

운영체제 과제 프로세스 실습

2013130890 영어영문학과 김승태

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

const int SEVEN_AND_A_HALF_MILLION_YEARS = 3;
const int A_DAY = 1;

// Allocated in data segment.
static int the_answer = 0;

int main(int argc, char* argv[]){
    // Allocated in stack segment.
    int arthur = 0;

    pid_t pid;

    switch(pid = fork()){ // pid = fork(void)
    default:
        // HINT: The parent process should fall into this scope.
        the_answer = 42;
        arthur = 6 * 9;
        sleep(SEVEN_AND_A_HALF_MILLION_YEARS);
        // The use of sleep() (in the code executed by the parent) in this program permits the child to be scheduled for the CPU before the parent.
        break;
    case 0:
        // HINT: The child process should fall into this scope.
        sleep(A_DAY * 2);
        break;
    case -1:
        printf("WTF?");
        return -1;
        break;
    }

    printf("My pid is %d (%s)\n", (long) getpid(), pid == 0 ? "child" : "parent");
    printf("The answer to the ultimate question of life the universe and everything is %d.\n", the_answer);
    printf("But Arthur replied that it was %d.\n", arthur);

    return 0;
}

```

```

(base) ryan_kim@gimseungtaeui-MacBookAir 01 % ./fork1
My pid is 2942 (child)
The answer to the ultimate question of life the universe and everything is 0.
But Arthur replied that it was 0.

My pid is 2941 (parent)
The answer to the ultimate question of life the universe and everything is 42.
But Arthur replied that it was 54.

```

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char* argv[])
{
    pid_t pid;
    int val = 1;

    printf("The value is %d\n", val);

    pid = fork();

    if(pid > 0){
        // HINT: The parent process should fall into this scope.
        val++;
    } else if(pid == 0) {
        // HINT: The child process should fall into this scope.
        sleep(1);
        val--;
    } else {
        printf("WTF?");
        return -1;
    }

    printf("The value is %d in %s.\n", val, pid == 0 ? "child" : "parent");

    return 0;
}

```

```

(base) ryan_kim@gimseungtaeui-MacBookAir 02 % ./fork2
The value is 1
The value is 2 in parent.
(base) ryan_kim@gimseungtaeui-MacBookAir 02 % The value is 0 in child.

```

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char* argv[]){
    printf("%s executing `ls -l`.\\n", "Before");

    // HINT: The /bin/ls -l should be executed.
    execl("/bin/ls", "ls", "-l", (char*)0);

    printf("%s executing `ls -l`.\\n", "After");

    return 0;
}
```

```
(base) ryan_kim@gimseungtaeui-MacBookAir 03 % ./fork3
Before executing `ls -l`.
total 112
-rwxr-xr-x  1 ryan_kim  staff  49472  4 28 11:27 fork3
-rw-r--r--  1 ryan_kim  staff    453  4 28 11:27 main.c
```

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char* argv[]){
    pid_t pid = fork();

    switch (pid)
    {
        default:
            // HINT: The parent process should fall into this scope.
            printf("I'm your father.\n");
            sleep(3);
            break;

        case 0:
            sleep(1);
            // HINT: The child process should fall into this scope.
            printf("I'm sorry, but I'm not Luke. I'm...");
            fflush(stdout);

            sleep(1); // for dramatic effect

            // HINT: The /usr/bin/whoami should be executed.
            execl("/usr/bin/whoami", "whoami", NULL);

        case -1:
            printf("WTF?");
            return -1;
            break;
    }
}

```

```

(base) ryan_kim@gimseungtaeui-MacBookAir 04 % ./fork4
I'm your father.
I'm sorry, but I'm not Luke. I'm...ryan_kim

```

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
|
//
// Expected output:
//
// It breaks my heart to see my fellow zealots suffer on the battlefield
// But what if we dragoons went to their rescue?
// Duh! Ra! Goon!
//

int main(int argc, char* argv[]){
    pid_t pid;
    int status;

    printf("It breaks my heart to see my fellow zealots suffer on the battlefield.\n");
    printf("But what if we dragoons went to their rescue?\n");

    printf("Duh! ");
    fflush(stdout);

    pid = fork();

    if(pid > 0){
        // HINT: The parent process should fall into this scope.
        wait(&status);
        printf("Goon!\n");
    } else if(pid == 0){
        // HINT: The child process should fall into this scope.
        printf("Ra! ");
    } else {
        printf("WTF?");
        return -1;
    }

    return 0;
}

```

```

(base) ryan_kim@gimseungtaeui-MacBookAir 05 % ./fork5
It breaks my heart to see my fellow zealots suffer on the battlefield.
But what if we dragoons went to their rescue?
Duh! Ra! Goon!

```

감사합니다