How to interact with OS

Lecture 6 from Intro to CS

Unix,

Terminal commands

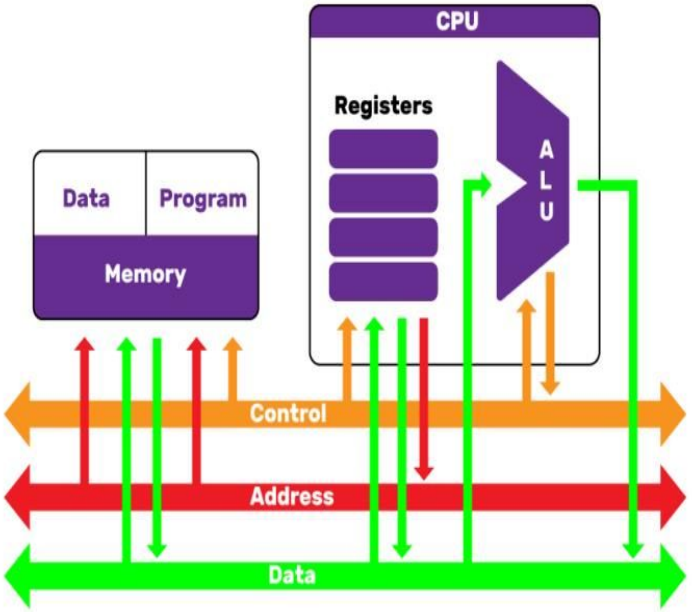
Last lecture: Process

Program: a binary file in hard disk

When we run a program,

- We need to load its data, instructions into memory
- And run on CPU

Process: The program that is loaded into memory



https://www.icdrex.com/the-brain-behind-the-machine-transistors-in-cpu-architecture/

Last lecture: Running one program vs running multiple program

Multi-tasking

- Computers can run many program at the same time
- Many programs use the same memory, CPU, I/O devices

Problems

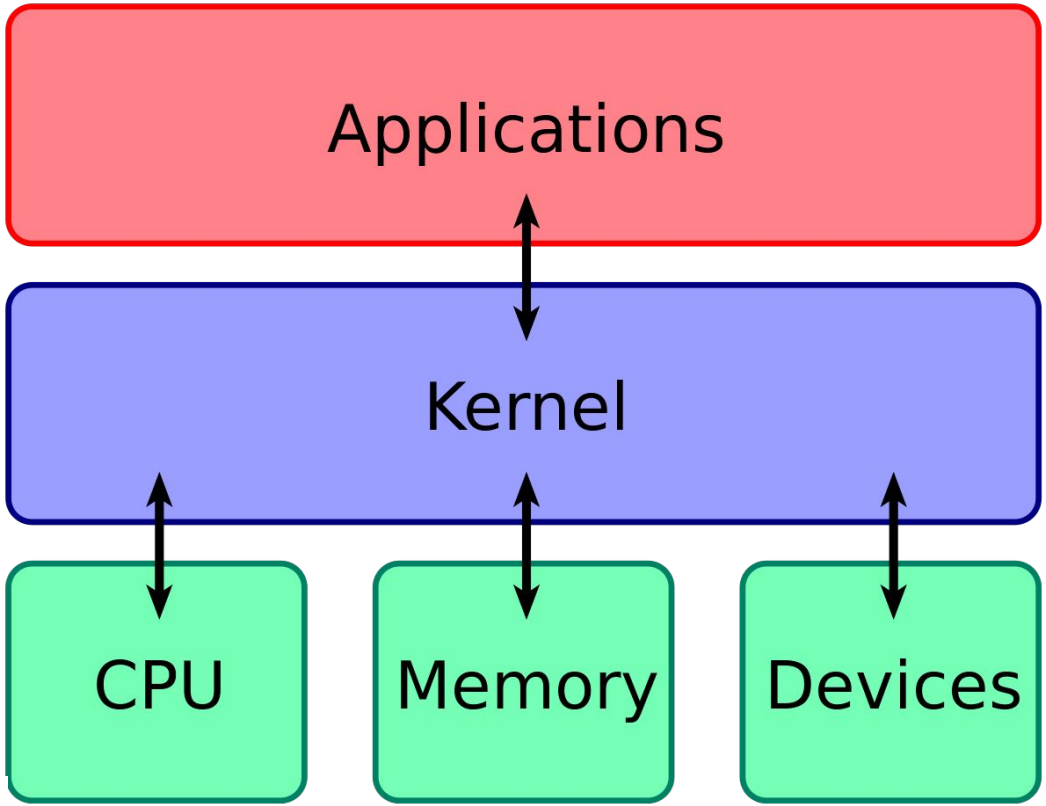
- One program can affect another
- One program can steal another's data
- There may be more than one user

Protection

- How to protect system resources (hardware)?
- How to prevent one program affecting another?
- ..

Last lecture:
Operating system

Kernel: the core of os that controls system resources



https://en.wikipedia.org/wiki/Kernel_(operating_system)

Content for this lecture

How to interact with os

The rest of the semester:

User apps and their needs

- Apps for writing code
  - Version control
- Apps for documentation
- Data storage
- networking

Unix OS

Some history

- 3rd time sharing system (2nd CTSS)
- Dev. started in 1969 at Bell Labs by Ken Thompson, Dennis Ritchie, Brian Kernighan, and others
  - Run on PDP-7 - Wikipedia and -9 computers
  - Originally written in assembly, later written in C
- Outside bell labs- 1973
- Late 1970s distributed, licensed to academic and commercial institutions
-

Origins and History of Unix, 1969-1995

https://en.wikipedia.org/wiki/Unix

Unix OS

The original spelling was “UNICS” (UNiplexed Information and Computing Service)

Ken (seated) and Dennis (standing) at a PDP-11 in 1972.

Ritchie observes:

- “What we wanted to preserve was not just a good environment in which to do programming, but a system around which a fellowship could form.
- We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication”.

1969 was also the year the ARPANET (the direct ancestor of today's Internet) was invented.

Content copied from [Origins and History of Unix, 1969-1995]

John Lions's 1976 A COMMENTARY ON THE SIXTH EDITION UNIX OPERATING SYSTEM on the Version 6 source code became the first serious documentation of the Unix kernel internals.

Ken Arnold “...back then you couldn't be a kernel hacker without a Lions.”

Origins and History of Unix, 1969-1995

first public report in 1974- 600 installations.

AT&T (the parent organization of Bell Labs) had been forbidden from entering the computer business.

- Unix couldn’t turn into commercial product

Unix was very popular in academia 1970s and 1980s

Douglas Comer: “Many universities contributed to UNIX.

At the University of Toronto, the department acquired a 200-dot-per-inch printer/plotter and built software that used the printer to simulate a phototypesetter.

At Yale University, students and computer scientists modified the UNIX shell.

At Purdue University, the Electrical Engineering Department made major improvements in performance, producing a version of UNIX that supported a larger number of users. Purdue also developed one of the first UNIX computer networks.

At the University of California at Berkeley, students developed a new shell and dozens of smaller utilities.

By the late 1970s, when Bell Labs released Version 7 UNIX, it was clear that the system solved the computing problems of many departments, and that it incorporated many of the ideas that had arisen in universities.

The end result was a strengthened system. A tide of ideas had started a new cycle, flowing from academia to an industrial laboratory, back to academia, and finally moving on to a growing number of commercial sites”

Origins and History of Unix, 1969-1995

1977-BSD released (Berkeley version of Unix)

1980-DARPA chose BSD as platform to implement TCP/IP

- TCP/IP implementation released with Berkeley 4.2 in 1983

1981-Microsoft made deal with IBM to market MS-DOS separately

- Mostly for PCs-cheaper machines instead of workstations

1983 System V Unix (AT&T try to commercialize but failed.

1983 Richard Stallman start writing GNU, completely a free clone of Unix

1985 GNU manifesto-(GNU Manifesto - Wikipedia)

The GNU Manifesto - GNU Project - Free Software Foundation

Standardization

System V and BSD

1985 -POSIX standards backed by IEEE

- Described intersection set of between System V and BSD calls

1988-IBM, HP, etc formed open software foundation against AT&T/Sun

The UNIX® Standard | www.opengroup.org

Origins and History of Unix, 1969-1995

Linux

1991-Linus Torvalds announced Linux project

By late 1993 Linux has

- X and internet capability
- GNU toolkits that provides high quality dev tools

Open source movement

1998 source code release for Netscape

That leads more into Linux

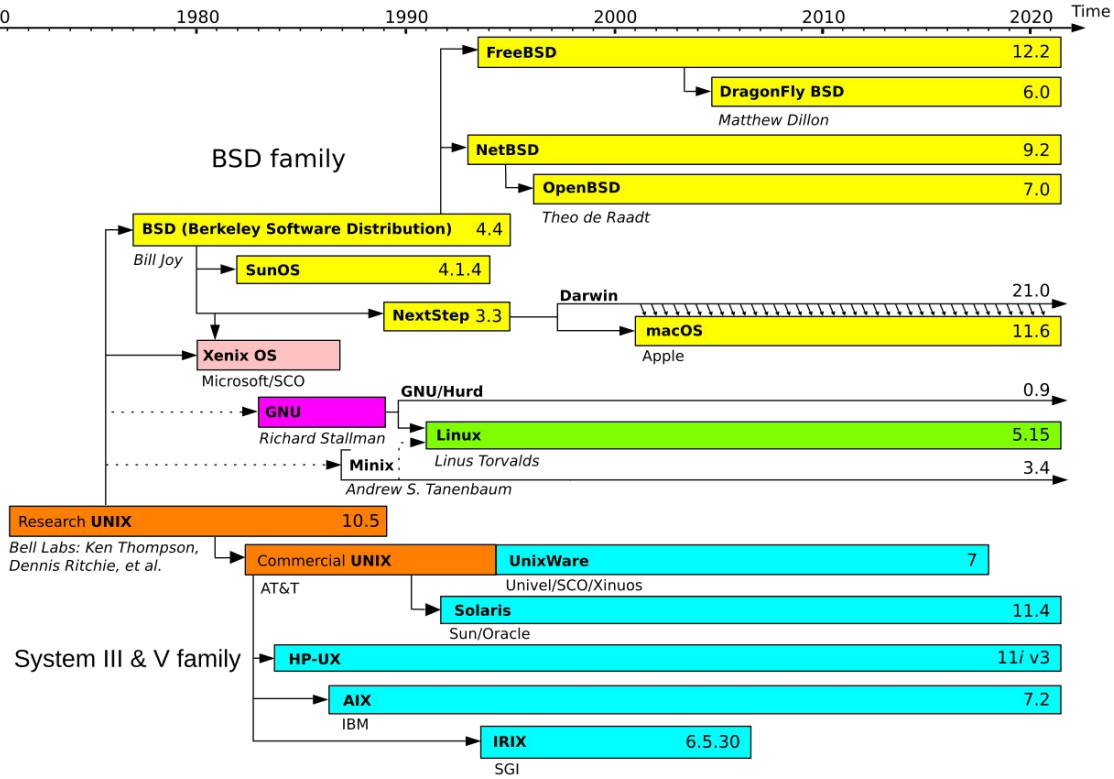
https://en.wikipedia.org/wiki/Open-source-software_movement

The Open-Source Movement: 1998 and Onward

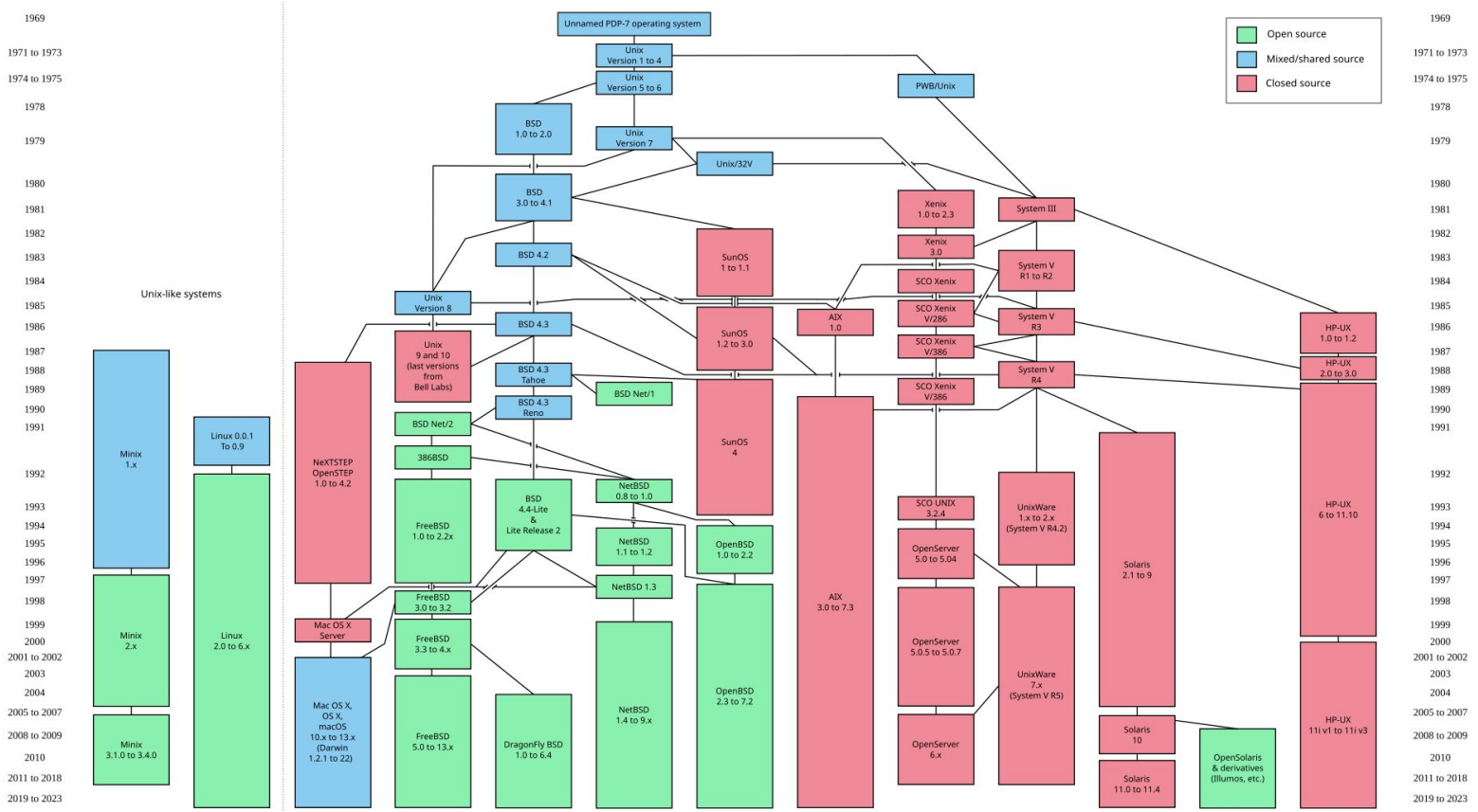
Unix-like OSes (*Nix)

An OS behaves similar to Unix

- Amoeba
- BSD
- Coherent
- Darwin
- DEMOS
- DNIX
- Domain/OS
- DYNIX
- GNU Hurd
- Linux
- LynxOS
- MINIX
- MNOS
- MOS
- NeXTSTEP
- QNX
- Redox
- RISC iX
- SOX
- SunOS
- SerenityOS
- Ultrix
- uNETix



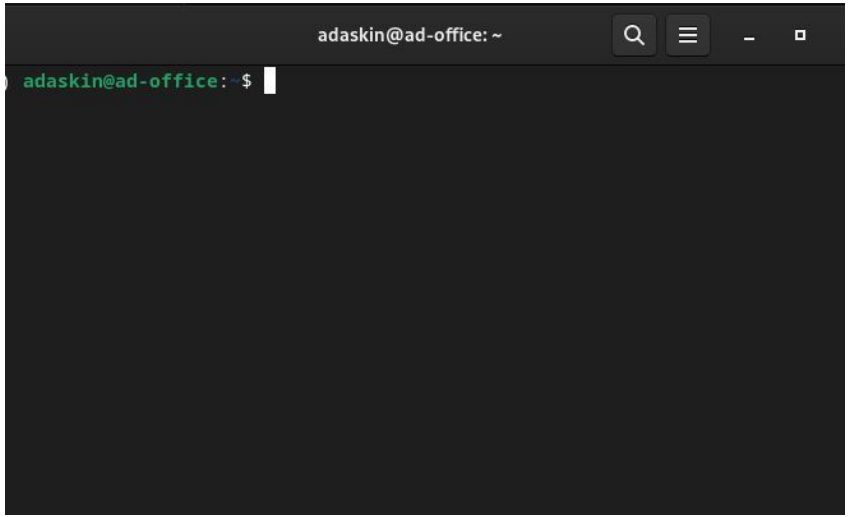
https://en.wikipedia.org/wiki/Unix-like



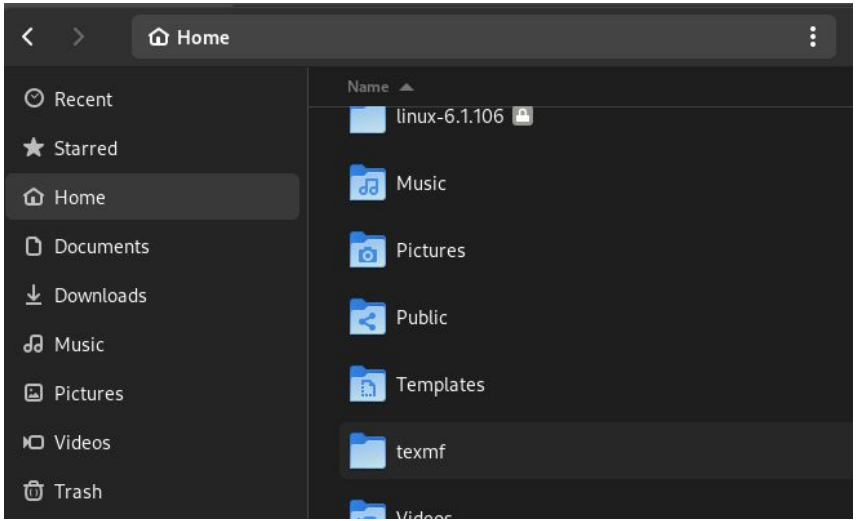
https://en.wikipedia.org/wiki/List_of_Unix_systems

OS user interface

Command Line (terminal, cmd)

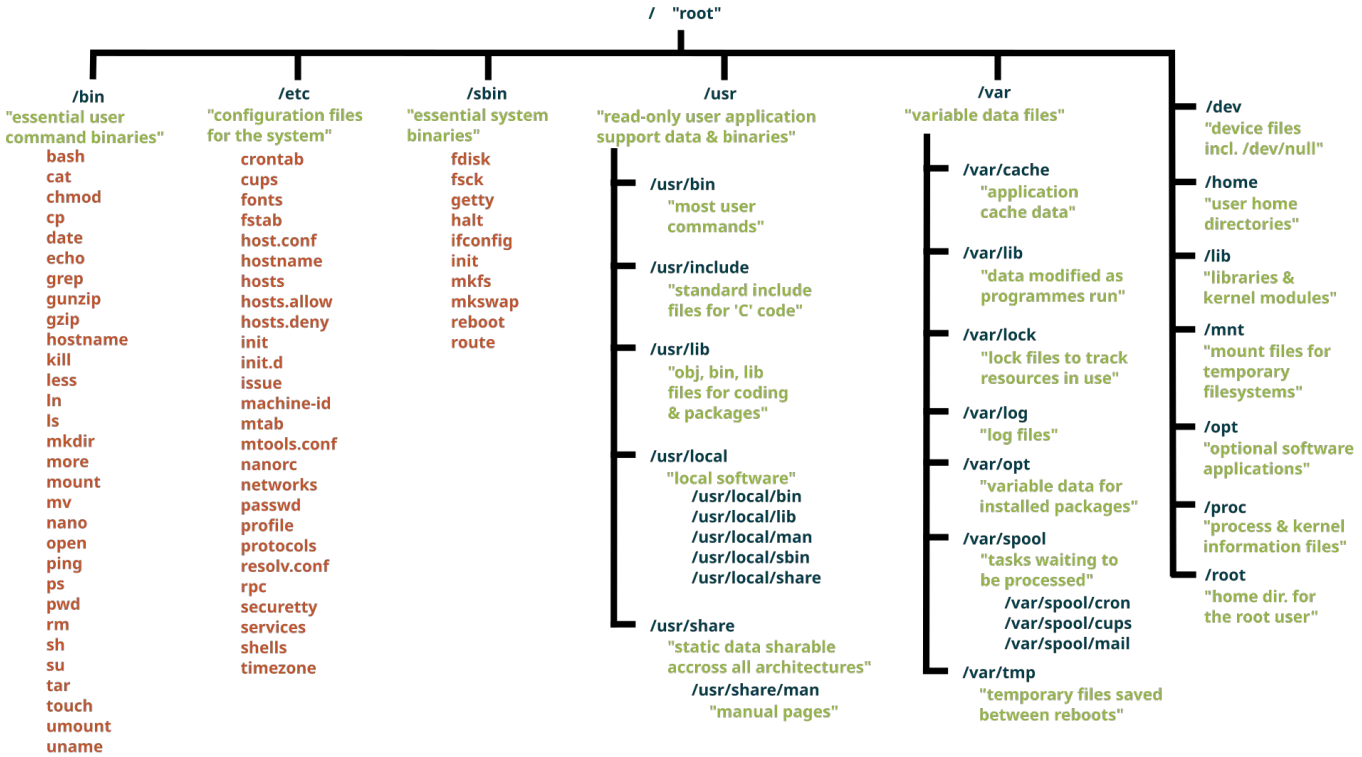


GUI



Other examples?

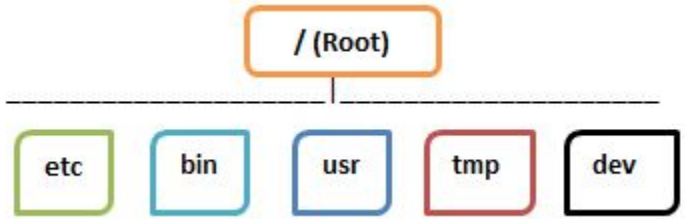
Unix-like OS file systems



Generic Files

Device files

Directory files



Users

Root user

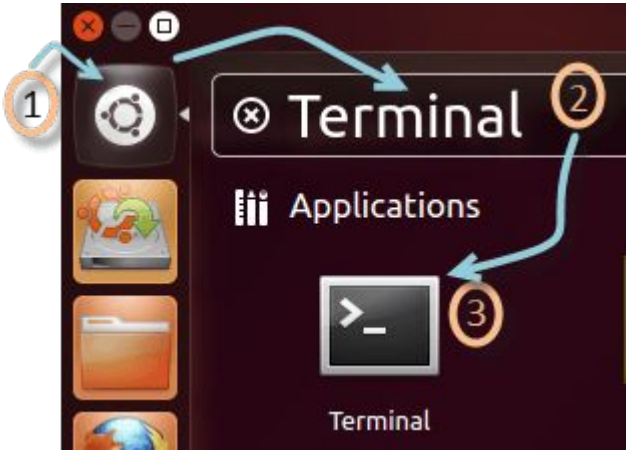
Regular user

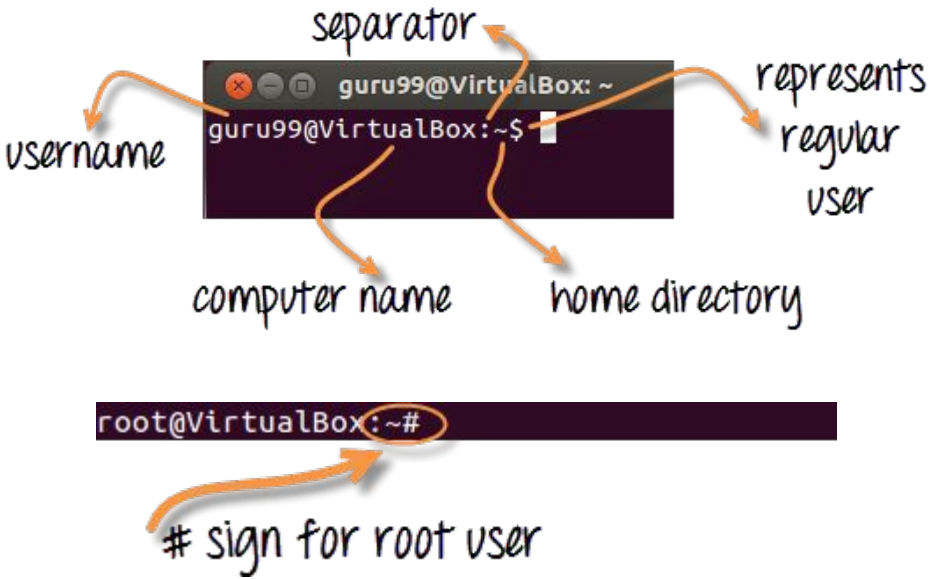
Service user

Terminal

See also [The Linux command line for beginners | Ubuntu]

CTRL + Alt + T to launch the Terminal





https://www.guru99.com/terminal-file-manager.html

Commands

- ls
  - Listing files
- pwd
  - Present working directory
- cd
  - Changing directory
- man
  - Manual
  - Man ls
  - ls –help

Absolute path

```
guru99@VirtualBox:~$ cd /home/guru99/Pictures
guru99@VirtualBox:~/Pictures$
```

Relative path

```
guru99@VirtualBox:~$ cd Downloads
guru99@VirtualBox:~/Downloads$
```

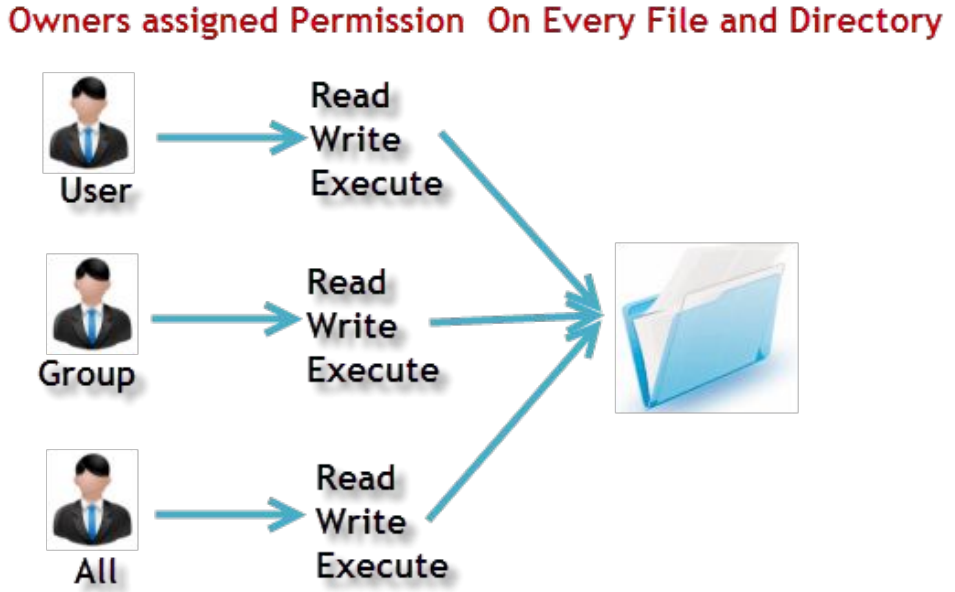
https://www.guru99.com/terminal-file-manager.html

- cat
  - concatenate files and print on the standard output
  - cat f1 f2 f3
- Touch
  - Change access time
  - Touch afile
- rm
  - Remove
  - Non-recoverable
  - rm -r directory
    - Remove directory
- Mv
  - Move
  - mv a ../b

- Head
  - Output the first part
  - Head -n 9
    - Print the first 9 lines
- Tail
  - Output the last lines
  - Tail -n 9
    - Print the last 9 lines
- Less or more
  - One screenfull at a time view for files

https://www.guru99.com/must-know-linux-commands.html

File ownerships



User

A user is the owner of the file.

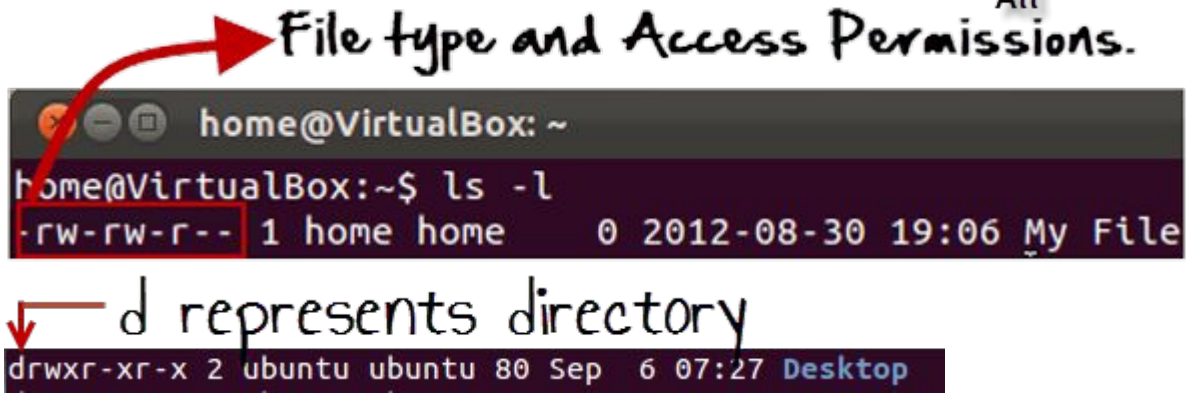
- By default, the person who created a file becomes its owner.

Group

A user- group can contain multiple users.

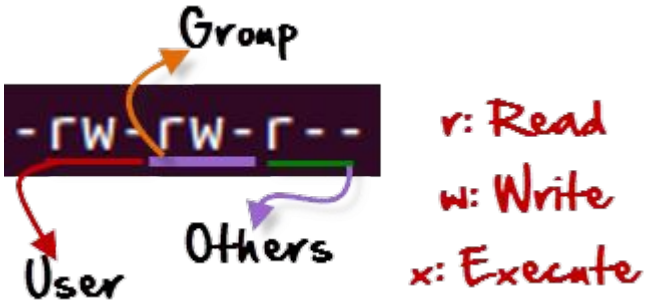
Other

Any other user who has access to a file.



https://www.guru99.com/file-permissions.html

Changing file permissions



Number	Permission Type	Symbol
0	No Permission	—
1	Execute	–x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r–
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

chmod permission filename

Checking Current File Permissions

```
ubuntu@ubuntu:~$ ls -l sample
-rw-rw-r-- 1 ubuntu ubuntu 15 Sep  6 08:00 sample
```

chmod 764 and checking permissions again

```
ubuntu@ubuntu:~$ chmod 764 sample
ubuntu@ubuntu:~$ ls -l sample
-rwxrw-r-- 1 ubuntu ubuntu 15 Sep  6 08:00 sample
```

https://www.guru99.com/file-permissions.html

Current File Permissions

```
home@VirtualBox:~$ ls -l sample
-rw-rw-r-- 1 home home 55 2012-09-10 10:59 sample
```

Setting permissions to the 'other' users

```
home@VirtualBox:~$ chmod o=rwx sample
home@VirtualBox:~$ ls -l sample
-rw-rw-rwx 1 home home 55 2012-09-10 10:59 sample
```

Adding 'execute' permission to the usergroup

```
home@VirtualBox:~$ chmod g+x sample
home@VirtualBox:~$ ls -l sample
-rw-rwxrwx 1 home home 55 2012-09-10 10:59 sample
```

Removing 'read' permission for 'user'

```
home@VirtualBox:~$ chmod u-r sample
home@VirtualBox:~$ ls -l sample
--w-rwxrwx 1 home home 55 2012-09-10 10:59 sample
```

https://www.guru99.com/file-permissions.html

Changing ownership

```
chown user filename

chown user:group filename

chgrp group_name filename
```

Check the current file ownership using ls -dl

```
guru99@VirtualBox:~$ ls -dl test1
-rwxrwxrwx 1 root cdrom 0 Oct  6 11:27 test1
```

Change the file owner to root . You will need sudo

```
guru99@VirtualBox:~$ sudo chgrp root test1
```

Group ownership changed to root

```
guru99@VirtualBox:~$ ls -dl test1
-rwxrwxrwx 1 root root 0 Oct  6 11:27 test1
```

```
guru99@VirtualBox:~$ groups
cdrom guru99 adm sudo dip plugdev lpadmin sambashare
guru99@VirtualBox:~$
```

```
guru99@VirtualBox:~$ newgrp cdrom
guru99@VirtualBox:~$ cat > test
this is a test to change group
^C
guru99@VirtualBox:~$ ls -dl test
-rw-rw-r-- 1 guru99 cdrom 31 Oct 11 16:39 test
guru99@VirtualBox:~$
```

https://www.guru99.com/file-permissions.html

Running commands with substitute user

- su
  - Defaults to root
- su username
-

- sudo
  - Defaults to super-user(root)
  - Execute commands as another user
- sudo rm afile
- sudo ls afile
-

Adding user

Adduser, addgroup

Or useradd, groupadd

Usermod: modify a user account

- Change home, name, password, group etc.

sudo useradd ayse

sudo id ayse

sudo passwd ayse

sudo useradd -m ayse

- Creates home directory

sudo userdel ayse

Searching/locating files

find [path] [expression]

- Search for files in a directory
- find . -name "example.txt"
-

Argument	Description	Example	Command
`-name`	Find files by name	Find files named `example.txt`	`find /path/to/search -name "example.txt"`
`-type`	Find files by type (`f`, `d`, `l`)	Find directories	`find /path/to/search -type d`
`-size`	Find files by size	Find files larger than 100MB	`find /path/to/search -size +100M`
`-mtime`	Find files by modification time (days)	Find files modified in the last 7 days	`find /path/to/search -mtime -7`
`-atime`	Find files by access time (days)	Find files accessed in the last 7 days	`find /path/to/search -atime -7`
`-ctime`	Find files by change time (days)	Find files changed in the last 7 days	`find /path/to/search -ctime -7`
`-exec`	Execute a command on found files	Delete files named `example.txt`	`find /path/to/search -name "example.txt" -exec rm {} \;`
`-delete`	Delete found files	Delete files larger than 100MB	`find /path/to/search -size +100M -delete`
`-user`	Find files by user	Find files owned by user `john`	`find /path/to/search -user john`
`-group`	Find files by group	Find files owned by group `admin`	`find /path/to/search -group admin`
`-perm`	Find files by permissions	Find files with `755` permissions	`find /path/to/search -perm 755`
`-mindepth`	Minimum search depth	Start search at least 2 directories deep	`find /path/to/search -mindepth 2`
`-maxdepth`	Maximum search depth	Search up to 3 directories deep	`find /path/to/search -maxdepth 3`
`-empty`	Find empty files or directories	Find empty directories	`find /path/to/search -type d -empty`
`-prune`	Exclude directories from search	Exclude `dir_to_exclude` directory	`find /path/to/search -path "dir_to_exclude" -prune -o -print`

https://www.serveracademy.com/blog/linux-find-command/

Searching patterns in a file

grep [options] pattern [*file ...*]

grep “istanbul” sehirler.txt sehirler2.txt sehirler3.txt

- -r
  - Search all files
- -i
  - Case sensitive
- -c
  - Count number of occurrences
- -e
  - Search for pattern
  - -e “pattern1” -e “pattern2”

`-i`: Ignore case distinctions.

`-v`: Invert the match to select non-matching lines.

`-c`: Count the number of matching lines.

`-l`: List filenames containing the match.

`-L`: List filenames that do not contain the match.

`-n`: Prefix each line of output with the line number.

`-H`: Print the filename for each match.

`-r` or `-R`: Read all files under each directory, recursively.

`-w`: Match whole words only.

`-x`: Match whole lines only.

`-E`: Use extended regular expressions (ERE).

`-F`: Interpret pattern as a list of fixed strings (fgrep).

`-q`: Quiet, do not write anything to standard output.

sed

Stream editor, perform text transformations

sed SCRIPT INPUTFILE…

-
- sed 's/hello/world/' input.txt
  - Replace all occurrences of ‘hello’ to ‘world’
- sed -i 's/hello/world/' file.txt
  - Edit files in place
- sed -n '45p' file.txt
  - Print only line 45
    - p for printing
    - -n for suppressing
- sed -n '1p ; $p' one.txt two.txt three.txt
  - Multiple input considered as a single stream
  - 1st line of one.txt
  - Last line of three

https://www.gnu.org/software/sed/manual/sed.html#Introduction

awk

AWK is an interpreted language is designed for text processing

- Mawk,
- Gawk, etc are implementations

An AWK program is a series of pattern action pairs,

condition { action }

condition { action }

```
/regex_pattern/ {
    # Actions to perform in the event the record (line) matches the above regex_pattern
    print 3+2
    print foobar(3)
    print foobar(variable)
    print sin(3-2)
}
```

https://en.wikipedia.org/wiki/AWK

examples

File advice

```
BEGIN { print "Don't Panic!" }
```

```
$ awk -f advice
```

```
$ awk 'BEGIN { print "Don\47t Panic!" }'
```

Mail-list file

```
Amelia       555-5553     amelia.zodiacusque@gmail.com    F
Anthony      555-3412     anthony.asserturo@hotmail.com   A
Becky        555-7685     becky.algebrarum@gmail.com      A
Bill         555-1675     bill.drowning@hotmail.com       A
Broderick    555-0542     broderick.aliquotiens@yahoo.com R
Camilla      555-2912     camilla.infusarum@skynet.be     R
Fabius       555-1234     fabius.undevicesimus@ucb.edu    F
Julie        555-6699     julie.perscrutabor@skeeve.com   F
Martin       555-6480     martin.codicibus@hotmail.com    A
Samuel       555-3430     samuel.lanceolis@shu.edu        A
Jean-Paul    555-2127     jeanpaul.campanorum@nyu.edu
R
```

```
$ awk '/li/ { print $0 }' mail-list
```

- Search “li”
- When lines with “li” found, they are printed
- $0 means current line

```
$ awk 'length($0) > 80' data
```

- Print every line longer than 80

https://www.gnu.org/software/gawk/manual/gawk.html#Running-gawk

Input Output Redirection

```
ls -al > listings
```

```
home@VirtualBox:~$ ls -al > listings
home@VirtualBox:~$ cat listings
total 324
drwxr-xr-x 26 home home  4096 2012-09-10 10:42 .
drwxr-xr-x  3 root root  4096 2012-09-01 19:43 ..
-rw-rw-r--  1 home home     0 2012-09-10 09:25 abc
```

https://www.guru99.com/linux-redirection.html

```
sed 's/hello/world/' input.txt > output.txt
```

Pipes

Command-1 | Command-2 | …| Command-N

```
cat contents.txt |grep file
```

```
cat sample | grep -v a | sort - r
```

- -v Shows all the lines that do not match the searched string

The contents of the 'sample' file

```
home@VirtualBox:~$ cat sample
Bat
Goat
Apple
Dog
First
Eat
Hide
```

Using 'grep' for searching Apple

```
home@VirtualBox:~$ cat sample | grep Apple
Apple
```

Using 'grep' for searching Eat

```
home@VirtualBox:~$ cat sample | grep Eat
Eat
```

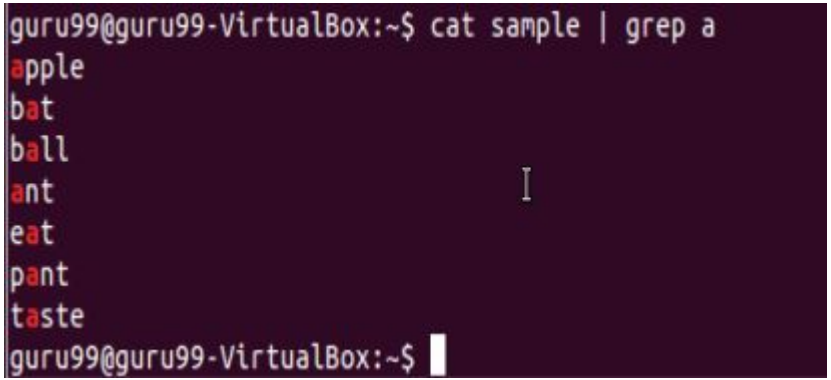
Filtered Results given to the next command

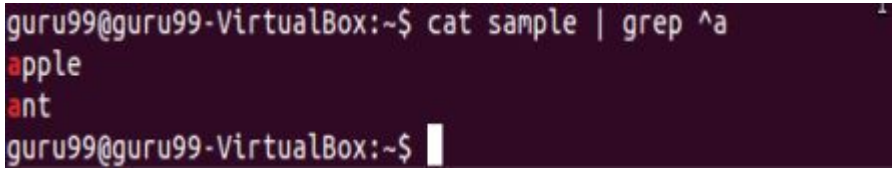
```
home@VirtualBox:~$ cat sample | grep -v a | sort -r
Hide
First
Dog
Apple
```

https://www.guru99.com/linux-pipe-grep.html

Regular expressions in terminal commands

- .
  - replaces any character
- ^
  - matches start of string
- $
  - matches end of string
- *
  - matches up zero or more times the preceding character
- \
  - Represent special characters
- ()
  - Groups regular expressions
- ?
  - Matches up exactly one character





https://www.guru99.com/linux-regular-expressions.html

```
guru99@guru99-VirtualBox:~$ cat sample | grep t
bat
ant
eat
pant
taste
```

```
guru99@guru99-VirtualBox:~$ cat sample|grep p
apple
pant
people
```

```
guru99@guru99-VirtualBox:~$ cat sample|grep -E p\{2}
apple
guru99@guru99-VirtualBox:~$
```

```
guru99@guru99-VirtualBox:~$ cat sample | grep t$
bat
ant
eat
pant
guru99@guru99-VirtualBox:~$
```

- {n} Matches the preceding character appearing ‘n’ times exactly
- {n,m} Matches the preceding character appearing ‘n’ times but not more than m
- {n, } Matches the preceding character only when it appears ‘n’ times or more
- \+ Matches one or more occurrence of the previous character
- \? Matches zero or one occurrence of the previous character

https://www.guru99.com/linux-regular-expressions.html

```
sed '/^foo/d ; s/hello/world/' input.txt > output.txt
```

```
sed -e '/^foo/d' -e 's/hello/world/' input.txt > output.txt
```

```
echo 's/hello/world/' > script2.sed

sed -e '/^foo/d' -f script2.sed input.txt > output.txt
```

https://www.gnu.org/software/sed/manual/sed.html