CS2106 Live Class Lec01: Function and classification of OS

$_/|/|U_C h@NgRu!$

May 10, 2021

1 The focus area of the module

1. OS structure and architecture
2. Process management
3. Memory management
4. File management
5. OS protection mechanism

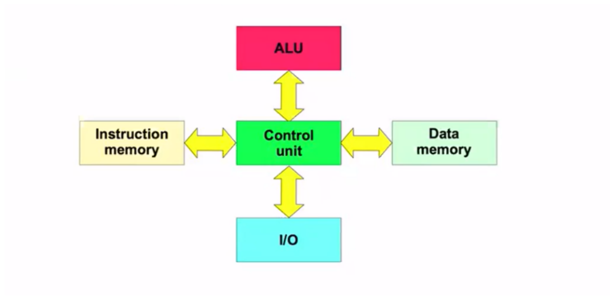
2 Operating system

A program that acts as an intermediary between a computer user/computer program and the computer hardware

3 History of OS

3.1 computer with no OS

ENIAC (no program, run machine manually)
Harvard Mark I(10 years after ENIAC, the program is in paper tickets with small holes



the harvard mark I: the seperate way for code and data

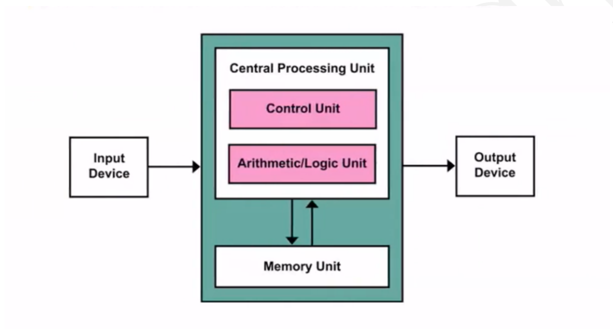
3.1.1 Advantage

minimal overhead (nothing to slow the speed)

3.1.2 disadvantage

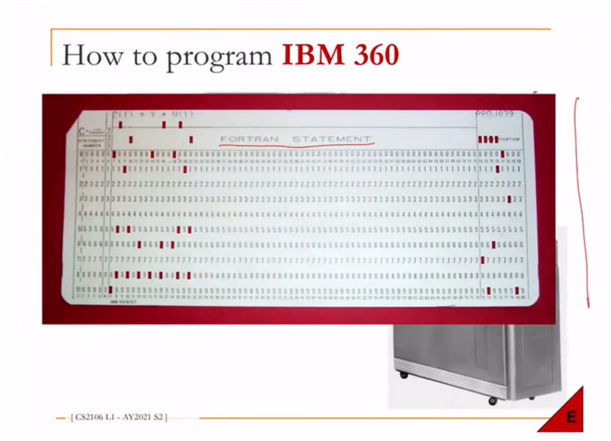
Not portable, inflexible
Inefficient use of computer resources

3.2 start of OS

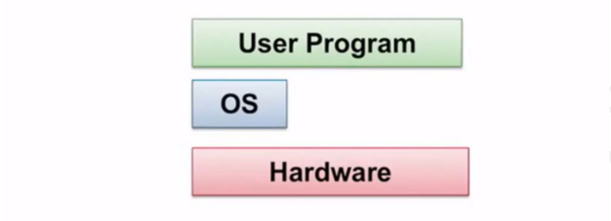


The von Neumann architecture memory are shared by instruction and data , which means one program can manage another
One piece of software can manipulate data and instruction of the other(which makes the OS possible)

3.3 The first OS: IBM 360


How to program **IBM 360**

punch card contains program compilers are needed to compile the program into binary
The difference here is that, the code of the IBM360 should be compiled
Later, the paper card is replaced by tapes(so that multiple program can be given at one time)



3.3.1 Batch OS

2Execute user program one at a time

3.3.2 User job

Still interact with hardware directly
With additional information for the OS

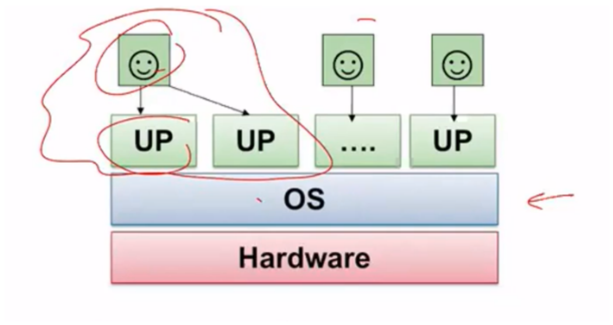
3.3.3 possible improvement for the IBM 360

1.Multiprogramming
2. Multiple user(windows cannot make)
3. time-sharing feature

3.3.4 time-sharing

allow multiple users to interact with machine using terminal
User job schedulling (illusion of concurrency) Famous example: CTSS developed at MIT 1960s ; Multics(1970, paren of unix)
time-sharing is a feature that unix behaves like

4 time-sharing OS



OS provides sharing of : CPU time, memory and storage
virtualization of hardware: each program execute as if it has all the resources itself(every thing is file)
The inventor of Unix and C:Dennis Ritchie(very very very strong man)

1. Allowing multiple program to run
2. time-sharing

C is the only high level language allows a intermediate contact with the hardware

5 personal computer

The first computer for individual: Apple I and Apple II

5.1 Windows model

multiple users, but only one at a time

5.2 Unix model

generally time-sharing

6 OS function: Abstraction

6.1 reason

1. large variation in hardware configurations(e.g number of CPU core, memory(different capacity and capability)); however hardware in the same category has well defined and common functionality
p.s: compacity: the ability to store data
p.s: compability(latency, bandwieth)

6.2 OS serves as an abstraction

1. hide the different low-level details
2. present the common high level functionality to users
In this way, users can use the same essential task in different computer with similar action without considering the different and low-level

6.2.1 Efficiency and portability

portability: even some hardware changed, the use of computer would not change
efficiency: can achieve multiple program without human intervention

7 OS function:Resource allocator

OS allocate computer resources while multiple programs should be allowed to execute simutaneously and some of them need the same resource at the same time(arbitary conflicting requests: for efficient and fair resource use)

8 OS function: control program

1. prevent errors and improper use of the computer
2. provides security and protection

8.1 Program can misuse the computer

Accidentally: due to coding bugs
Maliciously: virus, malware

8.2 Multiple users can share the computer

to sparate user space

9 Summary

9.1 Manage resources and coordination

Process synchronization, resource sharing

9.2 Simplify programming

abstraction of hardware, convenient services

9.3 Enforce usage polices

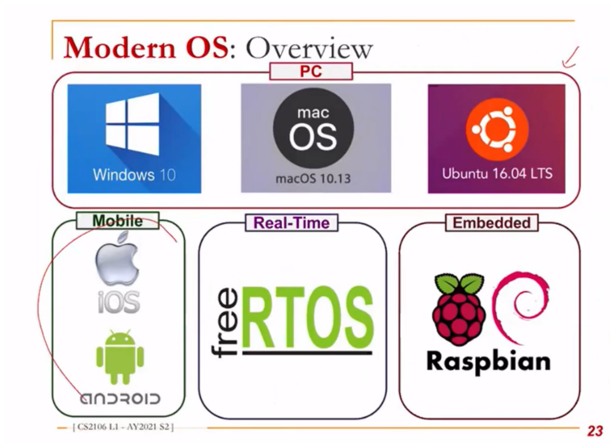
efficiency

9.4 User program Portability

across different hardware

9.5 Efficiency

sophisticated implementation
optimized for particular usage and hardware



Real-time system: the system must response within a very short time # The below part should be reviewed, the genius prof talks about too fast

10 OS: factors to judge a OS

10.1 Flexibility

10.2 Robustness

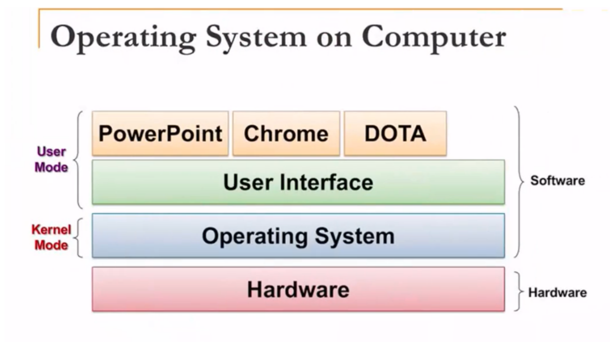
how easy it is to run if some parts of the hardware are changed

10.3 maintainability

10.4 Performance,scalability

how many program can be run at the same time

11 the location of OS in modern computer

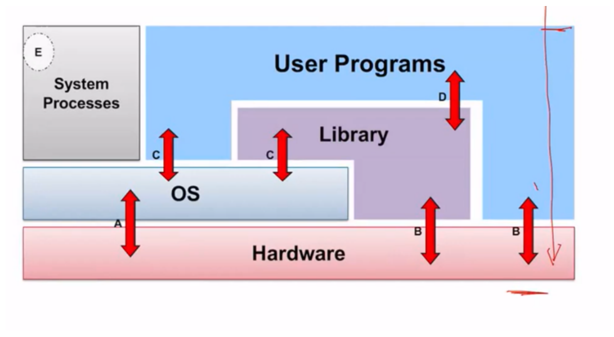


11.1 user moder

11.2 kernel model

can execute some privilege instruction(only the OS can do)
kernel mode is a state of the underlined processor

12 Generic OS: illustration



1. The hardware bottom, the user program top
2. usually the program can directly run on hardware
3. some instruction may need some library(such as math livrary); this library will execute directly in the hardware
4. Library may call the other library (e.g. malloc call sys call, sys call execute in hardware)
5. there are many system processes can call system process

IMPORTANT: OS is also a software

13 the program in specific can hve these operation

1. Deals with hardware issues
2. Provides **system call interface**
3. special code for **interrupt handler**, device driver

14 the way OS implement the OS order(e.g. open)

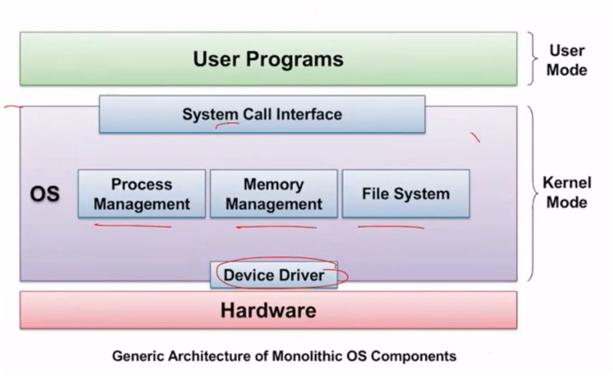
when a user program call OS to open a file, the OS itself cannot call others

15 The implementation of OS

In the past, assembly code
Now HLLs: C/C++(C is a high level language that can call hardware)

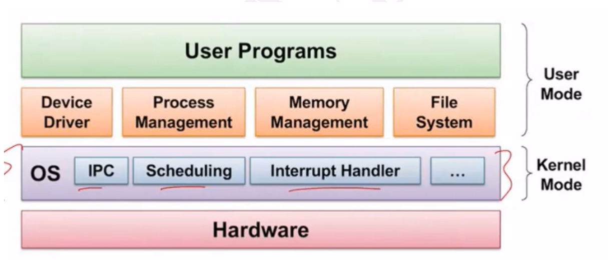
15.1 Monolithic

1. One BIG special program
2. good SE prinnciples are still possible



15.2 Microkernel

Take much works out of kernel and let kernel minimum complexity
1. Very small and clean
2. Only provides only basic and essential facilities
3. Inter-process communication(IPC)



Many functions(e.g. File system, memory management) are out of OS and when these functions need to communication, they should call inter-process communication(IPC), which is slow

15.3 layered system

Generalization of monolithic system
Organize the components into hierachy of layers: upper layer make use of lower layer; lower layer is the hardware while the highest layer is the user interface

15.4 Client-server system

Variation of microkernel
Two classes of processes: client process request service from server process; server process built on top of the microkernel; client and server processes can be on different machines

16 Virtual machine

16.1 Two function

1. Multisystem on the same machine
2. debug the OS itself(kernel development)

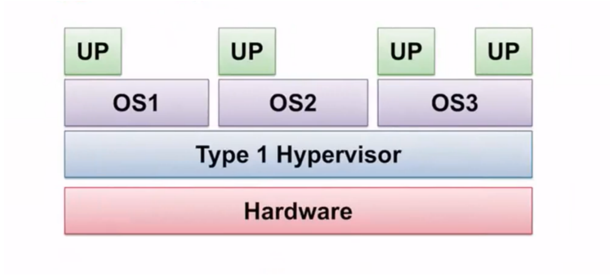
16.2 virtual machine

A software emulation of hardware
Virtualization of underlying hardware
Then normal oS can run on sop of it

16.3 hypervisors

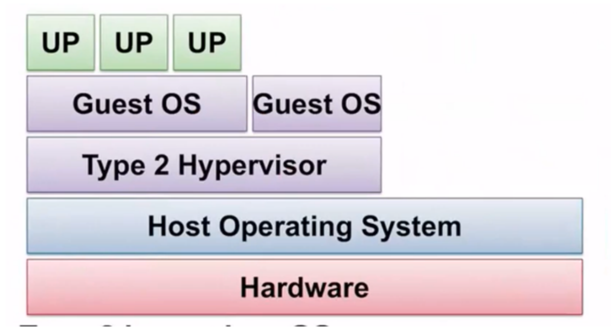
manage virtual machine
Something that similar to OS, but specialiesed to virtual machine

16.4 Type 1 hypervisor OS



Provides indivisual virtual machines to guesr OSes(the MAC with windows system)
Hypervisor is directly running on top of the hardware

16.5 Type 2 hypervisor



run in host OS(virtual Box)
guest OS runs inside virtual machine