What is the services that the OS provides to help the user ?

- **User interface** : Command-Line (CLI), Graphics User Interface (GUI), Batch
- **Program execution** - load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
- **I/O operations** : A running program may require I/O, which may involve a file or an I/O device
- **File-system manipulation** : programs need to read and write files and directories, create and delete them, search them .
- **Communications** : Processes may exchange information, on the same computer or between computers over a network
- **Error detection** : OS needs to be constantly aware of possible errors and correction .

what is the OS services that for ensuring the efficient operation of the system itself via resource sharing ?

- **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
- **Accounting -** To keep track of which users use how much and what kinds of computer resources
- **Protection and security :** The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other .

Explain CLI ?

- Command Line Interface (CLI) or command interpreter allows direct command entry
  - Sometimes implemented in kernel, sometimes by systems program
  - If the latter, adding new features doesn’t require shell modification

Explain GUI ?

- User-friendly desktop metaphor interface
- Usually mouse, keyboard, and monitor , Icons represent files, programs
- Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a folder)

What is system call ?

Programming interface to the services provided by OS .

How can system call used ?

Mostly access by programs via high-level application program interface(API) rather than direct system call use .

What is API ?

Application program interface used to specify the parameter and specification of system call .

List three example of API ?

Win32 API for windows .
POSIX API for POSIX-based system like UNIX, Linux and Mac
Java API for java virtual machine .

Explain system call implementation ?

Number associated with each system call , and system call interface(API) maintain a table indexed according to these number .

How system call used ?

System call interface (API) invokes intended system call in OS kernel and returns status of the system call and any return values .

What is the benefits of API and not in direct system call use ?

- The caller need know any thing about how the system call are implemented
- Just need to obey API and understand what OS will do as result call
- So the benefit is API hides most details of OS interface from programmer
- Other benefit is you can run any program on any machine that use the same API platform .

What is ways of passing system call parameters ?

- Simplest: pass the parameters in registers
  - In some cases, may be more parameters than registers
- Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register
- Parameters placed, or pushed, onto the stack by the program and

popped off the stack by the operating system

- Block and stack methods do not limit the number or length of parameters being passed

What is the types of system call ?

- Process control
- File management
- Device management
- Information maintenance
- Communications
- Protection

What is the system programs ?

- System programs provide a convenient environment for program development and execution
- Some of them are simply user interfaces to system calls; others are considerably more complex
- Most users' view of the operation system is defined by system programs, not the actual system calls

What is type of system programs

- **File manipulation** : Create, delete, copy, rename, print, dump, list, and generally manipulate files and directories

- **Status information** : Some ask the system for info - date, time, amount of available memory, disk space, number of users, Others provide detailed performance, logging, and debugging information

- **File modification** : Text editors to create and modify files, Special commands to search contents of files or perform transformations of the text .

- **Programming language support** : Compilers, assemblers, debuggers and interpreters sometimes provided

- **Program loading and execution** : Absolute loaders, relocatable loaders, linkage editors, and overlay-loaders, debugging systems for higher-level and machine language

- **Communications** : Provide the mechanism for creating virtual connections among processes, users, and computer systems, Allow users to send messages to one another’s screens, browse web pages, send electronic-mail messages, log in remotely, transfer files from one machine to another

- **Application programs**

How the operating system designed and implemented ?

- Design and Implementation of OS not “solvable”, but some approaches have proven successful
- Internal structure of different Operating Systems can vary widely
- Start by defining goals and specifications
- Affected by choice of hardware, type of system

What is types of operating system goals ?

- **User goals** : operating system should be convenient to use, easy to learn, reliable, safe, and fast
- **System goals** : operating system should be easy to design, implement, and maintain, as well as flexible, reliable, error-free, and efficient

What is the impotent principles that has be separated in operating systems?

- **Policy** : what will be done ?
- **Mechanism** : how to do it ?

Give example of separation between policy and mechanism in OS ?

In timer construction :

- **Policy** : the loop don't spend more than 5 msec .
- **Mechanism** : set timer and OS decrement the counter until it be zero then interrupt will be generated .

What is the benefit of separation between policy and mechanism ?

It allows maximum flexibility if policy decisions are to be changed later .

What is the main structures of operating systems ?

- Simple
- Layered

- Microkernel
- Modules

What is the simple structure of operating systems structures ?

- It is MS-DOS (Microsoft disk operating system)
- It is written to provide the most functionality in the available hardware.
- It isn't divided to module , so the interface and levels of operation are not separated .
- There isn’t has dual mode .

What is the layered approach ?

- Divide the operating system into number of layered .
- Each layer built on top of lower layers .
- The button layer (layer0) is the hardware .
- The highest layer is the user interface .
- With modularity layers are selected such that each uses functions (operations) and services of only lower-level layers
- **the main disadvantage:**
    - ✓ how to define the layer
    - ✓ Less efficient than other structure

What is the structure of UNIX operating system ?

- UNIX limited hardware functionality .
- The UNIX OS consists of two separated part : **System programs** and **kernel**

What is The kernel ?

It the program that running in every time on the computer , and Consists of everything below the system-call interface and above the physical hardware .

What is microkernel structure ?

- It means move as much from kernel to user space. (Win2000)
- The communication takes place between user modules using **message passing** not use shared memory.
- **Advantages** :
    - ○ Easier to extend microkernel because its size is small.
    - ○ Easier to port the OS to new architecture

    - More reliable **because** of less code is running in kernel mode
    - More secure
- **Disadvantages** :
    - Performance overhead of user space to kernel space communication
    - What part of kernel will be move.

That is the module structure ?

- Most modern operating systems implement kernel modules
- similar to layers but with more flexible and efficient.
- **Characteristic**
    - Uses object-oriented approach
    - Each core component is separate
    - Each talks to the others over known interfaces
    - Each is loadable as needed within the kernel

What is the virtual machine ?

The operating system host creates the illusion that a process has its own processor and (virtual memory)

What is the debugging

finding and fixing errors, or bugs

What is types of error and where it will be store ?

- **Log files**: contains information of error caused by OS
- **Core dump file**: contain information of error caused by application
- **Crash dumb** : contain information of error caused by kernel

What does Kernighan's Law say ?

"Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it".

What does DTrace tool in solaris do ?

Help in debugging. by allowing live instrumentation on production systems

What is booting ?

Starting a computer by loading the kernel .