**Lab Assignment 4**

**Aim:**  
To study and learn about various system calls.

**1. Process Management System Calls**

**a) fork()**

* Used to create a new process.
* The new process is a copy of the current (parent) process and is called the child process.
* Returns:
  + 0 to the child process.
  + PID of child to the parent process.
  + -1 on failure.

**b) exec()**

* Replaces the current process image with a new process image.
* Used after fork() to execute a new program in the child process.
* There are different variants like execl(), execp(), execv(), etc.

**c) wait()**

* Makes the parent process wait until all of its child processes have terminated.
* Returns the process ID of the terminated child process.

**d) exit()**

* Terminates the calling process.
* Returns a status code to the parent process (if wait() is used).
* Syntax: exit(status\_code);

**2. File Management System Calls**

**a) open()**

* Used to open a file and return a file descriptor.
* Syntax: int fd = open("filename", O\_RDONLY);

**b) read()**

* Reads data from a file into a buffer.
* Syntax: read(fd, buffer, size);

**c) write()**

* Writes data from a buffer to a file.
* Syntax: write(fd, buffer, size);

**d) close()**

* Closes an open file descriptor.
* Syntax: close(fd);

**3. Device Management System Calls**

**a) read()**

* Reads data from an input device or file descriptor.

**b) write()**

* Writes data to an output device or file descriptor.

**c) ioctl()**

* Performs device-specific input/output operations.
* Used to control hardware devices.
* Syntax: ioctl(fd, command, &arg);

**d) select()**

* Monitors multiple file descriptors to see if they are ready for I/O.
* Useful for non-blocking I/O operations.

**4. Network Management System Calls**

**a) socket()**

* Creates an endpoint for communication and returns a socket descriptor.
* Syntax: int sockfd = socket(domain, type, protocol);

**b) connect()**

* Used by client to connect to a server socket.
* Syntax: connect(sockfd, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr));

**c) send()**

* Sends data over a connected socket.
* Syntax: send(sockfd, buffer, len, flags);

**d) recv()**

* Receives data from a connected socket.
* Syntax: recv(sockfd, buffer, len, flags);

**5. System Information Management System Calls**

**a) getpid()**

* Returns the process ID of the calling process.

**b) getuid()**

* Returns the real user ID of the calling process.

**c) gethostname()**

* Retrieves the standard host name for the current machine.
* Syntax: gethostname(buffer, size);

**d) sysinfo()**

* Provides system information like uptime, load average, total/free memory, etc.
* Syntax: sysinfo(struct sysinfo \*info);