Review

- What is Operating System-extended machine, resource manager
- Computer System (Macroscopic view)– CPU, RAM, I/O Devices
- Instruction Cycle – fetch, decoding, execution
- History of Operating System
  - The First Generation – Vacuum Tubes
    - Human Operate Computer
  - The Second Generation – Transistors
    - No multiprogramming
    - Batch system
  - The Third Generation – IC (Integrated Circuits)
    - Multiprogramming
    - Time sharing
    - spooling
  - The Fourth Generation – LSI, VLSI ULSI
    - Personal computer
  - The Fifth Generation – VLSI ULSI
    - mobile computer (Smartphone)

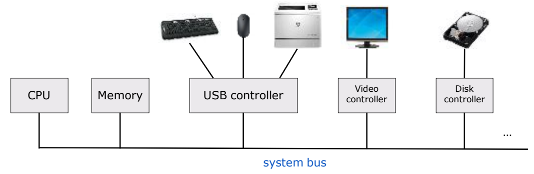
Preview

- Computer system architecture
  - CPU
  - Interrupt and Implementation
  - Memory Hierarchy
  - Input/Out devices
  - Buses – parallel and serial bues
  - Single and Multiprocessor System Types
- Multiprogramming
- Operating System Operation
  - Dual-Mode and Multimode Operations

Computer System Architecture

- A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access between I/O devices and shared memory. (CPU + Memory + I/O devices)
- Each device controller is in charge of a specific type of device. Depending on the controller, more than one device may be attached.
  - Ex) one system USB port can connect to a USB hub, where several devices can be connected
- A device controller maintains some local buffer storage and a set of special-purpose registers. The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage.
- Each I/O device has a device driver for each device controller. Device driver becomes part of operating system.

Computer System Architecture



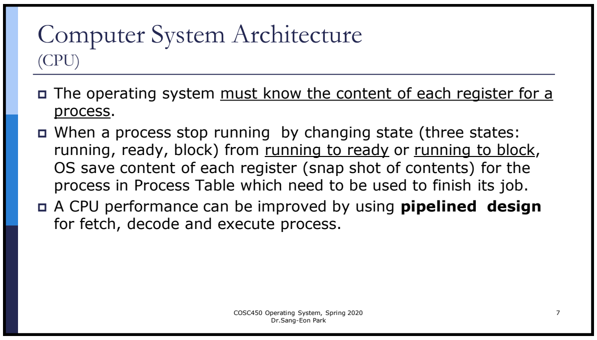
A typical personal computer system

Computer System Architecture
(CPU)

- The CPU is the brain of the computer.
- Each types of CPU has a specific instruction sets which can be used for executing each instruction.
- Since accessing RAM to get an instruction or data takes much longer than executing an instruction (Fetch cycle), all CPU contains set of registers to hold instructions, key variables and temporary result.
  - PC (program counter), IR(instruction register), Data register, Stack pointer,…
- It fetches instructions from memory and execute them.
- Basic instruction cycle executed by CPU
  1. Fetch a data (instruction or data) from memory to a register
  2. Decode the instruction
  3. Execute the instruction
- The instruction cycle is repeated until program finishes.

Computer System Architecture
(CPU)

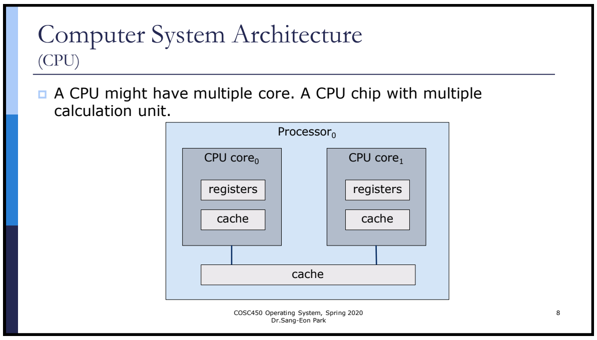
- A CPU composed with several components
  - ALU (Arithmetic Logic Unit)
  - Control Unit
  - Cache (depend on design, it might be outside)
  - Registers
    - General Registers –
      - Instruction registers
      - Data registers
    - Program Counter (PC)– address of next instruction
    - Stack Pointer – address of top of stack
    - Program Status Word (PSW) – save control information for each process (condition code bits, cpu priority, the mode..)



Computer System Architecture
(CPU)

- The operating system must know the content of each register for a process.
- When a process stop running by changing state (three states: running, ready, block) from running to ready or running to block, OS save content of each register (snap shot of contents) for the process in Process Table which need to be used to finish its job.
- A CPU performance can be improved by using **pipelined design** for fetch, decode and execute process.

COSC450 Operating System, Spring 2020
Dr.Sang-Eon Park
7



Computer System Architecture
(CPU)

- A CPU might have multiple core. A CPU chip with multiple calculation unit.

Processor$_0$

CPU core$_0$

registers

cache

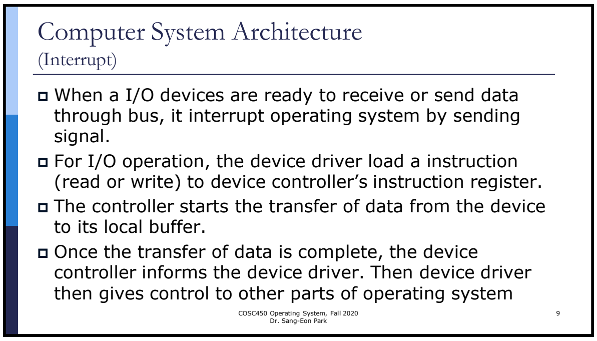
CPU core$_1$

registers

cache

cache

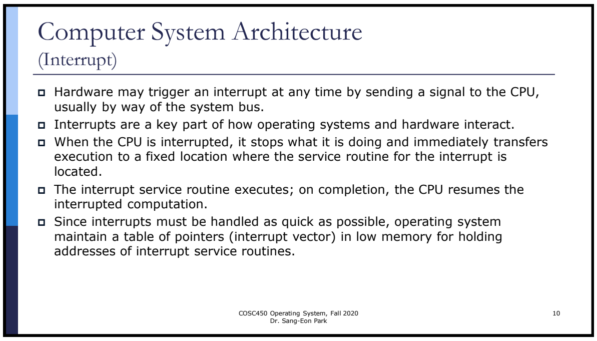
COSC450 Operating System, Spring 2020
Dr.Sang-Eon Park
8



Computer System Architecture
(Interrupt)

- When a I/O devices are ready to receive or send data through bus, it interrupt operating system by sending signal.
- For I/O operation, the device driver load a instruction (read or write) to device controller’s instruction register.
- The controller starts the transfer of data from the device to its local buffer.
- Once the transfer of data is complete, the device controller informs the device driver. Then device driver then gives control to other parts of operating system

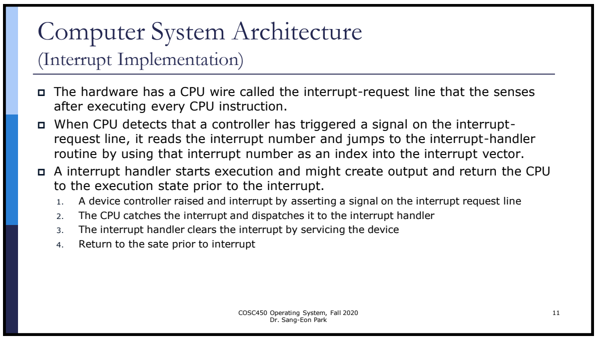
COSC450 Operating System, Fall 2020
Dr. Sang-Eon Park
9



Computer System Architecture
(Interrupt)

- Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus.
- Interrupts are a key part of how operating systems and hardware interact.
- When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location where the service routine for the interrupt is located.
- The interrupt service routine executes; on completion, the CPU resumes the interrupted computation.
- Since interrupts must be handled as quick as possible, operating system maintain a table of pointers (interrupt vector) in low memory for holding addresses of interrupt service routines.

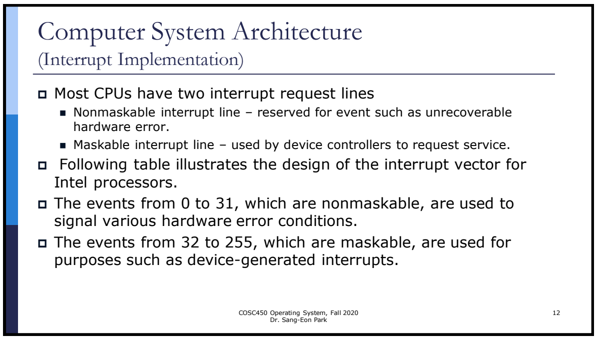
COSC450 Operating System, Fall 2020
Dr. Sang-Eon Park
10



Computer System Architecture
(Interrupt Implementation)

- The hardware has a CPU wire called the interrupt-request line that the senses after executing every CPU instruction.
- When CPU detects that a controller has triggered a signal on the interrupt-request line, it reads the interrupt number and jumps to the interrupt-handler routine by using that interrupt number as an index into the interrupt vector.
- A interrupt handler starts execution and might create output and return the CPU to the execution state prior to the interrupt.
  1. A device controller raised and interrupt by asserting a signal on the interrupt request line
  2. The CPU catches the interrupt and dispatches it to the interrupt handler
  3. The interrupt handler clears the interrupt by servicing the device
  4. Return to the sate prior to interrupt

COSC450 Operating System, Fall 2020
Dr. Sang-Eon Park
11

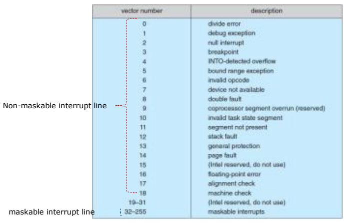


Computer System Architecture
(Interrupt Implementation)

- Most CPUs have two interrupt request lines
  - Nonmaskable interrupt line – reserved for event such as unrecoverable hardware error.
  - Maskable interrupt line – used by device controllers to request service.
- Following table illustrates the design of the interrupt vector for Intel processors.
- The events from 0 to 31, which are nonmaskable, are used to signal various hardware error conditions.
- The events from 32 to 255, which are maskable, are used for purposes such as device-generated interrupts.

COSC450 Operating System, Fall 2020
Dr. Sang-Eon Park
12

Computer System Architecture
(Interrupt Implementation)



vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

Non-maskable interrupt line

maskable interrupt line

Intel Processor event-vector table

Computer System Architecture
(Interrupt)

- In summary, interrupts are used throughout modern operating systems to handle asynchronous events.
- Device controllers and hardware faults raise interrupts.
- To enable the most urgent work to be done first, modern computers use a system of interrupt priorities.
- Because interrupts are used so heavily for time-sensitive processing, efficient interrupt handling is required for good system performance.

Computer System Architecture
(Memory)

- The CPU can load instructions only from RAM, so any programs must first be loaded into memory from the secondary memory to run.
- Storage hierarchy
  - Registers in CPU (volatile)
  - Cache memory – volatile
  - Main Memory (volatile) – RAM(Random Access Memory)
    - DRAM (Dynamic RAM) – build with capacitors.
    - SRAM (Static RAM) – Build with logic gates.
  - Secondary Memory (non-volatile)
    - HDD(Hard Disk Drive) - rigid metal or glass platters covered with magnetic recording material
    - SSD(Solid State Drive) - floating-gate transistors.
    - Magnetic Tape

Computer System Architecture
(Input /Output Device)

- A large portion of operating system code is dedicated to managing I/O, because of its importance to the reliability and performance of a system and the varying nature of the devices.
- The form of interrupt-driven I/O is fine for moving small amounts of data but can produce high overhead when used for bulk data movement.
- The solution is using DMA (Direct Memory Access). DMA controller can handle I/O independent from the CPU.
- DMA is a capability provided by some computer bus architectures that allows data to be sent directly from an attached device (such as a disk drive) to the memory.
- The CPU is freed from involvement with the data transfer, thus speeding up overall computer operation.

Computer System Architecture
(Input /Output Device)

- Operating System controls all I/O devices by
  - Issue commands (read/write) to Devices
  - Catch interrupts from Devices- when devices are ready to send or receive data.
  - Handle error
- OS also provide interface between the devices and the rest of the system

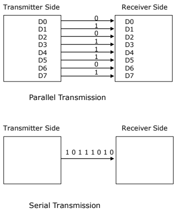
Computer System Architecture
(Input /Output Device)

- I/O units consist of a mechanical component, an electronic component and device driver.
  - **Mechanical Components** - Device itself.
  - **Electrical components**
    - Device controller (adapter).
    - On personal computer, it takes the **form of a chip on the** parent board or a printed circuit card that can be inserted into a PCI expansion slot.
  - **Device Driver (software)**– the standard interface between device Controller and Operating System

Computer System Architecture
(Buses)

- The bus in a PC is the <u>common pathway between the CPU and peripheral devices</u>.
  - **Parallel buses** <u>use slots on the motherboard</u> and provide multiple lines for data (32 bits, 64 bits) between the CPU and peripheral card. Cards plug into the bus inside the cabinet.
  - **Serial buses** <u>have external ports, and the cable</u> that plugs into them can connect to multiple devices.

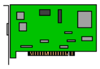
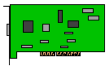
Computer System Architecture
(Buses)



Computer System Architecture
(Parallel Buses)

- Advantage and disadvantage of Parallel Buses
  - Advantage – It offers fast data communication between device.
  - Disadvantage – It supports short distance communication due to crosstalk between the parallel line. It cost more due to the multiple line and more pins to connect.
- Parallel Buses:
  - **Peripheral Component Interconnect** (PCI) – is available in 32- and 64-bit versions, is the <u>most popular bus architecture.</u>
  - **Accelerated Graphics Port** (AGP) - <u>designed for faster screen display</u>. If used, there is just one AGP slot on the motherboard

Computer System Architecture
(Parallel Buses)

PARALLEL PC BUSES	Bandwidth	
	Bits	Speed
PCI 	32 64	33MHz
AGP 	32	66-533 MHz

Computer System Architecture
(Serial Buses)

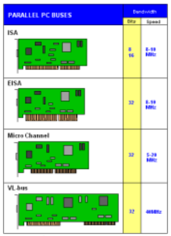
- Advantage and disadvantage of Serial Buses
  - Disadvantage – It offers slower data communication between device.
  - Advantage – It supports long distance communication. It cost less than parallel buses.
- Serial Bues
  - **Universal Serial Bus (USB) –** One USB port connects up to 127 peripherals. The first version of USB was designed for low-speed peripherals.
  - **FireWire (IEEE 1394)-** FireWire connects up to 63 peripheral devices and has been mostly used for digital camera connections

Computer System Architecture
(Parallel Buses: Earlier version)

Parallel Buses (Earlier)

- **Industry Standard Architecture** (ISA) **-** from the original PC. It was an 8-bit bus originally known as the PC bus and then the XT bus. It was later extended to 16 bits and became the AT bus.
- **Extended ISA** (EISA)- was a 32-bit extension of ISA created by major vendors to counter IBM's Micro Channel. EISA slots accepted both EISA and ISA cards, but clock speed was still at the slow ISA rate.
  Micro Channel (MCA) - IBM switched to the 32-bit Micro Channel with its PS/2 line in 1987, then later added back ISA. Eventually, it gave up Micro Channel for PCI.
  VESA Local Bus (VL) -The 32-bit VL-bus was introduced during the 486 era and offered higher speed than ISA. It then gave way to PCI.

Computer System Architecture
(Parallel Buses: Earlier version)



PARALLEL PC BUSES	Bandwidth	
	Bits	Speed
ISA	8 16	8-10 MHz
EISA	32	8-10 MHz
Micro Channel	32	5-20 MHz
VL-bus	32	40MHz

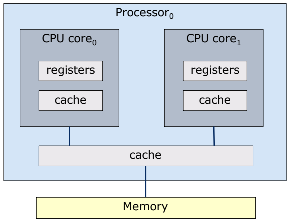
Computer System Architecture
(Single Processor Systems)

- A computer systems used a single processor containing one CPU with a single processing core.
- The core is the component that executes instructions and registers for storing data locally. The one main CPU with its core is capable of executing a general-purpose instruction set, including instructions from processes.
- These systems might have other special-purpose processors as well.
  - Form od device-specific processors such as disk, keyboard, graphic controllers.
- These special-purpose processors run a limited instruction set and do not run processes.
- OS manages these special-purpose processor by sending information about their next task and monitor their status.
  - Ex) DMA
  - A processor in the keyboard to convert the key strokes into code to be sent to CPU

Computer System Architecture
(Multiprocessor Systems)

- The primary advantage of multiprocessor systems is increased throughput. We expect to get more work done in less time by increasing the number of processors.
- The speed-up ratio with N processors is less than N, because of possible extra overhead for communicating between CPU’s for any calculation.
- The definition of multiprocessor has evolved over time and now includes multicore systems, in which multiple computing cores reside on a single chip.
- Multicore systems can be more efficient than multiple chips with single cores because on-chip communication is faster than between-chip communication.
- One chip with multiple cores uses significantly less power than multiple single-core chips, an important issue for mobile devices as well as laptops.

Computer System Architecture
(Multiprocessor Systems)



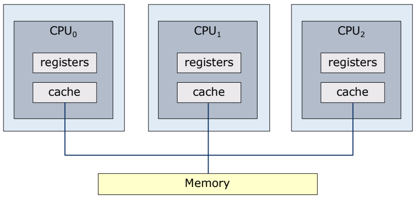
A dual-core design with two cores on the same chip

Computer System Architecture
(Multiprocessor Systems)

- Multiprocessor System Types

  Symmetric multiprocessing
  - Each CPU has its own set of registers and possibly cash memory. Each CPU performs all tasks including OS functions and user processes. All processors share a common physical memory over the system bus.
  - Virtually all modern operating systems—including Windows, macOS, and Linux, as well as Android and iOS mobile systems—support multicore SMP systems.
  - Adding additional CPUs to a multiprocessor system will increase computing power; however once we add too many CPUs, contention for the system bus becomes a bottleneck and performance begins to degrade.

Computer System Architecture
(Multiprocessor Systems)



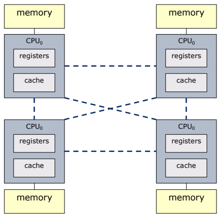
Symmetric Multiprocessing Architecture

Computer System Architecture
(Multiprocessor Systems)

- Multiprocessor System Types

  Non-Uniform memory access (NUMA)
  - Each CPU has its own local memory that is accessed via a small, fast local bus.
  - CPU are connected by a shared system interconnect, so that all CPUs share one physical address space.
  - The advantage: when a CPU accesses its local memory, not only is it fast, but there is also no contention over the system interconnect.
  - A potential drawback: increased latency when a CPU must access remote memory across the system interconnect, creating a possible performance penalty- OS need careful CPU scheduling and memory management to reduce overhead.

Computer System Architecture
(Multiprocessor Systems)



Non-Uniform Memory Access (NUMA)

Computer System Architecture
(Multiprocessor Systems)

- Multiprocessor System Types

  Clustered Systems
  - Two or more individual systems are connected locally and run as one system –loosely coupled.
  - Clustering is usually used to provide high-availability service—that is, service that will continue even if one or more systems in the cluster fail.
  - High availability can be obtained by adding a level of redundancy in the system. A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the others (over the network). If the monitored machine fails, the monitoring machine can take ownership of its storage and restart the applications that were running on the failed machine.
    - Asymmetric clustering- one machine is in hot-standby mode. Hot-standby host machine does just monitoring the active server. If that server fails, the hot-standby host becomes the active server.
    - Symmetric clustering – two or more hosts are running application and monitoring each other. This model is more efficient but it does require that more than one application be available to run.

Computer System Architecture
(Multiprocessor Systems)

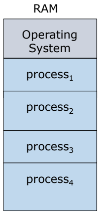
- Multiprocessor System Types

  Clustered Systems –continue.
  - Asymmetric clustering- one machine is in hot-standby mode. Hot-standby host machine does just monitoring the active server. If that server fails, the hot-standby host becomes the active server.
  - Symmetric clustering – two or more hosts are running application and monitoring each other. This model is more efficient but it does require that more than one application be available to run.
  - Since a cluster consists of several computer systems connected via a network, clusters can also be used to provide high-performance computing environments.
  - To take advantage of the cluster structure, the application must have been written specifically. A program is divided into separate components (using thread) that run in parallel on individual cores in a computer or computers in a cluster.

Operating System Operation
(Multiprogramming)

- Multiprogramming:
  - The operating system keeps several processes in memory simultaneously. The operating system (short term scheduler) picks and begins to execute one of these processes. Eventually, the process may have to wait for some task, such as an I/O operation, to complete.
  - Multiprogramming increases CPU utilization, as well as keeping users satisfied, by organizing programs so that the CPU always has one to execute.

Operating System Operation
(Multiprogramming)



Multiple jobs are loaded and OS provide processes concurrent running

Memory layout for a multiprogramming system

Operating System Operation
(Multitasking)

- Multitasking is a logical extension of multiprogramming.
- Multitasking is allowing a user to perform more than one computer task at a time.
- The operating system is able to keep track of where you are in these tasks and go from one to the other without losing information

Operating System Operation
(Dual-Mode and Multimode Operation)

- Since the OS and its users share resources, an OS must ensure that malicious program cannot cause other programs to execute incorrectly.
- In order to ensure the proper execution of the system, must distinguish between the execution of OS code and user code (kernel mode and user mode).
- The approach taken by most computer systems is to provide hardware support that allows differentiation among various modes of execution.
- A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1).
- When a process request a service via system call, the system must change mode from user to kernel mode to fulfil the request.

Operating System Operation
(Dual-Mode and Multimode Operation)

- The dual mode of operation provides us with the means for protecting the operating system from errant users (or hacker).
- This protection can be accomplished by designating some of the machine instructions (may cause harm as privileged instructions) to be executed only in kernel mode.
- The concept of modes can be extended to more than two modes.

Ex) Intel processors have four separate protection rings
  - Ring 0 : kernel mode
  - Ring 1 and 2: used for various OS service rarely used
  - Ring 3: user mode

Operating System Operation
(Dual-Mode and Multimode Operation)

- System calls provide the means for a user program to ask the operating system to perform tasks reserved for the operating system on the user program's behalf.
- A system call usually takes the form of a trap to a specific location in the interrupt vector. This trap can be executed by a generic trap instruction, although some systems have a specific syscall instruction to invoke a system call.
- When a system call is called, control passes through the interrupt vector to a service routine in the operating system, and the mode bit is set to kernel mode.
- The system-call service routine is a part of the operating system. The kernel examines the interrupting instruction to determine what system call has occurred; a parameter indicates what type of service the user program is requesting.
- The kernel verifies that the parameters are correct and legal, executes the request, and returns control to the instruction following the system call.

Operating System Operation
(Dual-Mode and Multimode Operation: System Call)

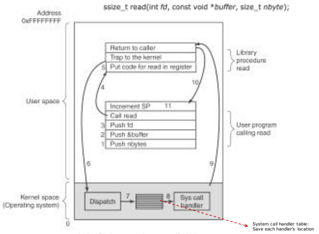
```
#include <unistd.h>
ssize_t read(int fd, const void *buffer, size_t nbyte);
```

Return number of byte read for ok, -1 for error

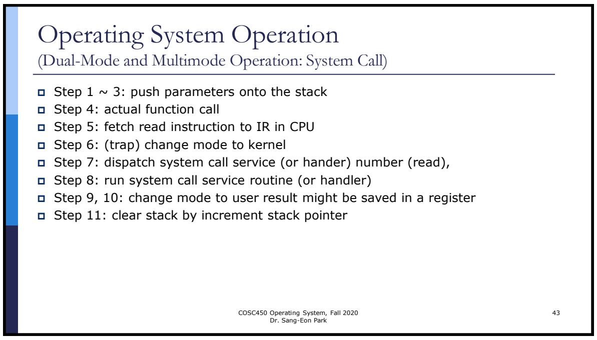
- The read() system call attempts to read *nbyte* bytes to the buffer pointed to by *buffer* from the file associated with the open file descriptor, *fd*.
- It returns the number of bytes actually read.

Operating System Operation
(Dual-Mode and Multimode Operation: System Call)

ssize_t read(int *fd*, const void **buffer*, size_t *nbyte*);



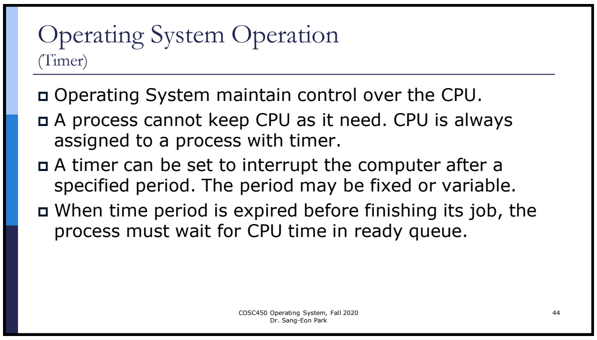
System call handler table:
Save each handler's location



Operating System Operation

(Dual-Mode and Multimode Operation: System Call)

- Step 1 ~ 3: push parameters onto the stack
- Step 4: actual function call
- Step 5: fetch read instruction to IR in CPU
- Step 6: (trap) change mode to kernel
- Step 7: dispatch system call service (or hander) number (read),
- Step 8: run system call service routine (or handler)
- Step 9, 10: change mode to user result might be saved in a register
- Step 11: clear stack by increment stack pointer



Operating System Operation

(Timer)

- Operating System maintain control over the CPU.
- A process cannot keep CPU as it need. CPU is always assigned to a process with timer.
- A timer can be set to interrupt the computer after a specified period. The period may be fixed or variable.
- When time period is expired before finishing its job, the process must wait for CPU time in ready queue.