

이번 과제의 목표는 wait() 함수를 대신하여 프로세스 간 의사소통을 하여

child 프로세스가 종료되었음을 parent 프로세스에게 전달하는 방법이다.

그 방법을 위해 text.txt 파일에 child 프로세스에서 내용을 적고

Parent 프로세스에서 그 파일을 읽어올때 child 프로세스에서 적은 내용과 비교를 하여 child 프로세스가 종료되었음을 확인하고자 한다.

이를 위해 문자열에 관한 연산과 파일 입출력에 관한 연산이 필요해 fcntl.h 와 string.h 헤더를 사용하였다.

상세한 코드에 대한 설명은 아래에 적어두었다.

코드 내용	코드 설명
<pre> #include <stdio.h> #include <unistd.h> #include <fcntl.h> #include <string.h> int main() { int flag = open("text.txt", O_CREAT O_RDWR O_TRUNC, S_IRWXU); int rc = fork(); if (rc == 0) { write(flag, "201600282", 9); printf("hello\n"); close(flag); } else if (rc > 0) { char buffer[9]; buffer[9] = '\0'; while (strcmp(buffer, "201600282\0") != 0) { flag = open("text.txt", O_CREAT O_RDWR, S_IRWXU); read(flag, buffer, 9); close(flag); } printf("goodbye\n"); } return 0; } </pre>	<pre> .printf 사용 .fork 함수 사용 .파일 입출력을 위해 사용 .문자열 비교를 위해 사용 . . . 읽기 & 쓰기 모드로 text.txt 생성후 open (사용자 모두 읽기 쓰기 실행 가능) . . child 프로세스일때 실행할 내용 작성 . . text.txt에 201600282 저장 (이때 작성내용 이 문자열 비교 기준이 될 예정) . . parent 프로세스일때 실행할 내용 작성 . . 문자열 buffer 생성 . buffer 문자열에 201600282가 들어올때까 지 반복 (wait 구현) . text 파일에서 buffer 문자열로 읽어옴 즉 child 프로세스에서 text파일에 작성을 완 료 해야 (child 프로세스의 마지막부분) while 문에서 빠져나올수 있다. . . </pre>

코드 내용 캡처

```
gisan@gisan-VirtualBox:~$ cat 201600282.c
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

int main()
{
    int flag = open("text.txt", O_CREAT | O_RDWR | O_TRUNC, S_IRWXU);
    int rc = fork();

    if (rc==0) {
        printf("hello\n");
        write(flag, "201600282", 9);
        close(flag);
    }
    else if (rc>0) {
        char buffer[9];
        buffer[9]='\0';
        while (strcmp(buffer, "201600282\0") != 0 ) {
            flag = open("text.txt", O_CREAT | O_RDWR, S_IRWXU);
            read(flag, buffer, 9);
            close(flag);
        }
        printf("goodbye\n");
    }
    return 0;
}
```

실행 결과

```
gisan@gisan-VirtualBox:~$ gcc 201600282.c
gisan@gisan-VirtualBox:~$ ./a.out
hello
goodbye
gisan@gisan-VirtualBox:~$
```