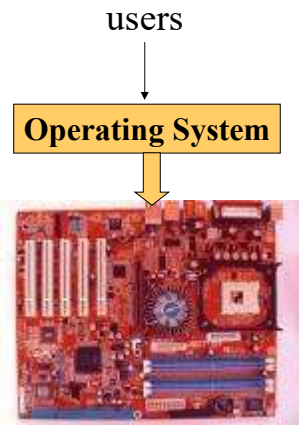
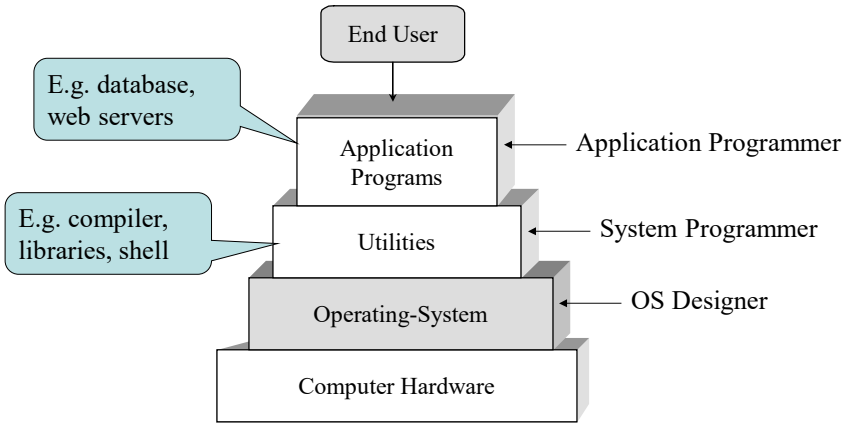
Operating Systems Overview

What is an Operating System?

- OS is a program
- OS provides interface between users and bare hardware
- OS manages resources: CPU(s), memory, disks, other I/O devices (e.g. printer, terminal, network devices)



Role of Operating Systems?



Evolution of OS

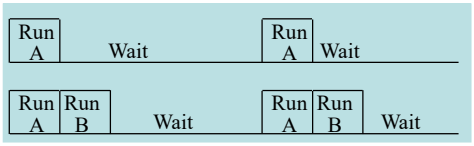
- First Generation (1945-1955)
  - **No operating system**
  - Each user is allocated some time slot, during the time slot, the user has exclusive access to machine
  - User interacts with bare machine using machine language
  - Inconvenient
  - Waste time due to the time slot assignment and set-up time

Evolution of OS

- Second Generation (1955-1965)
  - **Simple batch processing**
  - Requirements
    - Memory management and protection (OS and user code)
    - Timer (in case a program run into an infinite loop)
  - Overhead
    - OS layer
    - Low resource utilization
  - Driving force
    - User convenience

Evolution of OS

- Third Generation
  - (1955-1970)
  - **Multiprogramming**
    - Batch: programs are processed one at a time
    - Better: A different job can use CPU when current one goes to I/O
  - Advantage
    - Allow CPU to continue to work while an I/O request is pending
  - Further requirements
    - Interrupts (to indicate the termination of I/O)
    - Spooling for I/O
  - Driving force
    - Efficient use of system resources (were precious that time)



Evolution of OS

- Fourth Generation (1970 -)
  - Multiprogramming **Timesharing Systems**
  - Due to the demand of interactive processing
    - Goal: make each user feels like owning the system
    - *n* users share the system and each one gets 1/*n* of the system time
    - E.g., use round robin method to share the CPU
  - Further requirements
    - Timer interrupts (for time sharing)
  - Driving force
    - User friendliness

Evolution of OS

- What's current?
  - Personal Computers (1980)
    - User friendliness is the most important issue
    - The system is still multiprogramming systems
    - Most of them support multiple users
      - Via multiple terminal connections or networking
      - But less emphasis on the efficiency for multiple users
    - Who cares about performance: abundant resources???
    - Who cares about reliability: just reboot???

Evolution of OS

- What's current?
  - Parallel systems (1985 -)
    - Too costly (cannot have mass production)
    - Hard to evolve
  - Distributed systems (1980 -)
    - Communication latency
    - Very popular solution
  - Ubiquitous, pervasive, IoT, …
    - Mobile, hand-held computers
    - Special small devices: sensors, wearable devices
    - Embedded OS

What Do Operating Systems Do?

- CPU Management
  - Schedule CPU for many programs
    - System programs (e.g., OS, device drivers)
    - User programs (switch among them)
  - Handle process switch
  - Handle interrupts
- Memory Management
  - Allocate memory for many programs
  - Compute correct addresses for programs
  - Provide protection mechanisms
    - One program does not read from or write to the space of another

What Do Operating Systems Do?

- Device Management
  - Devices: disk, printer, monitor/keyboard, clock, etc.
  - Initiate I/O operation, forward or fetch data
  - Handle I/O interrupts (I/O interrupt handler)
- File System Management
  - Provide organized accesses to disk storage
    - users do not need to directly access physical storage locations
  - keep track of the file system organization (directory system)
  - provide access control

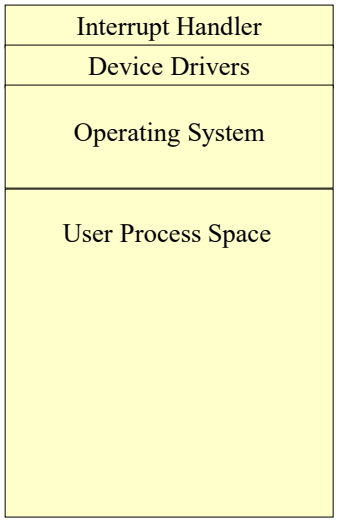
What Do Operating Systems Do?

- Summary of previous two slides
  - Manage many programs that run on the system
  - Provide support during the execution of these programs
    - CPU scheduling support, memory management support
    - When access I/O, device and file system support
- How do the programs run?
  - What are the differences between program files and data files?
    - After all, they are just 0's and 1's
  - When you execute a program (e.g., type "a.out"), what is done?

How Do the Programs Run?

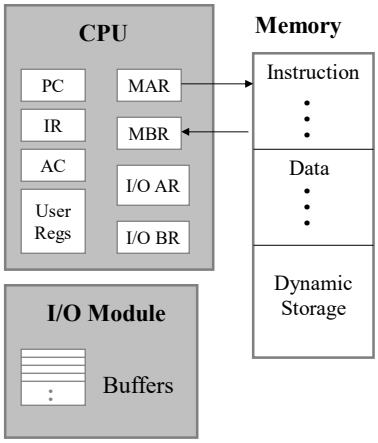
- System creates a process for the program
  - Create an entry in OS for the program
    - Put in ready queue (later)
  - Allocate space in memory for the program
    - Not just the code, but also the control block and data (next page)
    - Control block: OS space (later)
    - Instructions of the program: program space
    - Static data: data space
    - Dynamic data: dynamic address space (heap, not pre-allocated)
  - Put the program to ready queue, wait for execution
  - During execution
    - Fetch instruction, ...

How Do the Programs Run?

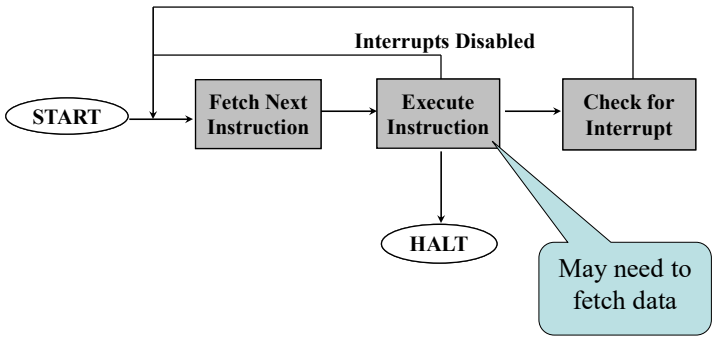


Review – Basic Computer Architecture

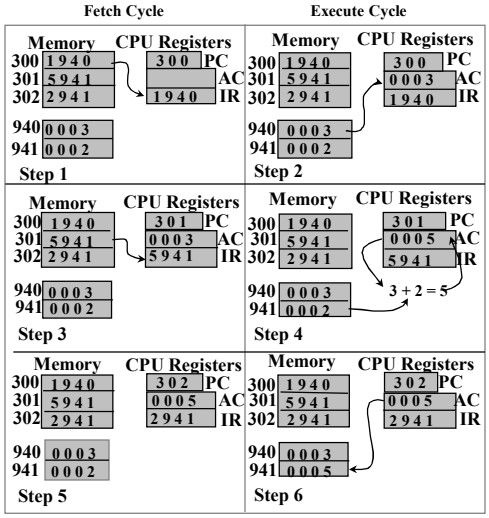
- CPU
  - ALU
  - Registers
    - Memory control registers
    - I/O control registers
    - Instruction related
      - PC, IR, AC



Review – Instruction Cycle



Review – Micro Instructions



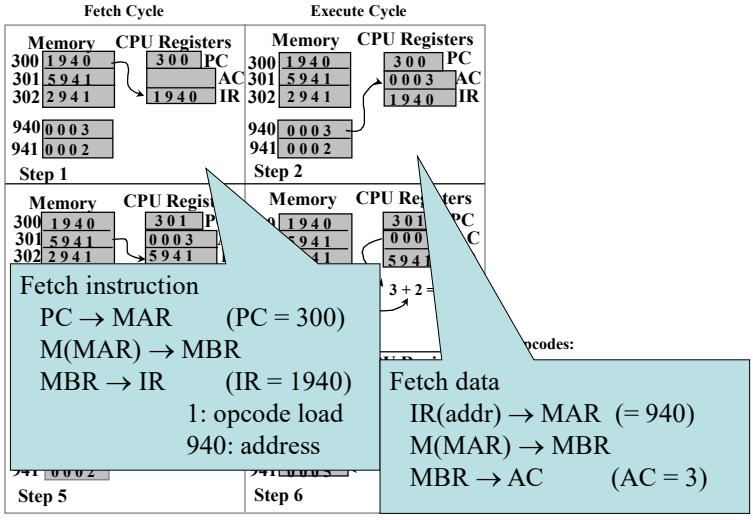
Opcodes:

0001 = Load AC from memory

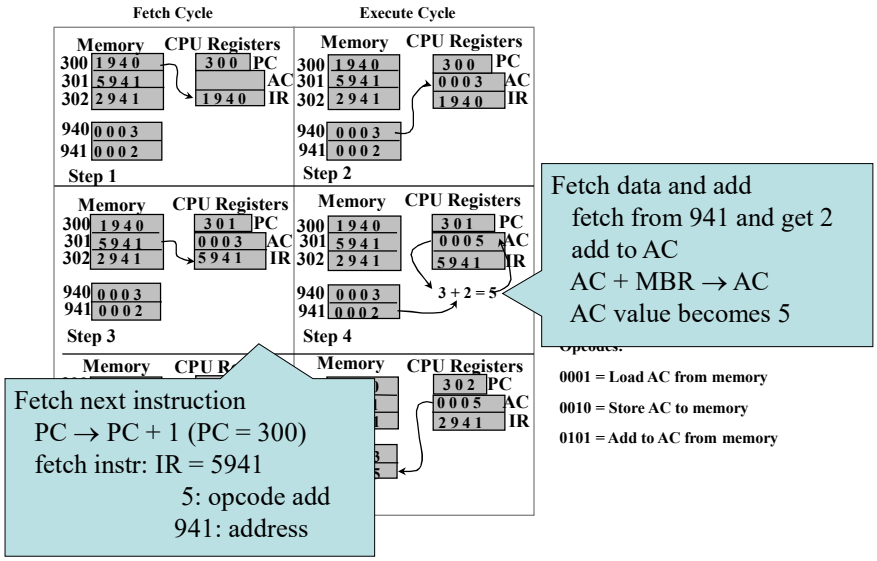
0010 = Store AC to memory

0101 = Add to AC from memory

Review – Micro Instructions



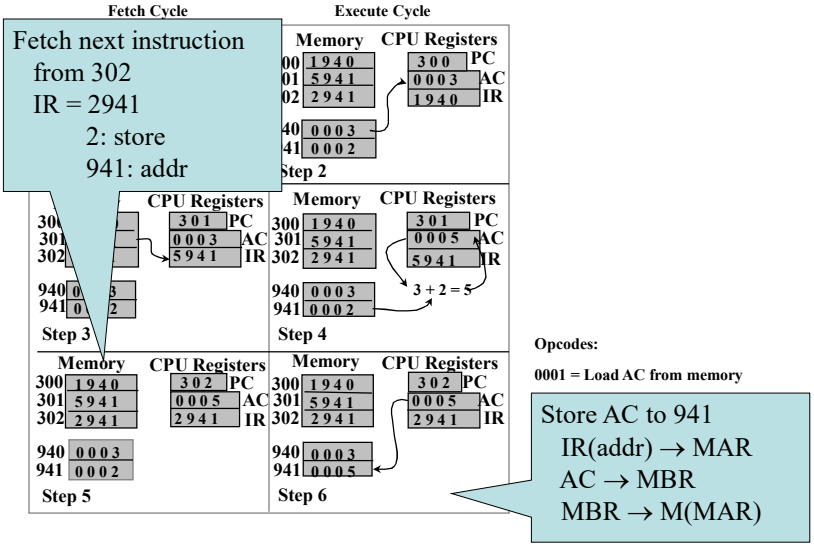
Review – Micro Instructions



Fetch next instruction
PC → PC + 1 (PC = 300)
fetch instr: IR = 5941
5: opcode add
941: address

Fetch data and add
fetch from 941 and get 2
add to AC
AC + MBR → AC
AC value becomes 5

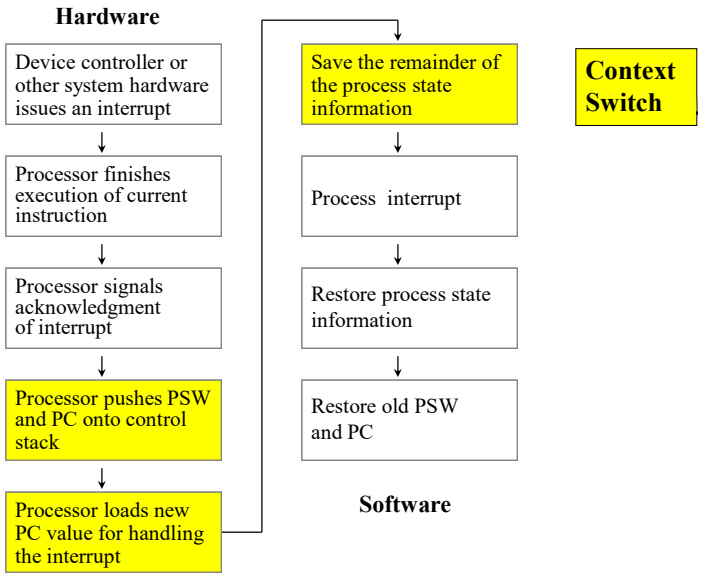
Review – Micro Instructions



Fetch next instruction
from 302
IR = 2941
2: store
941: addr

Store AC to 941
IR(addr) → MAR
AC → MBR
MBR → M(MAR)

Review – Interrupt Handling



What is to be Covered?

- OS manages resources
  - CPU scheduling
  - Memory management
  - Device management
    - Disk, terminal, clock
- OS provides file systems
  - File system organization and how to store files on disk
- Multi-programming concept
  - Processes and their states
  - Concurrent programming

What is to be Covered?

- Likely exam topics
  - Exam 1
    - Overview, processes & threads, processor scheduling, part of concurrent programming
  - Exam 2
    - Concurrent programming, part of memory management
  - Exam 3
    - Memory management, IO, file system, some other potential topics

Readings

- OS evolutions
  - 2.2
- OS functions
  - 2.1
- Reviews
  - 1.1-1.4