|  |  |
| --- | --- |
| What is the operating System ? |  |
| program that acts as intermediary between the user of a computer and the computer hardware   * **OS is resource allocator** : Manages all resources and Decides between conflicting requests for efficient and fair resource use * **OS is control program** **:** Controls execution of programs to prevent errors and incorrect use of the computer |  |

|  |  |
| --- | --- |
| What is the operating system goals ? |  |
| * + Execute user programs   + Make the computer system suitable to use   + Make solving user problems easier |  |

|  |  |
| --- | --- |
| What is the computer system component ? |  |
| * + **Hardware** : provides basic computing resources , CPU , memory , I/O   + **Operating system**   + **Application program** : define the ways in which the system resources are used to solve the computing problems , word , excel   + **System program** : compiler , database system , assembler   + **Users** |  |

|  |  |
| --- | --- |
| What is the different between program and process ? |  |
| It is the program image that is in RAM to be execution . process It is a unit of work within the system. Program is a passive entity, process is an active entity. |  |

|  |  |
| --- | --- |
| What is the kernel ? |  |
| The one program running at all times on the computer |  |

|  |  |
| --- | --- |
| What is CPU component ? |  |
| * Register * ALU * Control unit |  |

|  |  |
| --- | --- |
| What does the RAM content ? |  |
| Instruction or data |  |

|  |  |
| --- | --- |
| What is the bootstrap program ? |  |
| * It's the program the loaded typically in ROM and it known as firmware   + It Initializes all aspects of system   + Loads operating system kernel and starts execution |  |

|  |  |
| --- | --- |
| Explain the computer system operation ? |  |
| * One or more CPUs, device controllers connect through common bus providing access to shared memory * Concurrent execution of CPUs and devices competing for memory cycles * Each device controller is in charge of a particular device type * Each device controller has a local buffer * CPU moves data from/to main memory to/from local buffers * I/O is from the device to local buffer of controller * Controller is response fro move the date from/to devices * Device controller informs CPU that it has finished its operation by causing an **interrupt** |  |

|  |  |
| --- | --- |
| Explain the interrupt ? |  |
| * Signals the CPU to a temporary suspension of the current execution * Interrupt transfers control to the interrupt service routine ISR, through the interrupt vector, which contains the addresses of all the service routines * Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt* * Operation system is **interrupt driven** * Interrupt driven by hardware . |  |

|  |  |
| --- | --- |
| What is the different between interrupt and trap with example ? |  |
| **Trap** : software-generated interrupt caused either by an error or a user request **example** : division by zero , invalid memory access  **Interrupt** : some time hardware-generated **example:** problem in hard disk or Power outage |  |

|  |  |
| --- | --- |
| What is the types of interrupt handling ? |  |
| * **polling** * **vectored** interrupt system |  |

|  |  |
| --- | --- |
| What is the interrupt types ? |  |
| * **User** **interrupt** : user cancel the program before it finished * **Software** **interrupt** : deviation by zero * **Hardware** **interrupt** : power outage * **I/O interrupt** : Device controller informs CPU that it has finished * **Timer** **interrupt** : round robin algorithm |  |

|  |  |
| --- | --- |
| How can the operation system save the CPU state ? |  |
| storing registers and the program counter |  |

|  |  |
| --- | --- |
| What is the methods of I/O structure ? |  |
| * **After I/O starts, control returns to user program only upon I/O completion** * Wait instruction idles the CPU until the next interrupt * Wait loop (contention for memory access) * At most one I/O request is outstanding at a time * **After I/O starts, control returns to user program without waiting for I/O completion** * **System call** : request to the operating system to allow user to wait for I/O completion * **Device-status table :** contains entry for each I/O device indicating its type, address, and state * Operating system access into I/O device table to determine device status and to modify table entry |  |

|  |  |
| --- | --- |
| What is the direct memory access ? |  |
| * Used to allow high-speed I/O devices to transmit information at close to memory speeds * Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention * Only one interrupt is generated per block, rather than the one interrupt per byte |  |

|  |  |
| --- | --- |
| What is the storage structure in computer system ? |  |
| * **Main memory (RAM)** : array and volatile and only large storage media that the CPU can access directly * **Secondary storage** : extension of main memory that provides large nonvolatile storage capacity * **Magnetic disks** : rigid metal or glass platters covered with magnetic recording material |  |

|  |  |
| --- | --- |
| What does disk surface contain ? |  |
| Disk surface is logically divided into **tracks**, which are subdivided into **sectors** |  |

|  |  |
| --- | --- |
| What is the disk controller ? |  |
| It determines the logical interaction between the device and the computer |  |

|  |  |
| --- | --- |
| According to what the storages organized ? |  |
| * Speed * Cost * Volatility * Capacity |  |

|  |  |
| --- | --- |
| What is the caching ? |  |
| * **copying information(not process) into faster storage system that means Information in use copied from slower to faster storage temporarily :** * cache memory is cache for main memory * main memory is *cache* for secondary storage * **Faster storage (cache) checked first to determine if information is there**   + If it is, information used directly from the cache (fastest)**(cache hit)** * If not, data copied to cache and used there (cache miss) |  |

|  |  |
| --- | --- |
| Summarize in figure how the computer work ? |  |
| 1 |  |

|  |  |
| --- | --- |
| Who manage the transition between storage device ? |  |
| * Hard ware is response to the transition between cache, RAM and register * OS is response to transition between RAM and hard disk |  |

|  |  |
| --- | --- |
| Sort the storage device from small to large ? |  |
| * Register * Cache * RAM * Electronic disk * Magnetic disk * Optical disk * Magnetic tapes |  |

|  |  |
| --- | --- |
| What is the current computer system architecture ? |  |
| * Most systems use a single general-purpose processor * Most systems have special-purpose processors like multiprocessors, multi-cores and Clustered . |  |

|  |  |
| --- | --- |
| What is the advantages of multiprocessors (parallel, coupled) computer ? |  |
| * **Increased throughput** : increase the rate of process finished on unit time because of each processors work in one process . * **Economy of scale** : because of shared recourses such as I/O and RAM . * **Increased reliability** : continue to run even if one or more processor get failure . |  |

|  |  |
| --- | --- |
| What is the types of multiprocessing ? |  |
| **Asymmetric multiprocessing** : on of the processors is master and it monitor the other processors , and if one is failure it do its task .  **Symmetric multiprocessing** : all the processors are the same . |  |

|  |  |
| --- | --- |
| What is the ready queue ? |  |
| To run the process, this process should be in ready queue , this main this process hold all of its resources . |  |

|  |  |
| --- | --- |
| What is the different between multiprocessor and multi-cores architecture ? |  |
| **Multiprocessor** : every processor in different ship .  **Multi-cores** : all core are in the same ship . |  |

|  |  |
| --- | --- |
| What is the advantages of multi-core architecture on multiprocessor ? |  |
| * Faster because every core in the same ship . * Use less power . |  |

|  |  |
| --- | --- |
| What is the clustered system . |  |
| * Like multiprocessor systems, but multiple systems working together . * Usually sharing storage via a storage-area network (SAN) * Provides a high-availability service which survives failures |  |

|  |  |
| --- | --- |
| What is the types of clustered ? |  |
| * + - **Asymmetric clustering**: has one machine in hot-standby mode     - **Symmetric clustering** : has multiple nodes running applications, monitoring each other |  |

|  |  |
| --- | --- |
| What is the CPU utilization ? |  |
| Keep CPU busy as possible , and it apposite of idle . |  |

|  |  |
| --- | --- |
| What is the multiprogramming ? |  |
| * It used for efficiency . * Because single user cannot keep CPU busy all time . * **Multiprogramming** organize the process so CPU always has one to execute . |  |

|  |  |
| --- | --- |
| What is job scheduling ? |  |
| * Because subset of total jobs in the system is kept in the memory . * **Job scheduling** is part of OS and it used to select the process that will be loaded to memory from job pool . |  |

|  |  |
| --- | --- |
| What is the time sharing (multitasking) ? |  |
| It is logical extension of multiprogramming in which CPU switch jobs so frequently that users can interact with each job while it is running, creating **interactive** computing |  |

|  |  |
| --- | --- |
| What is the response time ? |  |
| It's the time required to get the first response o the process (not outing) |  |

|  |  |
| --- | --- |
| What is the requirements of interactive computing ? |  |
| * Response time should be < 1 second . * Each user has at least one program executing in memory **process** |  |

|  |  |
| --- | --- |
| What is CPU scheduling ? |  |
| * It is part of OS and it used to select a process from ready queue of the RAM to be execute . * It used if several jobs ready to run at the same time . |  |

|  |  |
| --- | --- |
| What is the swapping ? |  |
| * It used if the process don’t fit in the memory * It move the data from memory to VM in HD (swap out) * It move the date from VM in HD to memory (swap in) |  |

|  |  |
| --- | --- |
| What is virtual memory ? |  |
| It is in HD and it allows execution of processes not completely in memory . if the process don’t fit in the memory |  |

|  |  |
| --- | --- |
| What is the dual-mode operation ? |  |
| * It allow OS to protect itself and other system component . * It is divide the mode to : user mode – kernel mode . |  |

|  |  |
| --- | --- |
| What mode bit ? |  |
| * mode bit provided by hardware .   + - Provides ability to distinguish when system is running user code or kernel code     - Some instructions designated as **privileged**, only executable in kernel mode . (change the time) |  |

|  |  |
| --- | --- |
| Explain the transition between the processes ? |  |
| * Timer to prevent infinite loop * 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 |  |

|  |  |
| --- | --- |
| Explain the transition between user mode and kernel mode ? |  |
| **System call changes mode to kernel, return from call resets it to user** |  |

|  |  |
| --- | --- |
| Explain the process management ? |  |
| * Process need recourses to accomplish its task (CPU,RAM,I/O, files, data). * Process termination requires reclaim of any reusable resources . * Single threaded process has one counter , so the process executes instructions sequentially, one at a time, until completion . * Multi-threaded process has one program counter per thread |  |

|  |  |
| --- | --- |
| What does OS do to manage the process ? |  |
| * Creating and deleting both user and system processes : **load programs** * Suspending and resuming processes : **interrupt** * Providing mechanisms for process synchronization : **time sharing** * Providing mechanisms for process communication : **data sharing** * Providing mechanisms for deadlock handling : **cycle needing** |  |

|  |  |
| --- | --- |
| Explain 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 * Optimizing CPU utilization **(largest possible)** and computer response **(smallest possible)** to users |  |

|  |  |
| --- | --- |
| What does OS do to manage the memory ? |  |
| * 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 deallocating memory space as needed |  |

|  |  |
| --- | --- |
| Explain storage management ? |  |
| * + Abstracts physical properties of information storage to logical   storage unit - **file**   * + Each medium is controlled by device (i.e., disk drive, tape drive) * **File-System management**   + Files usually organized into directories **(folder)**   + Access control on most systems to determine who can access what |  |

|  |  |
| --- | --- |
| What does OS do to manage the storage ? |  |
| * Creating and deleting files and directories **(folder)** * Primitives to manipulate files and directories **(folder)** * Mapping files onto secondary storage * Backup files onto stable (non-volatile) storage media |  |

|  |  |
| --- | --- |
| Explain 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 * Entire speed of computer operation hinges on disk subsystem and its algorithms |  |

|  |  |
| --- | --- |
| What does OS do to manage the mass storage ? |  |
| * Free-space management * Storage allocation * Disk scheduling |  |

|  |  |
| --- | --- |
| Explain performance of different storage ? |  |
|  |  |

|  |  |
| --- | --- |
| What is cache coherency ? |  |
| Is used to ensure that all caches of all processes has the most recent value . |  |

|  |  |
| --- | --- |
| What is the I/O subsystem Responsibility ? |  |
| * Memory management of I/O including * **buffering** (storing data temporarily while it is being transferred), * **caching** (storing parts of data in faster storage for performance), * **spooling** (the overlapping of output of one job with input of other jobs) * General device-driver interface * Drivers for specific hardware devices |  |

|  |  |
| --- | --- |
| What is the different between 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 |  |