

NAME

chage – change user password expiry information

SYNOPSIS

chage [*options*] *LOGIN*

DESCRIPTION

The **chage** command changes password expiration information for a user. It sets the number of days between password changes and the date of the last password change. This information is used by the system to determine when the user must change their password.

OPTIONS

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

-d, --lastday *LAST_DAY*

Set the date when the password was last changed. The value can be specified as a date in the YYYY-MM-DD format or as a number of days since 1970-01-01. The date is interpreted using the UTC timezone. If the *LAST_DAY* is set to 0, the user is forced to change their password upon the next login.

Passing the value -1 or an empty string as the *LAST_DAY* clears the value and removes the password change requirement.

-E, --expiredate *EXPIRE_DATE*

Set the date on which the user's password expires and their account will no longer be accessible. The value can be specified as a date in the YYYY-MM-DD format or as a number of days since 1970-01-01. The date is interpreted using the UTC timezone. If the password expires, the user must contact the system administrator to regain access to the system.

For example, the following command sets an account to expire in 180 days:

```
chage -E $(date -d +180days +%F)
```

Passing the value -1 or an empty string as the *EXPIRE_DATE* removes the account expiration date.

-h, --help

Display help message and exit.

-i, --iso8601

When printing dates, use YYYY-MM-DD format.

-I, --inactive *INACTIVE*

Set the number of days of inactivity after a password has expired before the account is locked. The *INACTIVE* option is the number of days of inactivity. A user whose account is locked must contact the system administrator before being able to use the system again.

Passing the number -1 as the *INACTIVE* will remove an account's inactivity.

-l, --list

Show account aging information.

-m, --mindays *MIN_DAYS*

Set the minimum number of days between password changes to *MIN_DAYS*. A value of zero for this field indicates that the user may change their password at any time.

-M, --maxdays *MAX_DAYS*

Set the maximum number of days during which a password is valid. When *MAX_DAYS* plus *LAST_DAY* is less than the current day, the user will be required to change their password before being able to use their account. This occurrence can be planned for in advance by use of the **-W** option,

which provides the user with advance warning.

Passing the number `-1` as `MAX_DAYS` will remove checking a password's validity.

-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. No SELINUX support.

-P, --prefix *PREFIX_DIR*

Apply changes to configuration files under the root filesystem found under the directory *PREFIX_DIR*. This option does not chroot and is intended for preparing a cross-compilation target. Some limitations: NIS and LDAP users/groups are not verified. No SELINUX support.

-W, --warndays *WARN_DAYS*

Set the number of days of warning before a password change is required. The *WARN_DAYS* option is the number of days prior to the password expiring that a user will be warned their password is about to expire.

If none of the options are selected, **chage** operates in an interactive fashion, prompting the user with the current values for all of the fields. Enter the new value to change the field, or leave the line blank to use the current value. The current value is displayed between a pair of `[ ]` marks.

NOTE

The **chage** program requires a shadow password file to be available.

The chage program will report only the information from the shadow password file. This implies that configuration from other sources (e.g. LDAP or empty password hash field from the passwd file) that affect the user's login will not be shown in the chage output.

The **chage** program will also not report any inconsistency between the shadow and passwd files (e.g. missing x in the passwd file). The **pwck** can be used to check for this kind of inconsistencies.

The **chage** command is restricted to the root user, except for the **-l** option, which may be used by an unprivileged user to determine when their password or account is due to expire.

CONFIGURATION

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

FILES

`/etc/passwd`

User account information.

`/etc/shadow`

Secure user account information.

EXIT VALUES

The **chage** command exits with the following values:

0

success

1

permission denied

2

invalid command syntax

15

can't find the shadow password file

SEE ALSO

passwd(5), **shadow(5)**.