attack.

CONFIGURATION

The following configuration variables in /etc/login.defs change the behavior of this tool:

CONSOLE_GROUPS (string)
List of groups to add to the user's supplementary groups set when logging in on the console (as determined by the CONSOLE setting). Default is none.

Use with caution − it is possible for users to gain permanent access to these groups, even when not logged in on the console.

DEFAULT_HOME (boolean)
Indicate if login is allowed if we can't cd to the home directory. Default is no.

If set to *yes*, the user will login in the root (/) directory if it is not possible to cd to her home directory.

ENV_PATH (string)
If set, it will be used to define the PATH environment variable when a regular user login. The value is a colon separated list of paths (for example */bin:/usr/bin*) and can be preceded by *PATH=*. The default value is *PATH=/bin:/usr/bin*.

ENV_SUPATH (string)
If set, it will be used to define the PATH environment variable when the superuser login. The value is a colon separated list of paths (for example */sbin:/bin:/usr/sbin:/usr/bin*) and can be preceded by *PATH=*. The default value is *PATH=/sbin:/bin:/usr/sbin:/usr/bin*.

ERASECHAR (number)
Terminal ERASE character (*010* = backspace, *0177* = DEL).

The value can be prefixed "0" for an octal value, or "0x" for an hexadecimal value.

FAIL_DELAY (number)
Delay in seconds before being allowed another attempt after a login failure.

FAKE_SHELL (string)
If set, **login** will execute this shell instead of the users' shell specified in /etc/passwd.

HUSHLOGIN_FILE (string)
If defined, this file can inhibit all the usual chatter during the login sequence. If a full pathname is specified, then hushed mode will be enabled if the user's name or shell are found in the file. If not a full pathname, then hushed mode will be enabled if the file exists in the user's home directory.

KILLCHAR (number)
Terminal KILL character (*025* = CTRL/U).

The value can be prefixed "0" for an octal value, or "0x" for an hexadecimal value.

LOGIN_RETRIES (number)
Maximum number of login retries in case of bad password.

This will most likely be overridden by PAM, since the default pam_unix module has its own built in of

3 retries. However, this is a safe fallback in case you are using an authentication module that does not enforce PAM_MAXTRIES.

LOGIN_TIMEOUT (number)
Max time in seconds for login.

LOG_OK_LOGINS (boolean)
Enable logging of successful logins.

LOG_UNKFAIL_ENAB (boolean)
Enable display of unknown usernames when login failures are recorded.

Note: logging unknown usernames may be a security issue if an user enter her password instead of her login name.

TTYGROUP (string), **TTYPERM** (string)
The terminal permissions: the login tty will be owned by the **TTYGROUP** group, and the permissions will be set to **TTYPERM**.

By default, the ownership of the terminal is set to the user's primary group and the permissions are set to *0600*.

TTYGROUP can be either the name of a group or a numeric group identifier.

If you have a **write** program which is "setgid" to a special group which owns the terminals, define TTYGROUP to the group number and TTYPERM to 0620. Otherwise leave TTYGROUP commented out and assign TTYPERM to either 622 or 600.

TTYTYPE_FILE (string)
If defined, file which maps tty line to TERM environment parameter. Each line of the file is in a format something like "vt100 tty01".

USERGROUPS_ENAB (boolean)
If set to *yes*, **userdel** will remove the user's group if it contains no more members, and **useradd** will create by default a group with the name of the user.

FILES

/var/run/utmp
List of current login sessions.

/var/log/wtmp
List of previous login sessions.

/etc/passwd
User account information.

/etc/shadow
Secure user account information.

/etc/motd
System message of the day file.

/etc/nologin
Prevent non−root users from logging in.

/etc/ttytype
List of terminal types.

$HOME/.hushlogin
Suppress printing of system messages.

/etc/login.defs
Shadow password suite configuration.

SEE ALSO

mail(1), **passwd**(1), **sh**(1), **su**(1), **login.defs**(5), **nologin**(5), **passwd**(5), **securetty**(5), **getty**(8).