CSE 30341

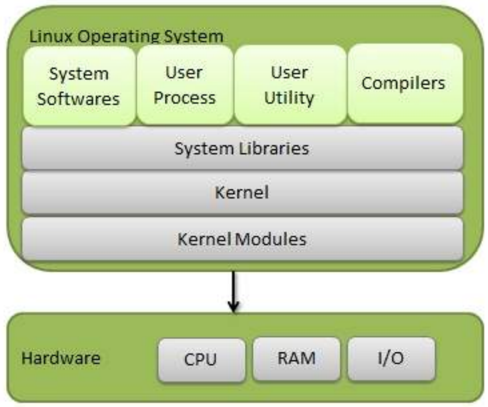
Operating System Principles

Lecture 2 – Introduction – Continued

Recap – Last Lecture

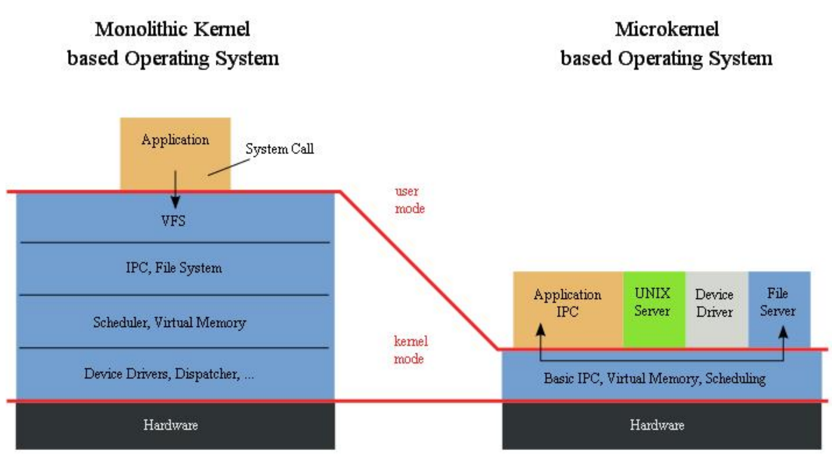
- What is an operating system & kernel?
- What is an interrupt?

OS - Kernel



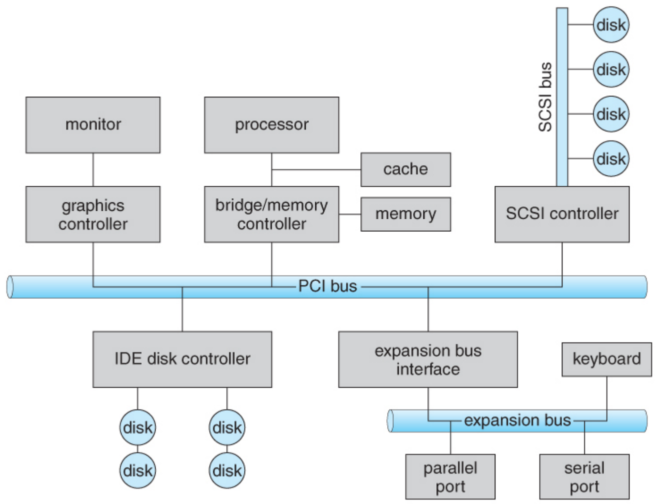
CSE 30341 – Operating System Principles 3

OS - Kernel

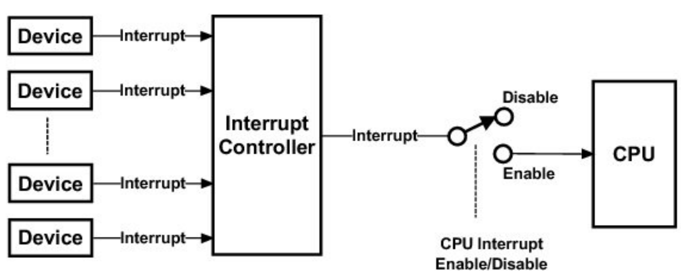


CSE 30341 – Operating System Principles 4

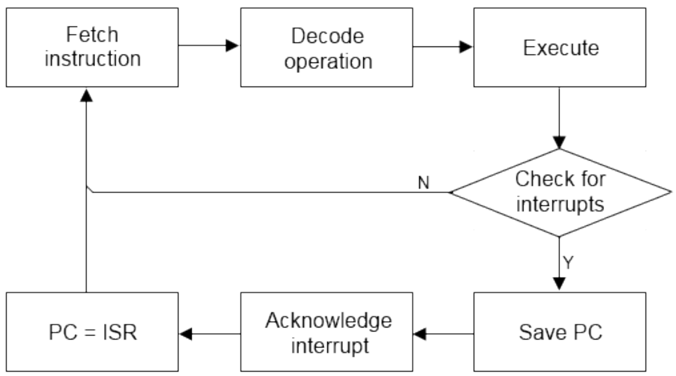
System Architecture



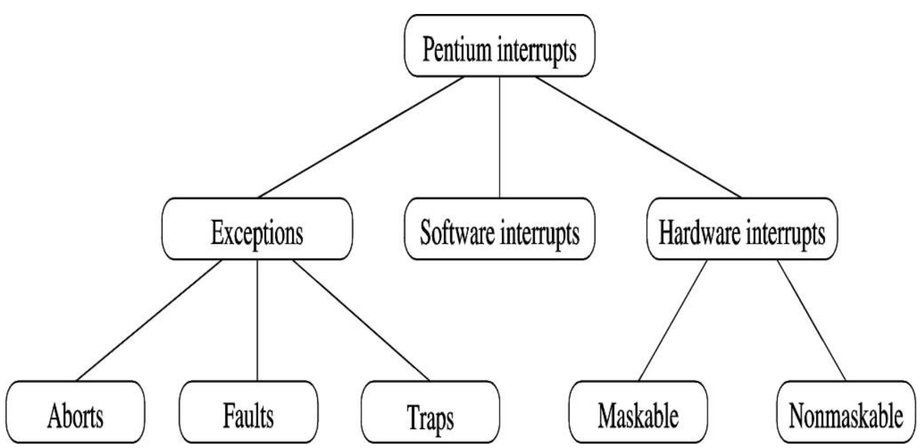
Interrupt



Interrupt



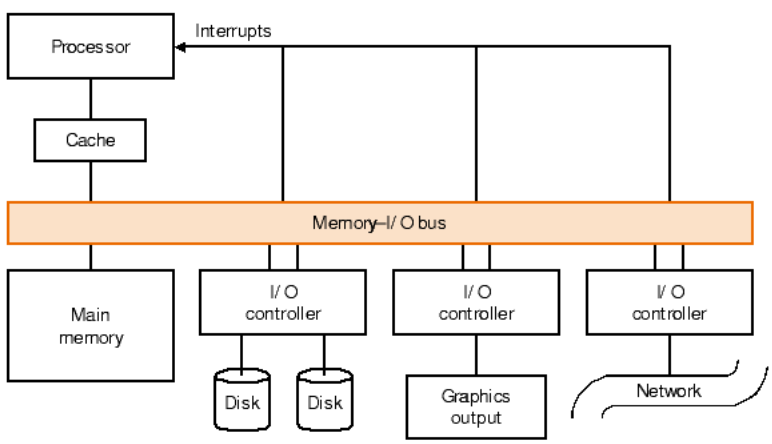
Interrupt



Input/Output – I/O

- Communication between CPU and “outside world”:
  - Storage
  - Network
  - Keyboard/mouse
  - Display
  - Printer
  - …

Input/Output – I/O



Interacting with I/O

- System-controlled:
  - "Write this chunk of data to block 8,783,486"
  - "Please give me the data from blocks 7,345,286 – 7,345,289"
- External events (system reacts):
  - The user is pressing the shift key
  - Block 3,285,001 appears to be bad
  - Data arrived over a network connection

Interacting with I/O

- Responsibilities of OS:
  - Hide peculiarities of hardware devices from the user
  - Manage hardware devices ("resources") efficiently
  - Prevent intentional/unintentional misuse

Interacting with I/O

- Application requests I/O from OS
  - Uses specific interface: **system calls**
  - **Blocking:** application will wait until I/O complete
  - **Non-blocking:** application will do something else in the meantime (and receive notification from OS when I/O complete)

Direct Memory Access (DMA)

- CPU responsible for data moving to/from I/O devices
- Alternative: let a separate controller do it (DMA)

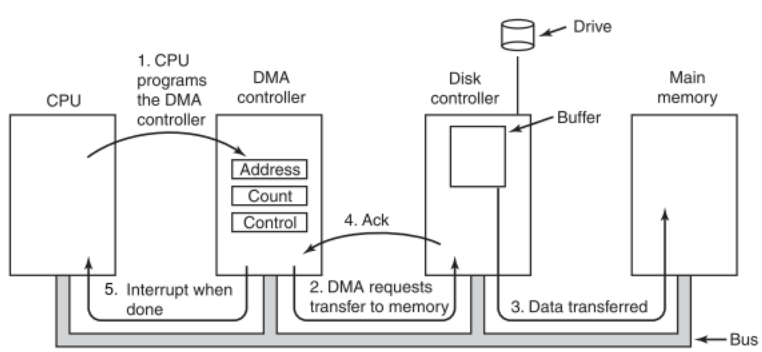
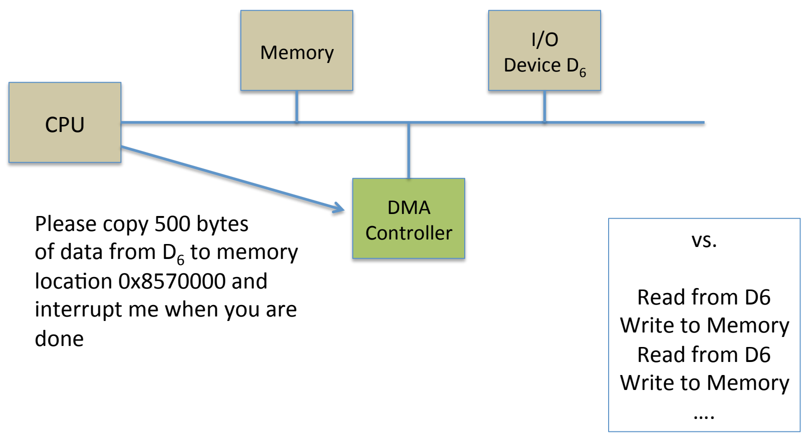


Figure 5-4. Operation of a DMA transfer.

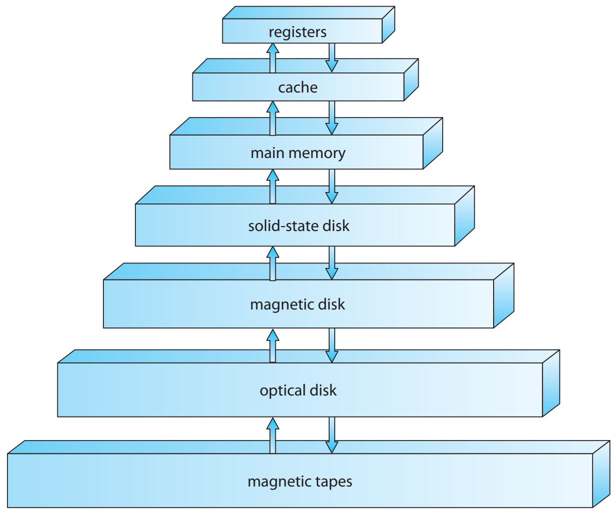
DMA Controller



Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access memory**
  - **Volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
  - Magnetic disks – rigid metal or glass platters covered with magnetic recording material
    - Disk surface - **tracks**, subdivided into **sectors**
    - The **disk controller** determines the logical interaction between the device and the computer
  - Solid-state disks – faster than magnetic disks, nonvolatile
    - Various technologies
    - Becoming more popular

Storage-Device Hierarchy



Storage Hierarchy

- Storage systems organized in hierarchy
  - Size
  - Speed
  - Cost
  - Volatility
- **Caching** – leverage faster storage system; higher layer can be cache for lower layer

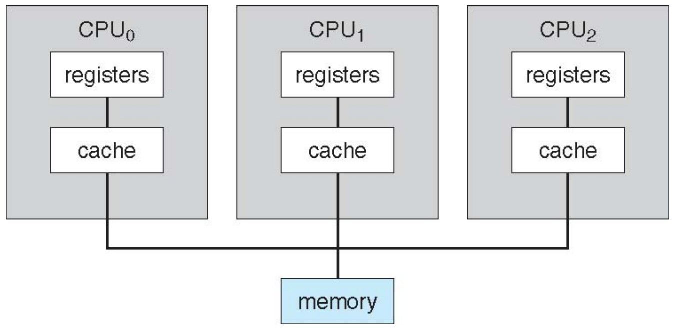
Caching

- One of the most important principles in systems
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

Computer-System Architecture

- **General-purpose** processors (CPU) versus **special-purpose** processors (controllers)
- **Multiprocessor** systems are now typical
  - Parallel systems, tightly-coupled systems
  - Advantages include:
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance

Symmetric Multiprocessing (SMP) Architecture

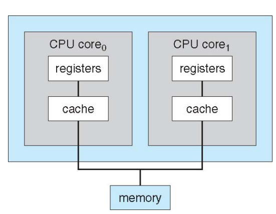


UMA – Uniform Memory Access
- All share the same memory on the same machine, same cost to access.
- May have a private cache

NUMA – Non-uniform Memory Access
- Each processor has its own memory

Multi-Core Design

- Multiple "cores" on same chip
  - On-chip communication is fast
  - Power consumption can be reduced

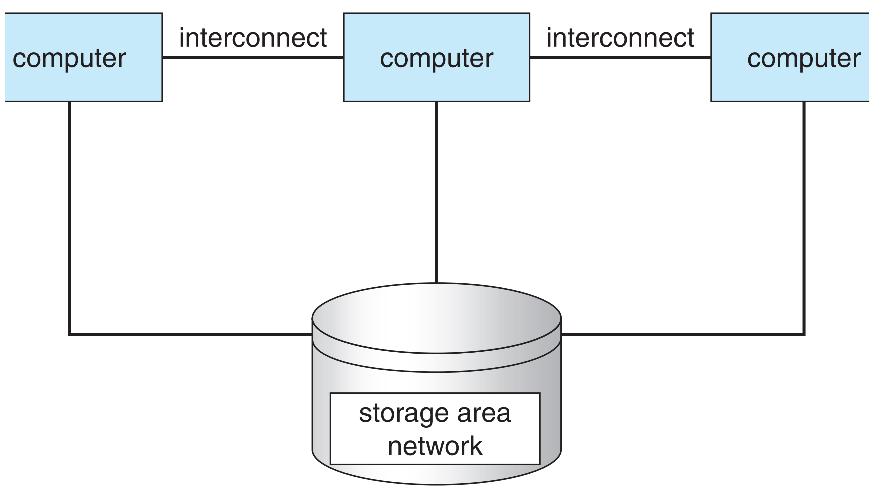


Clustered Systems

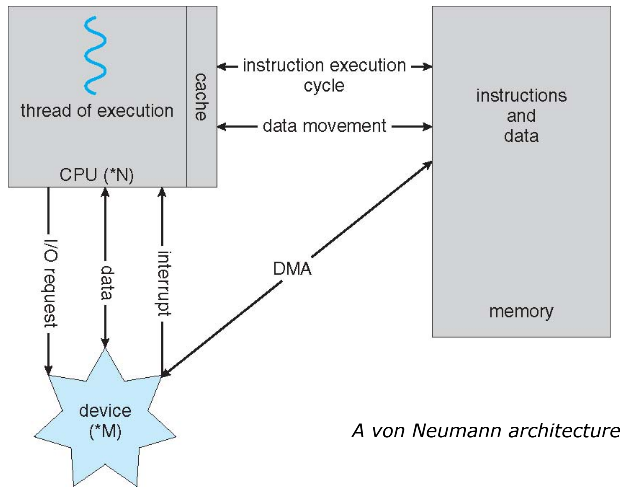
- Like multiprocessor systems, but multiple systems working together
  - Connected via **LAN** (local-area network)
  - Storage often shared via **SAN** (storage-area network)

  - Main reasons:
    - **High availability**
      - Asymmetric clustering (one machine in hot-standby mode)
      - Symmetric clustering (multiple machines running and monitoring each other)
    - **High performance (HPC)**
      - Applications must be written to exploit **parallelization**

Clustered Systems



Operating Systems Concepts

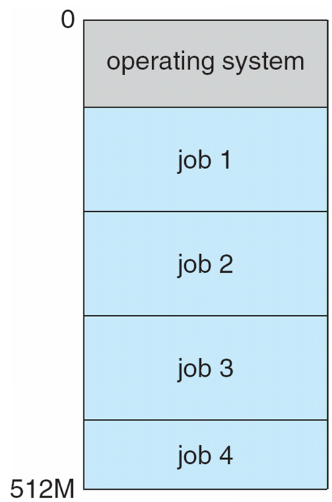


A von Neumann architecture

Operating Systems Concepts

- **Multiprogramming** (efficiency)
  - Single user cannot keep CPU and I/O devices busy at all times
  - Jobs (code & data) organized s.t. CPU always has at least one to execute
  - Subset of jobs kept in memory
  - When a job has to wait (e.g., for I/O), the OS switches to another job

Memory Layout for Multiprogrammed System



Operating Systems Concepts

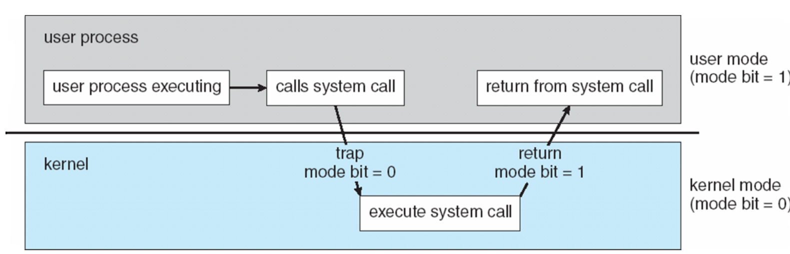
- **Timesharing (multitasking):**
  - Switching between jobs happens so frequently that users can interact with each job while it is running: **interactive computing**
  - **Response time** (e.g., < 1 second)
  - Each user has at least 1 program executing in memory (**process**)
  - If several jobs ready to run at the same time: **CPU scheduling**
  - If processes don't fit into memory: **swapping**
  - **Virtual memory** allows of execution of partially loaded processes

Operating Systems Concepts

- Access to resources needs to be controlled:
  - Simultaneous access
  - Unauthorized access
  - "Improper" access (e.g., too long)
- **Dual-mode operating systems**
  - User mode (application)
  - Kernel mode (OS and privileged instructions)
  - **Mode bit** indicates current mode (0 = kernel)
  - Transition via **system calls**

From User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management

- **Process** = program in execution!
  - Program = passive
  - Process = active
- Process needs resources (CPU, memory, I/O, initialization data, files, etc.)
- **Single-threaded process**: one program counter (PC)
- **Multi-threaded process:** one program counter per thread

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and de-allocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  – Free-space management
  – Storage allocation
  – Disk scheduling
- Some storage need not be fast
  – Tertiary storage includes optical storage, magnetic tape
  – Still must be managed – by OS or applications
  – Varies between WORM (write-once, read-many-times) and RW (read-write)

Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

- Movement between levels of storage hierarchy can be explicit or implicit

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks

Recap

- Key Points
  - What is DMA?
  - What is the memory hierarchy?
  - What is caching?
  - What is virtual memory?
  - What is a SAN?
  - What is the difference between kernel and user mode?