OPERATING SYSTEM(4ITRC2)

IT IV(Semester)

Submitted by

Name of Student – Akanksha Singh

Roll Number – 23I4106

Branch with Section – IT(B)

Submitted to

MRS_JASNEET KAUR

Assistant Professor, Department of Information Technology

Institute of Engineering and Technology

Devi Ahilya Vishwavidhyalaya, Indore (M.P.) India

(www.iet.dauniv.ac.in**)**

Session JAN-APRIL,2025

Exhaustive Study of System Call

1. Process Management System Calls

These system calls are responsible for creating, managing, and terminating processes.

a. `fork()`

- **Purpose**: Creates a new process by duplicating the calling process.
- **Example**:

```
CopyEdit
pid_t pid = fork();
if (pid == 0) {
printf("Child process\n");
} else {
printf("Parent process\n");
}
```

b. `exec()`

- **Purpose**: Replaces the current process image with a new program.
- **Example**:

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

c. `wait()`

- **Purpose**: Suspends the execution of the calling process until one of its child processes terminates.
- **Example**:

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

d. `exit()`

- **Purpose**: Terminates the calling process and returns a status code to the parent.
- **Example**:

```
CopyEdit
exit(0);
```

---

2. File Management System Calls

These calls allow user programs to interact with the file system—opening, reading, writing, and closing files.

a. `open()`

- **Purpose**: Opens a file and returns a file descriptor.
- **Example**:

```
CopyEdit
int fd = open("example.txt", O_RDONLY);
```

b. `read()`

- **Purpose**: Reads data from a file descriptor into a buffer.
- **Example**:

```
CopyEdit
char buffer[100];
read(fd, buffer, sizeof(buffer));
```

c. `write()`

- **Purpose**: Writes data from a buffer to a file descriptor.
- **Example**:

```
CopyEdit
write(fd, "Hello, World!", 13);
```

d. `close()`

- **Purpose**: Closes an opened file descriptor.
- **Example**:

```
CopyEdit
close(fd);
```

---

3. Device Management System Calls

These system calls are used to perform operations on device files, which are interfaces to hardware devices.

a. `read()`

- **Purpose**: Similar to file read, but used for reading from devices.
- **Example**:

```
CopyEdit
read(device_fd, buffer, sizeof(buffer));
```

b. `write()`

- **Purpose**: Sends data to a device.
- **Example**:

```
CopyEdit
write(device_fd, data, data_length);
```

c. `ioctl()`

- **Purpose**: Performs device-specific input/output operations.
- **Example**:

```
CopyEdit
ioctl(device_fd, SOME_IOCTL_COMMAND, &arg);
```

d. `select()`

- **Purpose**: Monitors multiple file descriptors, waiting until one or more become "ready".
- **Example**:

```
CopyEdit
fd_set readfds;
FD_ZERO(&readfds);
FD_SET(device_fd, &readfds);
select(device_fd + 1, &readfds, NULL, NULL, NULL);
```

---

4. Network Management System Calls

These are crucial for communication over networks, supporting socket-based communication in client-server models.

a. `socket()`

- **Purpose**: Creates a socket for network communication.
- **Example**:

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

b. `connect()`

- **Purpose**: Connects a socket to a remote address.
- **Example**:

```
CopyEdit
connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
```

c. `send()`

- **Purpose**: Sends data through a socket.
- **Example**:

```
CopyEdit
send(sockfd, "Hello", strlen("Hello"), 0);
```

d. `recv()`

- **Purpose**: Receives data from a socket.
- **Example**:

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

---

5. System Information Management System Calls

These calls allow access to system-level information such as process IDs, user IDs, hostname, and more.

a. `getpid()`

- **Purpose**: Returns the process ID of the calling process.

- **Example**:

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

b. `getuid()`

- **Purpose**: Returns the user ID of the calling process.
- **Example**:

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

c. `gethostname()`

- **Purpose**: Retrieves the hostname of the machine.
- **Example**:

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

d. `sysinfo()`

- **Purpose**: Provides information about the system like uptime, total RAM, etc.
- **Example**:

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