

Operating Systems Topics

Fundamentals of Operating Systems

How do uniprogrammed and multiprogrammed systems differ? Why were they developed?

How does a batch programming system differ from an interactive one?

What is the difference between kernel and user space?

What is a system call, and how does it differ from a system library or user library?

What is a kernel trap? A signal? An interrupt? What is an interrupt vector? How is each used?

***Know how a system call works and how its execution differs from a regular process invocation.**

Processes & Threads

What are the segments of a process in memory? What is the function of each?

What are the possible states of a process/thread? How do they differ? What do they mean?

Which segments are shared between processes, and which are not? Why/why not?

What are the advantages / disadvantages of threads compared to processes? When would we use each?

What are the main preemptive scheduling algorithms? How about non-preemptive? Which are better and under what circumstances? What would be use under an interactive system vs a batch system?

What is a semaphore? What are their functions / features? How are different types of semaphores different? How do they relate to / differ from mutex entities?

What are race conditions? How do we avoid them? Why does it matter?

What is a critical region? Why should we protect a critical region, and how would we do so?

What are the types of process synchronization? How do they differ? What are the benefits / drawbacks?

What are the different forms of interprocess communication and examples? What are the benefits / drawbacks of each? How do they play a role in synchronization?

What are the different types of schedulers? What role does each play? How do they interact with the various parts of the hardware and the operating system? What are their behaviors, and when do those behaviors come into play?

***Know the scheduling algorithms covered and how they would execute given a sample problem.**

***Know the major categories of synchronization problems, the problems they present, and how they can be resolved.**

***Know the difference between Safety, Concurrency, and Liveness issues.**

Memory

What is a memory page? How does it differ from a segment?

What are the memory management algorithms used by the memory scheduler? How do they differ? When would you use one over another?

How do memory addresses work? What is the difference between a logical and physical address?

What specialized hardware exists to aid in the computation of memory addresses? How does it work?

What are the different levels of memory? How do they impact speed, and how do they differ in monetary expense?

What is the difference between a cache miss and a page fault?

How does memory access work when attempting to access memory stored in cache? How does the fundamental process used for cache differ from the process used to detect a page fault?

What are the various ways that free memory can be tracked? How does this differ from Pret

What is fragmentation? Compaction? How do they work? (Do they work?)

What is the page table, and how is it used? What approaches can be used for very large tables?

How does a Translation Lookaside Buffer work?

What are the challenges of cache consistency, and how can they be addressed?

What is effective access time? How does it play a role, and how does it differ, in caching and paging systems?

What are the common page replacement approaches? How do they differ?

How do page faults work? What role does the memory management unit play? The kernel? Userspace?

How does segmentation play into memory management? How do segments & pages interact with each other?

File Systems

What are the expectations of long term storage? How do they differ from those of working memory?

What is a file, what are its primary functions, and what are its constituent parts? How about directories?

How are directories implemented? What are common functions used on directories?

What is a hard (normal) link, and how does it differ from a soft (symbolic) link?

What are the different ways files can be allocated? What are the benefits and drawbacks of each?

What is a file control block? A File allocation table? An inode? How do each of these work?

What is chained allocation? How does it modify indexed allocation?

What is the difference between single-indirect, double-indirect, and triple-indirect allocation?

How does allocation approach impact data access time?

What is a block? What is its function? How does block size impact efficiency and access time?

What approaches are frequently used to track unused blocks?

How can we check for file system consistency? Why does it matter? What incorrect states could we run into, and what are the possible solutions?

How does a block differ from a zone?

What is the purpose of a block cache, and how does one work?

What approaches can be used to support file system reliability? How do they differ? What are benefits / drawbacks of each?

IO Devices

What is the difference between a character (stream) device and a block device?

What is the function / purpose of a device controller?

What are the different ways that I/O can be managed/performed?

What are the differences between CPU-managed, co-processor-managed, memory-mapped, and Direct Memory Access (DMA) approaches to I/O? How does each work? What are the benefits and drawbacks of each?

How does the software approach to each type of I/O differ in practice?

What is the purpose and function of device drivers? How do they assist in getting data from the device to the user or system? What layers of the OS are involved?

How do clocks work? Keyboards? Mice?

How are characters displayed on a character display? How does scrolling work?

How are pixels displayed on a bitmap display? How does scrolling work?

***Know the typical bus layouts and variations on approaches used for I/O.**

Networking

What are the differences between the OSI and IP layering model? Why does this matter? How did it play a role in the development of IP networks?

What is the function and purpose of each layer of the IP protocol suite?

How is data formatted when sent over the network? How is this different from its native (local host) formatting?

What is the difference and purpose of device, network, and subnetwork designations, and how do they differ between IPv4 and IPv6?

How are address ranges designated? What do they represent?

What is the difference between unicast, broadcast, and multicast transmission?

What is the function of the IP (Internet Protocol) address? Specifically, what does it identify?

What is the function of the port in the transmission layer? Specifically, what does it identify? How does this differ in TCP (Transmission Control Protocol) and UDP (Uniform Datagram Protocol)?

What are the primary differences between TCP and UDP from a feature perspective? Why do these matter?

What is a socket? What is it used for?

How is a TCP connection established, and how is it terminated?

What is a partial message? Can a partial message occur in TCP? How about UDP?

What is the purpose of flow control? Which protocols implement flow control?

What is the receive window, and how does it work? How does it impact the send window?

What is the send window, and how does it work? What does it mean when a byte is "ACK"ed (acknowledged)?