CENG331 Study Guide!

The following guide provides you a list of terms and topics that you can check your knowledge. It's by no means complete. Some topics are mentioned in general terms since it is difficult to summarize or mention them. The size of the discussion is not an indication of its importance. I merely tried to type in the key concepts with the hope that they will be of help to your study. **We have decided to exclude some of the topics, despite covering them at a certain detail. No questions will be asked in the final from the topics marked in RED.**

Enjoy! Erol Sahin

- Birth of GUI, UNIX, Apple, Microsoft, GPL, Open source,
- Evolution of OS's
  - Just Libraries
  - Batch systems
  - Multi-programmins
  - Time-sharing
  - MULTICS and UNIX
  - PCs
- What services do we expect from OS?
  - Program execution
  - I/O operations
    - *Standardized interfaces to extremely diverse devices*
  - File system manipulation
    - *How do you read/write/preserve files?*
  - Communications
    - *Networking protocols/Interface with CyberSpace?*
  - User interface- Almost all operating systems have a user interface (UI)
  - Cross-cutting capabilities
    - Error detection & recovery
    - Resource allocation
    - Accounting
    - Protection
- System calls (versus systems programs)
- POSIX
- User and kernel modes, privileged instructions
- Interrupts (or exceptions)
  - Hardware
  - Software
- Interrupt handling
  - Interrupt vector (table)
  - Interrupt handler
- Asynchronous interrupts
  - I/O
  - Hard reset
  - Soft reset
- Synchronous interrupts
  - Traps

    - Faults
    - Aborts
- The concept of process
- Components of process state and PCB
    - Address space
    - Processor state
    - OS resources
- Process address space
- Execution states of a process and transitions
- Context switching
- Ready queue, Disk I/O queue
- Process creation
- Fork()
    - Parent, child,
    - Wait, exit, execve
- UNIX file abstraction
    - Stdin, stdout, stderr
    - File handles
    - File table
    - Shell redirection
    - Dup(), dup2()
- Inter-process communication (communication between processes)
    - Pipes
    - Shared memory
- The concept of a thread and TCB
    - Components of TCB
    - Context switch between threads
    - User-level threads
        - Address space
        - Setjmp, longjmp
    - Kernel-level threads
    - Preemption
    - Preemptive vs. nonpreemptive threads
        - Yield()
    - Pros and cons of user-level and kernel-level threads
- Syncronization problem
- Race condition
    - What's the reason?
    - When does it happen?
- Critical section
    - Mutual exclusion
    - Four requirements
        - Mutual exclusion
        - Progress
        - Bounded waiting (no starvation)
        - Performance
- Locks
    - Spinlocking - pros and cons
    - Enable/disable interrupts

    - Test-and-set instruction
- Software solutions
    - Peterson's algorithm
- Mutexes
- Limitation of locks
- Semaphores
    - Up, V(), signal()
    - Down, P(), wait()
    - Implementation w locks
- Monitors and condition variables
    - Monitor concept
    - Condition variables
    - Wait, notify, notifyAll
    - Mesa and Hoare implementations
- Implementation of locks, semaphores and monitors with each other
- Synchronization problems
    - Signalling
    - Rendezvous
    - Barrier
    - Producer-consumer problem
    - Reader-writers problem
- Synchronization "problems"
    - Deadlocks
        - Conditions
            - Mutual exclusion
            - Hold-and-wait
            - No preemption
            - Circular wait
        - Dining philosophers
            - Different solution strategies that aim to break one of the conditions
        - Avoidance
            - Resource allocation graph
                - Deadlock vs. cycles in the graph
            - Safe, and unsafe states
            - Banker's algorithm
    - Starvation
    - Priority inversion
        - Priority inheritance (or donation)
- Scheduling problem
    - Dispatcher vs. scheduler
    - Preemptive vs. non-preemptive scheduling
    - Scheduling policies
        - FCFS
        - RR
        - SJF
        - SRTF
        - Priority
        - **Lottery**

        - Multi-level Feedback Queues
        - Other issues for uniprocessor scheduling
        - Multi-processor scheduling issues
        - Real-time schedulers
    - Virtual Memory
        - Mechanisms
        - Memory management requirements
            - Protection
            - Fast translation
            - Fast context switching
        - Advantages
        - MMU
        - TLB
        - Fixed and variable partition (or segmentation) approaches to MMU
        - External and internal fragmentation
        - Paging for Virtual Memory
            - Page Table
                - PTE format
                - Page table base pointer
                - Multi-level page tables
                    - Page directory
            - Advantages
            - Page fault
            - Demand paging
                - Holes in virtual address space
                - Copy-on-write in fork
            - Page in /Page out of different sections of virtual address space
                - Swap,
                - **Zero page**
            - Memory-mapped file
            - Page replacement
                - Temporal and spatial locality in programs
                - **Inverted page table**
                - Policies
                    - OPT or MIN
                    - Random
                    - FIFO
                        - Belady's anomaly
                    - LRU
                        - Spatial and computational complexity
                        - Approximate LRU with reference bits
                        - Second-chance
                            - **N'th chance**
                - Swap file
                    - What is swapped, where
                - Performance evaluation
                    - Page fault freq.
                    - Working set
    - Disks and filesystems

- Overall characteristics of disks and its evolution in time
- **Disk I/O scheduling**
  - **Latency**
  - **Throughput**
  - **Policy**
    - **FIFO**
    - **SSTF**
    - **SCAN (elevator)**
    - **CSCAN**
    - **Pros and cons**
- **Boot sequence**
  - **MBR,**
  - **Disk partitions**
  - **OS boot loader**
- Filesystem
  - Requirements
  - File abstraction and operations
  - Directory abstraction and operations
  - File system mounting
- File implementation
  - Contiguous
  - Linked list
  - **FAT**
  - **inodes**
- Directory implementation
  - Windows
  - UNIX
- **Shared files**
  - **Hard and soft links in unix**
  - **Shortcuts**
- Free Block representation
  - Free Blocks
  - Bitmap
- Case studies
  - FAT
  - UNIX V7
  - **ISO9660**
  - **VFS**
- Corruption
  - Possible corruptions in filesystems
    - Consequences
    - Consistency check and fixes
  - **Journalling filesystems**
    - **Log'ging filesystem transactions**
    - **Checkpoint**
    - **Commit**
    - **Intent-to-commit**
    - **Recovery**

- **Backups**
  - **Physical dump**
  - **Logical dump**
    - **Issues such as holes,links and special files**
  - **Pros and cons of each approach**
- **Filesystem caching**
  - **Read-ahead**
  - **Placement**

- I/O systems
  - Port, bus, device controller
  - I/O instructions
    - Specialized, memory mapped, hybrid
  - **Device controller**
    - **Status, command, data-in, data-out Registers**
  - **Communication between device driver and device controller**
  - **I/O interfacing**
    - **Programmed I/O**
    - **Interrupt-driven I/O**
      - **Device driver and Interrupt handler cooperation**
        - **Detailed operation of device drivers and interrupt handlers and their overall structure**
      - **Interrupt priority, masking, chaining**
    - **DMA - detailed operation of**
    - **Uniform interfacing of device drivers**
  - Characteristic of I/O devices
    - Character vs. block
    - Blocking, non-blocking I/O calls
    - Etc.
  - **I/O scheduling**
  - **Buffering**
    - **reasons**
    - **buffering in user space**
    - **buffering in kernel space**
    - **double buffering in kernel space**
  - Caching
  - Spooling
  - Error handling

- Protection
  - Goals
  - Objects and domains
  - **Domain implementation**
    - **User**
    - **Process**
    - **Procedure**
    - **Setuid**
    - **Switching**
    - **Transfer of rights**
    - **Owner rights**

        - **Access matrix**
        - **Access list**
        - **capability list**
        - **lock-key**
    - Security
        - Goals
        - Passwords
        - Cryptography
        - Potential security threats
        - Document signing
        - **Sandbox'ing**
    - **Multiple processors**
        - **OS porting/implementation**
        - **Each CPU with its own OS**
        - **Master-slave multiprocessors**
        - **Symmetric multiprocessors**
        - **Issues related to hardware and OS**
            - **Synchronization**
            - **Scheduling**
    - Virtualization
        - Goal
        - **Type I hypervisors**
        - **Type II hypervisors**
        - **Paravirtualization**
        - **Other issues**