

process 01

| | |
|------|----------|
| </1> | default |
| </2> | case 0 |
| </3> | getpid() |
| </4> | 0 |

결과 스크린샷:

```
[1524] (Parent) I am old and calm.  
[1525] (Child) I am young and wild.
```

process 02

| | |
|------|----------|
| </1> | pid > 0 |
| </2> | pid == 0 |
| </3> | 0 |
| </4> | val |

결과 스크린샷:

```
The original value is 10  
The parent will now add 1 and the child will subtract 3  
The value of child is 7.  
The value of parent is 11.
```

process 03

| | |
|------|-------------------------|
| </1> | "/bin/pwd", "pwd", NULL |
|------|-------------------------|

결과 스크린샷:

```
where am I?  
/mnt/shared_folder
```

process 04

| | |
|------|---|
| </1> | default |
| </2> | case 0 |
| </3> | pid2 |
| </4> | child_name, name, getpid() |
| </5> | case 0 |
| </6> | grandchild_name, child_name, name, getpid() |

결과 스크린샷:

```
My name is Jeffrey and I am a parent.  
My name is Michael and my father is Jeffrey. My pid is 1553  
My name is Steven! My father's name is Michael and my grandpa is called Jeffrey. My pid is 1554
```

process 05

| | |
|------|------------|
| </1> | pid > 0 |
| </2> | &status |
| </3> | status/256 |
| </4> | pid == 0 |
| </5> | 3 |

결과 스크린샷:

```
Counting to three.  
1!  
2!  
3!
```

process 06

| | |
|------|------------------|
| </1> | pid > 0 |
| </2> | pid2 > 0 |
| </3> | pid, &status, 0 |
| </4> | pid2, &status, 0 |
| </5> | pid2 == 0 |
| </6> | pid == 0 |

결과 스크린샷:

```
counting to 5!  
1!  
2!  
3!  
4!  
5!
```

process 07

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
#define SLEEP_TIME 2

int main(int argc, char* argv[]){
    pid_t pid;
    int status;
    int parent_pid, child_pid, favorite_number;
    char favorite_fruit[] = "Apple";

    printf("Final Question!\n");

    pid = fork();

    //you are allowed to write your own printf
    switch(pid) {
        default:
            //hint: parent_pid =
            parent_pid = getpid();
            printf("[%d] I am a parent\n", parent_pid);
            //hint: how would this process get the result of another???
            wait(&status);
            printf("[%d] ....and my child's favorite number is %d\n", parent_pid, status/256);
            break;
        case 0:
            favorite_number = 5;
            //hint: child_pid =
            sleep(SLEEP_TIME);
            child_pid = getpid();
            printf("[%d] and I am a child!\n", child_pid);
            printf("[%d] my parent's favorite fruit is %s but he doesn't know that my favorite number is %d\n", child_pid, favorite_fruit);
            exit(favorite_number);
            break;
        case -1:
            printf("and I am ???");
            break;
    }

    return 0;
}

/*
Expected output:

Final Question!
[xxxx] I am a parent
[yyyy] and I am a child!
[yyyy] my parent's favorite fruit is Apple but he doesn't know that my favorite number is 5
[xxxx] ....and my child's favorite number is 5

***xxxx and yyyy are process IDs***
*/
```

결과 스크린샷:

```
Final Question!
[1626] I am a parent
[1627] and I am a child!
[1627] my parent's favorite fruit is Apple but he doesn't know that my favorite number is 5
[1626] ....and my child's favorite number is 5
```

설명:

line 23: parent가 자신의 pid를 얻는다.

line 26: child로부터 child의 favorite number를 status로 받아올때까지 대기한다.

line 27: status를 256으로 나눠 child의 favorite number를 알아내고, 이를 출력한다.

line 32: parent가 먼저 line 24까지 실행할 수 있도록 SLEEP_TIME만큼 sleep한다.

line 33: child가 자신의 pid를 얻는다.

line 35: parent의 favorite fruit와 child의 favorite number를 출력한다.

line 36: exit을 이용해 parent에 child의 favorite number*256의 값을 넘겨준다.