**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:**

A computer screen shot of a computer screen

Description automatically generated