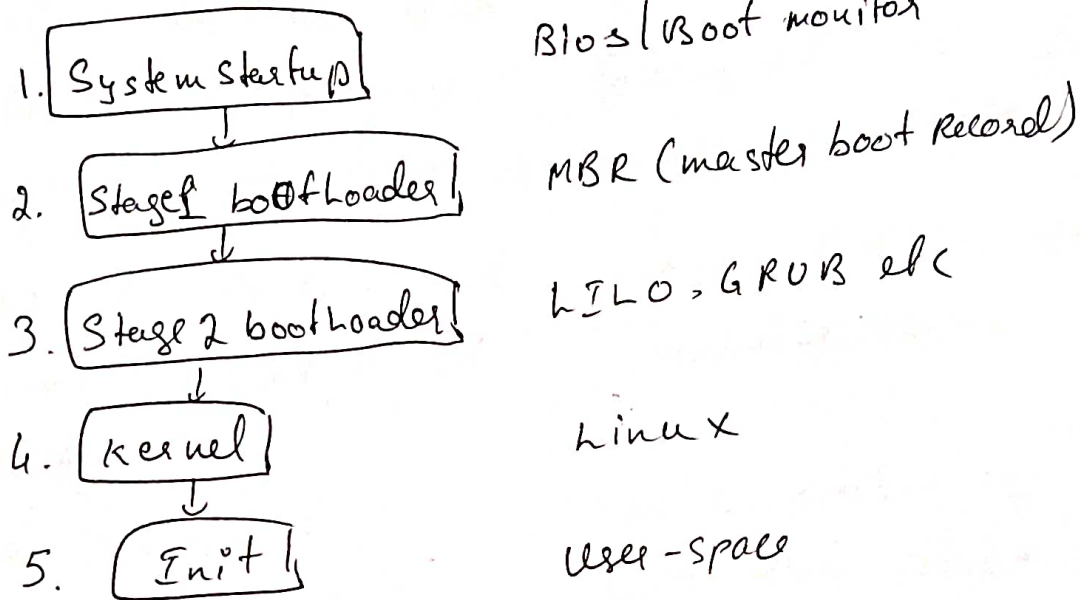


01)

Booting Process



1. System Startup → When a system is first booted / reset the processor executes code at a well known location. In PC - BIOS stored on a flash memory on mother board.

Q → Bios is a small piece of code (512 bytes) - 1st processor
 → Bios identifies HW (boot device) which is bootable and active
 → From this HW, linux is booted where MBR contains primary boot loader. MBR loaded into RAM, Bios yields control to it.

2. Stage 1 Boot Loader → job is to find and load secondary boot loader

→ MBR loads this Stage 1 boot loader

5 stage 2 boot loader -> Aply called as kernel loader
its task is to Load the linux kernel.

-> 1st & 2nd stage boot loader combined are called
LILO (Linux loader) or GRUB (Grand Unified Boot
Loader)

File system is consulted, default kernel
image and initial image are loaded into memory.

4. kernel -> At the head of the kernel image, a routine
does some HW setup and decompresses the
kernel image and place it in high memory.

-> kernel is then called and kernel boot begins

-> During this, initrd is copied into RAM and
mounted. This acts as temporary root file system
in RAM

5 init -> kernel starts the first user space application

-> compiled into C library

-> first prog /sbin/init

Q2) Function of OS

01) Security

- > Protects user data.
- > It prevents unauthorized access to programs and user data.

02) Job accounting

- > OS keeps track of time and resources used by various tasks.

03) Error detecting aids

- > Constantly monitors the system to detect errors and avoid malfunctioning of system.

04) Coordination between software and users.

- > OS co-ordinates and assigns interpreters, compilers, assemblers.

05) Memory management

- > The OS manages the primary memory.
- > It keeps track of memory, which bytes are used by which prog,
- > Allocate memory for the process, deallocate when the process has terminated.

06) Processor management

so it decides the order in which the processes have access to the processor.

→ keeps track of the status of processes

07) File management

→ A file system is organised into directories for efficient navigation and usage.

08) Monolithic and Micro Kernel difference with diagram

Micro Kernel

→ user services and kernel, are kept in separate address space

→ OS is complex to design

→ Micro Kernel is small in size

→ Easier to add new functionalities

→ Failure of one component does not affect the working of micro kernel.

→ Execution speed is low

Monolithic Kernel

Both user services and kernel services are kept in same address space.

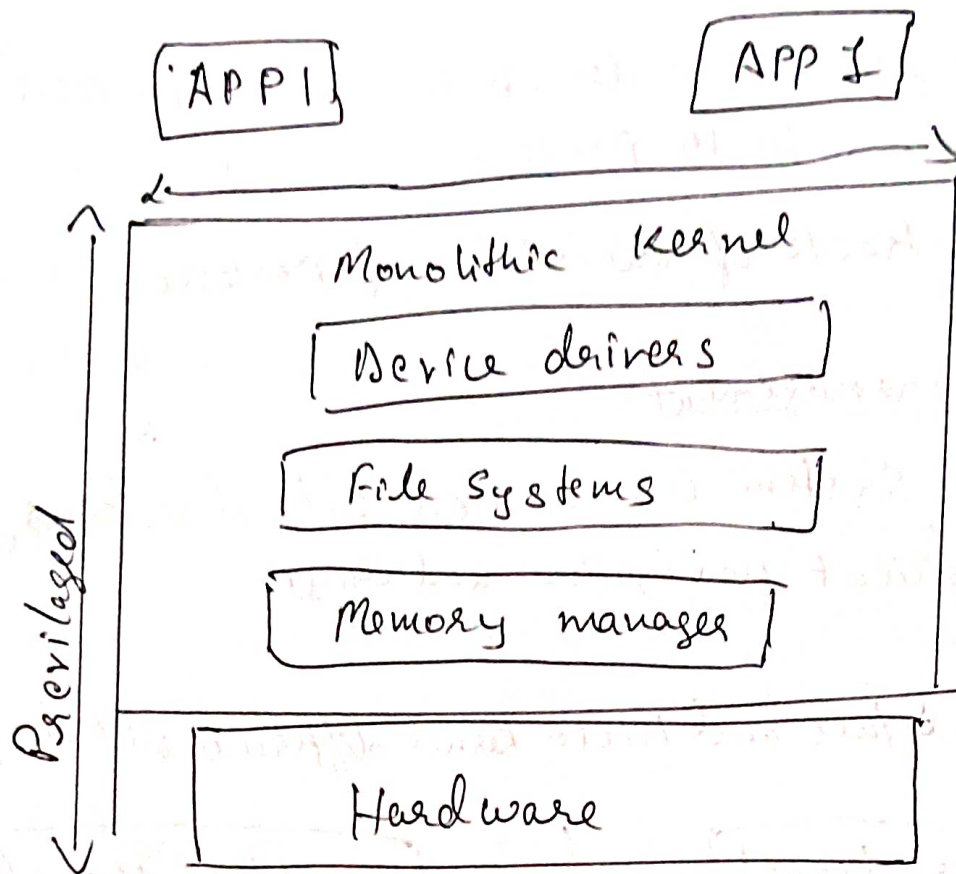
OS is easy to design and implement

→ Monolithic kernel are larger in size.

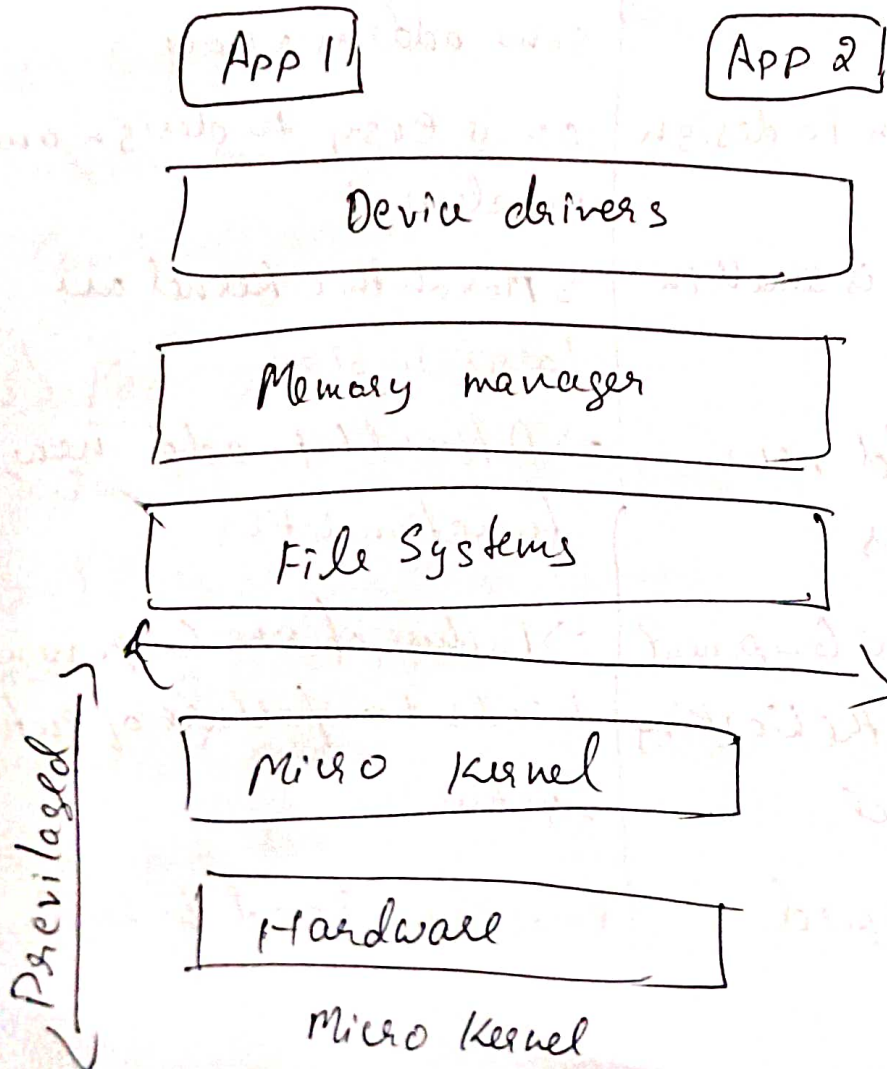
→ Difficult to add new functionalities

→ Failure of one component leads to failure of entire system.

Execution speed is high.



Monolithic Kernel



Micro Kernel

Q4) UEFI and Legacy boot

-> UEFI stands for unified extensible firmware interface. Most new motherboards consist of this type. It has more advantage than using BIOS.

Most importantly, it provides user friendly graphical user interface (GUI).

BIOS provides blue screen. BIOS cannot recognise large storage drives, UEFI provides a good alternative.

-> In regular BIOS that uses the keyboard to select the option UEFI allows controls via mouse.

UEFI contains Secure boot.

-> Legacy BIOS used legacy BIOS firmware. It stores a list of installed storage devices that are bootable.

BIOS performs POST (Power on Self-test)

Q5) Commands on Windows to check disk partition

-> open power shell terminal

-> Type diskpart

-> `Diskpart > List disk`

↳ It will list any detected disks.

-> `Diskpart > List volumes`

↳ It will list detected volumes

-> `Diskpart > List partition`

↳ This will list the current partitions on the device

Q6) Commands to check services in windows

-> List all services

↳ `sc queryen type = service state = all`

-> Search for specific service

↳ `sc queryen type = service state = all /find /i "Service`

name": myService

Q7) Steps to start or stop services in Windows

stop service

-> open start

-> search services

- Click on the service you want to start
- click start button
- Apply button

stop service

- open start
- search services
- click on the service you want to stop
- click stop
- click apply