Definitions

Lecture 6:

Deadlock: A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource
Mutual Exclusion: Only one process can use a resource
Hold and Wait: A process holding at least one resource is waiting to acquire additional resources
No preemption: a resource can be released only voluntarily be the process holding it, after that process has completed its task
Circular wait: there exists a set {P0, P1, P2, P3} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, Pn-1 is waiting for a resource held by Pn, Pn is waiting for a resource that is held by P0

Lecture 7:

Compile time: the duration of code execution where the developer is compiling code
Load Time: The duration of code execution when a piece of code is being loaded into main memory
Execution Time: The duration of code execution when a piece of code is running
Physical Address: Actual exact location of data in memory
Logical Address: Address that requires additional manipulation to get a Physical Address
Linking: Taking multiple object files and joining them into one large executable file
Dynamic Loading: By default, routine is not loaded. Only loaded when when it is called for the first time.
Dynamic Linking: Linking is Postponed until execution time.
Swapping: Moving a process out of main memory into a backing store. Expensive.
OS Kernel: Part of the OS that deals with hardware communication. Stored in low memory.
Partitioning: splitting main memory into smaller chunks to allocate to programs for code and data storage.
Internal Fragmentation: Main memory use is inefficient. Any program no matter how small occupies an entire portion.
External Fragmentation: Total memory space exists to satisfy a request but is not contiguous
Segmentation: Separates a program up into a collection of ‘segments’
Paging: Divide physical memory into fixed sized blocks called frames. (power of 2)

Lecture 8:

Virtual Memory: Separation of user logical memory from physical memory
Physical Memory: Actual physical memory
Lazy Swapper: never swap a page into memory unless page will be needed
Page Fault: When the OS tries to reference a page that is not mapped to a location in memory
Pure Demand Paging: Start process with no pages in memory

Lecture 9:

Transfer Rate: rate at which data flows between hard drive and computer
Positioning Time: Is the time to move the disk arm to desired cylinder
Head Crash: Results from the head making contact with the disk surface
Disk Scheduling: Purpose is to minimize seek time by allocating disk time between processes
NAS: Network Attached Storage
Mean-Time-to-Failure (MTTF): The average time it takes a disk to fail
Mean-Time-to-Repair (MTTR): The average time to replace a failed disk and restore data
Mirroring: Copy of a disk is duplicated on another disk
Disk Striping: Splitting the bits across multiple disks
Bit level striping: The bits of a byte are split across multiple disks
Block-Level striping: The blocks of a file byte are split across multiple disks
SCSI: Small Computer System Interface

Lecture 10:

Shared Lock: similar to reader lock - Several processes can acquire concurrently
Exclusive Lock: Similar to writer lock - Only one process can acquire
Absolute Pathing: path is absolute to the root of the filesystem
Relative Pathing: path relates from the currently file
Garbage Collection: Remove unused and old locks/data
Extent: A contiguous block of disks
Indexed Allocation: Each file has its own index blocks
Consistency Checking: Compares data in directory structure with data blocks on disk and tries to fix inconsistencies

Lecture 11:

Content

Deadlock:

Resource Graph:

- If graph contains no cycles, no deadlocks
- If graph contains a cycle
    - If only one instance per resource type, deadlock
    - If several instances per resource type, possibility of deadlock

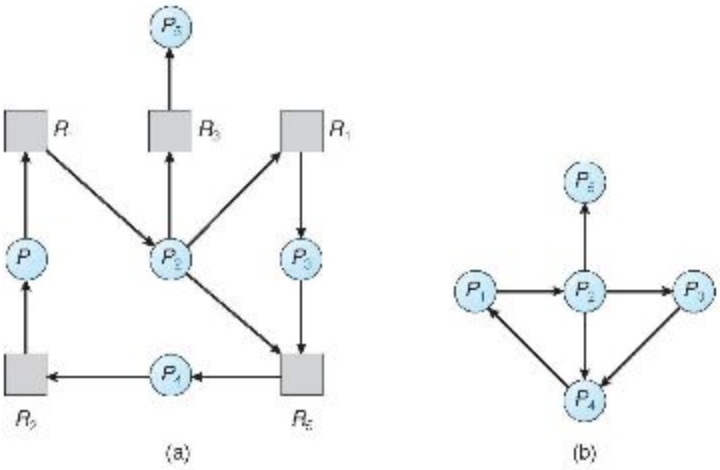
Deadlock avoidance:

- Mutual Exclusion: Must hold for non-sharable resources that can be accessed simultaneously by various processes. Therefore, cannot be used for prevention

- Hold and Wait: Must guarantee that whenever a process requests a resource, it does not hold any other resources.

Safe State:

System is in safe state if there exists a sequence <P1, P2 … Pn> of ALL the processes in the system such that for each Pi, the resources that Pi can still request can be satisfied currently available resources.



Fixed Partitioning:

Partition main memory into non overlapping sets called partitions, these can be of equal or unequal sizes. Any process whose size is less than or equal to the partition size can be loaded into memory.

If all partitions are occupied, the OS can swap a process out of a partition.

Fixed Partition Problems:
Main memory is inefficient. Any program, no matter how small occupies an entire partition. This is called *Internal Fragmentation*.
Unequal-size partitions lessen these problems but they still remain.

Dynamic allocation: Partitions are not used, instead processes are fit into “holes” in memory that are large enough.
External fragmentation: Total memory space exists to satisfy a request but its not contiguous.

Simple Segmentation:

Each section of a program (Code, Variables, Globals, Stack) is all allocated separately and filled into whatever gaps are available in the memory.

Advantages: the unit of allocation is:
Smaller than the entire program,

Segments can fit into gaps,
Easy to protect and share segments.

Disadvantages:
External fragmentation hasn't been eliminated. Holes still exist in memory

Paging:

Divide the physical memory into fixed-sized blocks called frames.

- Size usually a power of two.

Divide local memory into blocks of same size called pages.
To run a program of size *n* pages, we need to find *n* free frames and assign them to the program

- Keep track of all free frames
- Set up a page table to translate logical addresses to physical ones

Local addresses of paging consists of Page # and Offset.

Disk Scheduling:

FCFS:
First come first serve. Exactly what it sounds like.

SSTF:
Selects the request with the shortest distance from the current head position.

SCAN:
Disk arm starts at one end of the disk and services all requests until it hits the other end of the disk. At which point the direction of the arm reverses and satisfies other requests.

C-SCAN:
Similar to SCAN however when the arm reaches the end of the disk it jumps back to position 0 (start of the disk)

LOOK and C-LOOK:
Similar to the respective previous two however instead of going to the end of the disk it only goes as far as the last request in each direction.

RAID Structure:

RAID - Redundant Array of Inexpensive Disks. Increase the amount of disks to increase mean time to failure.

Disk Striping: Splitting the bits (or blocks) across multiple disks.

- Bit level striping: Bits are separated across the multiple disks
- Block Level Striping: Blocks are separated across the multiple disks
- E.G. 100 disks with MTTF of 100k hours
- Mean time to data loss is 100,000/100 = 1,000 H