## Lab Assignment 4

## Operating System Lab

## 4ITRC2|| Sweta Ahirwar || Roll No. 23I4073

### Aim:

To study and learn about various system calls.

### Objective:

Comprehensive study of different categories of Linux system calls.

---

### 1. Process Management System Calls

#### fork()

* - Creates a new process by duplicating the current process.
* - Returns 0 in the child process and child's PID in the parent.

#### exec()

* - Replaces the current process image with a new process image.
* - Common variants: execl(), execp(), execv(), etc.

#### wait()

* - Makes the parent process wait until any one of its child processes finishes execution.

#### exit()

* - Terminates the calling process.
* - Status code is returned to the parent process.

### 2. File Management System Calls

#### open()

* - Opens a file and returns a file descriptor.
* - Syntax: int open(const char \*pathname, int flags);

#### read()

* - Reads data from a file descriptor.
* - Syntax: ssize\_t read(int fd, void \*buf, size\_t count);

#### write()

* - Writes data to a file descriptor.
* - Syntax: ssize\_t write(int fd, const void \*buf, size\_t count);

#### close()

* - Closes an opened file descriptor.

### 3. Device Management System Calls

#### read() and write()

* - Same as file read/write. Used for I/O operations on devices too.

#### ioctl()

* - Used to manipulate underlying device parameters.
* - Syntax: int ioctl(int fd, unsigned long request, ...);

#### select()

* - Monitors multiple file descriptors.
* - Syntax: int select(int nfds, fd\_set \*readfds, fd\_set \*writefds, fd\_set \*exceptfds, struct timeval \*timeout);

### 4. Network Management System Calls

#### socket()

* - Creates an endpoint for communication.
* - Syntax: int socket(int domain, int type, int protocol);

#### connect()

* - Initiates a connection on a socket.
* - Syntax: int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

#### send()

* - Sends data on a socket.
* - Syntax: ssize\_t send(int sockfd, const void \*buf, size\_t len, int flags);

#### recv()

* - Receives data from a socket.
* - Syntax: ssize\_t recv(int sockfd, void \*buf, size\_t len, int flags);

### 5. System Information Management System Calls

#### getpid()

* - Returns the process ID of the calling process.

#### getuid()

* - Returns the user ID of the calling process.

#### gethostname()

* - Gets the name of the current host.
* - Syntax: int gethostname(char \*name, size\_t len);

#### sysinfo()

* - Retrieves overall system statistics.
* - Syntax: int sysinfo(struct sysinfo \*info);

---

### Conclusion:

This lab involved exploring different Linux system calls categorized under process, file, device, network, and system information management. Each of these system calls is fundamental for developing low-level applications and interacting directly with the Linux kernel.