

# Reference guide: Linux

## Google Cybersecurity Certificate

---

### Table of contents

[Navigate the file system](#)

[Read files](#)

[Manage the file system](#)

[Filter content](#)

[Manage users and their permissions](#)

[Get help in Linux](#)

---

### Navigate the file system

The following Linux commands are helpful when navigating the file system.

#### **cd**

Navigates between directories

```
cd reports
```

Navigates from the current working directory to its subdirectory `reports`

```
cd /home/analyst/reports
```

Navigates to the `reports` directory; the full path is required when `reports` is not a subdirectory of the current working directory

```
cd ..
```

Navigates to the directory that is one level above the current working directory

#### **ls**

Displays the names of the files and directories

```
ls
```

Displays the names of the files and directories in the current working directory

```
ls /home/analyst/reports
```

Displays the names of the files and directories in the `reports` directory; providing an argument that specifies the path to a directory is necessary to display the contents of a directory other than the user's current working directory

```
ls -a
```

Displays hidden files when displaying the names of files and directories inside the current working directory

```
ls -l
```

Displays permissions to files and directories in the current working directory; also displays other additional information, including owner name, group, file size, and the time of the last modification

```
ls -la
```

Displays permissions to files and directories in the current working directory, including hidden files; also displays other additional information, including owner name, group, file size, and the time of last modification

## **pwd**

Prints the working directory to the screen

```
pwd
```

Prints the working directory to the screen, such as `/home/analyst`

## **whoami**

Returns the username of the current user

```
whoami
```

Returns the username of the current user, such as `analyst` or `fgarcia`

## Read files

The following Linux commands are helpful when reading files.

### cat

Displays the content of a file

```
cat updates.txt
```

Displays the content of the `updates.txt` file

### head

Displays just the beginning of a file, by default 10 lines

```
head updates.txt
```

Displays only the first 10 lines of the `updates.txt` file

```
head -n 5 updates.txt
```

Displays only the first five lines of the `updates.txt` file; the `-n` option allows users to specify the number of lines to return

### less

Returns the content of a file one page at a time

```
less updates.txt
```

Returns the content of `updates.txt` one page at a time; the `less` command changes the terminal window to a display that allows users to easily move forward and backward through content

### tail

Displays just the end of a file, by default 10 lines

```
tail updates.txt
```

Displays only the last 10 lines of the `updates.txt` file

```
tail -n 5 updates.txt
```

returns only the last five lines of the `updates.txt` file; the `-n` option allows users to specify the number of lines to return

## Manage the file system

The following Linux commands are helpful when managing the file system.

### **cp**

Copies a file or directory into a new location; the file will not be removed from the previous location

```
cp permissions.txt /home/analyst/logs
```

Copies the `permissions.txt` file from the user's current working directory to the `logs` directory

### **mkdir**

Creates a new directory

```
mkdir network
```

Creates a new directory named `network` in the user's current working directory

```
mkdir /home/analyst/logs network
```

Creates a new directory named `network` in the `logs` directory; the full path is required when `logs` is not a subdirectory of the current directory

### **mv**

Moves a file or directory to a new location; the file is also removed from the previous location

```
mv permissions.txt /home/analyst/logs
```

Moves the `permissions.txt` file from the user's current working directory to the `logs` directory

```
mv permissions.txt perm.txt
```

Moves the `permissions.txt` file from the user's current working directory to the new file name `perm.txt` in the user's current working directory; this results in renaming the `permissions.txt` file as `perm.txt`

## nano

Opens or creates a file in the nano command-line file editor

```
nano permissions.txt
```

Opens an existing `permissions.txt` file in the nano file editor, or creates the `permissions.txt` file in the nano file editor if it doesn't already exist in the current working directory

## rm

Removes, or deletes, a file

```
rm permissions.txt
```

removes the `permissions.txt` file from the user's current working directory

```
rm home/analyst/reports/permissions.txt
```

Removes the `permissions.txt` file from the `reports` directory; the full path is required if the user's current working directory is not `reports`

## rmdir

Removes, or deletes, a directory; only removes directories if they are empty

```
rmdir network
```

Removes the empty `network` subdirectory of the user's current working directory from the file system

```
rmdir /home/analyst/logs/network
```

Removes the empty `network` directory from the file system; the full path is required when `network` is not a subdirectory of the current directory

## touch

Creates a new file

```
touch permissions.txt
```

Creates a new file named `permissions.txt` in the user's current working directory

```
touch /home/analyst/reports/permissions.txt
```

Creates a new file named `permissions.txt` in the `reports` directory; the full path is required if the user wants to create `permissions.txt` in any directory other than the current working directory

## Filter content

The following Linux commands are helpful when filtering content.

### find

Searches for directories and files that meet specified criteria

```
find /home/analyst/projects
```

Searches for all files starting at the `projects` directory

```
find /home/analyst/projects -name "*log*"
```

Searches for all files in the `projects` directory that contain the word `log` in the file name; the `-name` option searches for a specified string and is case-sensitive; the `*` wildcard represents zero or more unknown characters

```
find /home/analyst/projects -iname "*log*"
```

Searches for all files in the `projects` directory that contain the word `log` in the file name; the `-iname` option searches for a specified string and is not case-sensitive; the `*` wildcard represents zero or more unknown characters

```
find /home/analyst/projects -mtime -3
```

Searches for all files in the `projects` directory that have been modified within the past three days; the `-mtime` option bases its search for files or directories that were modified on days

```
find /home/analyst/projects -mmin -15
```

Searches for all files in the `projects` directory that have been modified within the past 15 minutes; the `-mmin` option bases its search for files or directories that were modified on minutes

## grep

Searches a specified file and returns all lines in the file containing a specified string

```
grep OS updates.txt
```

Searches the `updates.txt` file and returns all lines containing the string `OS`

## | (piping)

Sends the standard output of one command as standard input to another command for further processing; accessed using the pipe character (`|`)

```
ls /home/analyst/reports | grep users
```

Redirects the standard output of `ls /home/analyst/reports` to be standard input for the `grep users` command, meaning that `grep users` identifies files and subdirectories in the `/home/analyst/reports` directory that contain the string `users` within their file name

## Manage users and their permissions

The following Linux commands are helpful when managing user permissions. (Also review the subentries for `ls -l` and `ls -la` in the `ls` entry of the [Navigate the file system](#) section.)

## chmod

Changes permissions on files and directories

```
chmod u+rwx,g+rwx,o+rwx login_sessions.txt
```

Changes user (`u`), group (`g`), and other (`o`) permissions to add (+) read (`r`), write (`w`), and execute (`x`) permissions for the `login_sessions.txt` file

```
chmod g-rw bonuses.txt
```

Changes the group (`g`) permissions to remove (–) read (`r`) and write (`w`) permissions for the `bonuses.txt` file

```
chmod u=r,g=r,o=r login_sessions.txt
```

Changes user (u), group (g), and other (o) permissions to assign (=) read (r) permissions for the `login_sessions.txt` file

## chown

Changes ownership of a file or directory; used with `sudo`

```
sudo chown fgarcia access.txt
```

Changes the user owner of the `access.txt` file to `fgarcia`

```
sudo chown :security access.txt
```

Changes the group owner of `access.txt` to `security`; a colon (:) must be entered before the group name

## groupdel

Deletes a group from the system; used with `sudo`

```
sudo groupdel accounting
```

Deletes `accounting` as a group

## sudo

Temporarily grants elevated permissions to specific users; users must be in a sudoers file to use have access to `sudo`

```
sudo useradd fgarcia
```

Grants elevated permissions to the user running this command and so that this user can use the `useradd` command to add `fgarcia` as a new user to the system

## useradd

Adds a user to the system; used with `sudo`

```
sudo useradd fgarcia
```

Adds `fgarcia` as a new user to the system



```
sudo useradd -g security fgarcia
```

Adds `fgarcia` as a new user and uses the `-g` option to set their primary group as `security`

```
sudo useradd -G finance,admin fgarcia
```

Adds `fgarcia` as a new user and uses the `-G` option to add them to the supplemental groups of `finance` and `admin`

## userdel

Deletes a user from the system; used with `sudo`

```
sudo userdel fgarcia
```

Deletes `fgarcia` as a user

```
sudo userdel -r fgarcia
```

Deletes `fgarcia` as a user and deletes all files in their home directory

## usermod

Modifies existing user accounts; used with `sudo`

```
sudo usermod -g executive fgarcia
```

Uses the `-g` option to change the existing `fgarcia` user's primary group to the `executive` group

```
sudo usermod -G accounting fgarcia
```

Uses the `-G` option to replace any supplemental groups the existing `fgarcia` user is in with the supplemental `accounting` group; removes all other supplemental groups `fgarcia` is in

```
sudo usermod -a -G marketing fgarcia
```

Uses the `-a -G` options to add the existing `fgarcia` user to the supplemental `marketing` group; does not remove `fgarcia` from other supplemental groups

```
sudo usermod -d /home/garcia_f fgarcia
```

Uses the `-d` option to change the existing `fgarcia` user's home directory to `/home/garcia_f`

```
sudo usermod -L fgarcia
```

Uses the `-L` option to lock the existing `fgarcia` user's account so they cannot log in

```
sudo usermod -l garcia_f fgarcia
```

Uses the `-l` option to change the existing `fgarcia` user's login name to `garcia_f`

## Get help in Linux

The following Linux commands are helpful when getting help in Linux.

### **apropos**

Searches the manual page descriptions for a specified string

```
apropos password
```

Returns the manual pages of commands that contain the keyword `password`

```
apropos -a graph editor
```

Returns the manual pages of commands that contain both the keywords `graph` and `editor`; the `-a` option specifies to only return commands that contain all specified strings

### **man**

Displays information on other commands and how they work; the output is called a "man page," which is short for "manual page"

```
man chown
```

Displays detailed information about `chown` and how it works

### **whatis**

Displays a description of a command on a single line

```
whatis nano
```

Displays the description of `nano` on a single line