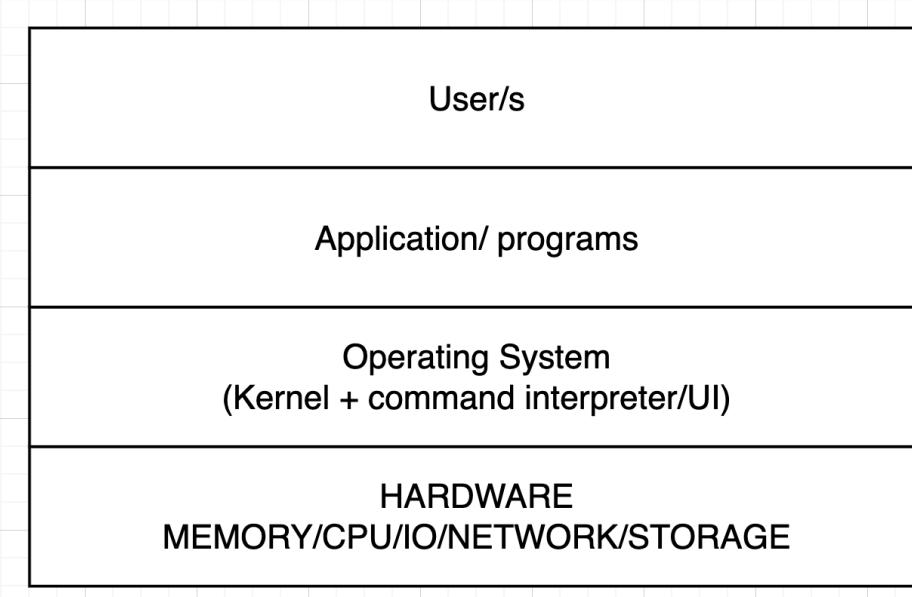


Operating System

- A Software to manage computer hardware
- Consists of
 - Kernel - communicates with and controls the hardware
 - Command interpreter / UI - Deals with the end user



Some of the most common operating systems for computers:

- macOS
- Linux (multitasking + PC or centralized operating system)
- MS Windows
- Unix (multitasking/ centralized operating system)
- MS DOS (single task os)

MS DOS

- command.com → command interpreter (processes internal commands)
- io.sys
- config.sys
- autoexec.bat (executes a bunch of commands to initialize your setup)
- A bunch of executable programs called external commands

Most common applications of olden days

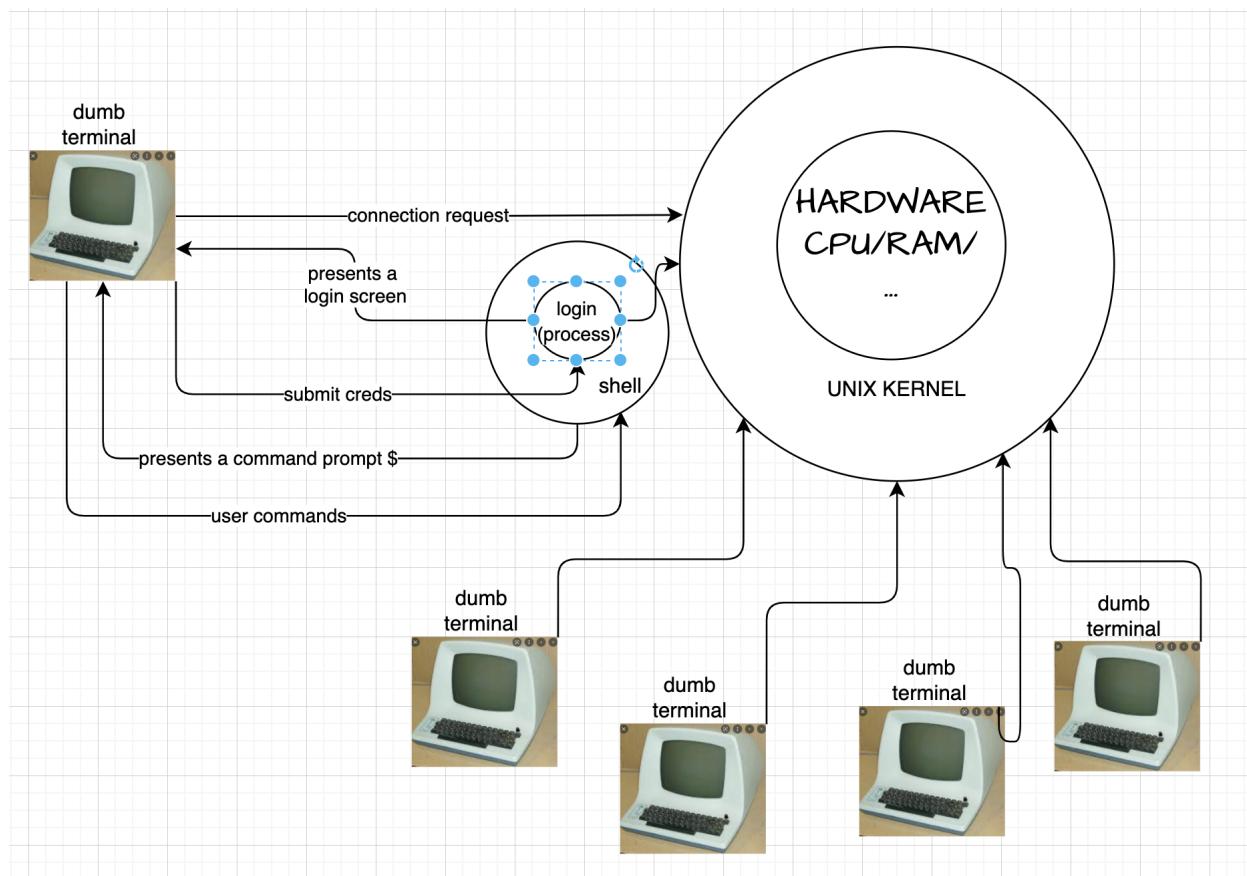
- Wordstar → word processor software (like MS-Word)
- Lotus 123 → spreadsheet software (like MS-Excel)
- Windows 3.11 → a GUI for managing your system/ applications

Responsibilities of an OS:

- Memory management
- I/O Management
- Network management
- CPU / Process management
- Disk/ Storage management
- User/Security management

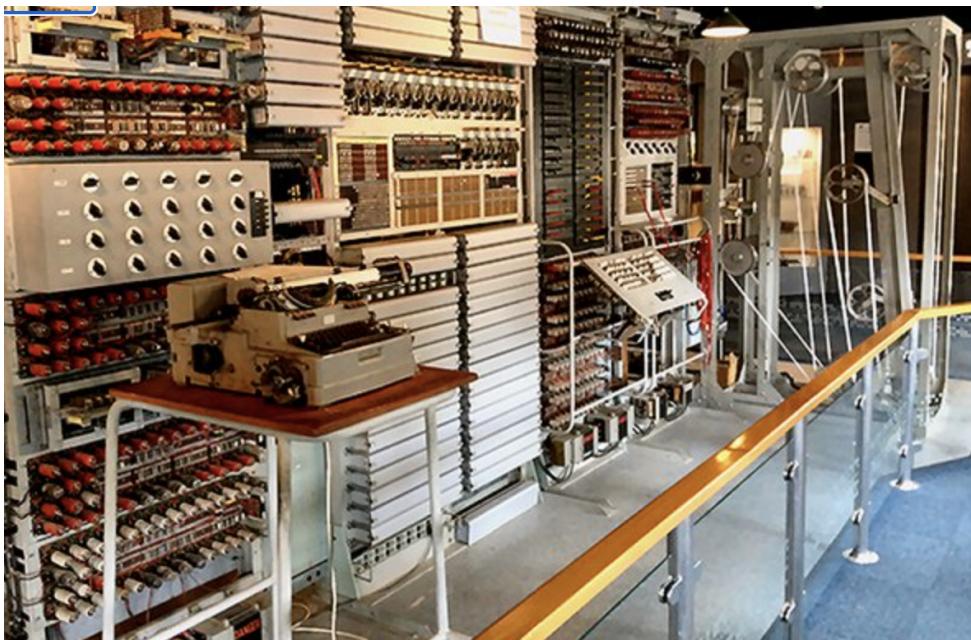
Unix Operating System

- Multi user operating system
- Multitasking operating system
- Centralized operating system



Generations of computer and OS

- 1st generation
 - 1945-55
 - Vacuum tubes as electronic parts
 - OS was implemented using plug boards / punch cards

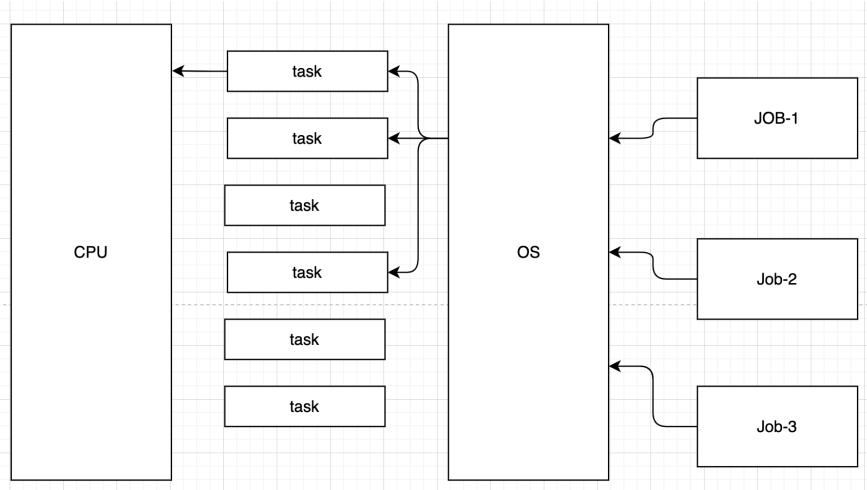


- 2nd generation
 - 1955-65
 - Transistors for electronic components
 - OS was implemented as batch system

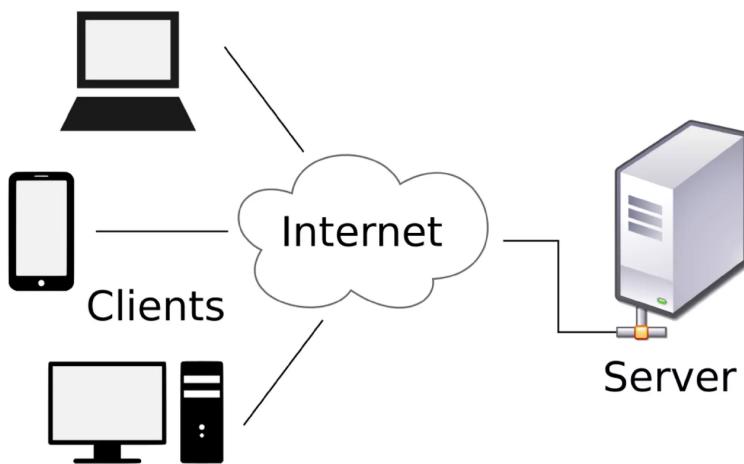
- 3rd generation
 - 1965-80
 - Integrated circuit chips for electronic components
 - Multiprogramming
- 4th generation
 - 1980 onwards
 - LSI
 - Different operating systems

Different types of OS

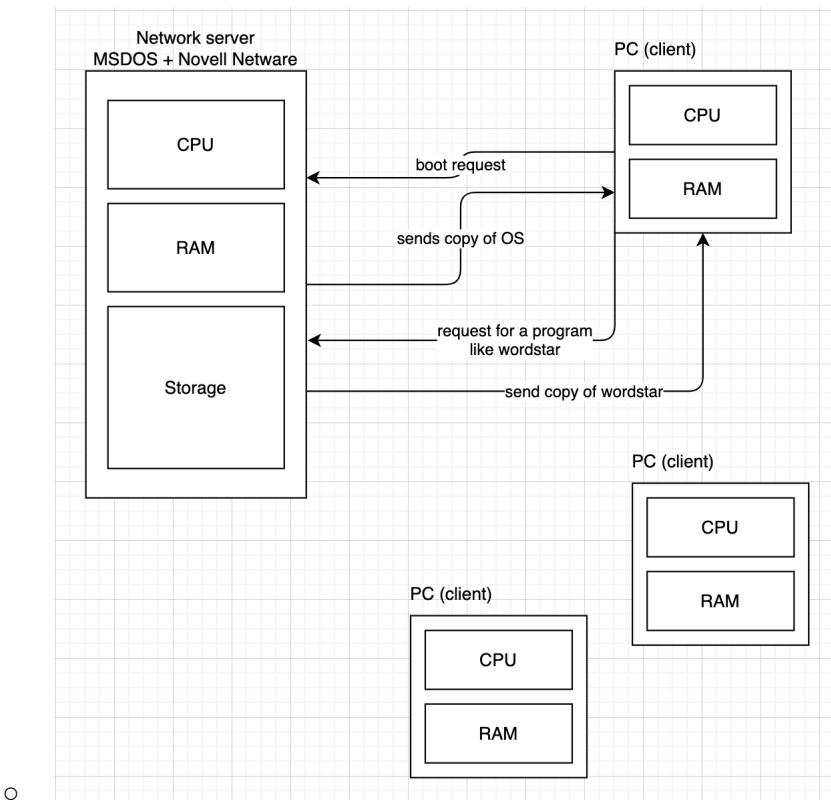
- Batch OS
 - Examples: Payroll system, banking application, controlling of huge machines in industries



- Time-shared operating system
 - Examples: Unix, Linux, Windows, Multics
 - Every task is given a fixed small amount of time (usually in milliseconds or nanoseconds)
 - At a given point in time, only one task is handled by the CPU
 - The task under execution by the CPU is processed for the given fixed time, and then will be suspended to give time to another task
- Distributed OS
 - Lots and lots of computers are interconnected in the internet world!
 - Think of the amount of unused CPU/RAM/Network etc collectively
 - What if we can aggregate all of the unused resources of these distributed computers and make them work like a single computer?
 - Leverages the computational power possessed by computers in the internet world



- Network OS



- Specialized network server is required
- Client computers do not have Storage
- OS and other software are loaded from the remote storage of the network server

-

Linux OS

- Developed by Linus Towalds in 1991



- Free
- Open Source
- Over 600 distributions of Linux exists
- Has two parts:
 - Kernel
 - Talks to and controls the hardware
 - Core part of OS
 - Manages resources using **drivers**
 - Bins and libs (programs and commands)
 - Special programs that help accessing the kernel (from other programs)
 - Provide System calls (functions/apis that can be called from other programs)
 - open() close() read() write()

Process Management

- A process is a program under execution
- A Program is a set of instructions
 - Developers write in high level language like Java, C or C++, etc
 - OS requires these instructions in *machine* (OS) language
- A Process is an active entity
- A Program is a passive entity
- A Process has:
 - PID - Process ID
 - Status (new, ready, run, wait, complete, ...)
 - CPU Registers (memory inside of CPU)
 - I/O Status
 - Schedule information (such as priority information)
 - Account information (such as user, group etc)

Different states of a process

- New
 - You just executed the command or the program
 - OS creates a running process for the same (allocates for example ID)
 - Added to a queue of a bunch of other already existing processes, and the process waits for its turn of the CPU time
 - Being prepared for execution
- Ready
 - Once the process is completely ready with all information like ID etc, the “Ready” is given to the process
 - As if the process “IS READY FOR EXECUTION”
 - Waiting...
 - Since one CPU handles one process at a time
- Running
 - CPU is currently executing/ interpreting/ handling the process
 - There will be a time cap as how long a process is handled by the CPU
 - If the task of the process is not finished with in the given time, the process state will be changed to “Ready”
 - If the task is done with in the given time, then the state of the process is changed to “Complete”
- Wait / Blocked
 - A Process becomes blocked when the process demands I/O access
 - For example, `/s -l | more`
 - The more command waits for user to enter RETURN or SPACE or ‘q’
 - Once the process is blocked, it can not be resumed back to the “Run” state. Instead, it gets the “Ready” state when the user input is complete.

Linux commands:

The linux operating system is case-sensitive. Once logged in, the user is taken automatically to a directory called HOME directory. The value of this is in an environment variable called HOME. To check what is my home directory, you can check the value of the HOME environment variable.

Basic commands:

- clear
 - clears the screen / console
 - CTRL+L or CTRL+K (for macOS CTRL+L or CMD+K)

- cal
 - displays the calendar for the current month/ year
 - cal <year>
 - cal <month> <year>
- pwd
 - displays the “present working directory”
- env
 - displays the list of all environment variables
- echo
 - echoes whatever is given after that
 - can be used to display the value of an environment variable
 - The variable should be prefixed with “\$” symbol
 - For example:
 - echo \$HOME
 - echo \$PATH
 - echo “My home directory is \$HOME”
- export
 - creates an environment variable
 - assign key=value without any additional space
 - If the value has spaces then use double quotes
 - For example:
 - export MY_NAME=”Vinod Kumar”
- unset
 - deletes an environment variable
- cd
 - change directory
 - cd .. → changes to the parent directory
 - cd → changes to the home directory
 - cd / → changes to the root directory
 - cd ../../ → changes to the parent of parent directory
- whoami
 - displays the username of the currently logged in user
- who
 - displays the list of all users
 - The option -H can be used to display the heading
 - Displays:
 - username
 - tty
 - time of login
 - IP address of the user’s terminal (SSH or Putty)
- write
 - used for sending a message to an intended recipient
 - press CTRL+D to mention EOF

- If the recipient has logged in using multiple terminals, the tty also can be specified
 - write u12
 - write u12 pts/16
 - echo Hello | write u3
 - ls -l | write u3
- wall
 - write all or broadcast
 - press CTRL+D to mention EOF
 - echo hello | wall
- mesg
 - displays and controls the receival of messages from other users
 - mesg -n → disables messages from other users (except root)
 - mesg -y → allows messages from other users
- mkdir
 - make directory or directories
 - mkdir python_files java_files
 - -p flag allows the creation of an entire tree structure of directories
 - mkdir -p java_files/java-basics/src/com/edgeverve/programs
- rm -r
 - can be used for deleting directories and subdirectories recursively
- ls
 - short form for list
 - lists all the files and directories in the current directory or the specified directory
 - -l option displays the same in long list format
 - 7 columns
 - column 1 made up of 10 letters (1 + 9)
 - the first letter denotes the kind of file (file or directory or link or ...)
 - the remaining 9 letters can be grouped in to set of 3 letters
 - rwx (for user) rwx (for group members) rwx (for others)
 - r → read, w → write, x → execute (for a directory, it is "cd")
 - column 2 → no. of links to the file/ directory
 - column 3 → name of the user who owns the file/ directory
 - column 4 → group name of the owner of the file/ directory
 - column 5 → size of the file (in bytes) or directory (in blocks)
 - column 6 → date/time
 - column 7 → file or directory name
 - -A or -a option can be used to list hidden files and directories
- cp
 - copy files and/or directories
- ln
 - create links between two files
 - one physical file, but multiple names