



Linux Essentials

Session-3

Using Directories and Listing Files



Kahoot!





Using the Command Line to Get Help



1 Man Pages





Man Pages

man [command]

A man page (short for manual page) is a form of software documentation usually found on a Unix or Unix-like operating system.

if we install a package to do some task, the man Page for that package will typically be installed at the same time. This gives us the ability to take a look at that documentation and make sure that we're using it in a manner consistent with its design.

The man page for a particular command is invoked by preceding the command with **man**.





Man Pages

man ls

```
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if none of
    -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576
        bytes; see SIZE format below

    -B, --ignore-backups
        do not list implied entries ending with ~

    -c
        with -lt: sort by, and show, ctime (time of last modification of file status information); with -l:
        show ctime and sort by name; otherwise: sort by ctime, newest first

    -C
        list entries by columns

    --color[=WHEN]
        colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

    -d, --directory
        list directories themselves, not their contents

Manual page ls(1) line 1 (press h for help or q to quit)
```

NAME

Program or Function name(s) followed by descriptions of functionality.

SYNOPSIS

A short overview of available options

DESCRIPTION

Detailed information about arguments and options.



2 Info Pages



Info Pages

info [command]

Info pages are additional documentation with more robust capability in detail. Info Page normally provides more detailed information about a command than its respective man page. Additionally, Info uses a structure for linking these pages together, and they may be assembled into a larger collection.

The info page for a particular command is invoked by preceding the command with **info**



Info Pages

info echo

```
Next: printf invocation, Up: Printing text
15.1 'echo': Print a line of text
=====

'echo' writes each given STRING to standard output, with a space between
each and a newline after the last one. Synopsis:

    echo [OPTION]... [STRING]...

Due to shell aliases and built-in 'echo' functions, using an
unadorned 'echo' interactively or in a script may get you different
functionality than that described here. Invoke it via 'env' (i.e., 'env
echo ...') to avoid interference from the shell.

The program accepts the following options. Also see *note Common
options::. Options must precede operands, and the normally-special
argument '--' has no special meaning and is treated like any other
STRING.

'-n'
    Do not output the trailing newline.

'-e'
    Enable interpretation of the following backslash-escaped characters
    in each STRING:

        '\a'      alert (bell)
        '\b'      backspace
        '\c'      produce no further output
        '\e'      escape
        '\f'      form feed
        '\n'      newline
        '\r'      carriage return
        '\t'

-----Info: (coreutils)echo invocation, 78 lines --Top-----
Welcome to Info version 6.5. Type H for help, h for tutorial.
```

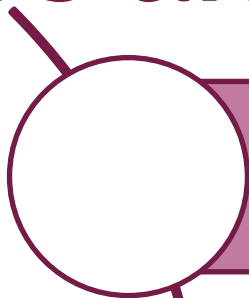


Files and Directories

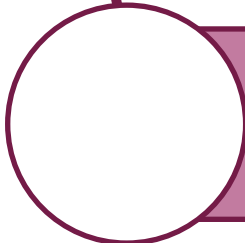




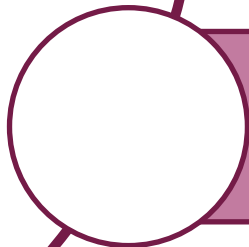
Files and Directories



The file system hierarchy standard (FHS) defines the structure of the file systems on Linux.



In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.



Most of these directories exist in all UNIX, however, they are not considered authoritative for platforms other than Linux.



Files and Directories

`/root` root directory of the root user

`/bin` Essential command binaries

`/boot` Boot loader files

`/dev` Essential device files

`/etc` Host-specific configuration files

`/home` Users' home directories

`/lib` Libraries essential for the binaries

`/mnt` Temporarily mounted filesystems.

`/opt` Optional application packages

`/proc` Contains information about system

`/sbin` Essential system binaries

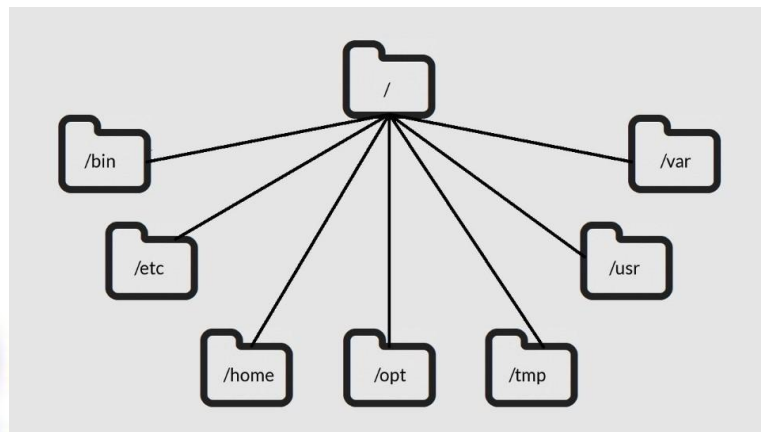
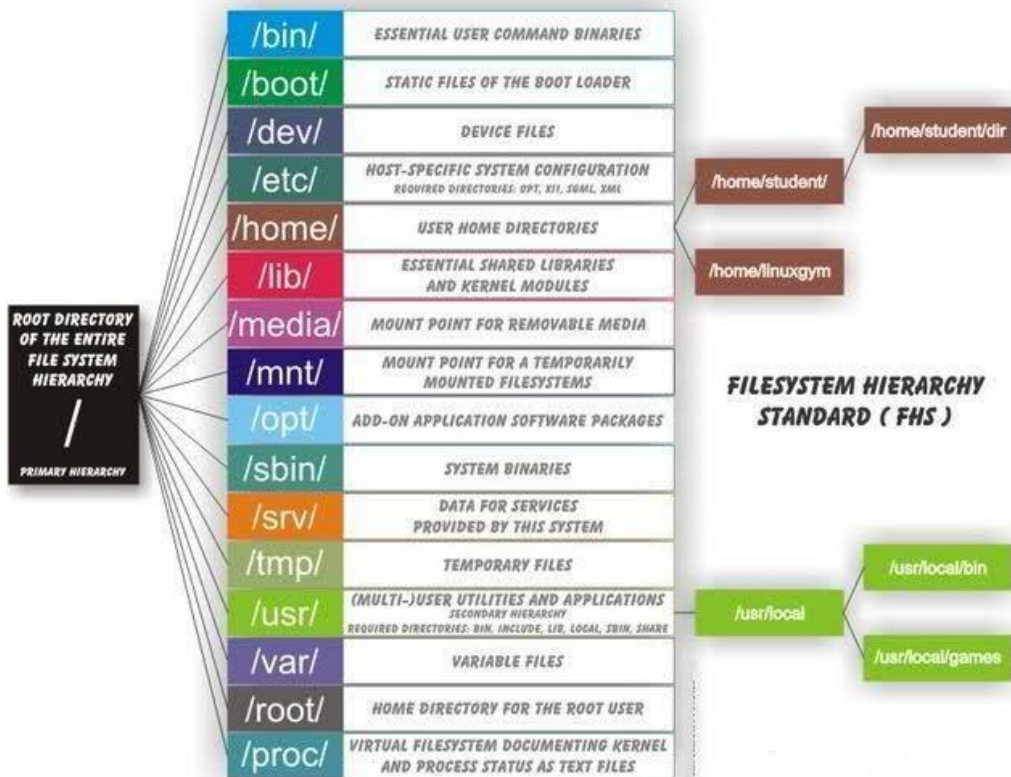
`/tmp` Temporary files

`/var` Variable data files

```
/      the root directory
/bin   user binaries
/boot  static boot files
/dev   device files
/etc   configuration files
/home  home directories
/lib   shared libraries
/mnt   temporary mount points
/opt   optional packages
/proc  kernel and process files
/root  root user home directory
/run   application state files
/sbin  system administration binaries
/srv   service data
/tmp   temporary files
/usr   user binaries
/var   variable data files
```



Files and Directories





File Permission

File Permission



Ownership

User

- A user is the owner of the file.

Group

- A user- group can contain multiple users.

Other

- Any other user who has access to a file.

Permission

Read

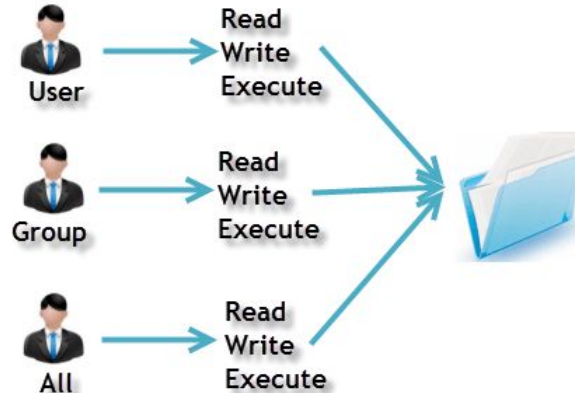
- This permission give you the authority to open and read a file.

Write

- The write permission gives you the authority to modify the contents of a file.

Execute

- you cannot run a program unless the execute permission is set.





File Permission

```
raymond@clarusway-linux: ~  
File Edit View Search Terminal Help  
raymond@clarusway-linux:~$ ls -l lesson.txt  
-rw-rw-r-- 1 raymond adm 8 Mar  2 21:19 lesson.txt  
raymond@clarusway-linux:~$
```

lesson.txt Properties

Basic Permissions Open With

Owner: Me

Access: Read and write

Group: adm

Access: Read and write

Others

Access: Read-only

Execute: ☐ Allow executing file as program

Security context: unknown

File Permission



```
gakeko2018@DESKTOP-JA07K2U:~$ ls
cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ ls -la
.  .. .bash_history .bash_logout .bashrc .local .profile .ssh cert.pem
gakeko2018@DESKTOP-JA07K2U:~$ ls -al
total 12
drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .
drwxr-xr-x 1 root      root      4096 Dec 25 18:19 ..
-rw-r--r-- 1 gakeko2018 gakeko2018 236 Jan 14 12:21 .bash_history
-rw-r--r-- 1 gakeko2018 gakeko2018 220 Dec 25 18:19 .bash_logout
-rw-r--r-- 1 gakeko2018 gakeko2018 3771 Dec 25 18:19 .bashrc
drwxrwxrwx 1 gakeko2018 gakeko2018 4096 Jan 13 09:38 .local
-rw-r--r-- 1 gakeko2018 gakeko2018 807 Dec 25 18:19 .profile
drwx----- 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .ssh
-r----- 1 gakeko2018 gakeko2018 1675 Jan 13 09:38 cert.pem
```

File type and Access Permissions

`-rw-r--r-- 1 gakeko2018 gakeko2018 807 Dec 25 18:19 .profile`

indicates File

`drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .`

d represents directory

User Group Others

r: Read
w: Write
x: Execute

-rw-rw-r--
no execute permission

r = read permission
w = write permission
x = execute permission
- = no permission



File Permission

Changing Permission with chmod Command

We can use the **chmod** command which stands for **change mode**.
we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

```
chmod permissions filename
```

```
chmod u=rwx,g=rx,o=r myfile
```

| Symbol | Permission Type |
|--------|--------------------|
| --- | No Permission |
| --x | Execute |
| -w- | Write |
| -wx | Execute+Write |
| r-- | Read |
| r-x | Read+Execute |
| rw- | Read+Write |
| rwx | Read+Write+Execute |



File Permission

zk@ubuntu:~/ASSIGNMENT/Lessons/HTML\$ ls -l

total 0

! !

| | | | | | | | | |
|-------------|---|----|----|---|-----|---|-------|----------|
| -rwx----- | 1 | zk | zk | 0 | Dec | 7 | 15:39 | cas.txt |
| -----rwx--- | 1 | zk | zk | 0 | Dec | 7 | 15:39 | html.txt |
| -----rwx | 1 | zk | zk | 0 | Dec | 7 | 15:39 | java.txt |
| -rwxrwxrwx | 1 | zk | zk | 0 | Dec | 7 | 17:10 | js.js |
| -rwxrw---x | 1 | zk | zk | 0 | Dec | 7 | 17:11 | k.txt |
| -r--r--r-- | 1 | zk | zk | 0 | Dec | 7 | 17:13 | l.txt |

File Permission



```
root@DESKTOP-4QQ1S5L:~# ls -l
total 0
-rw-rw-rw- 1 root root 0 Dec 29 17:53 file1
-r--r--rwx 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~# chmod 754 file2
root@DESKTOP-4QQ1S5L:~# ls -l file2
-rwxr-xr-- 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~#
```

754 code says;

- Owner can read, write and execute
- User's group can read and execute
- Other can only read

Permissions

read=4;

rwX

4 +2 +1

7

write=2;

r-X

4 +0 +1

5

execute=1

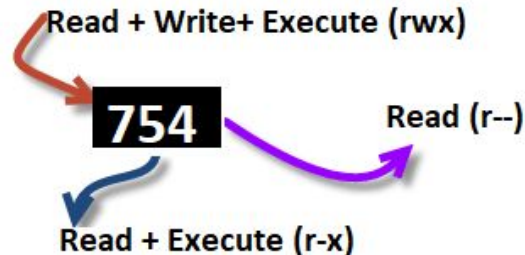
r--

4 +0 +0

4

`chmod u=rwx,g=rx,o=r myfile`

`chmod 754 myfile`



File Permission



| | Owner | | | Group | | | Other Users | | |
|--------|-------|---|---|-------|---|---|-------------|---|---|
| - or d | r | w | x | r | w | x | r | w | x |
| 4 | 2 | 1 | 4 | 2 | 1 | 4 | 2 | 1 | |
| 7 | | | 7 | | | 7 | | | |

Read + Write + Execute (rwx)

764

Read (r--)

Read + Write (rw-)

| | | | | | | | | | |
|-------------|-------------------|-------|------|-------------------|-------|------|------------------|-------|------|
| d | r | w | x | r | - | x | r | - | - |
| | read | write | exec | read | write | exec | read | write | exec |
| File type | Owner permissions | | | Group permissions | | | User permissions | | |
| (directory) | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 2 | 1 |
| | 7 | | | 5 | | | 4 | | |



Set permissions of myfile.txt to;

owner : full access

group : read and execute

others : no access

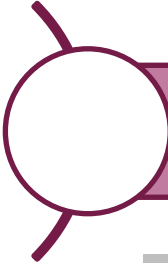


Students, write your response!



Ping & SSH Command

Ping Command



Ping or Packet Internet Groper is a network administration utility used to check the connectivity status between a source and a destination device.

ping host-name/IP

```
ping 54.93.34.220
```

```
gakeko2018@DESKTOP-JA07K2U:~$ ping 54.93.34.220
PING 54.93.34.220 (54.93.34.220) 56(84) bytes of data.
64 bytes from 54.93.34.220: icmp_seq=1 ttl=243 time=62.6 ms
64 bytes from 54.93.34.220: icmp_seq=2 ttl=243 time=93.5 ms
64 bytes from 54.93.34.220: icmp_seq=3 ttl=243 time=66.8 ms
64 bytes from 54.93.34.220: icmp_seq=4 ttl=243 time=67.6 ms
64 bytes from 54.93.34.220: icmp_seq=5 ttl=243 time=62.7 ms
64 bytes from 54.93.34.220: icmp_seq=7 ttl=243 time=84.6 ms
64 bytes from 54.93.34.220: icmp_seq=8 ttl=243 time=64.6 ms
64 bytes from 54.93.34.220: icmp_seq=9 ttl=243 time=72.0 ms
```


Ping Command



```
$ ping clarusway.com
```

```
Pinging clarusway.com [54.164.151.235] with 32 bytes of data:  
Reply from 54.164.151.235: bytes=32 time=132ms TTL=237  
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237  
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237  
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
```

```
Ping statistics for 54.164.151.235:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 130ms, Maximum = 132ms, Average = 130ms
```

```
$ ping www.google.com
```

```
Pinging www.google.com [172.217.169.132] with 32 bytes of data:  
Reply from 172.217.169.132: bytes=32 time=19ms TTL=116  
Reply from 172.217.169.132: bytes=32 time=18ms TTL=116  
Reply from 172.217.169.132: bytes=32 time=18ms TTL=116  
Reply from 172.217.169.132: bytes=32 time=19ms TTL=116
```

```
Ping statistics for 172.217.169.132:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 18ms, Maximum = 19ms, Average = 18ms
```



Ping Command

```
$ ping 54.164.151.235
```

```
Pinging 54.164.151.235 with 32 bytes of data:
```

```
Reply from 54.164.151.235: bytes=32 time=131ms TTL=237
```

```
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
```

```
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
```

```
Reply from 54.164.151.235: bytes=32 time=130ms TTL=237
```

```
Ping statistics for 54.164.151.235:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 130ms, Maximum = 131ms, Average = 130ms
```



SSH Command

- * ssh stands for “Secure Shell”.
- * It is a protocol used to securely connect to a remote server/system.

`ssh user@host(IP/Domain_name)`

```
ssh -i cert.pem ec2-user@54.93.34.220
```

```
gakeko2018@DESKTOP-JA07K2U:~$ ssh -i cert.pem ec2-user@54.93.34.220
The authenticity of host '54.93.34.220 (54.93.34.220)' can't be established.
ECDSA key fingerprint is SHA256:lvCnUtJiig4s2U4aojBonZOSbzGPBMOpB9yPPoGjVEo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.93.34.220' (ECDSA) to the list of known hosts.

 _ | _ | _ |
_| ( _ | _ | /   Amazon Linux 2 AMI
_| \ _ | _ |

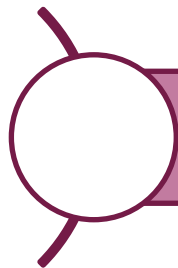
https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-35-15 ~]$
```



Whoami Command



Whoami Command



* Displays user, group and privileges information for the current user.

```
whoami
```

```
clarusway@Desktoop:~$ whoami  
clarusway
```



Text Editors





Vim Editor



Vim Editor



- Vim is a powerful text editor used in CLI (command line interface).
- Vim is an editor to create or edit a text file.

Insert Mode

- You cannot write text in command mode. To write text into a file, there is a dedicated insert mode. When you want to write something on a file, you must enter the insert mode.

Command Mode

- When you start Vim, you are placed in Command mode. In this mode, you can move across the screen, delete text and copy text.

```
VIM - Vi IMproved
      version 8.0.1453
      by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.ubuntu.com
Vim is open source and freely distributable

  Help poor children in Uganda!
type  :help iccf<insert>    for information

type  :q<insert>            to exit
type  :help<insert> or :h<insert> for on-line help
type  :help version<insert> for version info

0.0-1 All
```


Vim Editor



| Vim Command | Decription |
|---------------------|---|
| i | Enter insert mode |
| Esc | Enter command mode |
| x or Del | Delete a character |
| X | Delete character is backspace mode |
| u | Undo changes |
| Ctrl + r | Redo changes |
| y - yy | Copy / Copy a line |
| dd | Delete a line |
| p | Paste the content of the buffer |
| v | Enter visual mode for text selection |
| Shift v | Enter visual mode for selecting entire line |
| w | Move to end of a word |
| b | Move to beginning of a word |
| :%s/foo/bar/g | Search and replace all occurrences |
| Esc + :w | Save changes |
| Esc + :q (:wq, :q!) | quit Vim |



Nano Editor





Nano Editor

GNU nano is a small and friendly text editor.

Besides basic text editing, nano offers features like:

- undo/redo
- syntax coloring
- interactive search-and-replace
- auto-indentation
- line numbers
- word completion
- file locking, backup files
- internationalization support.

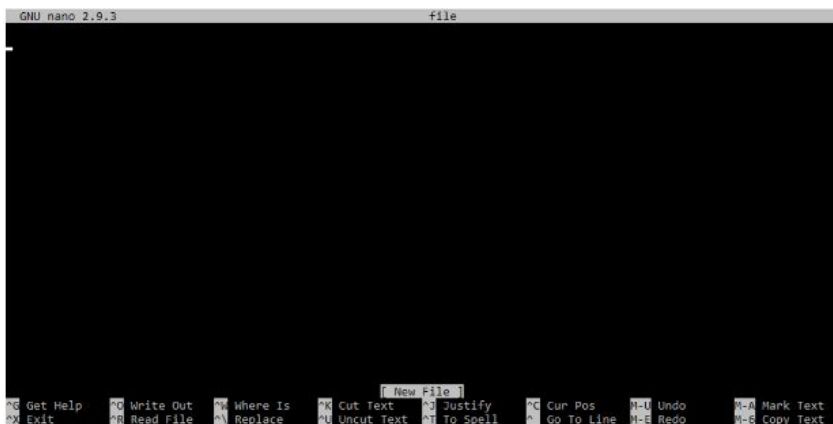




Nano Editor

- Unlike vi, nano is a modeless editor, which means that you can start typing and editing the text immediately after opening the file.
- To open an existing file or to create a new file, type nano followed by the file name.

```
nano filename
```



| Nano Command | Meaning |
|--------------|------------|
| Ctrl G | Get Help |
| Ctrl X | Exit |
| Ctrl O | Write Out |
| Ctrl R | Read File |
| Ctrl W | Where Is |
| Ctrl \ | Replace |
| Ctrl K | Cut Text |
| Ctrl U | Uncut Text |
| Ctrl J | Justify |
| Ctrl T | To Spell |
| Ctrl C | Cur Pos |
| Alt U | Undo |
| Alt E | Redo |



THANKS!

Any questions?