



ondia



Linux Plus for AWS and DevOps

Table of Contents



- ▶ **What We Learn**
- ▶ **Getting Help**
- ▶ **Text Editors**
- ▶ **File Management**

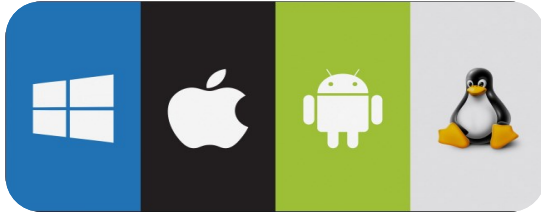


1

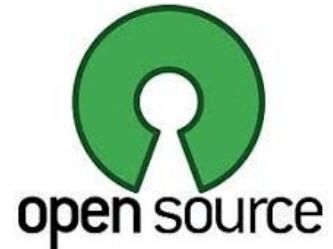
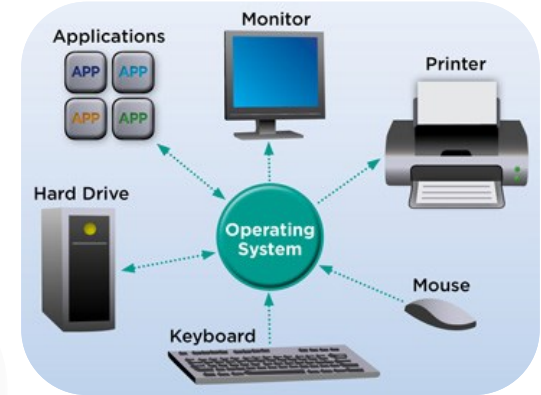
What We Learn

What is Linux?

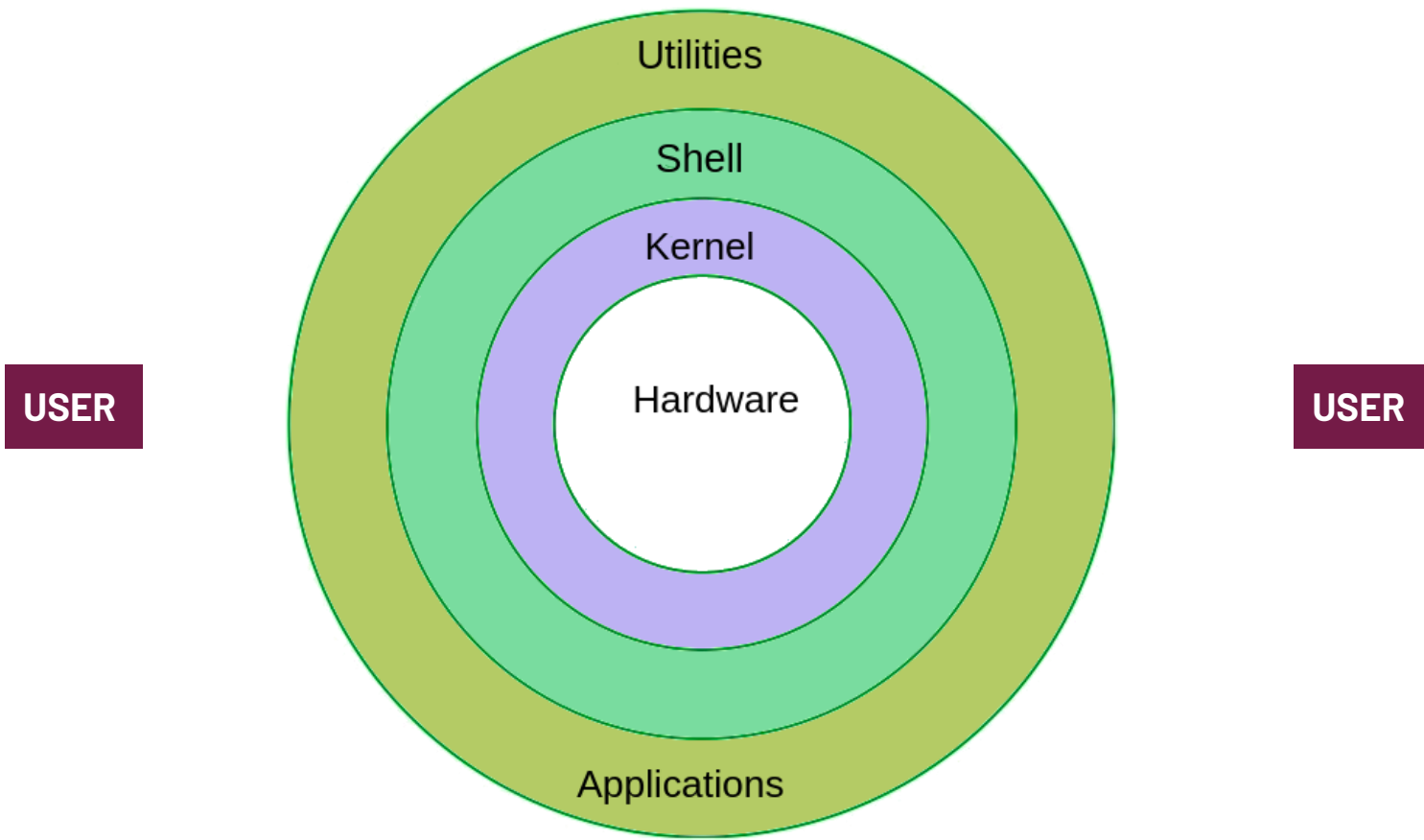
- Free
- Open-Source
- OS



What
is



Components of Linux

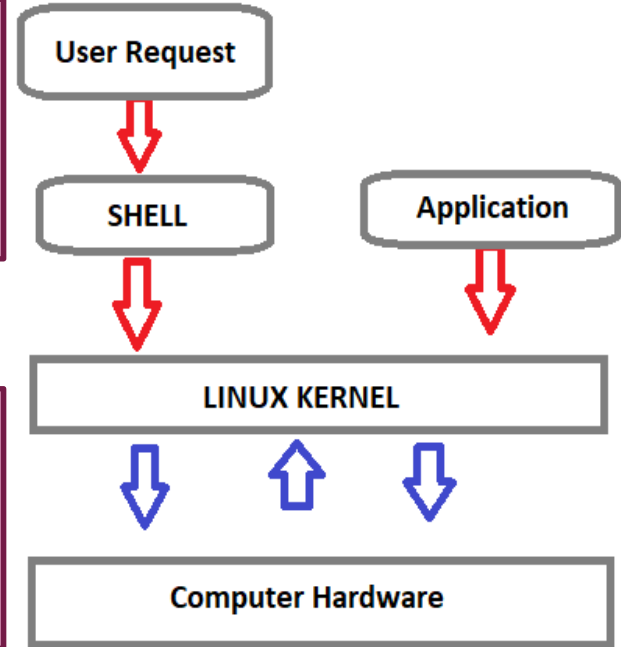


What is SHELL?



Shell is a **program** that **receives** the user's **commands** and **gives** them to the **operating system** to **process** and displays the output.

Bash (**B**ourne **A**gain **S**hell) is an **enhanced version** of Steve Bourne's first Unix shell application, and serves as the shell program on most Linux systems.



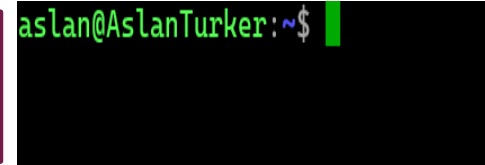
What is SHELL?



The standard Linux shell is both a **command-line interpreter** and a **programming language**.



The command prompt for Linux generally shows the current **user**, the current **host**, and the appropriate **directory**.



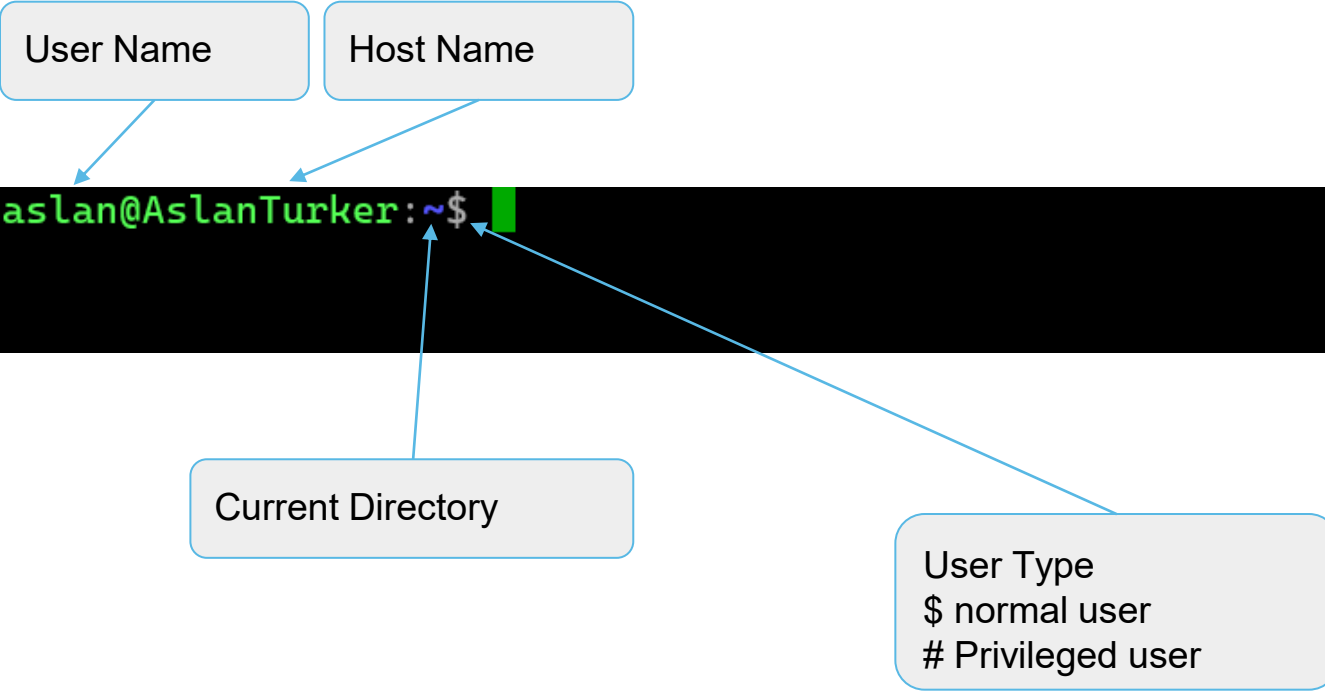
At the end of the prompt list, the **\$(dollar sign)** signifies the current user being unprivileged, and the device is ready to receive feedback.

\$ - you are normal user, you don't have all rights to execute

The input is sent for parsing and execution to the interpreter.

(root) you are super user. You have all the right to execute

Command Prompt



Basic Shell Commands



COMMAND	DESCRIPTION
pwd	show current path
ls	lists directory contents
cd	change (current) directory
mkdir	create a new directory
rmdir	delete an empty directory
touch	create a file
rm	delete a file

Basic Shell Commands



COMMAND	DESCRIPTION
cp	copy a file to another location
mv	move a file to another location
cat	show file contents
echo	print message to screen
clear	clear the terminal screen



2

Getting Help

Table of Contents



- ▶ **Man Pages**
- ▶ **Info Pages**
- ▶ **whatis command**
- ▶ **apropos command**
- ▶ **--help option**



2-1

Man Pages

Man Pages



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 **man command**.



`man <command>`

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

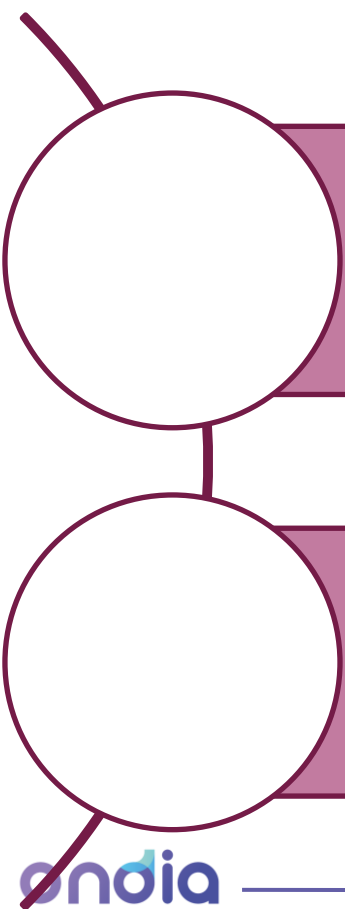
Detailed information about arguments and options.



2-2

Info Pages

▶ Info Pages



Info pages are **additional documentation** with more robust capability **in detail**. Info pages normally provide more detailed information about a command than its respective man page.

info <command>

The info page for a particular command is invoked by **info command**.



\$ info echo

```
Text: 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.
```



2-3 whatis command

whatis command



whatis

display one-line manual page names.

```
aslan@AslanTurker:~$ whatis ls
ls (1)          - list directory contents
aslan@AslanTurker:~$ whatis pwd
pwd (1)         - print name of current/working directory
aslan@AslanTurker:~$ whatis mv
mv (1)          - move (rename) files
aslan@AslanTurker:~$ █
```



2-4

apropos command

apropos command



apropos

search the manual page **names** and **descriptions**.

```
aslan@AslanTurker:~$ apropos pwd
pwd (1)          - print name of current/working directory
pwdx (1)         - report current working directory of a process
unix_chkpwd (8)  - Helper binary that verifies the password of the current user
aslan@AslanTurker:~$
```



--help Option

▶ --help Option



--help

gives a **short explanation** about how to use the command and a **list of available options**.

```
aslan@AslanTurker:~$ ls --help
Usage: ls [OPTION]... [FILE]...
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       with -l, scale sizes by SIZE when printing them;
                          e.g., '--block-size=M'; 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
                          change of file status information);
                          with -l: show ctime and sort by name;
                          otherwise: sort by ctime, newest first
```



3

Text Editors



3-1

vi/Vim Editor

▶ Vi/Vim Editor



- Vi is a text editor **originally created for the Unix** operating system.
- Vim (Vi IMproved) as its name suggests, is a **clone of Vi** and offers more features than Vi.

The reasons why we should use Vi/Vim editor.

- Vim is available on most linux distro's.
- Vim Uses Less Amount of System Resources.
- Vim Supports All Programming Languages and File Formats
- Vim is Very Popular in the Linux World

Vi/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.

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.

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.

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

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

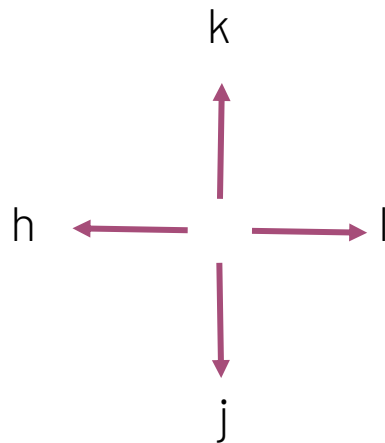
type  :q<Enter>            to exit
type  :help<Enter> or <F1>  for on-line help
type  :help version8<Enter> for version info

0.0-1 All
```

Vi/Vim Editor



Vim Command	Description
i	Enter insert mode
Esc	Enter command mode
x or Del	Delete a character
X	Delete character in backspace mode
u	Undo changes
Ctrl + r	Redo changes
yy	Copy a line
dd	Delete a line
p	Paste the content of the buffer
o	insert a blank line under the current cursor position.
:%s/foo/bar/g	Search and replace all occurrences
Esc + :w	Save changes
Esc + :wq or Esc + ZZ	Save and quit Vim





3-2

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



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



4-1 Files

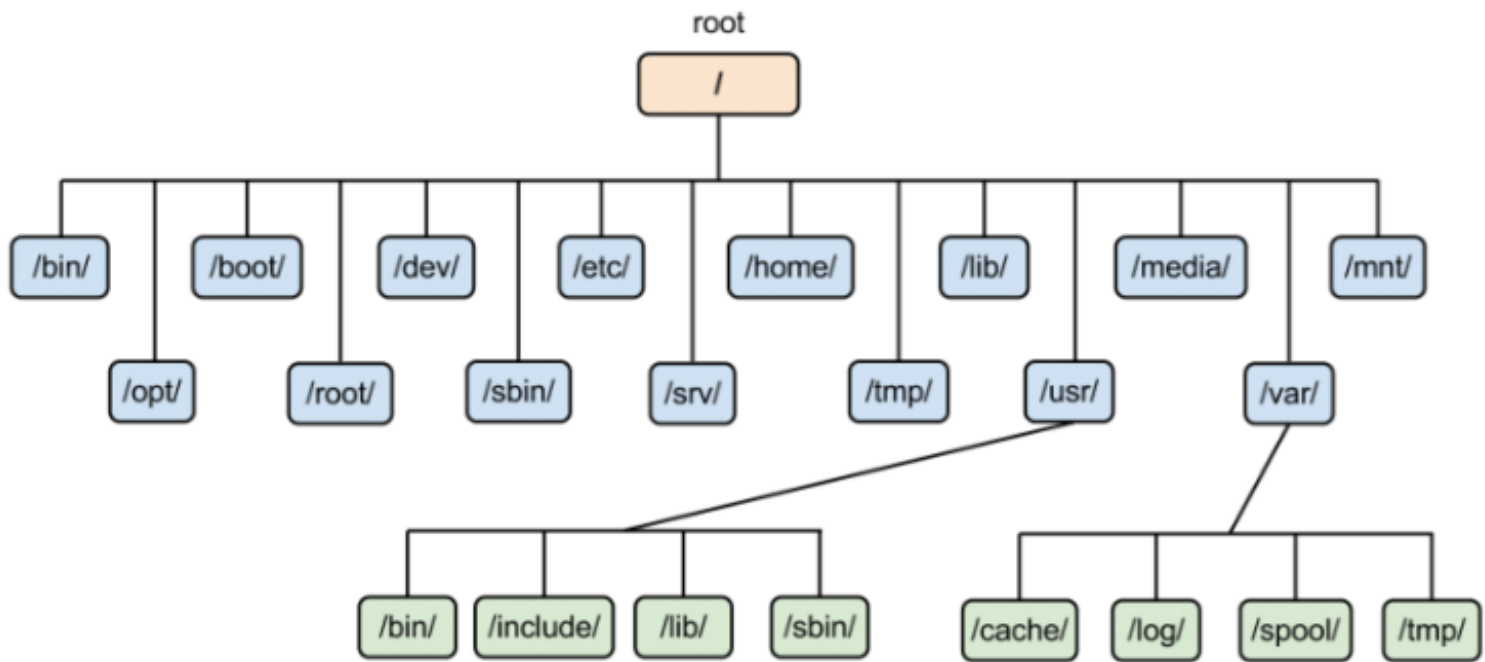


On a Linux system, everything is a file.

A Linux system makes no difference between a file and a directory, since a directory is just a file containing names of other files.

The tree of the file system starts at the trunk or slash, indicated by a forward slash (/). This directory, containing all underlying directories and files, is also called the root directory or “the root” of the file system.

ROOT Directory (/)



▶ ROOT Directory (/)



/bin	Essential command binaries
/boot	Static files of the boot loader
/dev	Device files
/etc	Host-specific system configuration
/home	Users' home directories
/lib	Essential shared libraries and kernel modules
/media	Mount point for removable media
/mnt	Mount point for mounting a filesystem temporarily
/opt	Add-on application software packages
/sbin	Essential system binaries
/srv	Data for services provided by this system
/tmp	Temporary files
/usr	Secondary hierarchy
/var	Variable data

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Character Device File
s	Socket File
p	Named Pipe
b	Block Device

-rw-----	Regular File
d rw-r--r--	Directory File
l rw-r--r--	Link File
c rw-r--r--	Character Device File
b rw-r--r--	Block Special File
s rw-r--r--	Socket File
p rw-r--r--	Named Pipe File



4-2

Viewing file properties



Viewing file properties



ls

On most Linux versions ls is aliased to color-ls by default. This feature allows to see the file type without using any options to ls.

Color	Meaning
Blue	directories
Red	compressed archives
White	text files
Pink	images
Cyan	links
Yellow	Devices
Green	Executables
flashing red	broken links

```
aslan@AslanTurker:~$ ls
archie.tar images.jpg linuxplus movies
aslan@AslanTurker:~$
```




4-3

Working with File Contents

Working with File Contents



head

output the first ten lines of a file.

```
aslan@AslanTurker:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
aslan@AslanTurker:~$
```

Working with File Contents



head -n

output the first n lines of a file.

```
aslan@AslanTurker:~$ head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
aslan@AslanTurker:~$
```

Working with File Contents



tail

output the last ten lines of a file.

```
aslan@AslanTurker:~$ tail /etc/passwd
systemd-network:x:998:998:systemd Network Management::/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization::/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver::/usr/sbin/nologin
uidd:x:103:103::/run/uidd:/usr/sbin/nologin
landscape:x:104:105::/var/lib/landscape:/usr/sbin/nologin
polkitd:x:990:990:User for polkitd::/usr/sbin/nologin
aslan:x:1000:1000:,,,:/home/aslan:/bin/bash
aslan@AslanTurker:~$
```

Working with File Contents



tail -n

output the last n lines of a file.

```
aslan@AslanTurker:~$ tail -5 /etc/passwd
systemd-resolve:x:991:991:systemd Resolver://usr/sbin/nologin
uidd:x:103:103:./run/uidd:/usr/sbin/nologin
landscape:x:104:105:./var/lib/landscape:/usr/sbin/nologin
polkitd:x:990:990:User for polkitd:./usr/sbin/nologin
aslan:x:1000:1000:,,,:/home/aslan:/bin/bash
aslan@AslanTurker:~$
```

Working with File Contents



cat

Display a file on the screen.

```
aslan@AslanTurker:~/linuxplus$ cat quotes.txt
1. "The only way to do great work is to love what you do."
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."
3. "Believe you can and you're halfway there."
4. "The best way to predict the future is to create it."
aslan@AslanTurker:~/linuxplus$
```

Working with File Contents



cat

One of the basic uses of cat is to concatenate files into a bigger (or complete) file.

```
aslan@AslanTurker:~/linuxplus$ echo this is file1 > file1
aslan@AslanTurker:~/linuxplus$ echo this is file2 > file2
aslan@AslanTurker:~/linuxplus$ echo this is file3 > file3
aslan@AslanTurker:~/linuxplus$ cat file1
this is file1
aslan@AslanTurker:~/linuxplus$ cat file2
this is file2
aslan@AslanTurker:~/linuxplus$ cat file3
this is file3
aslan@AslanTurker:~/linuxplus$ cat file1 file2 file3
this is file1
this is file2
this is file3
aslan@AslanTurker:~/linuxplus$ cat file1 file2 file3 > all
aslan@AslanTurker:~/linuxplus$ cat all
this is file1
this is file2
this is file3
aslan@AslanTurker:~/linuxplus$ █
```

Working with File Contents



cat

You can use cat to create **flat text files**.

```
aslan@AslanTurker:~/linuxplus$ cat winter.txt
Today is cold.
aslan@AslanTurker:~/linuxplus$ █
```

The **Ctrl d** key combination will send an **EOF (End of File)** to the running process ending the cat command.

Working with File Contents



more

view (but not modify) the contents of a text file one **screen at a time**.

```
aslan@AslanTurker:~/linuxplus$ more quotes.txt
1. "The only way to do great work is to love what you do."
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."
3. "Believe you can and you're halfway there."
4. "The best way to predict the future is to create it."
aslan@AslanTurker:~/linuxplus$
```

Working with File Contents



more -n

This option specifies an integer which is the screen size (in lines).

```
aslan@AslanTurker:~/linuxplus$ more -2 quotes.txt
1. "The only way to do great work is to love what you do."

--More--(23%)
```

Working with File Contents



less

Similar to more, less command allows you to view the contents of a file and navigate through file. The **main difference** between more and less is that **less** command is **faster** because it **does not load the entire file at once**.

```
aslan@AslanTurker:~/linuxplus$ less quotes.txt
```

```
1. "The only way to do great work is to love what you do."  
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."  
3. "Believe you can and you're halfway there."  
4. "The best way to predict the future is to create it."  
quotes.txt (END)
```

Working with File Contents



tac

concatenate and print files in reverse.

```
clarusway@DESKTOP-UN6T2ES:~$ cat count.txt
one
two
three
four
five
clarusway@DESKTOP-UN6T2ES:~$ tac count.txt
five
four
three
two
one
clarusway@DESKTOP-UN6T2ES:~$
```



4-4

Searching Files

Searching Files



find

search for files in a directory hierarchy.

```
find [starting-point...] [expression]
```

find

Find all the files whose name is linuxplus.txt in a current working directory.

```
aslan@AslanTurker:~/linuxplus$ find . -name linuxplus.txt
./linuxplus.txt
aslan@AslanTurker:~/linuxplus$
```

▶ Searching Files



find

Find all the files whose name is linuxplus.txt under /home directory.

```
aslan@AslanTurker:~$ find /home -name linuxplus.txt
/home/aslan/linuxplus/linuxplus.txt
aslan@AslanTurker:~$ █
```

Searching Files



find

Find all the files whose name is linuxplus.txt and contains both capital and small letters in /home directory.

```
aslan@AslanTurker:~$ find /home -iname linuxplus.txt
/home/aslan/linuxplus/Linuxplus.txt
/home/aslan/linuxplus/linuxplus.txt
aslan@AslanTurker:~$
```


Searching Files



find

Find all directories whose name is movies in /home directory.

```
aslan@AslanTurker:~$ find /home -type d -name movies
/home/aslan/movies
aslan@AslanTurker:~$
```

Searching Files



find

Find all txt files in a directory.

```
aslan@AslanTurker:~$ find . -type f -name "*.txt"
./linux.txt
./quotes.txt
./linuxplus/Linuxplus.txt
./linuxplus/count.txt
./linuxplus/linuxplus.txt
./winter.txt
aslan@AslanTurker:~$
```

Searching Files



grep

The grep, which stands for "global regular expression print," is used to search text.

```
grep [options] pattern [files]
```

Options	Description
-c	This prints only the number of lines that match a pattern
-h	Do not display the filenames headers.
-i	Ignores, case for matching
-l	Displays list of a filenames only.
-n	Display the matched lines and their line numbers.
-v	This prints out all the lines that do not matches the pattern

Searching Files



grep

The grep searches the given file for lines containing a match to the given strings or words.

```
aslan@AslanTurker:~/linuxplus$ cat quotes.txt
1. "The only way to do great work is to love what you do."
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."
3. "Believe you can and you're halfway there."
4. "The best way to predict the future is to create it."
aslan@AslanTurker:~/linuxplus$ grep "Believe" quotes.txt
3. "Believe you can and you're halfway there."
aslan@AslanTurker:~/linuxplus$
```

Searching Files



grep -n

Returns the result of lines matching the search string.

```
aslan@AslanTurker:~/linuxplus$ grep -n "Believe" quotes.txt
5:3. "Believe you can and you're halfway there."
```

grep -c

Returns the number of lines in which the results matched the search string.

```
aslan@AslanTurker:~/linuxplus$ grep -c "Believe" quotes.txt
1
aslan@AslanTurker:~/linuxplus$
```

Searching Files



grep -v

Returns the result of lines not matching the search string.

```
aslan@AslanTurker:~/linuxplus$ cat quotes.txt
1. "The only way to do great work is to love what you do."
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."
3. "Believe you can and you're halfway there."
4. "The best way to predict the future is to create it."
aslan@AslanTurker:~/linuxplus$ grep -v "Believe" quotes.txt
1. "The only way to do great work is to love what you do."
2. "Success is not final, failure is not fatal: It is the courage to continue that counts."
4. "The best way to predict the future is to create it."
aslan@AslanTurker:~/linuxplus$ █
```

THANKS!

Any questions?

