Operating Systems

Assignment 4

Name: Aryan Naik **Div:** AIDS-A

Roll No: 10 **PRN:** 12110057

Year: TY **Subject:** Operating Systems

Aim:

Write a program demonstrating use of different system calls.

1) process related system all:fork,wait,

2) file related:open ,read,write,close

3)protection: chmod

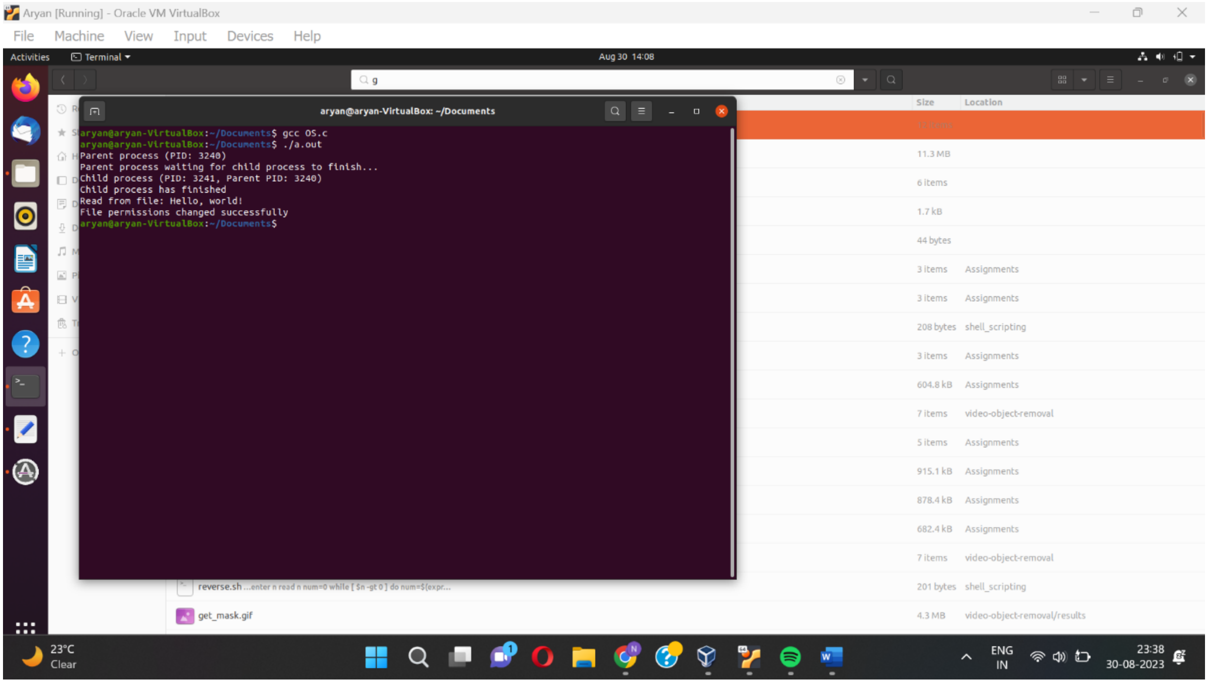
Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <sys/stat.h>
```

```
#include <fcntl.h>

int main() {
    // Process-related system calls
    pid_t child_pid;
    int status;

    printf("Parent process (PID: %d)\n", getpid());

    child_pid = fork();
    if (child_pid == -1) {
        perror("fork");
        exit(EXIT_FAILURE);
    }

    if (child_pid == 0) {
        // Child process
        printf("Child process (PID: %d, Parent PID: %d)\n", getpid(), getppid());
        exit(EXIT_SUCCESS);
    } else {
        // Parent process
        printf("Parent process waiting for child process to finish...\n");
        wait(&status);
        printf("Child process has finished\n");
```

```
}

// File-related system calls
int fd;
ssize_t bytes_read, bytes_written;
char buffer[100];

fd = open("example.txt", O_CREAT | O_WRONLY | O_TRUNC, 0644);
if (fd == -1) {
    perror("open");
    exit(EXIT_FAILURE);
}

bytes_written = write(fd, "Hello, world!\n", 14);
if (bytes_written == -1) {
    perror("write");
    exit(EXIT_FAILURE);
}
close(fd);

fd = open("example.txt", O_RDONLY);
if (fd == -1) {
    perror("open");
    exit(EXIT_FAILURE);
    }

    bytes_read = read(fd, buffer, sizeof(buffer));
    if (bytes_read == -1) {
        perror("read");
        exit(EXIT_FAILURE);
    }
    close(fd);

    printf("Read from file: %s", buffer);

    // Protection-related system call
    if (chmod("example.txt", S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH) == -1) {
        perror("chmod");
        exit(EXIT_FAILURE);
    }

    printf("File permissions changed successfully\n");

    return 0;
}
```

Output:



```
aryan@aryan-VirtualBox:~/Documents$ gcc OS.c
aryan@aryan-VirtualBox:~/Documents$ ./a.out
Parent process (PID: 3240)
Parent process waiting for child process to finish...
Child process (PID: 3241, Parent PID: 3240)
Child process has finished
Read from file: Hello, world!
File permissions changed successfully
aryan@aryan-VirtualBox:~/Documents$
```