

1.

```
solgae.yonsei.ac.kr - u17041@solgae: ~/prog VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main()
{
    pid_t pid;

    pid = fork();

    if(pid < 0) {
        fprintf(stderr, "Fork Failed\n");
        exit(-1);
    }
    else if(pid == 0) {
        printf("I am the child %d\n", pid);
        execlp("/bin/ls", "ls", NULL);
    }
    else {
        printf("I am the parent %d\n", pid);
        wait(NULL);

        printf("Child Complete\n");
        exit(0);
    }
}
```

```
u17041@solgae:~/prog$ ls -l
total 24
-rw-r--r-- 1 u17041 u_student 375 4월 9 14:35 Fig_3-9.c
-rwxr-xr-x 1 u17041 u_student 17000 4월 9 14:39 a.out
u17041@solgae:~/prog$ ./a.out
I am the parent 973155
I am the child 0
Fig_3-9.c a.out
Child Complete
u17041@solgae:~/prog$
```

Fig 3-9 를 실행하면 위와 같이 나온다. $pid > 0$ 일때, 프로세스 ID가 pid인 프로세스에게 시그널을 전달한다. $pid == 0$ 일때, 호출한 프로세스와 같은 프로세스 그룹ID의 모든 프로세스에게 시그널을 전달한다. $pid < 0$ 일때, pid의 절댓값에 해당하는 프로세스 그룹 ID의 모든 프로세스에게 시그널을 전달한다.

```

1  #include <stdio.h>
2  #include <Windows.h>
3
4  int main( void ) {
5      STARTUPINFO si;
6      PROCESS_INFORMATION pi;
7
8      ZeroMemory(&si, sizeof(si));
9      si.cb = sizeof(si);
10     ZeroMemory(&pi, sizeof(pi));
11
12     if (!CreateProcess(NULL,
13         "C:\\WINDOWS\\system32\\mspaint.exe",
14         NULL,
15         NULL,
16         NULL,
17         FALSE,
18         0,
19         NULL,
20         NULL,
21         &si,
22         &pi)) {
23         printf("CreateProcess failed(%d).\\n", GetLastError());
24         return -1;
25     }
26
27     WaitForSingleObject(pi.hProcess, INFINITE);
28
29     CloseHandle(pi.hProcess);
30     CloseHandle(pi.hThread);
31 }

```

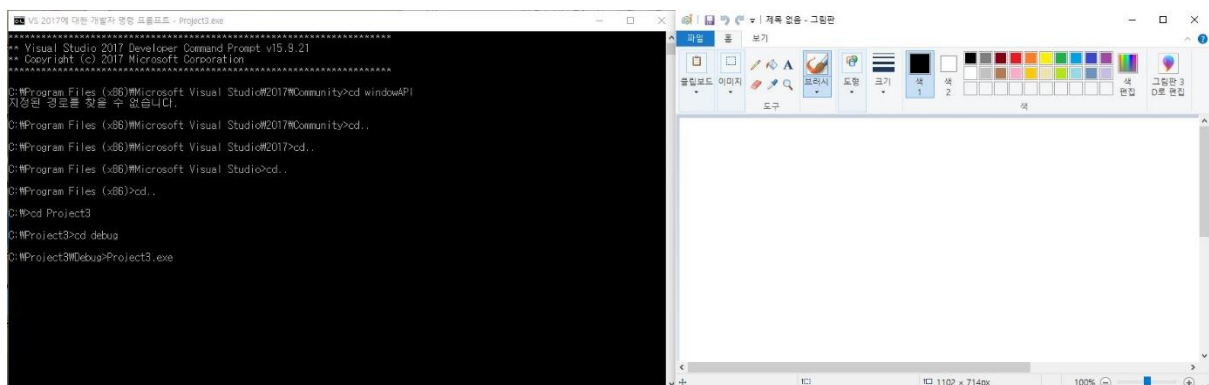


Fig 3-11 을 실행하면 콘솔창과 그림판이 실행된다. CreateProcess 함수를 통해 그림판 프로세스를 생성하고, 이 프로세스가 생성될 메모리 할당을 위해 Zeromemory 함수를 사용했다.

2.

1계열: 인자를 열거하는 방식이 나열형

v 계열: 인자를 열거하는 방식이 배열형

p가 안 붙은 계열: 경로를 지정해주면 현재/절대경로를 기준으로 찾게 됨

```
solgae.yonsei.ac.kr - u17041@solgae: ~/prog VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

void print_char(int ch);

int main()
{
    pid_t pid;
    int i,j;

    pid = fork();

    if(pid < 0) {
        fprintf(stderr, "Fork Failed\n");
        exit(-1);
    } else if(pid == 0) {
        print_char('-');
        exit(0);
    } else {
        print_char('0');
        exit(0);
    }
}

void print_char(int ch)
{
    int i,j;

    for(i = 0; i < 100; i++) {
        putchar(ch);
        fflush(stdout);
        for(j = 0; j < 1000; j++);
    }
}
```

[illegible]

4.

I/O bound 는 프로세스를 I/O 작업하는데 많은 시간을 소모하는 방식이다. CPU bound process 는 프로세스의 I/O 요청을 적게 받고, 대부분의 시간을 CPU 를 사용해 작업을 하는 방식이다. long

term scheduler 는 디스크에 있는 프로세스들을 메모리로 적재하는 역할을 한다. 생성된 프로세스들 중 어떤 프로세스에 메모리를 할당하여 준비 큐(Ready Queue)로 보내는 것으로 I/O bound 와 CPU bound process 를 적절히 혼합하여 메모리에 적재할 프로세스를 선택한다.

5.

입출력 완료상태, 로그인 대기 중 화면, 버튼 클릭을 기다리는 메시지 박스 같은 것들이 있다.

6.

Init 프로세스가 새로운 부모가 된다. init 프로세스는 고아 프로세스를 종료시키기 위해 wait 함수를 호출한다. 이 과정은 고아 프로세스가 좀비 프로세스가 되는 것을 방지한다.

7.

첫째로, 공유 메모리 모델이 있다. 이 모델은 두 개 이상의 프로세스들이 주소 공간의 일부를 공유하며, 공유한 메모리 영역에 읽기/쓰기를 통해 통신을 수행하는 모델이다. 둘째로, 메시지 전달 모델이다. 이 모델은 커널을 경유하여 고정길이 메시지, 가변길이 메시지를 송/수신자끼리 주고 받으며, 커널에서는 데이터를 버퍼링하는 모델이다.