Unix Process Control System calls

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid;

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

    if (pid < 0) {
        // Fork failed
        fprintf(stderr, "Fork failed\n");
        return 1;
    }
else if (pid == 0) {
        // Child process
        printf("Child process: My PID is %d\n", getpid());
        sleep(20);
```

```
        printf(“Executing the child process ….\n”);
        execlp("/bin/ls", "ls", NULL); // Example: executing 'ls' command
        // execlp failed only if the specified command doesn't exist
        perror("execlp");
        exit(1);
    }
else {
        // Parent process
        printf("Parent process: My PID is %d\n", getpid());
        printf("child process sleeping for 20 seconds....\n");
        // Wait for the child to finish
        wait(NULL);
        printf("Parent process: Child process has terminated\n");
    }
    return 0;
}
```