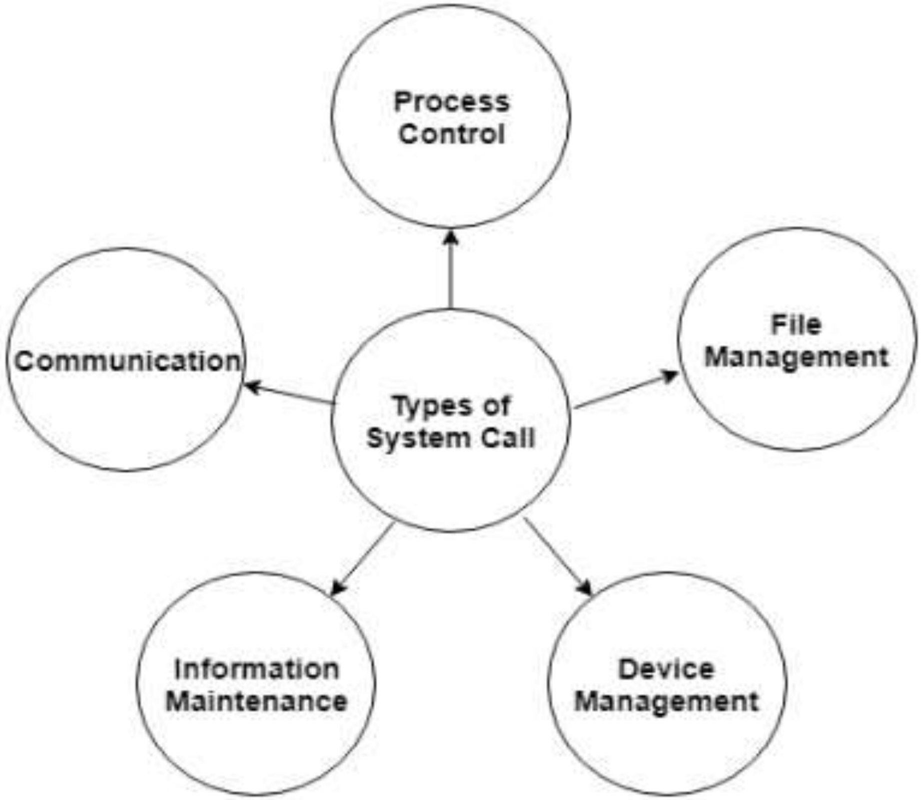
System call

- In computing, a **system call** is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on.
- (The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system. The kernel facilitates interactions between hardware and software components.)
- A system call is a way for programs to **interact with the operating system**. A computer program makes a system call when it makes a request to the operating system’s kernel. System call **provides** the services of the operating system to the user programs via Application Program Interface(API).
- It provides an interface between a process and operating system to allow user-level processes to request services of the operating system. System calls are the only entry points into the kernel system. All programs needing resources must use system calls.

- **Services Provided by System Calls :**
- Process creation and management
- Main memory management
- File Access, Directory and File system management
- Device handling(I/O)
- Protection
- Networking, etc.

Types of system call:-

- There are mainly five types of system calls. These are explained in detail as follows:



- **Process Control**
- These system calls deal with processes such as process creation, process termination etc.
- **File Management**
- These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.
- **Device Management**
- These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.
- **Information Maintenance**
- These system calls handle information and its transfer between the operating system and the user program.
- **Communication**
- These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

- **1) Process and job control**
- The category includes the system call to end or abort the running program, to load and execute the program, to create new process or terminate the existing one, to get the process attributes(quality) and to set them. Another set of the system calls are helpful in debugging a program and to dump the memory.
- **Process Control**
- 1. End, Abort
- 2. Load, Execute
- 3. Create Process, Terminate Process
- 4. Get and Set Process attributes

- **2) File manipulation**
- Systems calls are required to read and delete the file, to open them and to close them. In order to perform the read, write and reposition operations we need the system calls. To read and determine the attributes of the files we need system calls.
- **File manipulation**
- 1. Create File, Delete file
- 2. Open and close file
- 3. Read, write and reposition
- 4. Get and set file attributes

- **3) Device management**
- In order to use a device, we first request the device, after using it we have to release it. Once the device has been requested we can read, write and reposition the device.
- **Device manipulation**
- 1. Request and release the devices
- 2. Read, write and reposition
- 3. Get and set Device attributes

- **4) Information maintenance**
- Many system calls exist(happen/go/live) for the purpose of transferring information between the user program and operating system such as a call to return the current time and date.
- **Information Maintenance**
- 1. Get/Set time or date
- 2. Get/Set system date
- 3. Get/Set process/file/device attributes

Eg.	Windows	linux

Process Control	CreateProcess()
	ExitProcess()
	WaitForSingleObject()
		fork()
		exit()
		wait()

File Manipulation	CreateFile()
	ReadFile()
	WriteFile()
	CloseHandle()
		open()
		read()
		write()
		close()

Device Manipulation	SetConsoleMode()
	ReadConsole()
	WriteConsole()
		ioctl()
		read()
		write()

Information Maintenance	GetCurrentProcessID()
	SetTimer()
	Sleep()
		getpid()
		alarm()
		sleep()

Communication	CreatePipe()
	CreateFileMapping()
	MapViewOfFile()
		pipe()
		shmget()
		mmap()

Protection	SetFileSecurity()
	InitlializeSecurityDescriptor()
	SetSecurityDescriptorGroup()
		chmod()
		umask()
		chown()

1.Process Control:-

- A process or the job which is currently run in the system always want to load and execute another program. The command interpreter executes a program for example when the user clicks on the mouse button.
- (A command interpreter is the part of a computer OS that understands and executes commands that are entered interactively by a human being or from a program. In some OS System the Command Interpreter is called the shell.)
- Now the question arises that when the loaded program terminates when the control return. When any new program terminates in the system the memory the currently running program must be saved.

- So in this way we can efficiently create a program which calls another program and both are running concurrently.
- For the multiprogramming, we have created jobs or process. So the system calls for this purpose is called create a process or submit the job. When a new job, process or the group of the process is created then we have to manage the execution of these process.

2. File management

- In the file management system, we should be able to create or delete the files. For creating a file the system call requires the names of the file and some attribute of the files.
- When the file is created we should be able to open and use these files. We can also read-write or reposition the files. After all the work is done we need to close the file which indicates that the file is no longer need.

- We can also use these operations for directories also. In the file management system, we should be able to determine the values of various attributes for the file and directories.
- There are various file attributes like name, types, size, location, accounting information and so on. The two important system calls get attribute and file set attribute is required for this function.

3. Device management

- When the program is running in the system it needs some resources such as memory, input, output, and so on. The resources are only granted when they are available and not held by another process.
- If the resources are not free the program has to wait for the resources. Files are the abstract or virtual devices. So there are many system calls which are needed for the files and also for the devices.

- If there are multiple users then first it requests for the devices and after finishing the work with these devices we should release it. When the devices are allocated to the user then we can perform different operations like reading, write and execute.

4. Information management

- In information management, various system calls are used to communicate the information between the operating system and user program.
- For example, some system has a system call forget the current time or date.
- There are many another system call like for the number of the user, different versions of operating system., free space in memory etc.

5. Communication

- There are mainly two models for communication. First is the message passing model in this the information is exchanged between the process and this communication facility is provided by the operating system.
- When the process wants to communicate first the other communicator should be know so that the communication is established and they can communicate.
- Second is the shared memory model in this model different process shared memory and communicate with each other.