

USERMOD(8)

System Management Commands

USERMOD(8)

NAME

usermod - modify a user account

SYNOPSIS

usermod [options] LOGIN

DESCRIPTION

The **usermod** command modifies the system account files.

OPTIONS

The options which apply to the **usermod** command are:

-a, --append

Add the user to the supplementary group(s). Use only with the **-G** option.

-b, --badname

Allow names that do not conform to standards.

-c, --comment COMMENT

update the comment field of the user in `/etc/passwd`, which is

-d, --home HOME_DIR

The user's new login directory.

If the **-m** option is given, the contents of the current home directory will be moved to the new home directory, which is created if it does not already exist. If the current home directory does not exist the new home directory will not be created.

-e, --expiredate EXPIRE_DATE

The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD. Integers as input are interpreted as days after 1970-01-01.

An input of **-1** or an empty string will blank the account expiration field in the shadow password file. The account will remain available with no date limit.

This option requires a **/etc/shadow** file. A **/etc/shadow** entry will be created if there were none.

-f, --inactive INACTIVE

defines the number of days after the password exceeded its maximum age during which the user may still login by immediately replacing

is stored in the shadow password file. An input of 0 will disable an expired password with no delay. An input of -1 will blank the respective field in the shadow password file. See shadow(5) for more information.

This option requires a /etc/shadow file. A /etc/shadow entry will be created if there were none.

-g, --gid GROUP

The name or numerical ID of the user's new primary group. The group must exist.

Any file from the user's home directory owned by the previous primary group of the user will be owned by this new group.

The group ownership of files outside of the user's home directory must be fixed manually.

The change of the group ownership of files inside of the user's home directory is also not done if the home dir owner uid is different from the current or new user id. This is a safety measure for special home directories such as /.

-G, --groups GROUP1[,GROUP2,...[,GROUPN]]]

Each group is separated from the next by a comma, with no intervening whitespace. The groups must exist.

If the user is currently a member of a group which is not listed, the user will be removed from the group. This behaviour can be changed via the `-a` option, which appends the user to the current supplementary group list.

-l, --login NEW_LOGIN

The name of the user will be changed from `LOGIN` to `NEW_LOGIN`. Nothing else is changed. In particular, the user's home directory or mail spool should probably be renamed manually to reflect the new login name.

-L, --lock

Lock a user's password. This puts a `!` in front of the encrypted password, effectively disabling the password. You can't use this option with `-p` or `-U`.

Note: if you wish to lock the account (not only access with a password), you should also set the `EXPIRE_DATE` to 1.

-m, --move-home

moves the content of the user's home directory to the new location.

will not be created.

This option is only valid in combination with the `-d` (or `--home`) option.

`usermod` will try to adapt the ownership of the files and to copy the modes, ACL and extended attributes, but manual changes might be needed afterwards.

`-o, --non-unique`

allows to change the user ID to a non-unique value.

This option is only valid in combination with the `-u` option. As a user identity serves as key to map between users on one hand and permissions, file ownerships and other aspects that determine the system's behavior on the other hand, more than one login name will access the account of the given UID.

`-p, --password PASSWORD`

defines a new password for the user. `PASSWORD` is expected to be encrypted, as returned by `crypt(3)`.

Note: Avoid this option on the command line because the password (or encrypted password) will be visible by users listing the

The password will be written in the local `/etc/passwd` or `/etc/shadow` file. This might differ from the password database configured in your PAM configuration.

You should make sure the password respects the system's password policy.

-r, --remove

Remove the user from named supplementary group(s). Use only with the `-G` option.

-R, --root CHROOT_DIR

Apply changes in the `CHROOT_DIR` directory and use the configuration files from the `CHROOT_DIR` directory. Only absolute paths are supported.

-P, --prefix PREFIX_DIR

Apply changes within the directory tree starting with `PREFIX_DIR` and use as well the configuration files located there. This option does not chroot and is intended for preparing a cross-compilation target. Some limitations: NIS and LDAP users/groups are not verified. PAM authentication is using the host files. No SELINUX support.

-s, --shell SHELL

changes the user's login shell. An empty string for SHELL blanks the field in `/etc/passwd` and logs the user into the system's default shell.

-u, --uid UID

The new value of the user's ID.

This value must be unique, unless the `-o` option is used. The value must be non-negative.

The user's mailbox, and any files which the user owns and which are located in the user's home directory will have the file user ID changed automatically.

The ownership of files outside of the user's home directory must be fixed manually.

The change of the user ownership of files inside of the user's home directory is also not done if the home dir owner uid is different from the current or new user id. This is a safety measure for special home directories such as `/`.

No checks will be performed with regard to the `UID_MIN`, `UID_MAX`,

-U, --unlock

Unlock a user's password. This removes the '!' in front of the encrypted password. You can't use this option with -p or -L.

Note: if you wish to unlock the account (not only access with a password), you should also set the `EXPIRE_DATE` (for example to 99999, or to the `EXPIRE` value from `/etc/default/useradd`).

-v, --add-subuids FIRST-LAST

Add a range of subordinate uids to the user's account.

This option may be specified multiple times to add multiple ranges to a user's account.

No checks will be performed with regard to `SUB_UID_MIN`, `SUB_UID_MAX`, or `SUB_UID_COUNT` from `/etc/login.defs`.

-V, --del-subuids FIRST-LAST

Remove a range of subordinate uids from the user's account.

This option may be specified multiple times to remove multiple ranges to a user's account. When both `--del-subuids` and `--add-subuids` are specified, the removal of all subordinate uid

No checks will be performed with regard to SUB_UID_MIN, SUB_UID_MAX, or SUB_UID_COUNT from /etc/login.defs.

-w, --add-subgids FIRST-LAST

Add a range of subordinate gids to the user's account.

This option may be specified multiple times to add multiple ranges to a user's account.

No checks will be performed with regard to SUB_GID_MIN, SUB_GID_MAX, or SUB_GID_COUNT from /etc/login.defs.

-W, --del-subgids FIRST-LAST

Remove a range of subordinate gids from the user's account.

This option may be specified multiple times to remove multiple ranges to a user's account. When both --del-subgids and --add-subgids are specified, the removal of all subordinate gid ranges happens before any subordinate gid range is added.

No checks will be performed with regard to SUB_GID_MIN, SUB_GID_MAX, or SUB_GID_COUNT from /etc/login.defs.

defines the SELinux user to be mapped with LOGIN. An empty string ("") will remove the respective entry (if any). Note that the shadow system doesn't store the selinux-user, it uses semanage(8) for that.

CAVEATS

You must make certain that the named user is not executing any processes when this command is being executed if the user's numerical user ID, the user's name, or the user's home directory is being changed. usermod checks this on Linux. On other operating systems it only uses utmp to check if the user is logged in.

You must change the owner of any crontab files or at jobs manually.

You must make any changes involving NIS on the NIS server.

CONFIGURATION

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

LASTLOG_UID_MAX (number)

Highest user ID number for which the lastlog entries should be updated. As higher user IDs are usually tracked by remote user identity and authentication services there is no need to create a

No `LASTLOG_UID_MAX` option present in the configuration means that there is no user ID limit for writing lastlog entries.

`MAIL_DIR` (string)

The mail spool directory. This is needed to manipulate the mailbox when its corresponding user account is modified or deleted. If not specified, a compile-time default is used. The parameter `CREATE_MAIL_SPOOL` in `/etc/default/useradd` determines whether the mail spool should be created.

`MAIL_FILE` (string)

Defines the location of the users mail spool files relatively to their home directory.

The `MAIL_DIR` and `MAIL_FILE` variables are used by `useradd`, `usermod`, and `userdel` to create, move, or delete the user's mail spool.

`MAX_MEMBERS_PER_GROUP` (number)

Maximum members per group entry. When the maximum is reached, a new group entry (line) is started in `/etc/group` (with the same name, same password, and same GID).

The default value is 0, meaning that there are no limits in the

This feature (split group) permits to limit the length of lines in the group file. This is useful to make sure that lines for NIS groups are not larger than 1024 characters.

If you need to enforce such limit, you can use 25.

Note: split groups may not be supported by all tools (even in the Shadow toolsuite). You should not use this variable unless you really need it.

SUB_GID_MIN (number), SUB_GID_MAX (number), SUB_GID_COUNT (number)

If `/etc/subuid` exists, the commands `useradd` and `newusers` (unless the user already have subordinate group IDs) allocate `SUB_GID_COUNT` unused group IDs from the range `SUB_GID_MIN` to `SUB_GID_MAX` for each new user.

The default values for `SUB_GID_MIN`, `SUB_GID_MAX`, `SUB_GID_COUNT` are respectively `100000`, `600100000` and `65536`.

SUB_UID_MIN (number), SUB_UID_MAX (number), SUB_UID_COUNT (number)

If `/etc/subuid` exists, the commands `useradd` and `newusers` (unless the user already have subordinate user IDs) allocate `SUB_UID_COUNT` unused user IDs from the range `SUB_UID_MIN` to `SUB_UID_MAX` for each

The default values for SUB_UID_MIN, SUB_UID_MAX, SUB_UID_COUNT are respectively 100000, 600100000 and 65536.

FILES

/etc/group

Group account information

/etc/gshadow

Secure group account informatio.

/etc/login.defs

Shadow password suite configuration

/etc/passwd

User account information

/etc/shadow

Secure user account information

/etc/subgid

Per user subordinate group IDs

/etc/subuid

Linux UBUNTU Manual Pages

SEE ALSO

**chfn(1), chsh(1), passwd(1), crypt(3), gpasswd(8), groupadd(8),
groupdel(8), groupmod(8), login.defs(5), subgid(5), subuid(5),
useradd(8), userdel(8).**

shadow-utils 4.13

05/30/2024

USERMOD(8)