4ITRC2 Operating system Lab

Lab Assignment 4

Name: Pawan Kasde

Roll Number: 23I4051

Aim : To study and learn about various system calls

To perform: COmprehensive study of different categories of Linus system calls,catergorized as

---

1. Process Management System Calls

fork(): Used to create a new process by duplicating the current process. The new process is called the child process.

```
pid_t pid = fork();
```

exec(): Replaces the current process image with a new process image.

```
execl("/bin/ls", "ls", NULL);
```

wait(): Makes the parent process wait until all of its child processes have terminated.

```
int status;
wait(&status);
```

exit(): Terminates the calling process.

exit(0);

---

2. File Management System Calls

open(): Opens a file and returns a file descriptor.

int fd = open("file.txt", O_RDONLY);

read(): Reads data from a file descriptor.

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

write(): Writes data to a file descriptor.

write(fd, buffer, strlen(buffer));

close(): Closes an opened file descriptor.

close(fd);

---

3. Device Management System Calls

read(): Reads data from a device file.

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

write(): Writes data to a device file.

write(fd, buffer, strlen(buffer));

ioctl(): Device-specific input/output operations.

ioctl(fd, COMMAND, &arg);

select(): Monitors multiple file descriptors to see if any of them is ready for I/O.

select(nfds, &readfds, &writefds, &exceptfds, &timeout);

---

4. Network Management System Calls

socket(): Creates an endpoint for communication.

int sockfd = socket(AF_INET, SOCK_STREAM, 0);

connect(): Initiates a connection on a socket.

connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr));

send(): Sends data on a socket.

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

recv(): Receives data from a socket.

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

---

5. System Information Management System Calls

getpid(): Gets the process ID of the calling process.

```
pid_t pid = getpid();
```

getuid(): Gets the real user ID of the calling process.

```
uid_t uid = getuid();
```

gethostname(): Gets the standard host name for the current machine.

```
char hostname[1024];
gethostname(hostname, sizeof(hostname));
```

sysinfo(): Retrieves overall system statistics.

```
struct sysinfo info;
sysinfo(&info);
```