NAME

su − run a command with substitute user and group ID

SYNOPSIS

su [options] [**−**] [*user* [*argument*...]]

DESCRIPTION

su allows to run commands with a substitute user and group ID.

When called without arguments, **su** defaults to running an interactive shell as *root*.

For backward compatibility, **su** defaults to not change the current directory and to only set the environment variables **HOME** and **SHELL** (plus **USER** and **LOGNAME** if the target *user* is not root). It is recommended to always use the **−−login** option (instead of its shortcut **−**) to avoid side effects caused by mixing environments.

This version of **su** uses PAM for authentication, account and session management. Some configuration options found in other **su** implementations, such as support for a wheel group, have to be configured via PAM.

su is mostly designed for unprivileged users, the recommended solution for privileged users (e.g. scripts executed by root) is to use non-set-user-ID command runuser(1) that does not require authentication and provide separate PAM configuration. If the PAM session is not required at all then the recommend solution is to use command setpriv(1).

OPTIONS

−c, **−−command**=*command*
: Pass *command* to the shell with the **−c** option.

−f, **−−fast**
: Pass **−f** to the shell, which may or may not be useful, depending on the shell.

−g, **−−group**=*group*
: Specify the primary group. This option is available to the root user only.

−G, **−−supp−group**=*group*
: Specify a supplemental group. This option is available to the root user only. The first specified supplementary group is also used as a primary group if the option **−−group** is unspecified.

−, **−l**, **−−login**
: Start the shell as a login shell with an environment similar to a real login:

  - clears all the environment variables except **TERM** and variables specified by **−−whitelist−environment**
  - initializes the environment variables **HOME**, **SHELL**, **USER**, **LOGNAME**, and **PATH**
  - changes to the target user's home directory
  - sets argv[0] of the shell to '**−**' in order to make the shell a login shell

−m, **−p**, **−−preserve−environment**
: Preserve the entire environment, i.e. it does not set **HOME**, **SHELL**, **USER** nor **LOGNAME**. This option is ignored if the option **−−login** is specified.

−P, **−−pty**
: Create pseudo-terminal for the session. The independent terminal provides better security as user does not share terminal with the original session. This allow to avoid TIOCSTI ioctl terminal injection and another security attacks against terminal file descriptors. The all session is also possible to move to background (e.g. "su --pty - username -c application &"). If the pseudo-terminal is enabled then su command works as a proxy between the sessions (copy stdin and stdout).

  This feature is EXPERIMENTAL for now and may be removed in the next releases.

−s, **−−shell**=*shell*
Run the specified *shell* instead of the default. The shell to run is selected according to the following rules, in order:

- the shell specified with **−−shell**
- the shell specified in the environment variable **SHELL**, if the **−−preserve−environment** option is used
- the shell listed in the passwd entry of the target user
- /bin/sh

If the target user has a restricted shell (i.e. not listed in /etc/shells), the **−−shell** option and the **SHELL** environment variables are ignored unless the calling user is root.

−−session−command=*command*
Same as **−c** but do not create a new session. (Discouraged.)

−w, **−−whitelist−environment**=*list*
Don't reset environment variables specified in comma separated *list* when clears environment for **−−login**. The whitelist is ignored for the environment variables **HOME**, **SHELL**, **USER**, **LOGNAME**, and **PATH**.

−V, **−−version**
Display version information and exit.

−h, **−−help**
Display help text and exit.

SIGNALS

Upon receiving either **SIGINT**, **SIGQUIT** or **SIGTERM**, **su** terminates its child and afterwards terminates itself with the received signal. The child is terminated by SIGTERM, after unsuccessful attempt and 2 seconds of delay the child is killed by SIGKILL.

CONFIG FILES

su reads the */etc/default/su* and */etc/login.defs* configuration files. The following configuration items are relevant for su(1):

FAIL_DELAY (number)
Delay in seconds in case of an authentication failure. The number must be a non-negative integer.

ENV_PATH (string)
Defines the PATH environment variable for a regular user. The default value is */usr/local/bin:/bin:/usr/bin*.

ENV_ROOTPATH (string)
ENV_SUPATH (string)
Defines the PATH environment variable for root. The default value is */usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin*.

ALWAYS_SET_PATH (boolean)
If set to *yes* and −−login and −−preserve−environment were not specified **su** initializes **PATH**.

The environment variable PATH may be different on systems where /bin and /sbin are merged into /usr.

EXIT STATUS

su normally returns the exit status of the command it executed. If the command was killed by a signal, **su** returns the number of the signal plus 128.

Exit status generated by **su** itself:

1 Generic error before executing the requested command

126	The requested command could not be executed
127	The requested command was not found

FILES

/etc/pam.d/su	default PAM configuration file
/etc/pam.d/su-l	PAM configuration file if −−login is specified
/etc/default/su	command specific logindef config file
/etc/login.defs	global logindef config file

NOTES

For security reasons **su** always logs failed log-in attempts to the btmp file, but it does not write to the lastlog file at all. This solution allows to control **su** behavior by PAM configuration. If you want to use the pam_lastlog module to print warning message about failed log-in attempts then the pam_lastlog has to be configured to update the lastlog file as well. For example by:

```
session  required  pam_lastlog.so nowtmp
```

SEE ALSO

setpriv(1), login.defs(5), shells(5), **pam(8), runuser(8)**

HISTORY

This **su** command was derived from coreutils' **su**, which was based on an implementation by David MacKenzie. The util-linux has been refactored by Karel Zak.

AVAILABILITY

The su command is part of the util-linux package and is available from Linux Kernel Archive.