**Lab assignment 4**

**1. Process Management System Calls**

These system calls are used to create, execute, and manage processes.

**fork():**

Creates a new child process identical to the parent.

pid\_t pid = fork();

**exec():**

Replaces the current process with a new program.

execl("/bin/ls", "ls", "-l", NULL);

**wait():**

Waits for a child process to terminate.

wait(NULL);

**exit():**

Terminates the process.

exit(0);

**2. File Management System Calls**

Used for handling file operations like open, read, write, and close.

**open():**

Opens a file and returns a file descriptor.

int fd = open("file.txt", O\_RDONLY);

**read():**

Reads data from a file.

read(fd, buffer, sizeof(buffer));

**write():**

Writes data to a file.

write(fd, "Hello", 5);

**close():**

Closes an opened file.

close(fd);

**3. Device Management System Calls**

Used for interacting with hardware devices.

**read() / write():**

Same as file management but used with device files (e.g., /dev/sda).

**ioctl():**

Device-specific input/output control operations.

ioctl(fd, command, &arg);

**select():**

Monitors multiple file descriptors to see if I/O is possible.

select(nfds, &readfds, NULL, NULL, &timeout);

**4. Network Management System Calls**

Used for creating and managing network connections.

**socket():**

Creates a network socket.

int sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

**connect():**

Connects the socket to a remote server.

connect(sockfd, (struct sockaddr\*)&server, sizeof(server));

**send():**

Sends data over the socket.

send(sockfd, msg, strlen(msg), 0);

**recv():**

Receives data from the socket.

recv(sockfd, buffer, sizeof(buffer), 0);

**5. System Information Management System Calls**

Used to retrieve system-related information.

**getpid():**

Returns the process ID.

pid\_t pid = getpid();

**getuid():**

Returns the user ID.

uid\_t uid = getuid();

**gethostname():**

Retrieves the system's hostname.

gethostname(name, size);

**sysinfo():**

Retrieves system statistics.

struct sysinfo info;

sysinfo(&info);