**OS**

**UNIT 1**

**OVERVIEW OF OPERATING SYSTEM**

**OS functions and characteristics**

An **Operating System** acts as a communication bridge (interface) between the user and computer hardware. The purpose of an operating system is to provide a platform on which a user can execute programs in a convenient and efficient manner.

An operating system is a piece of software that manages the allocation of computer hardware. The coordination of the hardware must be appropriate to ensure the correct working of the computer system and to prevent user programs from interfering with the proper working of the system. Example: Just like a boss gives orders to his employee, in a similar way we request or pass our orders to the Operating System. The main goal of the Operating System is to thus make the computer environment more convenient to use and the secondary goal is to use the resources in the most efficient manner.

**What is an** **Operating System?**

An operating system is a program on which application programs are executed and acts as a communication bridge (interface) between the user and the computer hardware.

The main task an operating system carries out is the allocation of resources and services, such as the allocation of memory, devices, processors, and information. The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

**Important functions of an operating System:**

1. Security – The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.
2. Control over system performance – Monitors overall system health to help improve performance. records the response time between service requests and system response to having a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.
3. Job accounting – Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.
4. Error detecting aids – The operating system constantly monitors the system to detect errors and avoid the malfunctioning of a computer system.
5. Coordination between other software and users – Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.
6. Memory Management – The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory.
7. Processor Management – In a multi-programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling. An Operating System performs the following activities for processor management.

Keeps track of the status of processes. The program which performs this task is known as a traffic controller. Allocates the CPU that is a processor to a process. De-allocates processor when a process is no more required.

1. Device Management – An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps track of all devices connected to the system. designates a program responsible for every device known as the Input/Output controller. Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.
2. File Management – A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file, and more… These facilities are collectively known as the file system.

**The Operating System provides certain services** to the users which can be listed in the following manner:

1. Program Execution: The Operating System is responsible for the execution of all types of programs whether it be user programs or system programs. The Operating System utilizes various resources available for the efficient running of all types of functionalities.
2. Handling Input/Output Operations: The Operating System is responsible for handling all sorts of inputs, i.e, from the keyboard, mouse, desktop, etc. The Operating System does all interfacing in the most appropriate manner regarding all kinds of Inputs and Outputs.   
   For example, there is a difference in the nature of all types of peripheral devices such as mice or keyboards, the Operating System is responsible for handling data between them.
3. Manipulation of File System: The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.
4. Error Detection and Handling: The Operating System is responsible for the detection of any type of error or bugs that can occur while any task. The well-secured OS sometimes also acts as a countermeasure for preventing any sort of breach to the Computer System from any external source and probably handling them.
5. Resource Allocation: The Operating System ensures the proper use of all the resources available by deciding which resource to be used by whom for how much time. All the decisions are taken by the Operating System.
6. Accounting: The Operating System tracks an account of all the functionalities taking place in the computer system at a time. All the details such as the types of errors that occurred are recorded by the Operating System.
7. Information and Resource Protection: The Operating System is responsible for using all the information and resources available on the machine in the most protected way. The Operating System must foil an attempt from any external resource to hamper any sort of data or information.

**Characteristics of Operating System**

* Memory Management — It keeps tracks of primary memory i.e what part of it are in use by whom, what part are not in use etc. Allocates the memory when the process or program request it.
* Processor Management — Allocate the processor(CPU) to a process. Deallocate processor when processor is no longer required.
* Device Management — Keep tracks of all devices. This is also called I/O controller. Decides which process gets the device when and for how much time.
* File Management — Allocates the resources. De-allocates the resource. Decides who gets the resources.
* Security — By means of passwords & similar other techniques, preventing unauthorized access to programs & data.
* Error-detecting aids — Production of dumps, traces, error messages and other debugging and error-detecting methods.

Historical evolution of operating systems

The first operating systems were developed in the 1950s, when computers could only run one program at a time. Later in the following decades, computers began to include more and more software programs, sometimes called libraries, that came together to create the start of today's operating systems

**Categories**

* [Linux](http://lloogg.com/category/linux/)
* [Mac](http://lloogg.com/category/mac/)
* [Site Data](http://lloogg.com/category/site-data/)
* [System](http://lloogg.com/category/system/)
* [Windows PC](http://lloogg.com/category/windows-pc/)
* The first computers did not have operating systems. Each program that was running on these first computers had to include all the code needed to run on the computer, communicate with the connected hardware and perform the calculation that the program was intended to perform. This situation made even the simplest programs become very complex.
* In response to this problem, the owners of the central computers began to develop system software that facilitated the writing and execution of the programs included in the computer, and thus the first operating systems were born.
* The first operating system was created by General Motors in 1956 to run a single IBM central computer. In the 1960s, IBM was the first computer manufacturer to take on the task of developing operating systems and began distributing operating systems included in its computers.
* The first operating systems were developed in the 1950s, when computers could only run one program at a time. Later in the following decades, computers began to include more and more software programs, sometimes called libraries, that came together to create the start of today’s operating systems.
* In the late 1960s, the first version of the Unix operating system was developed. Written in programming language C, and available for free during its early years. Unix easily adapted to the new systems and quickly achieved wide acceptance.
* Many modern operating systems, including Apple OS X and all different versions of Linux, date back or rely on the Unix OS.
* Microsoft Windows was developed in response to an IBM request for an operating system to run its range of personal computers or PCs.
* The first operating system created by Microsoft was not called Windows, it was called MS-DOS and it was built in 1981 when it bought the 86-DOS operating system from Seattle Computer Products and modified it to meet IBM requirements.
* The Windows name was first used in 1985 when a graphical user interface was created and paired or joined with the MS-DOS.
* Today Apple, OS X, Microsoft Windows and the various forms of Linux (including Android) dominate the vast majority of the modern operating systems market, as we saw earlier.
* Video Operating Systems
* Here is a video that explains very well everything about operating systems in a very simple way and which OS to choose. Once you have studied the operating system, we recommend you do the following exercise in the form of a Test: Test Operating Systems.
* Most software programs are designed to work with the operating system of a single company, for example only Windows (Microsoft) or only macOS (Apple).
* A software will clearly indicate what operating systems it supports and will be very specific if necessary including the version or versions of that OS it supports. For example, a video production software program might say that it is compatible with Windows 10, Windows 8 and Windows 7, but is not compatible with earlier versions of Windows such as Windows Vista and XP.
* Software developers also usually release additional versions of their software that work with other operating systems or different versions. Returning to the example of the video production program, that company could also launch another version of the program with exactly the same features but to work with macOS, the place with Windows.
* It is also important to know if your operating system is 32-bit or 64-bit. It’s a common question that they ask you when downloading software Below you can see how to know if your computer is 32bit or 64bit in Windows.

**Issues in operating system design**

**Design Goals in Operating Systems**

* Concurrent Systems. Modern operating systems should be able to handle multiple users as well as multiple devices at the same time.
* Security and Privacy.
* Resource Sharing.
* Future Hardware and Software Changes.
* Portable Operating Systems.
* Backward Compatibility.
* No Specific Type of Users.

**What are operating system design issues?**

The important issues related to Operating system are transparency, flexibility, reliability, performance, scalability, naming, replication, synchronization, security. Let us understand the Different Types of Advanced Operating Systems to understand the Major Design Issues better.

A design issue often represents a challenging problem through the operational requirements and capabilities of a robotic system. A design issue is usually a very important item of the design process that will lead to specific design solutions.

What are design issues of mobile operating system?

Explanation of some of the main problems identified : It's performance is slow due to bad memory management. Background running apps. Internal storage is full which slows down the mobile performance. Cybercriminals have increasingly targeted Android-based and other mobile devices with SMS Trojans, Malwares, and Viruses

**User’s view of the OS**

The user viewpoint focuses on how the user interacts with the operating system through the usage of various application programs. In contrast, the system viewpoint focuses on how the hardware interacts with the operating system to complete various tasks

**1. Single User View Point**

Most computer users use a monitor, keyboard, mouse, printer, and other accessories to operate their computer system. In some cases, the system is designed to maximize the output of a single user. As a result, more attention is laid on accessibility, and resource allocation is less important. These systems are much more designed for a single user experience and meet the needs of a single user, where the performance is not given focus as the multiple user systems.

**2. Multiple User View Point**

Another example of user views in which the importance of user experience and performance is given is when there is one mainframe computer and many users on their computers trying to interact with their kernels over the mainframe to each other. In such circumstances, memory allocation by the CPU must be done effectively to give a good user experience. The client-server architecture is another good example where many clients may interact through a remote server, and the same constraints of effective use of server resources may arise.

**3. Handled User View Point**

Moreover, the touchscreen era has given you the best handheld technology ever. Smartphones interact via wireless devices to perform numerous operations, but they're not as efficient as a computer interface, limiting their usefulness. However, their operating system is a great example of creating a device focused on the user's point of view.

**4. Embedded System User View Point**

Some systems, like embedded systems that lack a user point of view. The remote control used to turn on or off the TV is all part of an embedded system in which the electronic device communicates with another program where the user viewpoint is limited and allows the user to engage with the application.

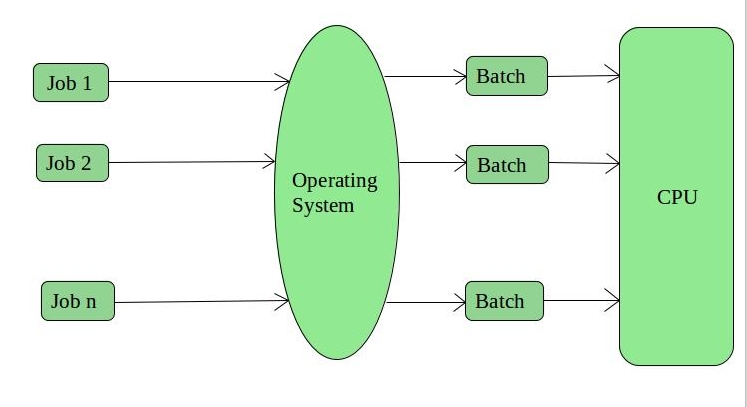
**Types of OS :**

An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. resource manager. Thus, the operating system becomes an interface between user and machine.

Types of Operating Systems: Some widely used operating systems are as follows-

**1. Batch Operating System –**

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.



**Advantages of Batch Operating System:**

* It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
* Multiple users can share the batch systems
* The idle time for the batch system is very less
* It is easy to manage large work repeatedly in batch systems

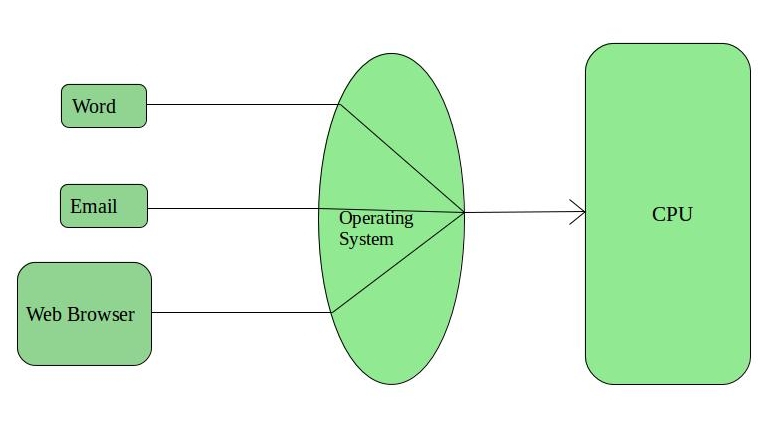
**Disadvantages of Batch Operating System:**

* The computer operators should be well known with batch systems
* Batch systems are hard to debug
* It is sometimes costly
* The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

**2. Time-Sharing Operating Systems –**

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



**Advantages of Time-Sharing OS:**

* Each task gets an equal opportunity
* Fewer chances of duplication of software
* CPU idle time can be reduced

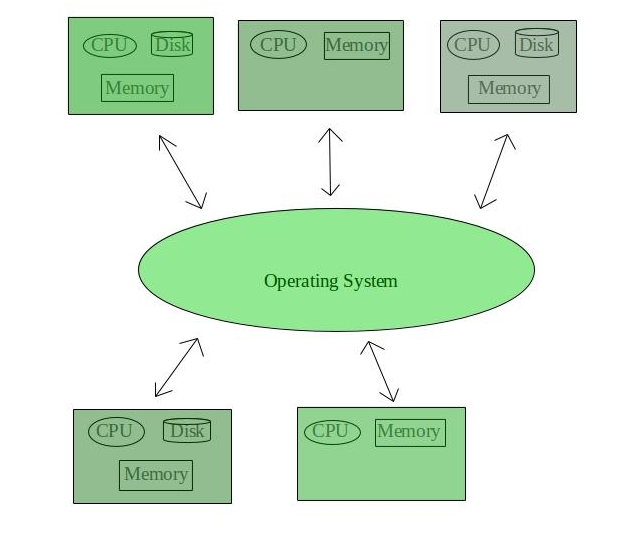
**Disadvantages of Time-Sharing OS:**

* Reliability problem
* One must have to take care of the security and integrity of user programs and data
* Data communication problem

Examples of Time-Sharing OSs are: Multics, Unix, etc.

**3. Distributed Operating System –**

These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as loosely coupled systems or distributed systems. These system’s processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network. 



**Advantages of Distributed Operating System:**

* Failure of one will not affect the other network communication, as all systems are independent from each other
* Electronic mail increases the data exchange speed
* Since resources are being shared, computation is highly fast and durable
* Load on host computer reduces
* These systems are easily scalable as many systems can be easily added to the network
* Delay in data processing reduces

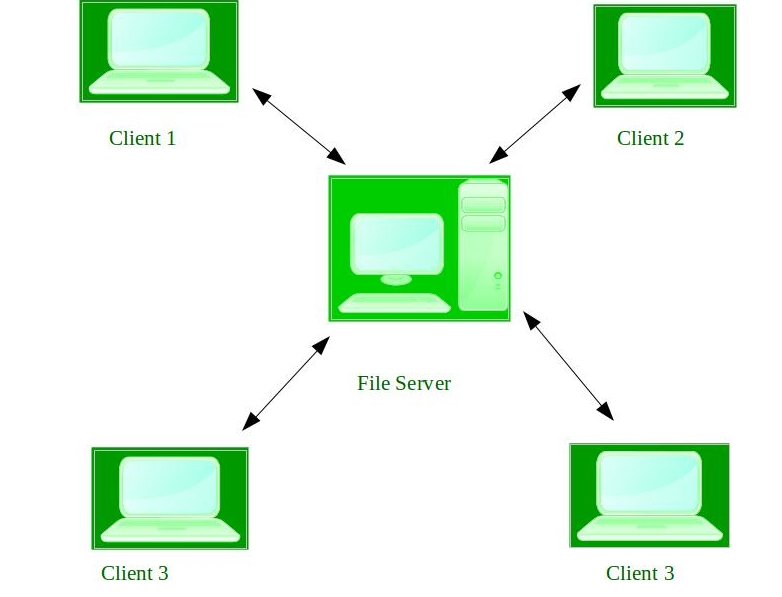
**Disadvantages of Distributed Operating System:**

* Failure of the main network will stop the entire communication
* To establish distributed systems the language which is used are not well defined yet
* These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples of Distributed Operating System are- LOCUS, etc.

**4. Network Operating System –**

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that’s why these computers are popularly known as tightly coupled systems.



**Advantages of Network Operating System:**

* Highly stable centralized servers
* Security concerns are handled through servers
* New technologies and hardware up-gradation are easily integrated into the system
* Server access is possible remotely from different locations and types of systems

**Disadvantages of Network Operating System:**

* Servers are costly
* User has to depend on a central location for most operations
* Maintenance and updates are required regularly

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

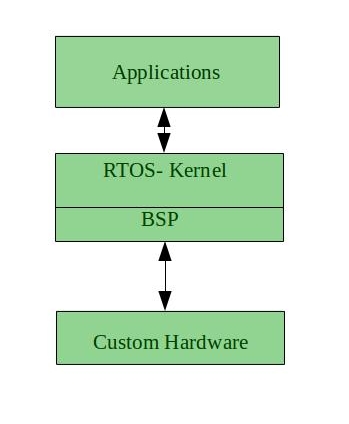
**5. Real-Time Operating System –**

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time.

Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Two types of Real-Time Operating System which are as follows:

* Hard Real-Time Systems: These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.
* Soft Real-Time Systems: These OSs are for applications where for time-constraint is less strict.



**Advantages of RTOS:**

* Maximum Consumption: Maximum utilization of devices and system, thus more output from all the resources
* Task Shifting: The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
* Focus on Application: Focus on running applications and less importance to applications which are in the queue.
* Real-time operating system in the embedded system: Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
* Error Free: These types of systems are error-free.
* Memory Allocation: Memory allocation is best managed in these types of systems.

**Disadvantages of RTOS:**

* Limited Tasks: Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
* Use heavy system resources: Sometimes the system resources are not so good and they are expensive as well.
* Complex Algorithms: The algorithms are very complex and difficult for the designer to write on.
* Device driver and interrupt signals: It needs specific device drivers and interrupts signals to respond earliest to interrupts.
* Thread Priority: It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

**What is a system call ?**

A system call is a request for service that a program makes of the kernel. The service is generally something that only the kernel has the privilege to do, such as doing I/O

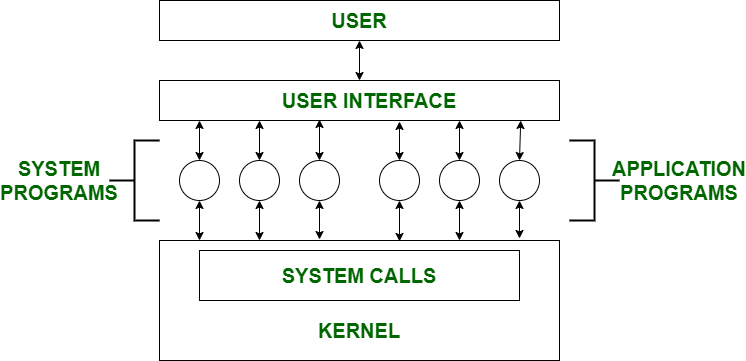
What are the various types of system calls?

Types of System Calls

* Process Control. These system calls deal with processes such as process creation, process termination etc.
* File Management.
* Device Management.
* Information Maintenance.
* Communication.

**System programs**

System Programming can be defined as the act of building Systems Software using System Programming Languages. According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs. Some of the System Programs are simply user interfaces, others are complex. It traditionally lies between the user interface and system calls.



So here, the user can only view up-to-the System Programs he can’t see System Calls. System Programs can be divided into these categories:

1. File Management – A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in the computer system, its management includes the process of creating, modifying and deleting files.
   * It helps to create new files in the computer system and placing them at specific locations.
   * It helps in easily and quickly locating these files in the computer system.
   * It makes the process of sharing files among different users very easy and user-friendly.
   * It helps to store files in separate folders known as directories.
   * These directories help users to search files quickly or to manage files according to their types of uses.
   * It helps users to modify the data of files or to modify the name of files in directories.
2. Status Information – Information like date, time amount of available memory, or disk space is asked by some users. Others providing detailed performance, logging, and debugging information which is more complex. All this information is formatted and displayed on output devices or printed. Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

1. File Modification – For modifying the contents of files we use this. For Files stored on disks or other storage devices, we used different types of editors. For searching contents of files or perform transformations of files we use special commands.

1. Programming-Language support – For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users. It provides all support to users. We can run any programming language. All languages of importance are already provided.

1. Program Loading and Execution – When the program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program. Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system.

1. Communications – Virtual connections among processes, users, and computer systems are provided by programs. Users can send messages to another user on their screen, User can send e-mail, browsing on web pages, remote login, the transformation of files from one user to another.

Examples

* Android
* Anti-virus
* Disk formatting
* Computer language translators

**Bash shell scripting : commands and shell as a scripting language**

Bash is a command language interpreter. It is widely available on various operating systems and is a default command interpreter on most GNU/Linux systems.

Bash Shell Scripting : It is widely available on various operating systems and is a default command interpreter on most GNU/Linux systems. The name is an acronym for the 'Bourne-Again SHell'. Shell is a macro processor which allows for an interactive or non-interactive command execution

**Bash shell commands**

* print the current working directory ( pwd )
* navigate between directories on your computer ( cd )
* create new directories ( mkdir )
* print a list of files and subdirectories within directories ( ls )
* delete files ( rm ) and directories ( rm -r )

**Scripting**

Scripting allows for an automatic commands execution that would otherwise be executed interactively one-by-one.

A shell script is a computer program designed to be run by the Unix shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. Typical operations performed by shell scripts include file manipulation, program execution, and printing text

Common scripting languages include: Shell scripts - sh, bash, csh, tcsh. Other scripting languages - TCL, Perl, Python.