

## Lecture 10: Exec, Exit, Wait

1. exec : execve, execl, execlp, ....

A program calls exec to transform itself to another.

1) algorithm of exec

- remove old body
- load new body
- adjust process descriptor

2) function prototype of execve

```
y=execve(fname, argv, envp); // change to fname with additional arguments specified in
                             // argv[] and additional environment variables specified in
                             // envp[]. returns -1 if error.
```

```
y=execve("/aa/bb", k, 0); // change to /aa/bb with additional arguments in k
```

## 2. Homework

1-1) Try below and explain why the output is "I am ex1" when ex2 runs.

ex1.c:

```
#include <stdio.h>
void main(){
    printf("I am ex1\n");
}
```

ex2.c:

```
#include <stdio.h>
#include <unistd.h>
void main(){
    execve("./ex1",0 , 0); // change to ./ex1 with no additional argument
    printf("I am ex2\n");
}
```

```
$ gcc -o ex1 ex1.c
```

```
$ gcc -o ex2 ex2.c
```

```
$ ex2
```

```
I am ex1
```

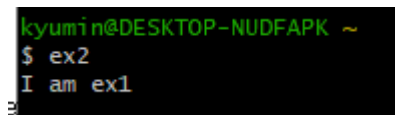
위 코드의 경우 `execve`가 실행되지 않고 “I am ex2” 메시지가 출력된다. 그 이유는 `execve`의 두 번째 argument로 0이 들어갔기 때문이다.

아래의 코드와 같이 포인터 배열의 끝을 나타내기 위해서 마지막 포인터에 0을 저장해줘야 하는 것은 맞다. `execve`의 두 번째 argument에는 포인터 주소가 들어가야한다. 그러나 0이 들어갔으므로 작동하지 않는 것이다.

For Cygwin, the above code will not work. Change as below:

ex2.c:

```
#include <stdio.h>
#include <unistd.h>
void main(){
    char *argv[5];
    argv[0]="./ex1";
    argv[1]=0;
    execve(argv[0], argv, 0); // change to ./ex1 with no additional argument
    printf("I am ex2\n");
}
```

A terminal window with a black background and green text. The prompt is 'kyumin@DESKTOP-NUDFAPK ~'. The user enters '\$ ex2' and the output is 'I am ex1'.

Ex2 프로그램을 실행 중 `execve`가 실행된다면 ex2의 바다가 지워지고, ex1이 메모리에 로드되므로 ex1 프로그램이 실행된다.

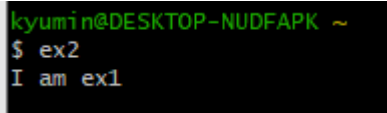
To check whether `execve` succeeds or not:

```
#include <stdio.h>
#include <unistd.h>
void main(){
    char *argv[5];
    argv[0]="./ex1";
    argv[1]=0;
    int y=execve(argv[0], argv, 0); // change to ./ex1 with no additional argument
    if (y < 0){                // we have an error in execve
        perror("err:");        // show the reason of error
    }
```

```

        exit(0);
    }
    printf("I am ex2\n");
}

```



```

kyumin@DESKTOP-NUDFAPK ~
$ ex2
I am ex1

```

Execve가 실행이 되면서 ex2의 바디는 제거가 되고, ex1이 실행이 되면서 “I am ex1”이 출력된다. Execve가 실패한다면 오류 메시지가 출력된다.

1-2) Run myexec below. Explain the result.

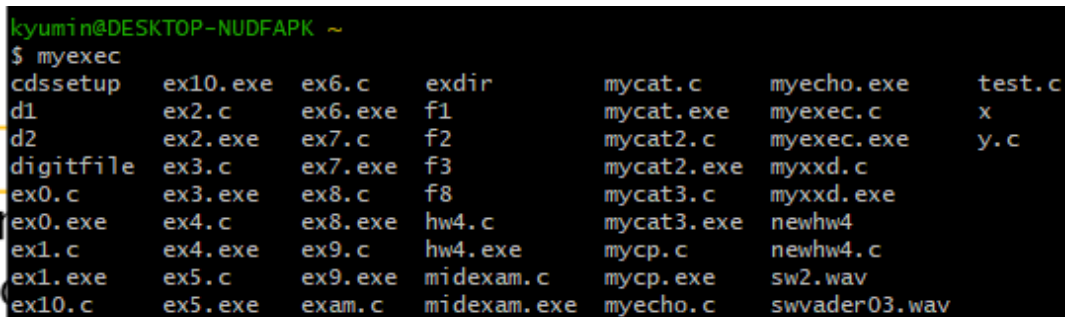
myexec.c:

```

#include <stdio.h>
#include <unistd.h>
void main(){
    char *k[10];
    k[0]="/bin/ls";
    k[1]=0;          // make sure you put 0 at the end
    execve(k[0], k, 0); // change to /bin/ls with no additional argument
}

```

The above program will exec to /bin/ls and print the listing of files in the current directoy.



```

kyumin@DESKTOP-NUDFAPK ~
$ myexec
cdssetup  ex10.exe  ex6.c    exdir      mycat.c    myecho.exe  test.c
d1        ex2.c     ex6.exe  f1         mycat.exe  myexec.c    x
d2        ex2.exe  ex7.c    f2         mycat2.c   myexec.exe  y.c
digitfile ex3.c     ex7.exe  f3         mycat2.exe myxxd.c
ex0.c     ex3.exe  ex8.c    f8         mycat3.c   myxxd.exe
ex0.exe   ex4.c     ex8.exe  hw4.c      mycat3.exe newhw4
ex1.c     ex4.exe  ex9.c    hw4.exe    mycp.c     newhw4.c
ex1.exe   ex5.c     ex9.exe  midexam.c  mycp.exe   sw2.wav
ex10.c    ex5.exe  exam.c   midexam.exe myecho.c   swvader03.wav

```

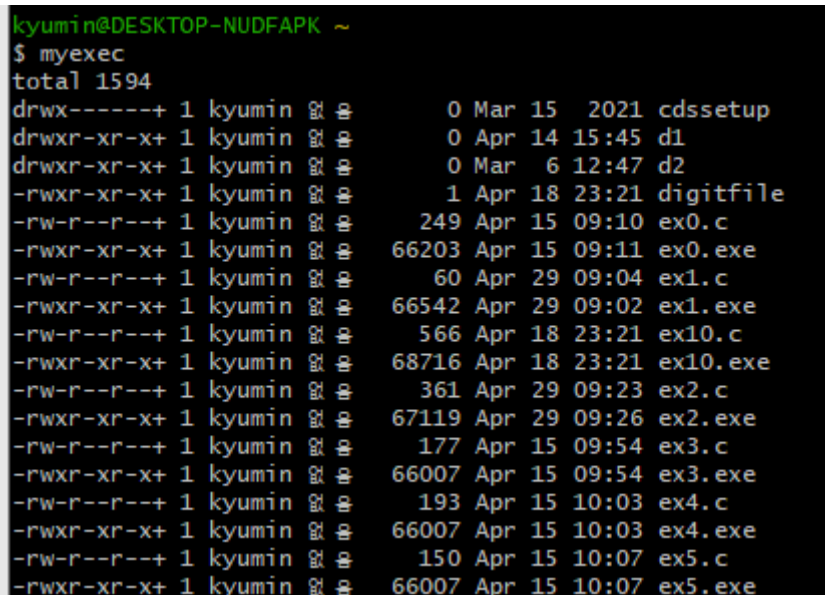
프로그램 실행 시 ls가 실행된다. 현재 위치의 파일과 디렉토리가 출력되는 것을 볼 수 있다.

1-3) Run myexec below. Explain the result.

myexec.c:

```
#include <stdio.h>
#include <unistd.h>
void main(){
    char *k[10];
    k[0]="/bin/ls";
    k[1]="-l";      // l is small L not number 1
    k[2]=0;         // make sure you put 0 at the end
    execve(k[0], k, 0); // change to "/bin/ls -l"
}
```

The above program will exec to “/bin/ls -l” and print a long listing of files in the current directoy.



```
kyumin@DESKTOP-NUDFAPK ~
$ myexec
total 1594
drwx-----+ 1 kyumin 없 음      0 Mar 15  2021 cdssetup
drwxr-xr-x+ 1 kyumin 없 음      0 Apr 14 15:45 d1
drwxr-xr-x+ 1 kyumin 없 음      0 Mar  6 12:47 d2
-rwxr-xr-x+ 1 kyumin 없 음      1 Apr 18 23:21 digitfile
-rw-r--r--+ 1 kyumin 없 음      249 Apr 15 09:10 ex0.c
-rwxr-xr-x+ 1 kyumin 없 음    66203 Apr 15 09:11 ex0.exe
-rw-r--r--+ 1 kyumin 없 음      60 Apr 29 09:04 ex1.c
-rwxr-xr-x+ 1 kyumin 없 음    66542 Apr 29 09:02 ex1.exe
-rw-r--r--+ 1 kyumin 없 음      566 Apr 18 23:21 ex10.c
-rwxr-xr-x+ 1 kyumin 없 음    68716 Apr 18 23:21 ex10.exe
-rw-r--r--+ 1 kyumin 없 음      361 Apr 29 09:23 ex2.c
-rwxr-xr-x+ 1 kyumin 없 음    67119 Apr 29 09:26 ex2.exe
-rw-r--r--+ 1 kyumin 없 음      177 Apr 15 09:54 ex3.c
-rwxr-xr-x+ 1 kyumin 없 음    66007 Apr 15 09:54 ex3.exe
-rw-r--r--+ 1 kyumin 없 음      193 Apr 15 10:03 ex4.c
-rwxr-xr-x+ 1 kyumin 없 음    66007 Apr 15 10:03 ex4.exe
-rw-r--r--+ 1 kyumin 없 음      150 Apr 15 10:07 ex5.c
-rwxr-xr-x+ 1 kyumin 없 음    66007 Apr 15 10:07 ex5.exe
```

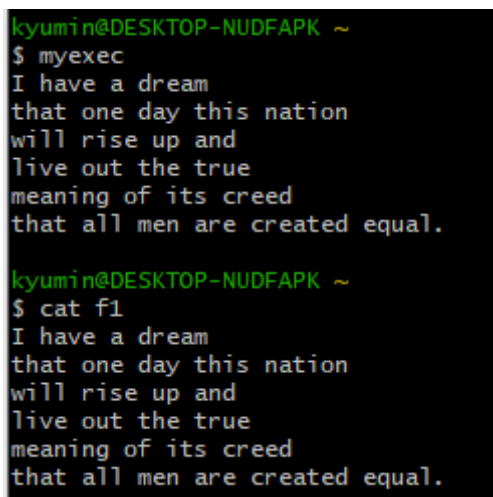
파일 이름뿐만 아니라 추가 argument로 “-l”이 들어왔다. 그래서 ls 실행 시 더 자세한 정보가 출력되는 것을 볼 수 있다.

1-4) Run myexec below and explain the result.

myexec.c:

```
void main(){
    char *x[10];
    x[0]="/bin/cat";
    x[1]="f1";
    x[2]=0;           // argument list should end with 0
    execve(x[0], x, 0); // change to /bin/cat with one argument f1
}
```

The above program will exec to "/bin/cat f1" which will print the contents of f1.



```
kyumin@DESKTOP-NUDFAPK ~
$ myexec
I have a dream
that one day this nation
will rise up and
live out the true
meaning of its creed
that all men are created equal.

kyumin@DESKTOP-NUDFAPK ~
$ cat f1
I have a dream
that one day this nation
will rise up and
live out the true
meaning of its creed
that all men are created equal.
```

추가 argument로 “f1”이 들어왔다. 그래서 f1의 파일 내용이 출력되는 것을 볼 수 있다.  
Cat 명령어로 출력해본 결과와 비교해보면 동일하게 출력된 것을 볼 수 있다.

2) Change myexec such that it execs to “/bin/ls -l -a”. Note “l” is small L, not number 1.

```
#include <stdio.h>
#include <unistd.h>
void main(){
    char *x[10];
    x[0]="/bin/ls";
    x[1]="-l";
    x[2]="-a"; // argument
    x[3]=0;;
    execve(x[0], x, 0); // d
}
```

```
kyumin@DESKTOP-NUDFAPK ~
$ myexec
total 1658
drwxrwx---+ 1 Administrators  0 Apr 29 09:39 .
drwxrwx---+ 1 kyumin         SYSTEM  0 Apr 27 17:08 ..
-rw-----+ 1 kyumin         6001 Apr 26 20:49 .bash_history
-rw-r--r--+ 1 kyumin         49 Apr 7 18:03 .bash_profile
-rw-r--r--+ 1 kyumin         20 Apr 7 18:01 .bashrc
-rw-r--r--+ 1 kyumin         0 Apr 23 14:59 .chapttest
-rwx-----+ 1 kyumin         51 Nov 16 2021 .gitconfig
drwx-----+ 1 kyumin         0 Jul 15 2021 .idlerc
-rw-----+ 1 kyumin         20 Apr 26 20:49 .lessht
-rw-----+ 1 kyumin        12288 Mar 1 17:49 .test.c.swp
-rw-----+ 1 kyumin        16667 Apr 29 09:39 .viminfo
drwx-----+ 1 kyumin         0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin         0 Apr 14 15:45 dl
drwxr-xr-x+ 1 kyumin         0 Mar 6 12:47 d2
-rwxr-xr-x+ 1 kyumin         1 Apr 18 23:21 digitfile
-rw-r--r--+ 1 kyumin         249 Apr 15 09:10 ex0.c
-rwxr-xr-x+ 1 kyumin        66203 Apr 15 09:11 ex0.exe
-rw-r--r--+ 1 kyumin         60 Apr 29 09:04 ex1.c
-rwxr-xr-x+ 1 kyumin        66542 Apr 29 09:02 ex1.exe
```

추가 argument로 “- l”과 “-a”를 추가해줬다. 숨겨진 파일들의 정보를 자세하게 출력한다.

3) Change myexec such that it execs to “/bin/cp f1 f2”.

```
kyumin@DESKTOP-NUDFAPK ~  
$ myexec  
  
kyumin@DESKTOP-NUDFAPK ~  
$ cat f1  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.  
  
kyumin@DESKTOP-NUDFAPK ~  
$ cat f2  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.
```

포인터 x[0]는 “/bin/cp”, x[1]은 “f1”, x[2]는 “f2”로 수정해줬다. Cp f1 f2와 동일한 결과가 나온다. 정상적으로 내용이 복사된 것을 볼 수 있다.

4) Change myexec such that it runs “/bin/ls -l” and prints “job done” after “/bin/ls -l” is finished.

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

void main(){
    char *x[10];

    x[0] = "/bin/ls";
    x[1] = "-l";
    x[2] = 0;

    int F;
    F = fork();
    if(F == 0){
        execve(x[0], x, 0);
    }
    usleep(100000);
    printf("job done\n");
}
```

Ls -l이 실행된 후 마지막에 job done을 출력하는 문제다. 그러나 execve 실행 시 현재 프로그램의 아래 코드는 실행이 되지 않는다. 그래서 fork를 이용했다. 자식 프로세스만 execve를 실행하도록 만들었다. 그리고 printf는 아래줄에 있으므로 자식 프로세스는 printf가 실행되지 않고, 부모 프로세스에서만 printf가 실행된다. 이 때 ls -l이 실행 중에 printf가 실행되면 “job done” 메시지가 ls 출력 결과 중간에 들어가는 문제가 발생할 수 있으므로 usleep 함수를 이용해 중간에 딜레이를 줬다.

```
kyumin@DESKTOP-NUDFAPK ~
$ myexec
total 1594
drwx-----+ 1 kyumin root 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin root 0 Apr 14 15:45 d1
drwxr-xr-x+ 1 kyumin root 0 Mar 6 12:47 d2
-rwxr-xr-x+ 1 kyumin root 1 Apr 18 23:21 digitfile
-rw-r--r--+ 1 kyumin root 249 Apr 15 09:10 ex0.c
-rwxr-xr-x+ 1 kyumin root 66203 Apr 15 09:11 ex0.exe
-rw-r--r--+ 1 kyumin root 60 Apr 29 09:04 ex1.c
-rwxr-xr-x+ 1 kyumin root 66542 Apr 29 09:02 ex1.exe
-rw-r--r--+ 1 kyumin root 566 Apr 18 23:21 ex10.c
-rwxr-xr-x+ 1 kyumin root 68716 Apr 18 23:21 ex10.exe
-rw-r--r--+ 1 kyumin root 361 Apr 29 09:23 ex2.c
-rwxr-xr-x+ 1 kyumin root 67119 Apr 29 09:26 ex2.exe
-rw-r--r--+ 1 kyumin root 177 Apr 15 09:54 ex3.c
-rwxr-xr-x+ 1 kyumin root 66007 Apr 15 09:54 ex3.exe
-rw-r--r--+ 1 kyumin root 193 Apr 15 10:03 ex4.c
```



```
-rwxr-xr-x+ 1 kyumin 없 음 67631 Apr 11 16:14 myxxd.exe
-rwxr-xr-x+ 1 kyumin 없 음 430 Mar 28 16:12 newhw4
-rwxr-xr-x+ 1 kyumin 없 음 430 Mar 28 16:13 newhw4.c
-rwxr-xr-x+ 1 kyumin 없 음 15156 Apr 24 10:26 sw2.wav
-rwx-----+ 1 kyumin 없 음 30268 Mar 31 18:54 swvader03.wav
-rw-r--r--+ 1 kyumin 없 음 526 Apr 24 09:15 test.c
-rw-r--r--+ 1 kyumin 없 음 128652 Apr 1 09:05 x
-rw-r--r--+ 1 kyumin 없 음 54 Mar 14 12:57 y.c
job done
```

마지막에 “job done” 메시지가 정상적으로 출력된 것을 볼 수 있다.

5) Change myexec such that it reads a command and execs to the given command. Your code should be able to exec any command. Read the command line with gets() or fgets() and use strtok() to extract all arguments and perform exec to run the command. Remember you have to remove the last character('\n') if you use fgets.

```
$ myexec
command> /bin/cat f1
....myexec execs to "/bin/cat f1"
$ myexec
command> /bin/cp f1 f2
....myexec execs to "/bin/cp f1 f2"
```

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

void main(){
    char *x[10];

    char pattern[100];
    fgets(pattern, 100, stdin);
    pattern[strlen(pattern) - 1] = 0;
    x[0] = strtok(pattern, " ");
    for(int i = 1;; i++){
        x[i] = strtok(NULL, " ");
        if(x[i] == NULL) break;
    }

    execve(x[0], x, 0); // change to /bin/

}
```

Fget로 입력을 받는 경우 마지막 문자에 개행 문자가 들어가므로, 0으로 초기화해줬다. 그리고 strtok 함수를 사용해 x에 단어별로 저장해줬다. Execve에서 입력된 x의 명령어를 실행해준다. 그리고 마지막에 x[i]는 NULL이 들어가므로 따로 0을 저장할 필요는 없다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ myexec  
/bin/cat f1  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.  
  
kyumin@DESKTOP-NUDFAPK ~  
$ myexec  
/bin/cp f1 f2
```

입력한 내용에 따라 그 명령어가 실행되는 것을 볼 수 있다.

6) Same as 5), but myexec will repeat the process forever until you stop the program with “ctrl-c”. Also display the result of strtok as shown below.

```
$ myexec
command> /bin/cat f1
argc: 2 argv[0]: /bin/cat argv[1]: f1 argv[2]:0
....display the contents of f1
command> /bin/cp f1 f2
argc: 3 argv[0]: /bin/cp argv[1]: f1 argv[2]: f2 argv[3]: 0
....copy f1 to f2
command> /bin/cat f2
.....
```

```
void main(){
    char *x[10];
    char pattern[100];

    while(1){
        int i = 0;

        fgets(pattern, 100, stdin);
        pattern[strlen(pattern) - 1] = 0;

        x[0] = strtok(pattern, " ");
        for(i = 1;; i++){
            x[i] = strtok(NULL, " ");
            if(x[i] == NULL) break;
        }

        printf("argc: %d ", i);
        for(int j = 0; j < i; j++) printf("argv[%d]: %s ", j, x[j]);
        printf("argv[%d]: %d\n", i, x[i]);

        int F;
        F = fork();
        if(F == 0) execve(x[0], x, 0);
    }
}
```

우선 입력된 argument의 수는 for 문에서 사용한 I 값을 이용했다. Strtok 구하는 과정을 진행하면 I 값은 단어 개수가 된다.

그리고 printf를 통해 x 배열의 문자열들을 출력해줬다. 그리고 마지막 x의 경우 NULL이므로 따로 printf를 사용해 0이 출력되도록 만들었다. 그리고 fork를 이용해 프로세스 복제 후 자식 프로세스에서만 execve가 실행되도록 만들었다.

이 모든 과정을 while 문을 통해 무한 반복되도록 만들었다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ myexec  
/bin/cat f1  
argc: 2 argv[0]: /bin/cat argv[1]: f1 argv[2]: 0  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.  
/bin/cp f1 f2  
argc: 3 argv[0]: /bin/cp argv[1]: f1 argv[2]: f2 argv[3]: 0  
/bin/cat f2  
argc: 2 argv[0]: /bin/cat argv[1]: f2 argv[2]: 0  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.
```

Argument 값들이 정상적으로 출력되었고, cat과 cp 모두 정상적으로 작동한 것을 볼 수 있다.

7) Same as 6), but change the prompt to the current location and '\$' as follows. You may need "getcwd". Also add code to detect "exit" command and stop the program when the user types "exit".

```
$ myexec
[/home/sp1/12345]$ /bin/cat f1
argc: 2 argv[0]: /bin/cat argv[1]: f1 argv[2]: 0
....display the contents of f1
[/home/sp1/12345]$ /bin/cp f1 f2
argc: 3 argv[0]: /bin/cp argv[1]: f1 argv[2]: f2 argv[3]: 0
....copy f1 to f2
[/home/sp1/12345]$ /bin/cat f2
.....
[/home/sp1/12345]$ exit
bye
$
```

```

void main(){
    char *x[10];
    char pattern[100];

    while(1){
        int i = 0;

        //현재 위치 저장 및 출력
        x[0] = getcwd(NULL, BUFSIZ);
        printf("[%s]$ ", x[0]);
        fgets(pattern, 100, stdin);
        pattern[strlen(pattern) - 1] = 0;

        x[0] = strtok(pattern, " ");

        //exit 입력된 경우 중단
        if(strcmp(x[0], "exit") == 0){
            printf("bye\n");
            break;
        }

        //토큰 추출
        for(i = 1;; i++){
            x[i] = strtok(NULL, " ");
            if(x[i] == NULL) break;
        }

        printf("argc: %d ", i);
        for(int j = 0; j < i; j++) printf("argv[%d]: %s ", j, x[j]);
        printf("argv[%d]: %d\n", i, x[i]);

        int F;
        F = fork();
        if(F == 0) execve(x[0], x, 0);
        usleep(200000);
    }
}

```

현재 위치를 출력하기 위해서 우선 `getcwd`를 사용하여 `x[0]`에 저장하고 출력했다. 그리고 입력 받은 내용을 토큰으로 나눈다. 그리고 토큰들을 출력한다. 여기서 첫번째 토큰이 “exit” 이라면 “bye” 메시지를 출력하고 프로그램을 바로 중단한다.

중단되지 않았다면 `fork`를 이용해 프로세스를 복사하고, 자식 프로세스에서는 `execve`를 실행한다.

그리고 `execve`가 끝나기 전에 부모 프로세스가 다시 실행되는 것을 막기위해서 `usleep`을 이용하여 딜레이를 줬다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ vi myexec.c  
  
kyumin@DESKTOP-NUDFAPK ~  
$ gcc -g -o myexec myexec.c  
  
kyumin@DESKTOP-NUDFAPK ~  
$ myexec  
[/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data]$ /bin/cat f1  
argc: 2 argv[0]: /bin/cat argv[1]: f1 argv[2]: 0  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.  
[/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data]$ /bin/cp f1 f2  
argc: 3 argv[0]: /bin/cp argv[1]: f1 argv[2]: f2 argv[3]: 0  
[/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data]$ /bin/cat f2  
argc: 2 argv[0]: /bin/cat argv[1]: f2 argv[2]: 0  
I have a dream  
that one day this nation  
will rise up and  
live out the true  
meaning of its creed  
that all men are created equal.  
[/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data]$ exit  
bye  
  
kyumin@DESKTOP-NUDFAPK ~  
$
```

명령어는 정상적으로 작동하고, exit 입력 시 정상적으로 프로그램이 종료된 것을 볼 수 있다.