

Implementation of process management using system calls

Program

```
#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>


int main() {

    pid_t child_pid;


    // Fork a child process
    child_pid = fork();


    if (child_pid < 0) {
        // Fork failed
        perror("Fork failed");
        return 1;
    }
    else if (child_pid == 0) {
        // This is the child process
        printf("Child process: PID=%d\n", getpid());


        // Execute a new program in the child process
        execl("/bin/ls", "ls", "-l", NULL);


        // exec only returns if an error occurs
        perror("execl failed");
        return 1;
    }
    else {
```

```
// This is the parent process

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


int status;

if (wait(&status) == -1) {
    perror("wait failed");
    return 1;
}


printf("Child process finished.\n");


// If you had opened any file descriptors, you could close them here
// close(file_descriptor);
}


return 0;
}
```