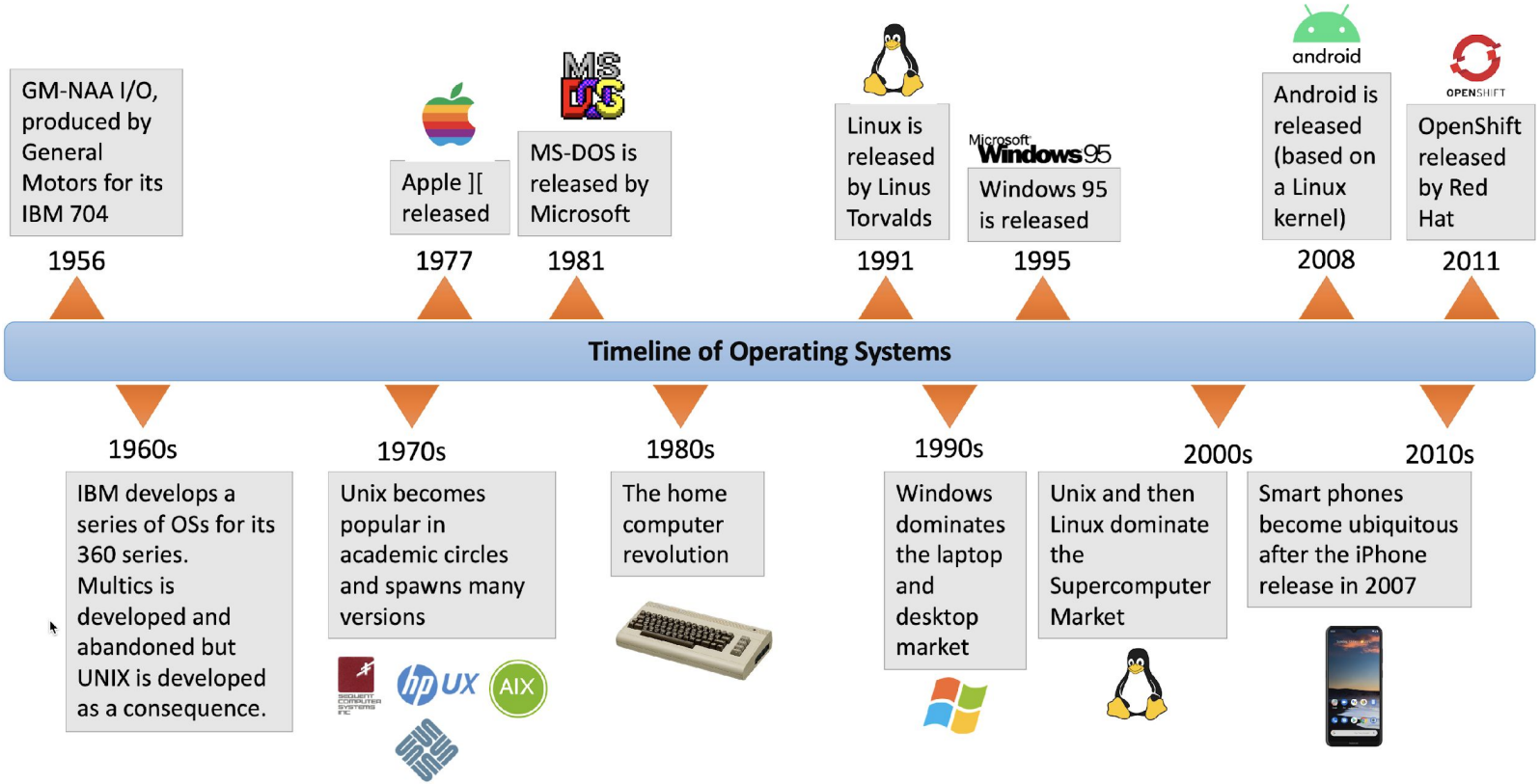
Timeline of OS

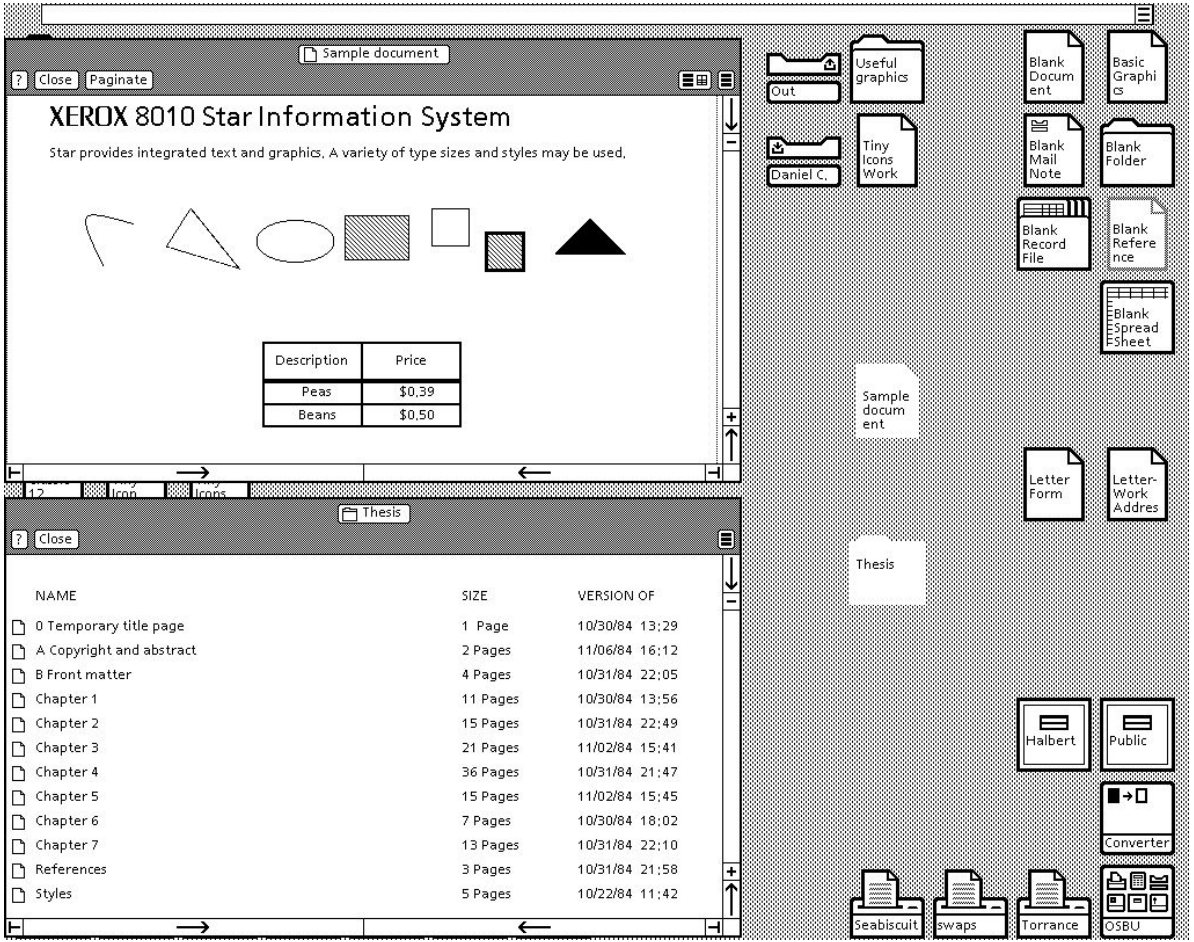


Timeline of OS

- IBM 704 was the first mass-produced computer
- GM-NAA (General Motors-North American Aviation) I/O is the OS used in the IBM 704
- There was no UI in the OS as we know now
- It was mostly based on command prompts (terminals)

Timeline of OS

- Xerox 8010 Star (released in 1981) was the first system that was referred to as a fully integrated desktop computer including applications and a GUI



Timeline of OS

- Bill Gates got lucky to sign the so-called “Deal of the Century” with IBM to design a new OS for IBM machines
- The deal allowed Microsoft to add an OS with $50 per PC and IBM did not own copyright for the OS
- This allowed Microsoft to start the their OS dominance on the PC domain

Timeline of OS

- Developed by Apple Computer, Inc for their new product, the Macintosh home PC, The Macintosh 128K, was released in 1984
- was widely advertised (the famous 1984 commercial is available below).
- Mac OS was the first OS with a GUI built-in
- It was also one of the first consumer computers with mouse!
- Even though Microsoft introduced mouse in their PCs a bit earlier
- The use of GUI with mouse was not Steve Job's idea, the idea was taken during his visit in Xerox PARC (Palo Aalto Research Center)
- Jobs reportedly traded US $1 million in stock options to Xerox for a detailed tour of their facilities and current projects
- One of the things Xerox showed Jobs was the Alto, which sported a GUI and a three-button mouse

Operating Systems

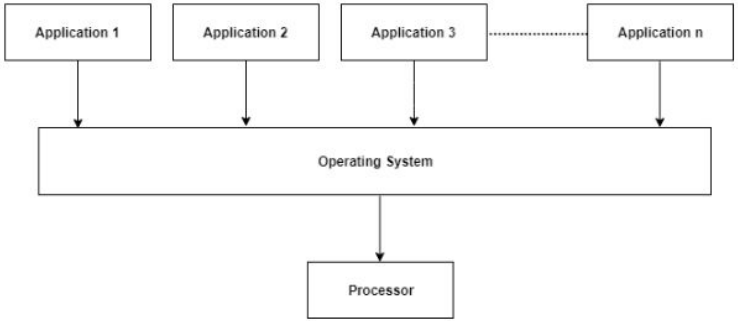
OS Architecture

FBA

Operating System Architecture

Single-Processor Systems:

- One main CPU capable of executing a general-purpose instruction set.
- Almost all single processor systems have other special-purpose processors (device-specific processors), which run a limiter instruction set.

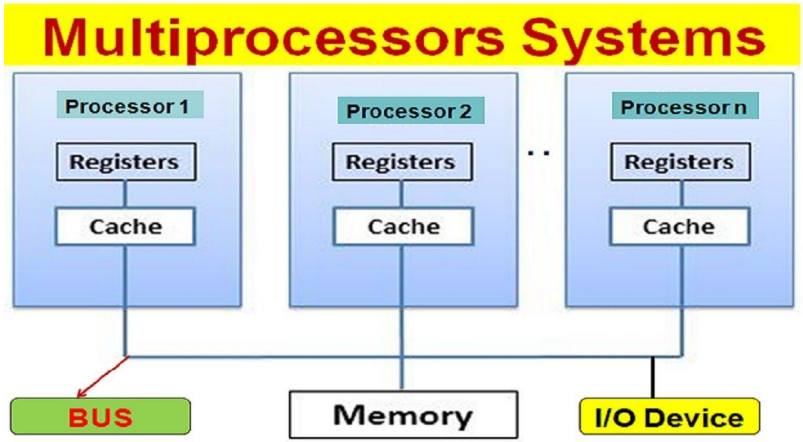


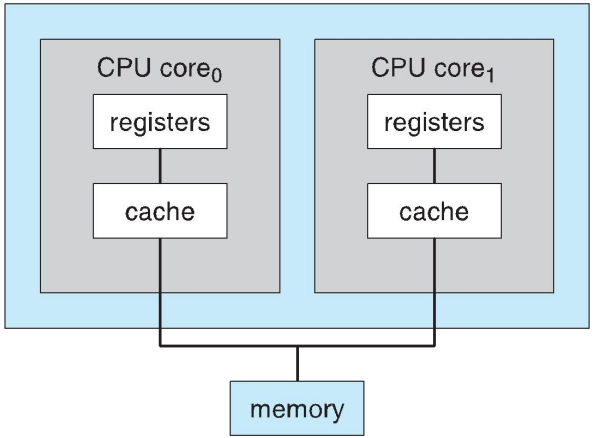
Single Processor System

Operating System Architecture

Multiprocessor Systems (parallel systems or multicore systems):

- Such systems have two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices.



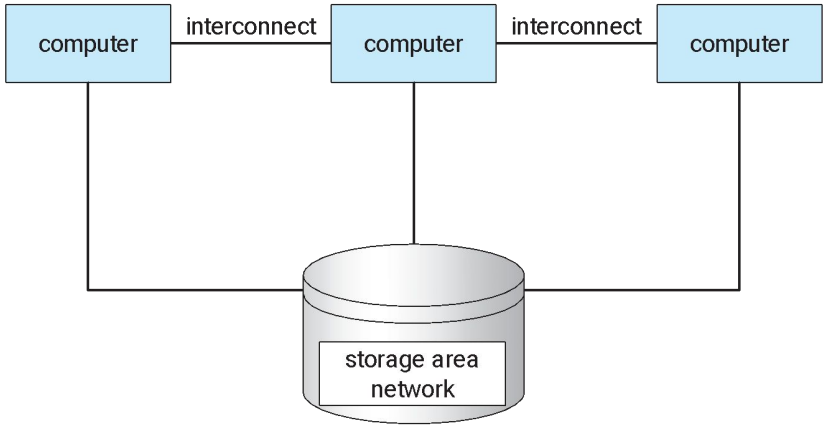


A dual-core system

Operating System Architecture

Clustered Systems:

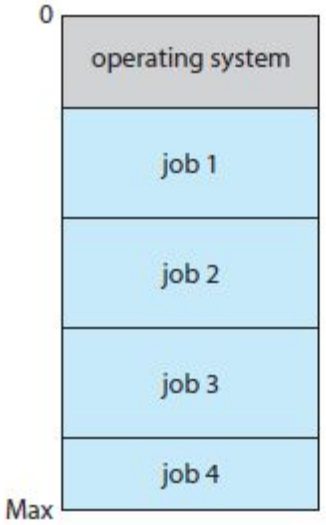
- Special kind of multiprocessor system which gathers together multiple CPUs. They are composed of two or more individual systems- or nodes - joined together.
- Clustered computers share storage and are closely linked via a local-area network (LAN) or a faster interconnect, such as InfiniBand



Operating System Structure

Multiprogramming:

- Increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.
- OS keeps several jobs in memory simultaneously. As main memory is small, it keeps the jobs on the disk (job pool) which waits there to be allocated in the main memory.
- OS picks and begins to execute one of the jobs in memory. Eventually, the job may have to wait for some task. OS picks another job from the pool to execute while the previous one waits.



Memory layout for a multiprogramming system.

Requirements of Multiprogramming

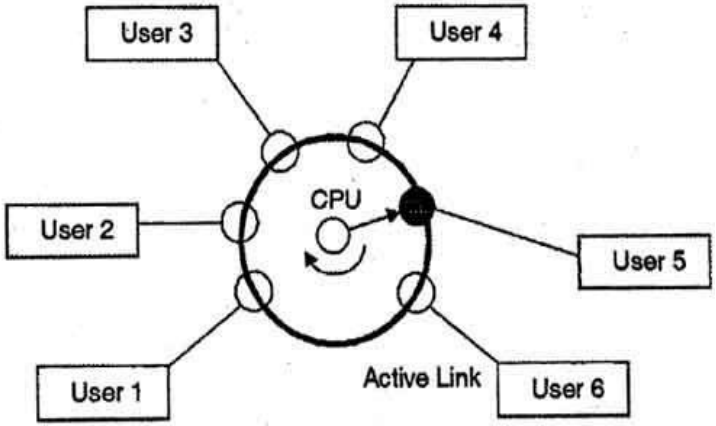
- **Job Scheduling:** If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them.
- When OS selects a job from the job pool, it loads that job into memory for execution. Having several programs in memory at the same time requires some form of memory management.
- **CPU Scheduling:** if several jobs are ready to run at the same time, the system must choose which job will run first.
- Running multiple jobs concurrently requires that their ability to affect one another be limited in all phases of the operating system.

★ If processes don't fit in memory, swapping moves them in and out to run from main memory achieving this goal is virtual memory.

Operating System Structure

Time Sharing:

- CPU executes multiple jobs by switching among them.
- Switches occur so frequently that the users can interact with each program while it is running.
- requires an interactive computer system, which provides direct communication between the user and the system.
- Response time should be short.



Operating System Operations

- Modern operating systems are interrupt driven
- Events are almost always signaled by the occurrence of an interrupt or a trap
- A trap (or an exception) is a software-generated interrupt
- For each type of interrupt, separate segments of code determine what action should be taken
- Errors can occur when one erroneous program modify another program, the data of another program, or even the operating system itself.
- A properly designed operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.

Types of System Call

Type	Windows OS	Linux OS
Process Control	**CreateProcess**() **ExitProcess**() **WaitForSingleObject**()	**fork**() **exit**() **wait**()
File Manipulation	**CreateFile**() **ReadFile**() **WriteFile**() **CloseHandle**()	**open**() **read**() **write**() **close**()
Device Manipulation	**SetConsoleMode**() **ReadConsole**() **WriteConsole**()	**ioctl**() **read**() **write**()

Types of System Call

Type	Windows OS	Linux OS
Information Maintenance	**GetCurrentProcessID**() **SetTimer**() **Sleep**()	**getpid**() **alarm**() **sleep**()
Communication	**CreatePipe**() **CreateFileMapping**() **MapViewOfFile**()	**pipe**() **shm_open**() **mmap**()
Protection	**SetFileSecurity**() **InitlializeSecurityDescriptor**() **SetSecurityDescriptorGroup**()	**chmod**() **umask**() **chown**()

System Programs

System programs, also known as system utilities, provide a convenient environment for program development and execution.

These system programs provide -

- File management
- Status information
- File modification
- Programming-language support
- Program loading and execution
- Communications
- Background services

System Boot

- When power initialized on system, execution starts at a fixed memory location.
  - ➔ Firmware ROM used to hold initial boot code

- Operating system must be made available to hardware so hardware can start it.
  - ➔ Small piece of code – bootstrap loader, stored in ROM locates the kernel, loads it into memory, and starts it
  - ➔ Sometimes two-step process where boot block at fixed location loaded by ROM code, which loads bootstrap loader from disk

- Common bootstrap loader, GRUB, allows selection of kernel from multiple disks, versions, kernel options

- Kernel loads and system is then running.

Operating Systems

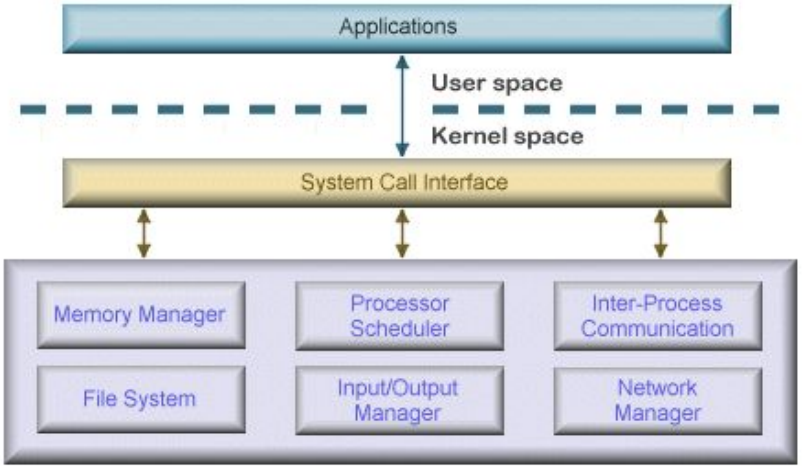
OS Structures

FBA

OS Structure

Simple/Monolithic structure:

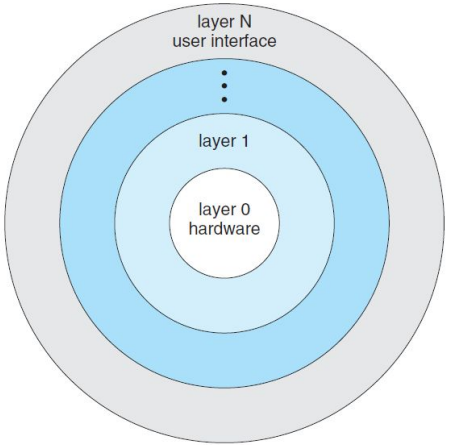
- Earliest and most common architecture
- Every component of OS is in the kernel and can communicate with each other directly
- Complex and Large (millions line of code ,) hard to maintain



OS Structure

Layered structure:

- OS is divided into layers.
- Each layer can use services of its lower layers.
- Easy to debug and develop.
- Less efficient as each layer adds some overhead



OS Structure

Microkernel structure:

- Moves as much from kernel into user space
- Communication takes place between user modules using message passing
- Easier to extend a microkernel
- More reliable(less code running in kernel mode) and secure
- Performance overhead