

아래 문제에 대해 **1) 소스코드 캡처 2) 실행 결과 3) 코드 및 실행결과에 대한 설명** 을 제시해야 합니다. word 나 pdf 로 제출하기 바랍니다. 모든 것이 오픈이지만 다른 사람에게서 도움을 받아서는 안됩니다.

mycp 프로그램에 -m 옵션을 추가하여 파일 하나를 여러 파일에 복사할 수 있도록 하시오. 단 아래 조건을 충족시켜야 합니다.

- 1) **nested loop(루프안에 다른 루프가 들어가는 루프)를 사용하면 안됩니다.** 함수를 사용해서 다중루프 사용을 피하기 바랍니다.
- 2) -m 옵션의 경우 복사하는 파일(아래 예에서 f1)은 하나이지만 복사가 되는 파일들(아래 예에서 f2, f3, f4)의 갯수는 제한이 없습니다. 즉 사용자가 지정한 파일 이름들이 몇개가 될지라도 그 파일들에 모두 복사해야 합니다.
- 3) 아주 큰 파일도 복사할 수 있어야 합니다. 따라서 루프를 돌면서 원본 파일의 내용을 다 읽어가면서 복사해야 합니다.

```
$ mycp f1 f2
```

는 f1을 f2에 복사

```
$ mycp -m f1 f2 f3 f4
```

는 f1을 f2, f3, f4에 각각 복사

```
$ mycp -m f1 f2 f3 f4 f5 f6
```

는 f1을 f2, f3, f4, f5, f6 에 각각 복사

1) 소스코드 캡처 및 설명

```
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void stdcp(char *second, char *third);
void m_op(int argc, char *argv[]);

void main(int argc, char *argv[]){
    if(strcmp(argv[1], "-m")==0)
        m_op(argc, argv);
    else
        stdcp(argv[1], argv[2]);
}

void stdcp(char *second, char *third){
    int x1, x2;
    char buf[20];
    int y;

    x1 = open(second, O_RDONLY, 00777);
    x2 = open(third, O_RDWR | O_CREAT | O_TRUNC, 00777);

    if (x1==-1 || x2 == -1){
        perror("error in open");
        exit(1);
    }

    for(;;){
        y = read(x1, buf, 20);
        if (y==0) break;
        write(x2, buf, y);
    }
}

void m_op(int argc, char *argv[]){
    int i;
    for(i=3; i<argc; i++){
        stdcp(argv[2], argv[i]);
    }
}
~
~
```

Command line argument를 활용하여 코드를 작성한다.

먼저 기본적인 복사를 해주는 함수인 stdcp와

-m 옵션을 통해 하나의 파일을 여러 개의 파일에 복사해주는 m_op함수를 사용하기 위해

main 함수 전에 함수 프로토타입을 작성한다.

[main]

main함수에서는 옵션을 선택하고, 옵션에 따라 함수를 실행하는 코드를 작성한다.

문제 경우는 옵션이 -m일 때와 아무런 옵션이 없을 때로 나뉜다.

입력받은 argv[1]값이 "-m"일 때 m_op함수를 실행하고, 그렇지 않을 때는 stdcp함수를 실행한다.

argv[1]에 저장된 값이 "-m"을 확인하기 위해 strcmp를 사용한다. strcmp로 받은 값이 일치하면 0을 return한다. m_op함수에는 아규먼트의 개수를 나타내는 argc와 전달받은 아규먼트인 argv를 모두 넘겨준다.

stdcp에서는 실제 실행창에서 mycp 이후에 오는 아규먼트 두 개만 필요하므로 argv[1], argv[2]를 stdcp함수의 second, third라는 포인터 변수에 넘겨준다.

[stdcp]

argv[1]은 second에 argv[2]는 third에 저장된다. 실행창에서 mycp f1 f2를 한다고 가정할 때, second 포인터 변수에 "f1"이, third 포인터 변수에 "f2"가 저장된다.

코드에 필요한 변수들을 선언한다.

second 파일은 읽기 모드로 open하고, 해당 파일의 번호를 x1에 return한다.

third 파일은 읽고 쓸 수 있고, 해당 파일이 없다면 새롭게 만들며, 해당 파일에 값이 이미 존재한다면 기존 값을 지워주는 모드로 open한다. 해당 파일 번호를 x2에 return한다.

open 과정에서 error가 발생하면 -1을 return하는데,

조건문을 통해 x1 = -1 혹은 x2 = -1 일 때, error가 발생하였으므로 해당 문장을 출력하고 종료시킨다.

error가 발생하지 않았다면

무한루프를 통해 x1(second 파일)을 최대 20byte씩 읽고 이를 buf에 저장한다.

실제 read한 bytes가 y에 return 된다.

write함수를 통해 read 과정에서 저장된 buf 값을 y bytes만큼 읽어 x2(third 파일)에 저장한다.

루프에서 원본 파일의 내용을 다 읽어가면서 복사해야 하므로 아주 큰 파일도 복사가 가능하다.

x1 파일을 모두 읽어 실제 read한 bytes가 0이 될 때, 파일을 모두 읽었다는 뜻이므로 무한 루프를 빠져나간다.

이 과정을 통해 second 파일의 모든 값이 third 파일에 저장된다.

[m_op]

main에서 command line argument를 통해 얻은 argc값과 argv값을 모두 넘겨받는다.

정수형 변수 i를 선언한다.

argv[2]의 파일을 복사하여 argv[3], argv[4] ..., 입력받은 아규먼트 만큼의 파일에 붙여넣기한다.

실행창에서 mycp -m f1 f2 f3 f4를 한다고 가정할 때, argv[2]인 f1을 복사하여 argv[3], argv[4], argv[5]인 f2, f3, f4에 붙여넣기한다.

결국 argv[2]의 파일을 복사하여 각각의 파일에 붙여넣기 하는 것이므로

반복문을 통해서 복사 대상이 되는 파일을 지정하고, 그 값을 stdcp 함수에 넘겨준다.

for문을 통해 복사 내용이 존재하는 argv[2] 파일과 붙여넣을 대상이 되는 argv[3] 파일부터 순서대로 stdcp 함수에 반복하여 넘겨준다.

2) 실행 결과 및 설명

```
[12180626@linuxer1 ~]$ vi mycp.c
[12180626@linuxer1 ~]$ gcc -o mycp mycp.c
[12180626@linuxer1 ~]$ echo 12180626 > f1
[12180626@linuxer1 ~]$ cat f1
12180626
[12180626@linuxer1 ~]$ cat f2
I have a dream
that one day
this nation will rise up.
live out the true meaning of its creed.
I have a dream
that one day
[12180626@linuxer1 ~]$ mycp f1 f2
[12180626@linuxer1 ~]$ cat f2
12180626
```

vi 편집기를 통해 mycp.c 코드를 작성하였다.

C프로그래밍을 사용하였으므로 gcc -o mycp mycp.c를 사용한다. mycp.c를 컴파일하여 변환된 기계어를 mycp에 저장한다.

작성한 mycp가 잘 구현되는지 확인하기 위해 echo 12180626 > f1을 활용하여 12180626이라는 값을 f1파일에 저장한다.

cat f1을 통해 값이 잘 저장됨을 확인하였고, cat f2를 통해 기존 f1과 f2에 다른 값이 저장되어있음을 알 수 있다.

mycp f1 f2 를 실행시킨 뒤 cat f2를 통해 값을 확인하면 f1파일의 내용이 잘 복사됨을 알 수 있다.

```
[12180626@linuxer1 ~]$ cat f3
I have a dream
that one day
this nation will rise up.
live out the true meaning of its creed.
I have a dream
that one day
[12180626@linuxer1 ~]$ cat f4
I have a dream
that one day
this nation will rise up.
live out the true meaning of its creed.
I have a dream
that one day
[12180626@linuxer1 ~]$ mycp -m f1 f2 f3 f4
[12180626@linuxer1 ~]$ cat f3
12180626
[12180626@linuxer1 ~]$ cat f4
12180626
```

위와 같이 f3, f4 파일내용을 cat 명령어를 통해 확인하였다. f1파일 내용과 다름을 알 수 있다.

mycp -m f1 f2 f3 f4를 통해 f1 파일을 복사하여 f2, f3, f4파일에 붙여넣었다.

이후 cat명령어를 통해 f1파일이 잘 복사되었음을 알 수 있다.

```
[12180626@linuxer1 ~]$ cat f5
I have a dream
that one day
this nation will rise up.
live out the true meaning of its creed.
I have a dream
that one day
[12180626@linuxer1 ~]$ cat f6
I have a dream
that one day
this nation will rise up.
live out the true meaning of its creed.
I have a dream
that one day
[12180626@linuxer1 ~]$ mycp -m f1 f2 f3 f4 f5 f6
[12180626@linuxer1 ~]$ cat f5
12180626
[12180626@linuxer1 ~]$ cat f6
12180626
```

위와 같이 f5, f6 파일내용을 cat 명령어를 통해 확인하였다. f1파일 내용과 다름을 알 수 있다.

mycp -m f1 f2 f3 f4 f5 f6를 통해 f1 파일을 복사하여 f2, f3, f4, f5, f6파일에 붙여넣었다.

이후 cat명령어를 통해 f1파일이 잘 복사되었음을 알 수 있다.