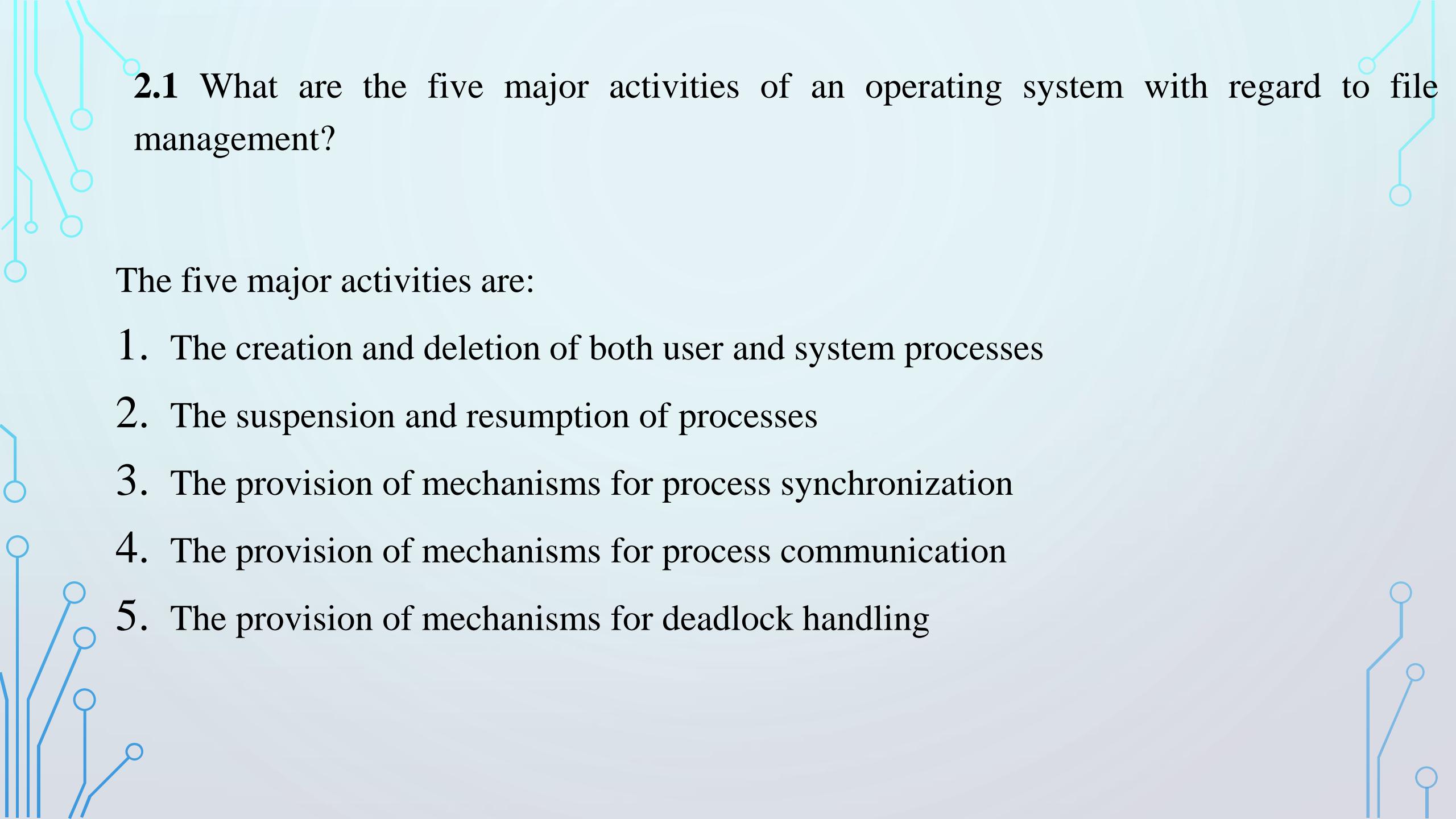


OPERATING SYSTEM

IT-41033

CHAPTER – 2

Q&A



2.1 What are the five major activities of an operating system with regard to file management?

The five major activities are:

1. The creation and deletion of both user and system processes
2. The suspension and resumption of processes
3. The provision of mechanisms for process synchronization
4. The provision of mechanisms for process communication
5. The provision of mechanisms for deadlock handling

2.2 What are the three major activities of an operating system with regard to memory management?

The three major activities are:

- a. Keep track of which parts of memory are currently being used and by whom.
- b. Decide which processes are to be loaded into memory when memory space becomes available.
- c. Allocate and deallocate memory space as needed.

2.7 What is the purpose of the command interpreter? Why is it usually separate from the kernel?

It reads commands from the user or from a file of commands and executes them, usually by turning them into one or more system calls. It is usually not part of the kernel since the command interpreter is subject to changes.

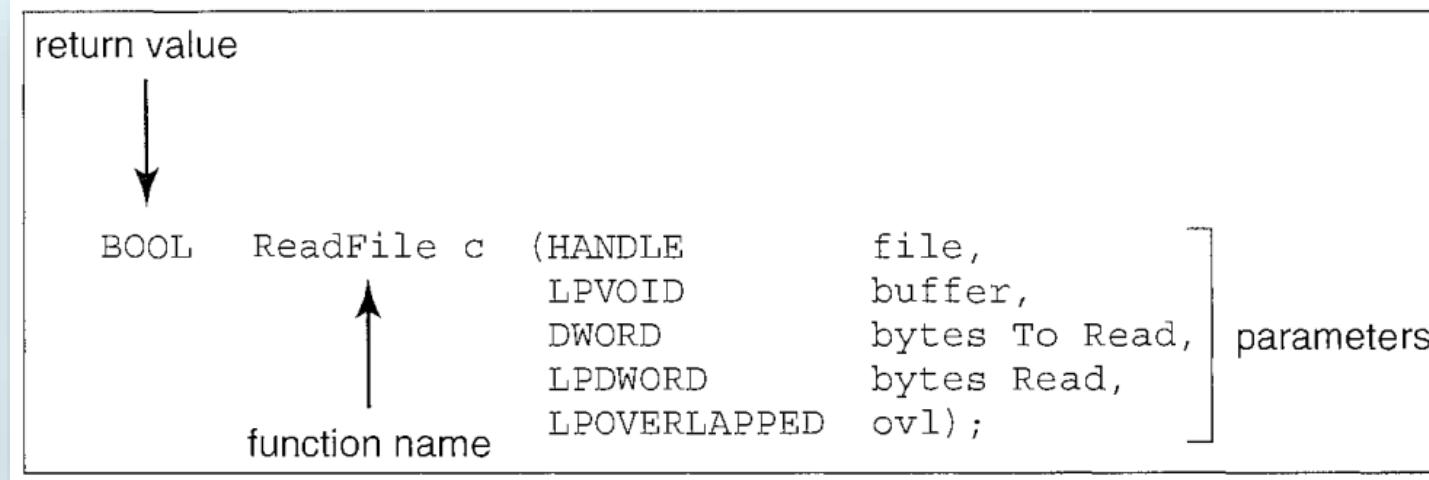
2.10 What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

2.14 What system calls have to be executed by a command interpreter or shell in order to start a new process?

In Unix systems, a *fork* system call followed by an *exec* system call need to be performed to start a new process. The *fork* call clones the currently executing process, while the *exec* call overlays a new process based on a different executable over the calling process.

Q. Describe about the standard API to consider the ReadFile() function in the Win32 API - a function for reading from a file.



A description of the parameters passed to ReadFile()

- HANDLE file – the file to be read
- LPVOID buffer – a buffer where the data will be read into and written from
- DWORD bytesToRead – the number of bytes to be read into the buffer
- LPDWORD bytesRead – the number of bytes read during the last read
- LPOVERLAPPED ovl – indicates if overlapped I/O is being used

Q. What is the purpose of system programs? Describe these categories.

It is known as system utilities, provides a convenient environment for program development and execution. It is bundle of useful system calls. It provides basic functionality to users so that users do not need to write their own programs to solve common problems.

File management: These programs create, delete, copy, rename, print, dump, list and manipulate files and directories.

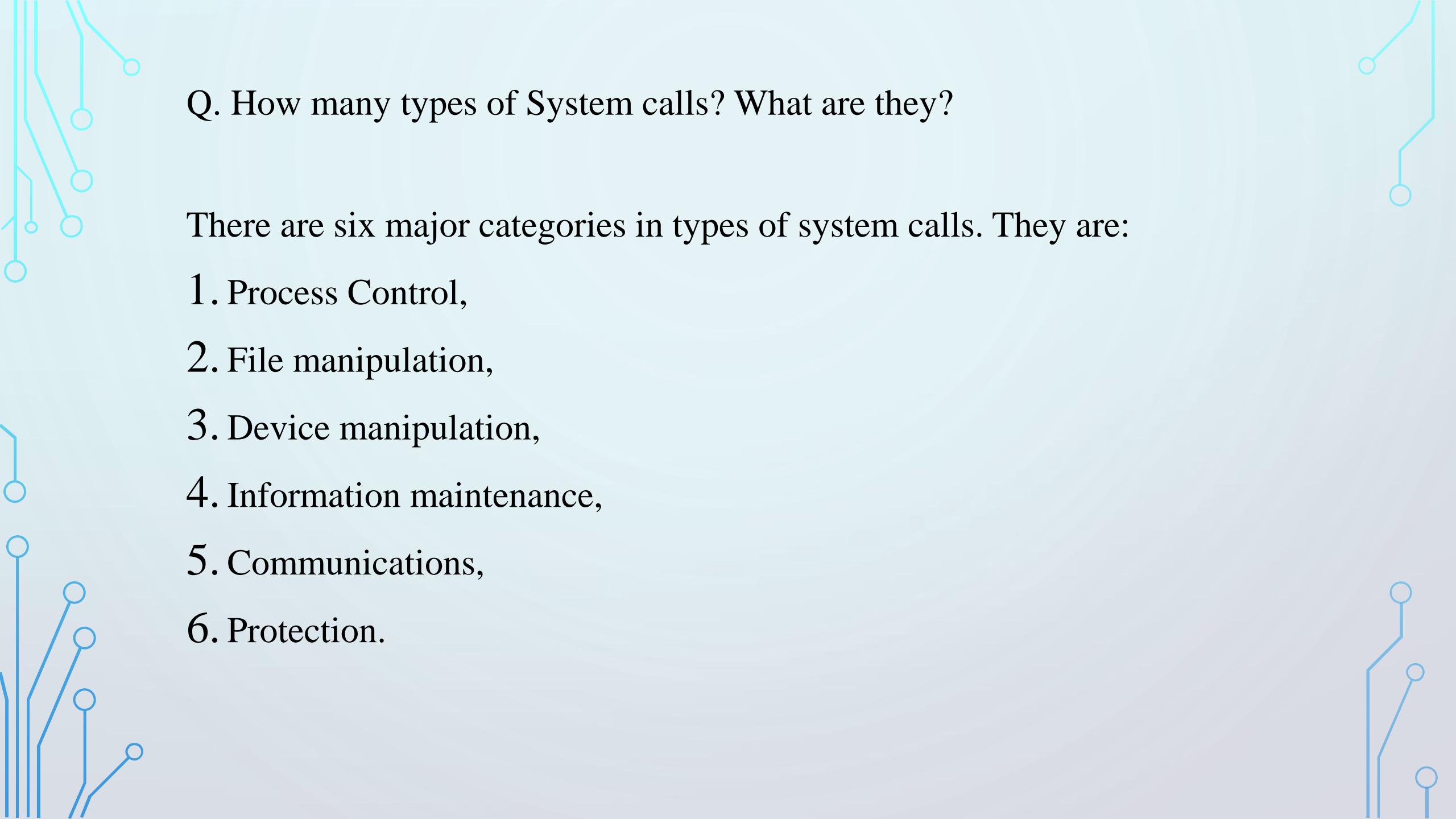
Status information: Some programs ask the system for the date, time, amount of available memory or disk space, number of users, or similar status information. Others are more complex, providing detailed performance, logging, and debugging information. Typically, these programs format and print the output to the terminal or other output devices or files or display it in a window of the GUI. Some systems support a registry, which is used to store and retrieve configuration information.

File modification: Several text editors may be available to create and modify the content of files stored on disk or other storage devices and to search contents of files or perform transformations of the text.

Programming-language support: Compilers, assemblers, debuggers, and interpreters for common programming languages (such as C, C++, Java, Visual Basic, and PERL) are provided to the user with the operating system.

Program loading and execution: Once a program is assembled or compiled, it must be loaded into memory to be executed. The system may provide absolute loaders, relocatable loaders, linkage editors, and overlay loaders.

Communications: These programs provide the mechanism for creating virtual connections among processes, users, and computer systems. They allow users to send messages to one another's screens, to browse Web pages, to send electronic-mail messages, to log in remotely, or to transfer files from one machine to another.



Q. How many types of System calls? What are they?

There are six major categories in types of system calls. They are:

1. Process Control,
2. File manipulation,
3. Device manipulation,
4. Information maintenance,
5. Communications,
6. Protection.

Types of system call (cont ...)

1. Process Control

- End, abort
- Load, execute
- Create process, terminate process
- Get process attributes, set process attributes
- Wait for time
- Wait event, signal event
- Allocate and free memory

2. File management

- Create file, delete file
- Open, close
- Read, write, reposition
- Get file attributes, set file attributes

3. Device management

- Request device, release device
- Read, write, reposition
- Get device attributes, set device attributes
- Logically attach or detach devices

4. Information maintenance

- Get time or date, set time or date
- Get system data, set system data
- Get process file, or device attributes
- Set process file, or device attributes

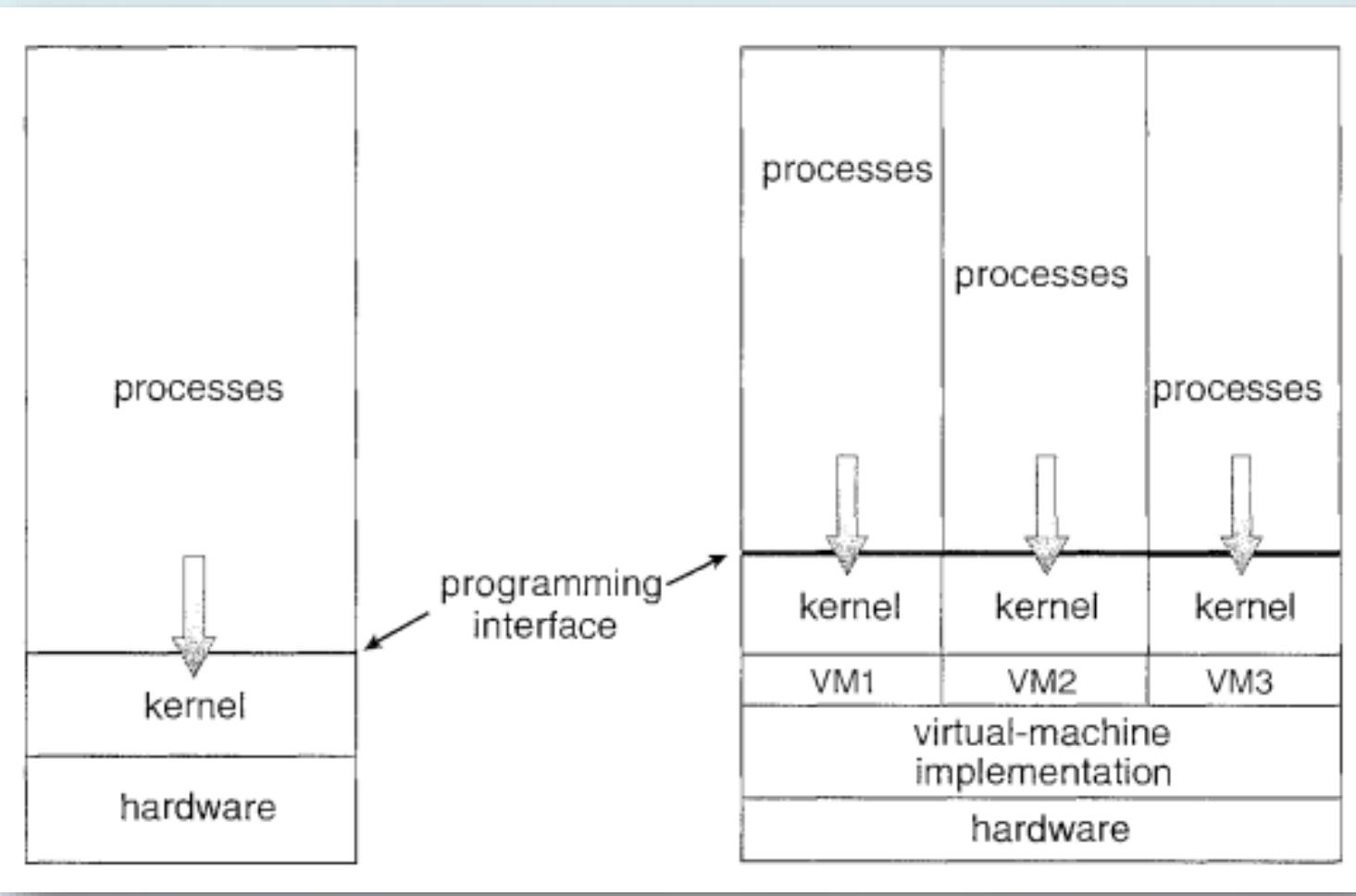
5. Communication

- Create, delete communication connection
- Send, receive messages
- Transfer status information
- Attach or detach remote devices

Q. Explain virtual machines.

Virtual machine is to abstract the hardware of a single computer(the CPU, memory, disk drives, network interface cards, and so on) into several different execution environments. It creates the illusion that each separate execution environments is running its own private computer. The virtual machine provides an interface that is identical to the underlying bare hardware. A major difficulty with the VM virtual machine approach involved disk systems. Suppose that the physical machine had three disk drives but wanted to support seven virtual machines. It could not allocate a disk drive to each virtual machine, because the virtual machine software itself needed substantial disk space to provide virtual memory and spooling. The solution was to provide virtual disks-termed minidisks in IBM's VM operating system -that are identical in all respects except size. The system implemented each minidisk by allocating as many tracks on the physical disks as the minidisk needed.

virtual machines (cont...)



(a) Non-virtual machine

(b) Virtual machine

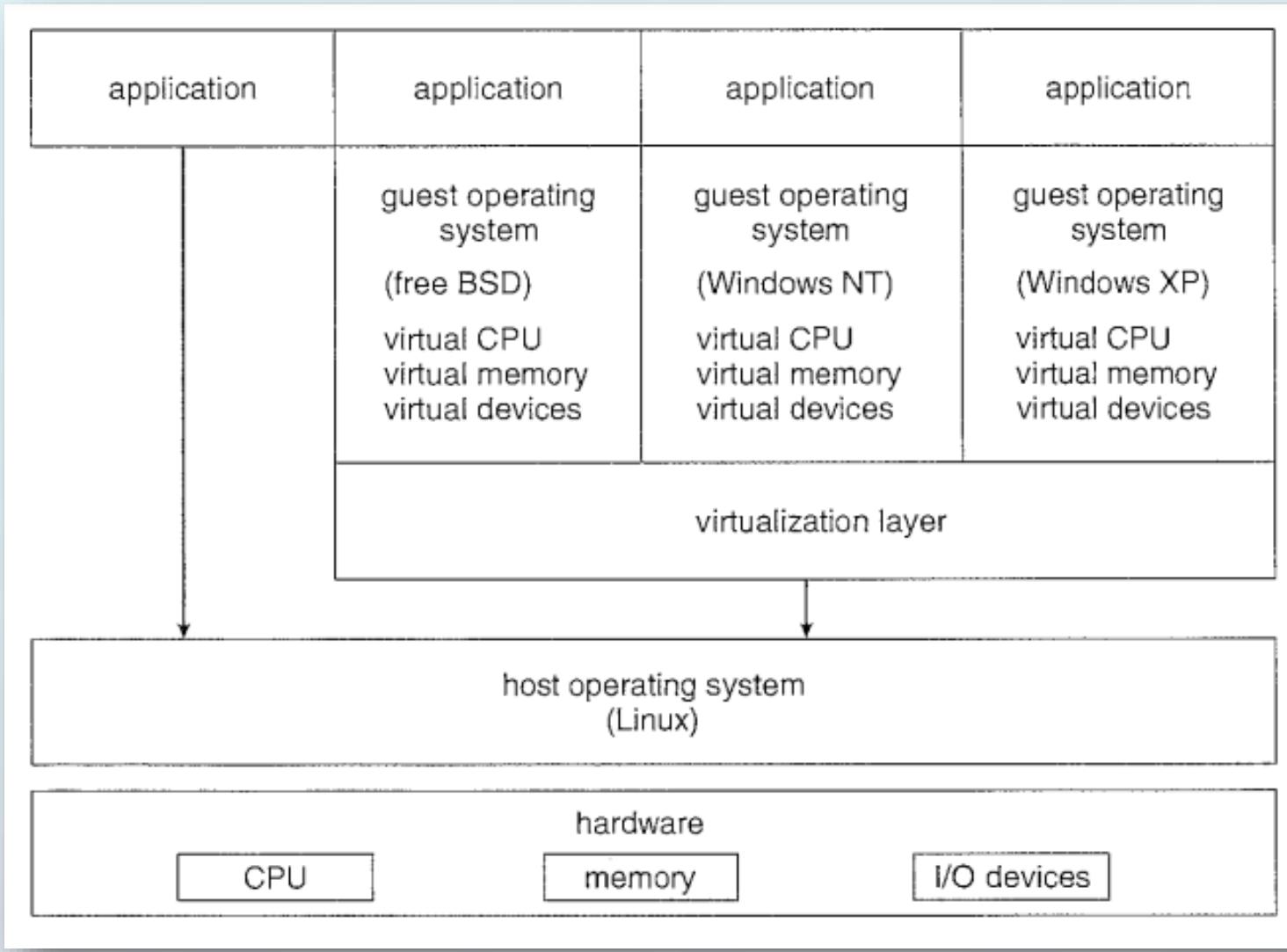
Q. Explain VMware.

VMware Workstation is a popular commercial application that abstracts Intel X86 and compatible hardware into isolated virtual machines. VMware Workstation runs as an application on a host operating system such as Windows, Linux and allows this host system to concurrently run several different guest operating systems as independent virtual machines.

The architecture of such a system is in figure. Linux is running as the host operating system; and FreeBSD, Windows NT, and Windows XP are running as guest operating system. The virtualization layer is the heart of VMware, as it abstracts the physical hardware into isolated virtual machines running as guest operating system. Each virtual machine has its own virtual CPU, memory, disk drives, network interfaces, and so forth.

The physical disk the guest owns and manages is really just a file within the file system of the host operating system. To create an identical guest instance, we can simply copy the file. Copying the file to another location protects the guest instance against a disaster at the original site. Moving the file to another location moves the guest system. These scenarios show how virtualization can improve the efficiency of system administration as well as system resource use.

VMware (cont...)



VMware Architecture