Introduction to OS

Introduction

MOS 1.1 – 1.3

Mahmoud El-Gayyar

elgayyar@ci.suez.edu.eg

Why an Operating Systems course?

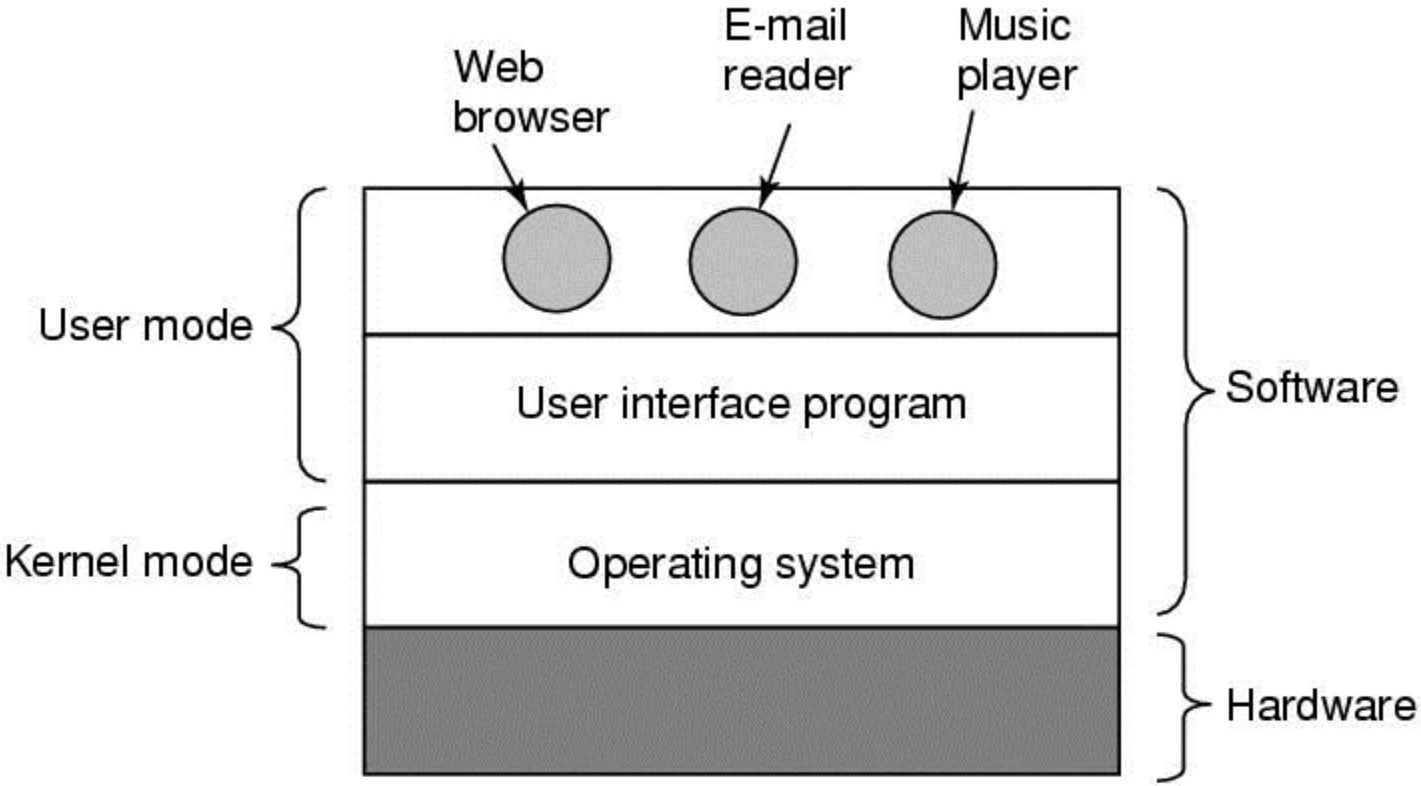
- Understanding of inner workings of “systems”
- OS-related assignments you may see in real life:–
  - “Design and develop an operating system to do THIS” (not very likely)
  - “Select an operating system for an application that will do THAT” (probable)
  - “Design and develop this application on THAT system to exploit its special features” (likely)

Outline

- ***What is an Operating System?***
- History of OS?
- Computer Hardware Review

What is an Operating System?

Where is the software?



OS – It is a Control Program

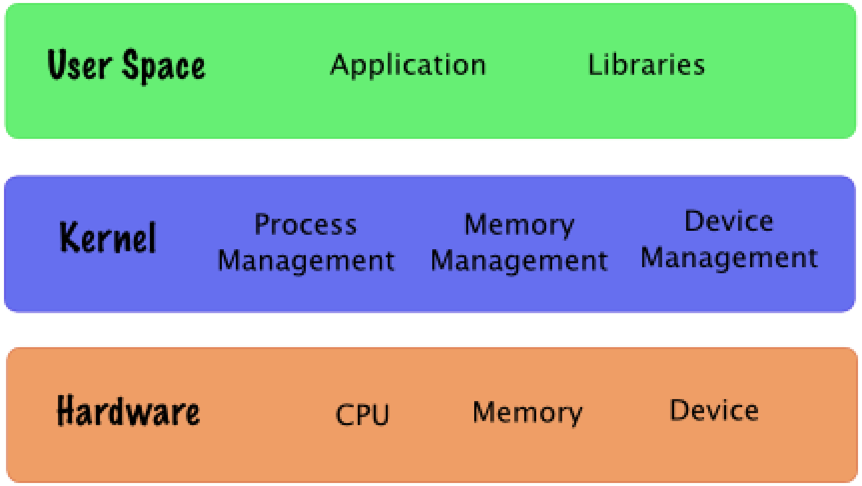
- Provide the rules for the how the machine will operate:
- Control the operation of the I/O devices
- Ensure smooth running of the machine

OS – It's a Protection Layer

- Make the machine more robust—less scope for a bug to have devastating consequences
  - OS does everything programs can't be trusted to do
  - OS makes programs play nice with others

OS – It’s a Resource Manager

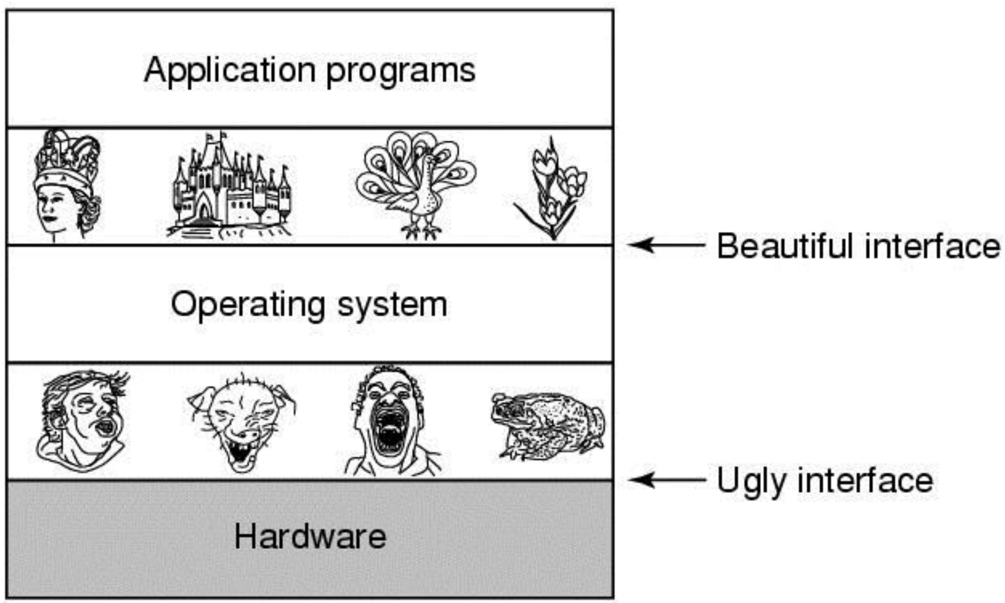
- Bottom-up View
- It manages physical resources:
  - Processor
  - Memory
  - Storage devices
  - Network devices
  - etc. . .



- Example: Multiplexing (Sharing)
  - Time !!
  - Space !!

OS – It is a Abstraction Layer

- Make the machine "nicer", easier to program, higher level. . . (top-down view)
- Hide some of the complications of the machine



Abstraction

The most important word in this course!

Four fundamental Abstractions

- Processes & threads
    - This Course (Tanenbaum, Chapter 2)
- Virtual memory
    - This Course (Tanenbaum, Chapter 3)
- Files & persistent storage
    - This Course (Tanenbaum, Chapter 4)
- Sockets & connections
    - Computer Networks Course

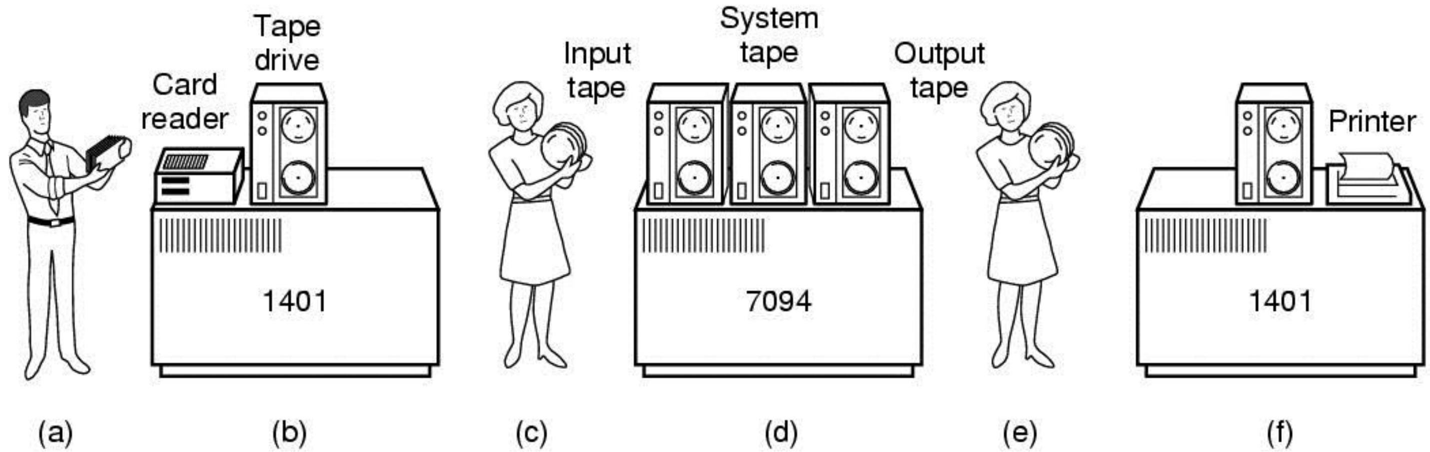
Outline

- What is an Operating System?
- ***History of OS?***
- Computer Hardware Review

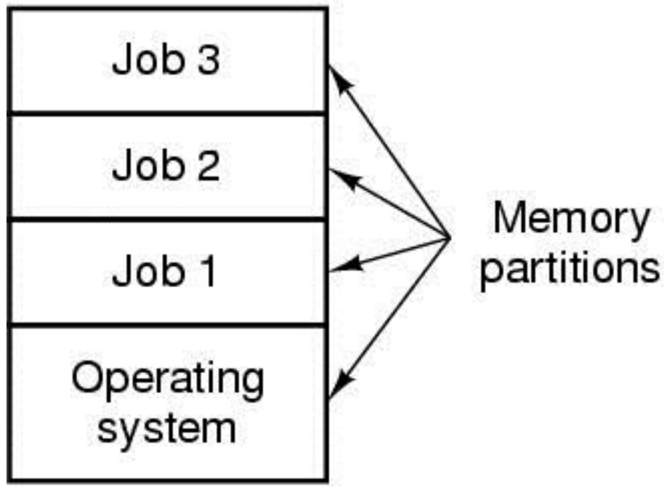
History of Operating Systems

- The first true digital computer was designed by the English mathematician Charles Babbage (1792-1871).
- Generations:
  - (1945–55) Vacuum Tubes (no OS, pure machine language)
  - (1955–65) Transistors and Batch Systems
  - (1965–1980) ICs and Multiprogramming
  - (1980–Present) LSI → Personal Computers, Tablets, Phones

(2nd) Transistors and Batch Systems



(3rd) ICs and Multiprogramming



- Time Sharing – CTSS (Compatible time sharing system)
- MULTICS (MULTiplexed Information and Computing Service)
- Unix (Ken Thompson)
- Linux (Linus Torvalds, Finland Student)

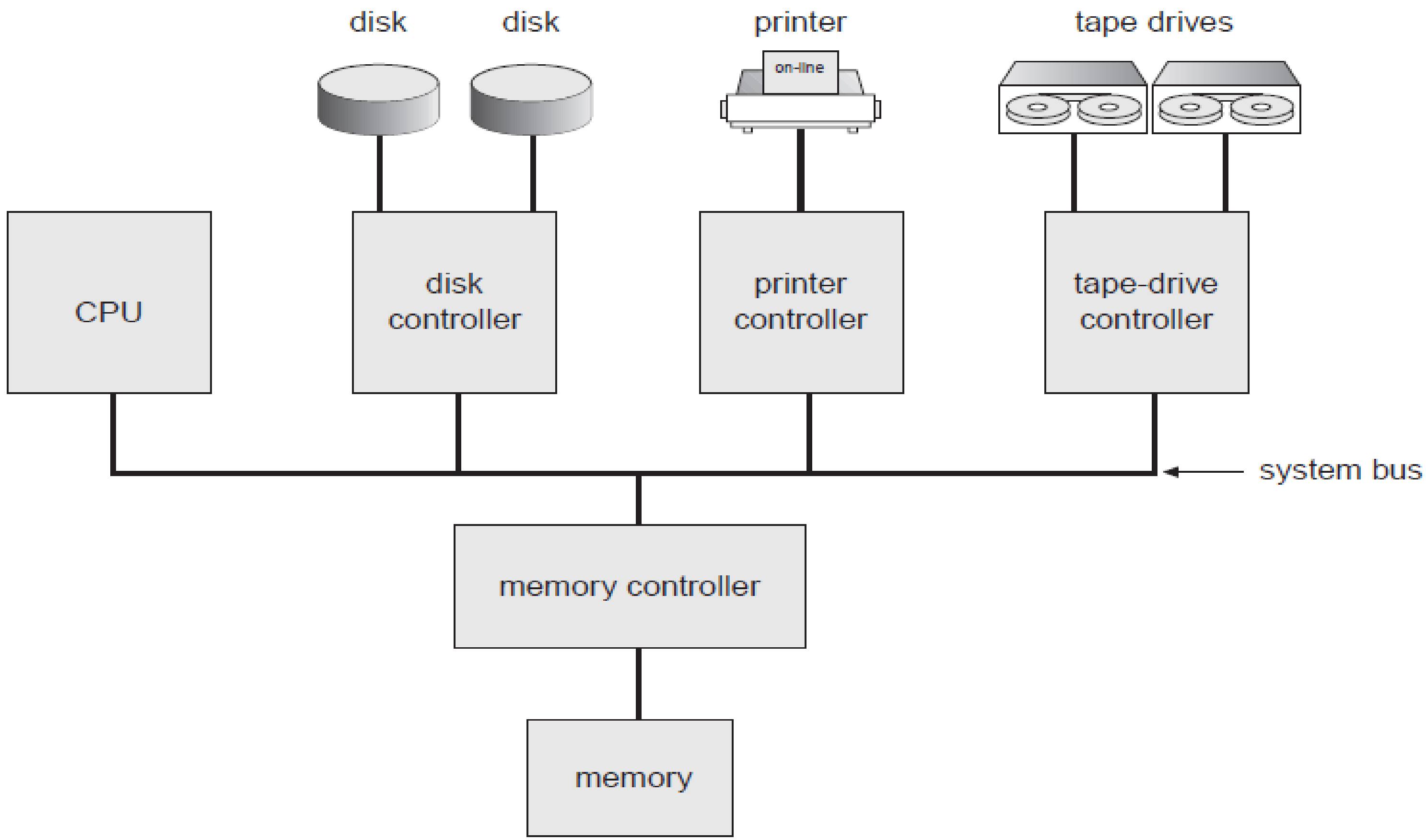
(4th) Large Scale Integrated Circuits

- PCs
- Network Operating Systems
- Distributed Operating Systems

Outline

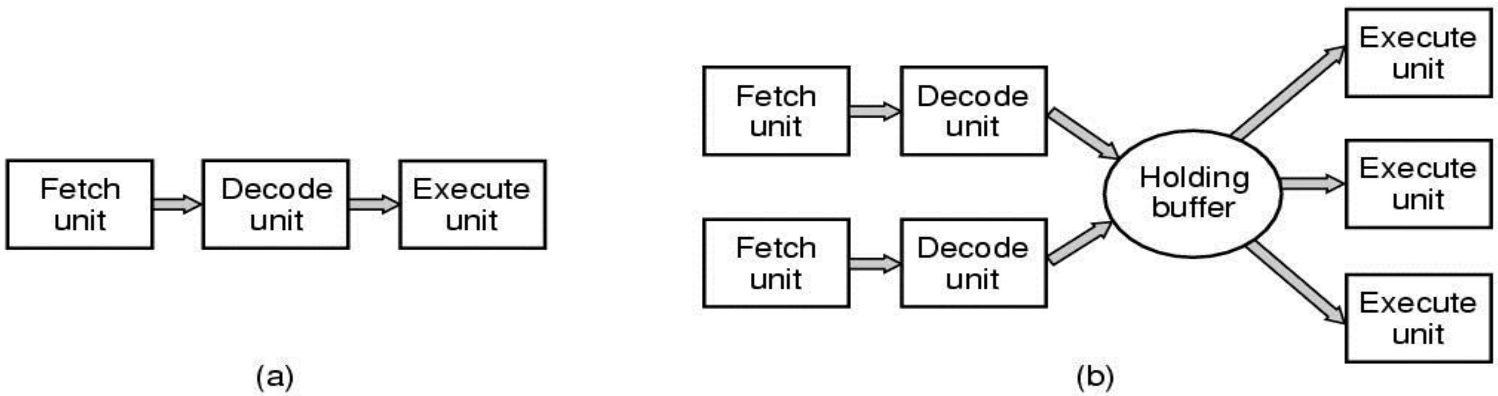
- What is an Operating System?
- History of OS?
- ***Computer Hardware Review***

Computer Hardware



CPU Pipelining

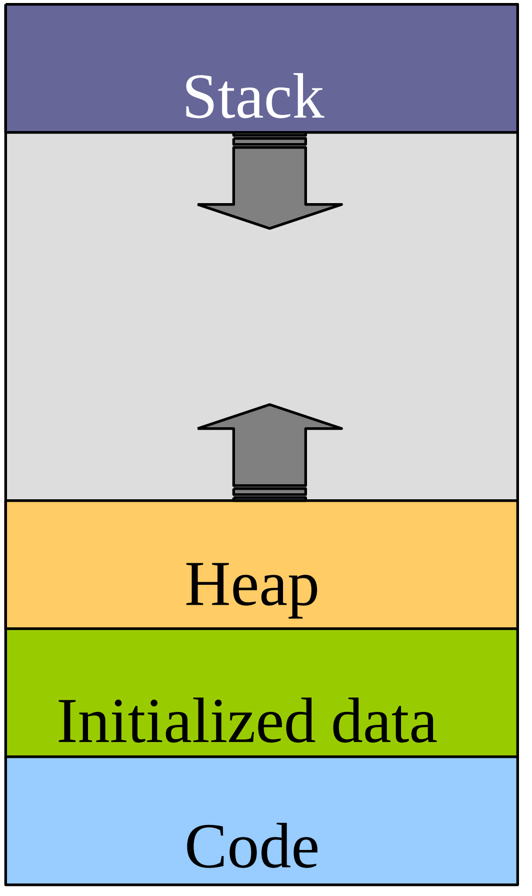
- Need to perform computation!



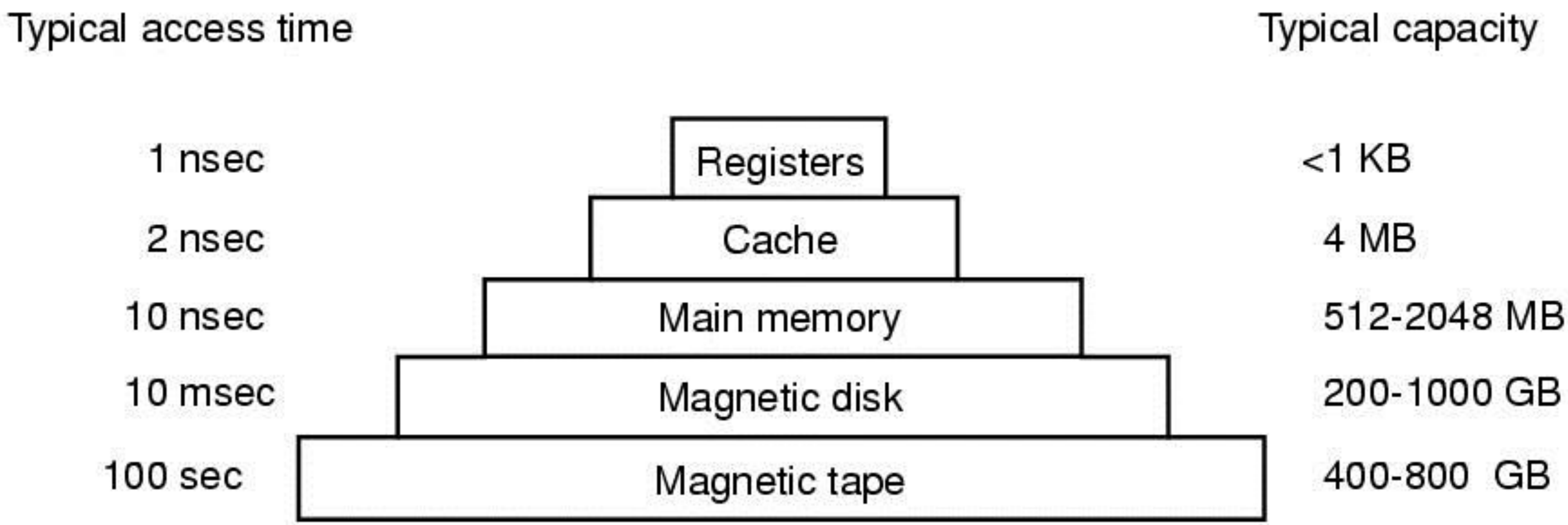
- Memory contains program instructions and program data
- Processor registers maintain processor state. Registers include:
  - General purpose (address & data) registers
  - Instruction pointer (aka program counter)
  - Stack pointer(s)
  - Control and status registers

Computer Program

- Four segments
  - Code/Text – instructions
  - Data – initialized global variables
  - Stack
  - Heap
- Why?
  - Separate code and data
  - Stack and heap go towards each other



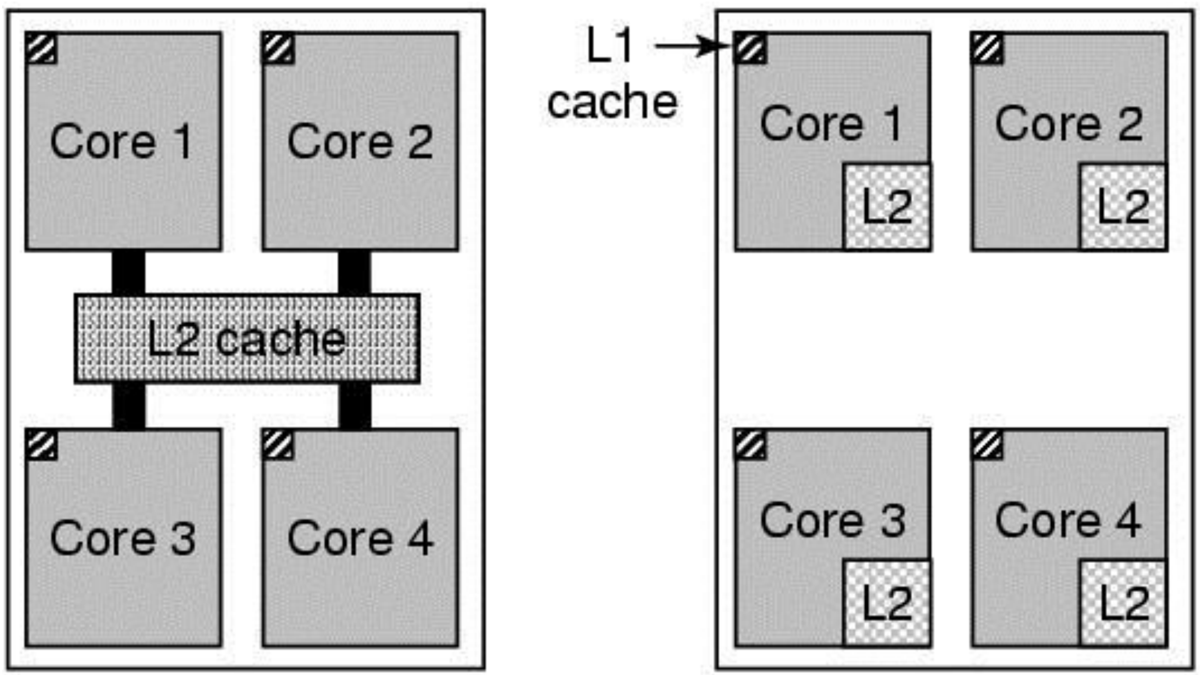
Memory Hierarchy



Cache Memory

- Main memory is divided into cache lines (64 bytes)
  - 0-63 in line 0, 64-127 in line 1,…
- The most heavily used cache lines are kept in a high-speed cache located inside or very close to the CPU.
- When program reads a word-cache hardware checks to see if in cache.
  - If so, then have a cache hit (2 cycles).
  - Otherwise, make request of main memory over the bus (expensive)
- Cache is expensive and is therefore limited in size
- Can have cache hierarchies
  - two or even three levels of cache, each one slower and bigger than the one before it.

Multithreaded and Multicore Chips



Intel

AMD

Memory

- RAM
- ROM
  - Keeps bootstrap OS loader
- EEPROM (Electrically Erasable PROM) and flash memory
  – Can be re-written but slowly
  – e.g: as disk in portable music players
  – If it is erased too many times, it wears out.
- CMOS
  – volatile.
  – hold the current time and date. The CMOS memory and the clock circuit that increments the time in it are powered by a small battery.
  – draws so little power that the original factory installed battery often lasts for several years.

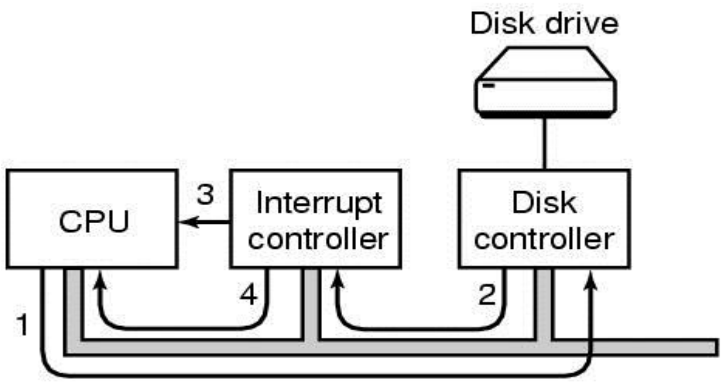
I/O Devices

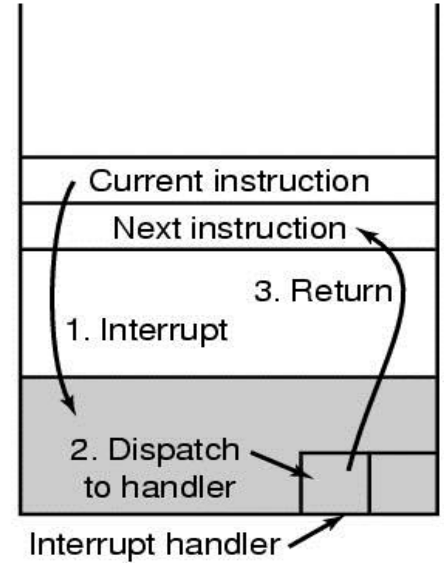
- Controller runs a device-accepts commands from the OS and executes them
- ***Device Driver***: OS software that talks to controller-gives commands, accepts responses
  - Driver runs in kernel mode
- Three modes of communication:
  - Polling
  - Interrupts
  - DMA

I/O by Polling Device

- Driver issues command to controller
- Driver polls device until it is ready
- e.g. Send character to printer controller and poll until it is ready to accept the next character
  - Big use of CPU (Busy waiting)
  - Called programmed I/O - not really used any more

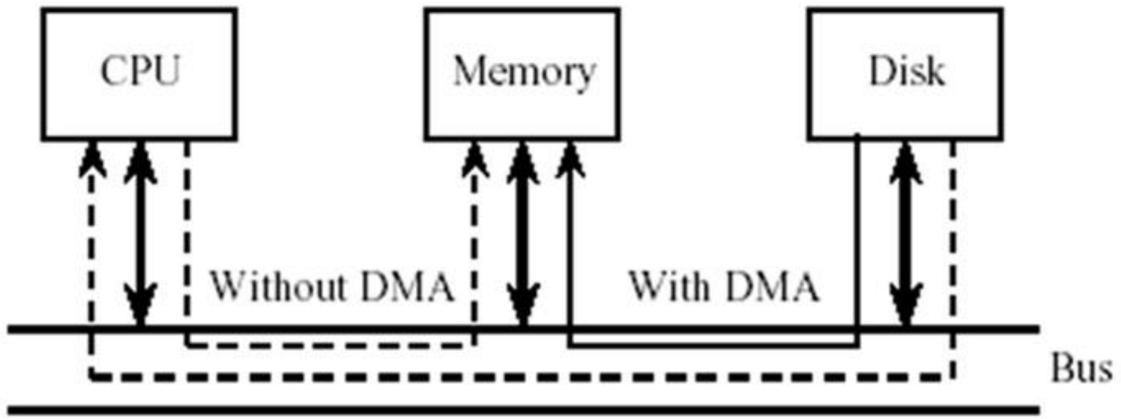
I/O by Interrupts





- Generate an interrupt when I/O is finished.
- e.g. When character is finished being printed, interrupt CPU. Allows CPU to do something else while character is being printed
- This part of memory that holds the interrupt handler is called the ***interrupt vector***.

I/O by DMA



DMA transfer from disk to memory bypasses the CPU.

- Special (controller) chip
- Avoids using the CPU as part of the transfer to/from memory
- CPU tells chip to set up transfer and take care of it
- Chip does as it told and interrupts CPU when it is finished

Review

- What is Multiprogramming?

- On early computers, every byte of data read or written was handled by the CPU (i.e., there was no DMA). What implications does this have for multiprogramming?