NAME

newgrp – log in to a new group

SYNOPSIS

Command

/usr/bin/newgrp [– | –l] [*group*]

sh Built-in

newgrp [*argument*]

ksh Built-in

† **newgrp** [*argument*]

AVAILABILITY

SUNWcsu

DESCRIPTION

Command

The **newgrp** command logs a user into a new group by changing a user's real and effective group ID. The user remains logged in and the current directory is unchanged. The execution of **newgrp** always replaces the current shell with a new shell, even if the command terminates with an error (unknown group).

Any variable that is not exported is reset to null or its default value. Exported variables retain their values. System variables (such as **PS1**, **PS2**, **PATH**, **MAIL**, and **HOME**), are reset to default values unless they have been exported by the system or the user. For example, when a user has a primary prompt string (**PS1**) other than **$** (default) and has not exported **PS1**, the user's **PS1** will be set to the default prompt string **$**, even if **newgrp** terminates with an error. Note that the shell command **export** (see **sh**(1) and **set**(1)) is the method to export variables so that they retain their assigned value when invoking new shells.

With no operands and options, **newgrp** changes the user's group IDs (real and effective) back to the group specified in the user's password file entry. This is a way to exit the effect of an earlier **newgrp** command.

A password is demanded if the group has a password and the user is not listed in **/etc/group** as being a member of that group. The only way to create a password for a group is to use **passwd**(1), then cut and paste the password from **/etc/shadow** to **/etc/group**. Group passwords are antiquated and not often used.

sh Built-in

Equivalent to **exec newgrp** *argument* where *argument* represents the options and/or operand of the **newgrp** command.

ksh Built-in

Equivalent to **exec /bin/newgrp** *argument* where *argument* represents the options and/or operand of the **newgrp** command.

On this man page, **ksh**(1) commands that are preceded by one or two † (daggers) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.

4. Words, following a command preceded by †† that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the **=** sign and word splitting and file name generation are not performed.

OPTIONS

The following option is supported:

–l

– change the environment to what would be expected if the user actually logged in again as a member of the new group.

OPERANDS

The following operand is supported:

group A group name from the group database or a non-negative numeric group ID. Specifies the group ID to which the real and effective group IDs will be set. If *group* is a non-negative numeric string and exists in the group database as a group name (see **getgrnam**(3C)), the numeric group ID associated with that group name will be used as the group ID.

argument **sh** and **ksh** only. Options and/or operand of the **newgrp** command.

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **newgrp**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

If **newgrp** succeeds in creating a new shell execution environment, whether or not the group identification was changed successfully, the exit status will be the exit status of the shell. Otherwise, the following exit value is returned:

>0 An error occurred.

FILES

/etc/group system's group file

/etc/passwd system's password file

SEE ALSO

login(1), **ksh**(1), **set**(1), **sh**(1), **intro**(2), **getgrnam**(3C), **group**(4), **passwd**(4), **environ**(5)