Operating System Services.

1) User interface → command Line Interface (CLI)
   → Graphical user interface (GUI)

2) program Execution.
3) I/O operation
4) File system manipulation
5) communication (communication between process)
6) Error detection.
7) Resource Allocation
8) Accounting
   ↓
   which user use how much & what kind of resources
   → with this data computing. scientists can improve
9) Protection & security

User operating system Interface

→ CLI (or) command Interpreter
→ GUI

* Some OS include the command Interpreter in kernel.
* Others like windows XP, UNIX treats as Special program
  (windows XP → cmd prompt; UNIX → Terminal)

* command interpreter are also known as shells
  → Bourne shell
  → C shell
  → Bourne - Again shell (BASH)
  → Korn shell (etc).

System calls

→ they provide an interface to the services that are made available by an Operating system.

User mode:

→ If a program is executing in user mode then it doesn't have direct access to resources.

→ If programcrash in user than entire system doesn't crash.

Kernal mode:

→ has direct access to resources.

→ it is in privileged mode.

→ If program crash in kernel mode then entire system crashes.

→ The call program makes when it tries to switch from User mode to kernel mode to access resourses.

→ System calls are generally available as routines written in C & C++.

Types of system calls.

1) process control: for controlling process

→ end, abort
→ load, execute
→ create process, terminate process
→ get process attributes, set process attributes
→ wait for time
→ wait event, signal event.
→ allocate and free memory

2) File manipulation:

→ create file, delete file
→ open, close
→ read, write, reposition
→ get file attributes, set file attributes

System calls.

3) Device manipulation:

→ request device, release device
→ read, write, reposition
→ get device attributes, set device attributes
→ logically attach or detach devices.

4) Information Maintence.

→ get time or date, set time or date
→ get system data, set system data
→ get process, file, or device attributes
set process, file or device attributes

5.) Communication.

→ create, delete communication connection
→ send, receive messages
→ transfer status information
→ attach or detach remote devices.

System programs.

→ They are above OS in hirracky
→ System programs provide a convenient environment for program development and execution.

System program.

→ File managment
- create, delete, copy
- rename, print, dump
- List,

→ Status information
- Date, time
- Amount of available memory
- Number of users
- detailed performance
- logging & debugging info

→ File modification.

→ Programming language support
- compilers
- Assemblers
- Debuggers
- Interpreters.

→ Program Loading and execution.
- → Absolute loaders
- → Relocatable loaders
- → Linkage editors
- → overlay loaders

→ Communications
- creating virtual connections
- to browse webpages
- send electronic-mail message
- send messages.

OS Design & Implementation.

User goals
- Convenient to use
- Easy to learn & use
- Reliable
- Safe & fast

System goals
- Easy to design, implement
- maintains operate
- Flexible, reliable
- Error free & efficient.

mainly
- choice of hardward
- type of system

Mechanism → how to do something.

Policies → what will be done.

* seperation of policy from mechanism.

Implementation.

→ initially OS were written in assembly languages

→ but now they are written in higher-level languages like C, C++ etc.

* MS-DOS → Intel 8088 assembly language
  ↳ so, it supports only intel family

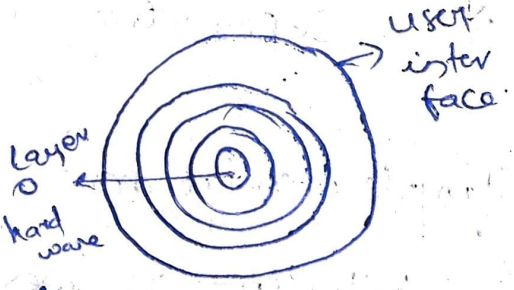
* Linux → C

Structures of operating systems

→ Simple structure. (MS-DOS)
- → Base hardware is accessed by everything
- → more vulnerable
- → when program fail entire system crashes.

→ Monolithic structure (earlier UNIX OS)
- → most of the functionalites are packed into one level i.e, kernel.
- → Implementation, debugging is difficult.

→ Layered structure.
- → Broken down different functionalites into layers
- → easy to implement and debugg.
- → difficulit in designing.
- → not very efficient.



→ Micro-Kernals :

→ we have small kernel → it provides core functionalities

→ we remove non-essential components in kernel and we implement them as system & user level programs

→ message passing for communication in micro-kernals

→ most of functionalities are implemented in user mode

→ performance decrease due to increased system overhead.

→ Modules → see video.

Virtual Machines :

→ creates the illusion that each separate execution environment is running its own private computer.

→ Virtual machine software runs in kernel mode

→ Virtual machine itself runs in User mode.

→ Virtual user mode
→ Virtual kernel mode } → physical User mode

Operation System Generation & System Boot.

* System generation (SYSGEN)
  ↳ OS will be generated in such a way that it will suitable to specific computer which your are using.

→ Sysgen program must determine:
- → What CPU is to be used?
- → How much memory is available?
- → what devices are available.
- → What OS options are desired?

System Boot

Booting → starting a computer by loading the kernel.

First thing that runs when computer is on.

→ Bootstrap program → it locates kernel.
↳ ROM

Firmware → ROM

→ Both OS & Bootstrap

→ EPROM solved the problem

→ Small devices.

changing something we need to change rom chip.