

```

/*
 * prog.c
 * Assignment 1 - CSI 3131 Summer 2024
 * Name: Saja Elkurehi
 * Student ID: 300288667
 */
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1
int main(void)
{
    char write_msg[BUFFER_SIZE];
    char read_msg[BUFFER_SIZE];
    int fd1[2], fd2[2];
    pid_t pid;
    /* create the pipe */
    if (pipe(fd1) == -1 || pipe(fd2) == -1)
    {
        fprintf(stderr, "Pipe failed");
        return 1;
    }
    /* fork a child process */
    pid = fork();
    if (pid < 0)
    { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    if (pid > 0)
    { /* parent process */
        /* close the unused end of the pipe */
        close(fd1[READ_END]);
        close(fd2[WRITE_END]);

        /* parent sends the first message */
        strcpy(write_msg, "Greetings son");

        /* write to the pipe */

```

```

write(fd1[WRITE_END], write_msg,
strlen(write_msg) + 1);
sleep(2);

/* parent reads response from child */
read(fd2[READ_END], read_msg, BUFFER_SIZE);
printf("Parent read: %s\n", read_msg);
sleep(2);

/* parent sends the third message */
strcpy(write_msg, "How are you son");
write(fd1[WRITE_END], write_msg, strlen(write_msg) + 1);
sleep(2);

/* Parent reads response from child */
read(fd2[READ_END], read_msg, BUFFER_SIZE);
printf("Parent read: %s\n", read_msg);

/* close the write end of the pipe */
close(fd1[WRITE_END]);
close(fd2[READ_END]);
}
else
{ /* child process */
/* close the unused end of the pipe */
close(fd1[WRITE_END]);
close(fd2[READ_END]);

/* read from the pipe */
read(fd1[READ_END], read_msg,
BUFFER_SIZE);
printf("Child Read: %s\n", read_msg);
sleep(2);

/* Child sends the second message */
strcpy(write_msg, "Hello dad");
write(fd2[WRITE_END], write_msg, strlen(write_msg) + 1);
sleep(2);

/* Child reads third message from parent */
read(fd1[READ_END], read_msg, BUFFER_SIZE);
printf("Child read: %s\n", read_msg);

```

```
sleep(2);

/* Child sends the fourth message */
strcpy(write_msg, "Fine dad thanks");
write(fd2[WRITE_END], write_msg, strlen(write_msg) + 1);

/* close the read end of the pipe */
close(fd1[READ_END]);
close(fd2[WRITE_END]);
}
return 0;
}
```

```
sajaelkurtehi@Sajas-MacBook-Air Assignment1 % gcc -o prog prog.c
```

```
sajaelkurtehi@Sajas-MacBook-Air Assignment1 % ./prog
```

```
Child Read: Greetings son
```

```
Parent read: Hello dad
```

```
Child read: How are you son
```

```
Parent read: Fine dad thanks
```

```
sajaelkurtehi@Sajas-MacBook-Air Assignment1 %
```