

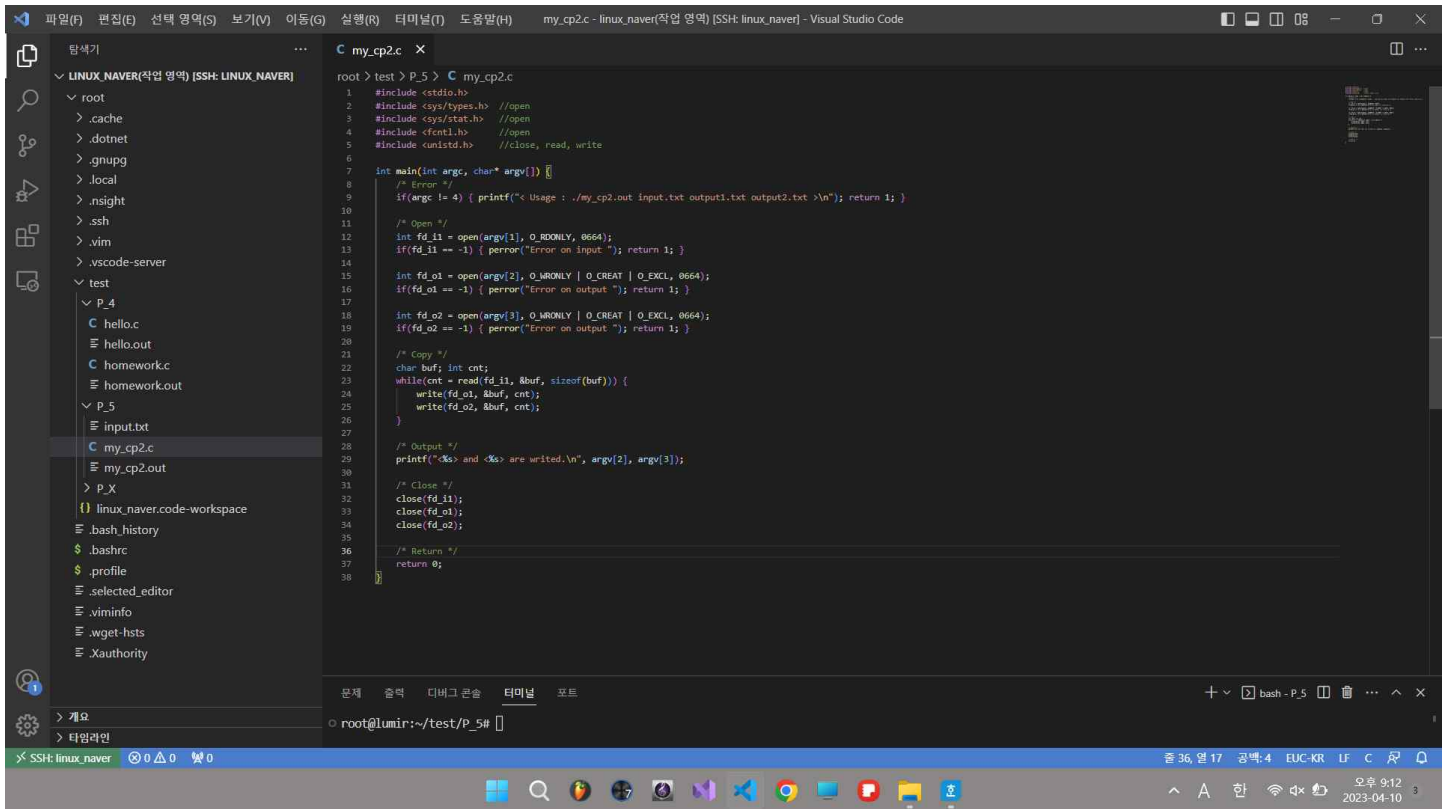
201818716

컴퓨터공학부

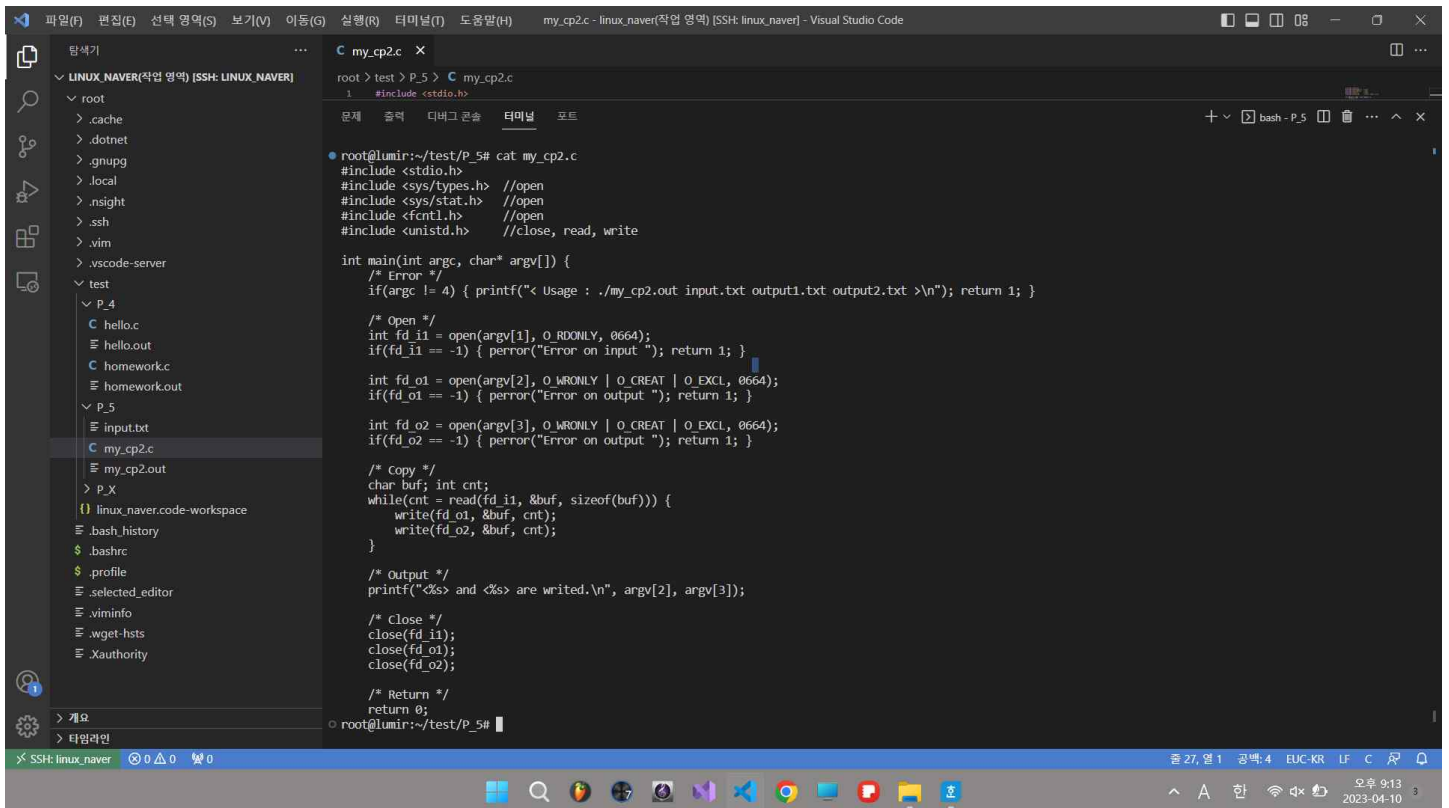
김용현

※ 실습 간 Putty · Xshell 대신, VScode상의 SSH 확장프로그램을 이용하여 실습 서버에 접근하였습니다.

[소스코드]



```
1 #include <stdio.h>
2 #include <sys/types.h> //open
3 #include <sys/stat.h> //open
4 #include <fcntl.h> //open
5 #include <unistd.h> //close, read, write
6
7 int main(int argc, char* argv[]) {
8     /* Error */
9     if(argc != 4) { printf("< Usage : ./my_cp2.out input.txt output1.txt output2.txt >\n"); return 1; }
10
11     /* Open */
12     int fd_i1 = open(argv[1], O_RDONLY, 0664);
13     if(fd_i1 == -1) { perror("Error on input "); return 1; }
14
15     int fd_o1 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, 0664);
16     if(fd_o1 == -1) { perror("Error on output "); return 1; }
17
18     int fd_o2 = open(argv[3], O_WRONLY | O_CREAT | O_EXCL, 0664);
19     if(fd_o2 == -1) { perror("Error on output "); return 1; }
20
21     /* Copy */
22     char buf; int cnt;
23     while(cnt = read(fd_i1, &buf, sizeof(buf))) {
24         write(fd_o1, &buf, cnt);
25         write(fd_o2, &buf, cnt);
26     }
27
28     /* Output */
29     printf("< %s and %s are writed.\n", argv[2], argv[3]);
30
31     /* Close */
32     close(fd_i1);
33     close(fd_o1);
34     close(fd_o2);
35
36     /* Return */
37     return 0;
38 }
```



```
root@lumir:~/test/P_5# cat my_cp2.c
#include <stdio.h>
#include <sys/types.h> //open
#include <sys/stat.h> //open
#include <fcntl.h> //open
#include <unistd.h> //close, read, write

int main(int argc, char* argv[]) {
    /* Error */
    if(argc != 4) { printf("< Usage : ./my_cp2.out input.txt output1.txt output2.txt >\n"); return 1; }

    /* Open */
    int fd_i1 = open(argv[1], O_RDONLY, 0664);
    if(fd_i1 == -1) { perror("Error on input "); return 1; }

    int fd_o1 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, 0664);
    if(fd_o1 == -1) { perror("Error on output "); return 1; }

    int fd_o2 = open(argv[3], O_WRONLY | O_CREAT | O_EXCL, 0664);
    if(fd_o2 == -1) { perror("Error on output "); return 1; }

    /* Copy */
    char buf; int cnt;
    while(cnt = read(fd_i1, &buf, sizeof(buf))) {
        write(fd_o1, &buf, cnt);
        write(fd_o2, &buf, cnt);
    }

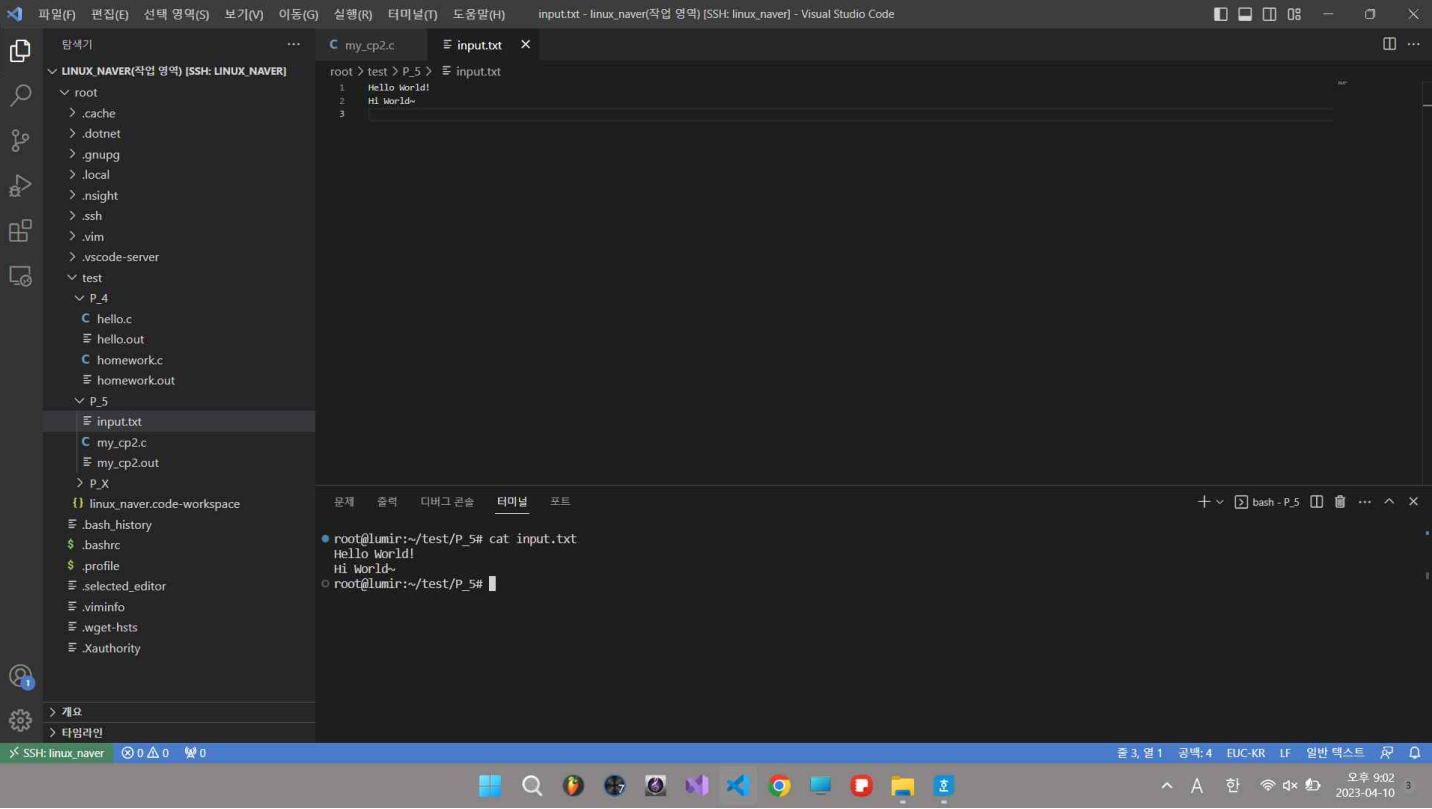
    /* Output */
    printf("< %s and %s are writed.\n", argv[2], argv[3]);

    /* Close */
    close(fd_i1);
    close(fd_o1);
    close(fd_o2);

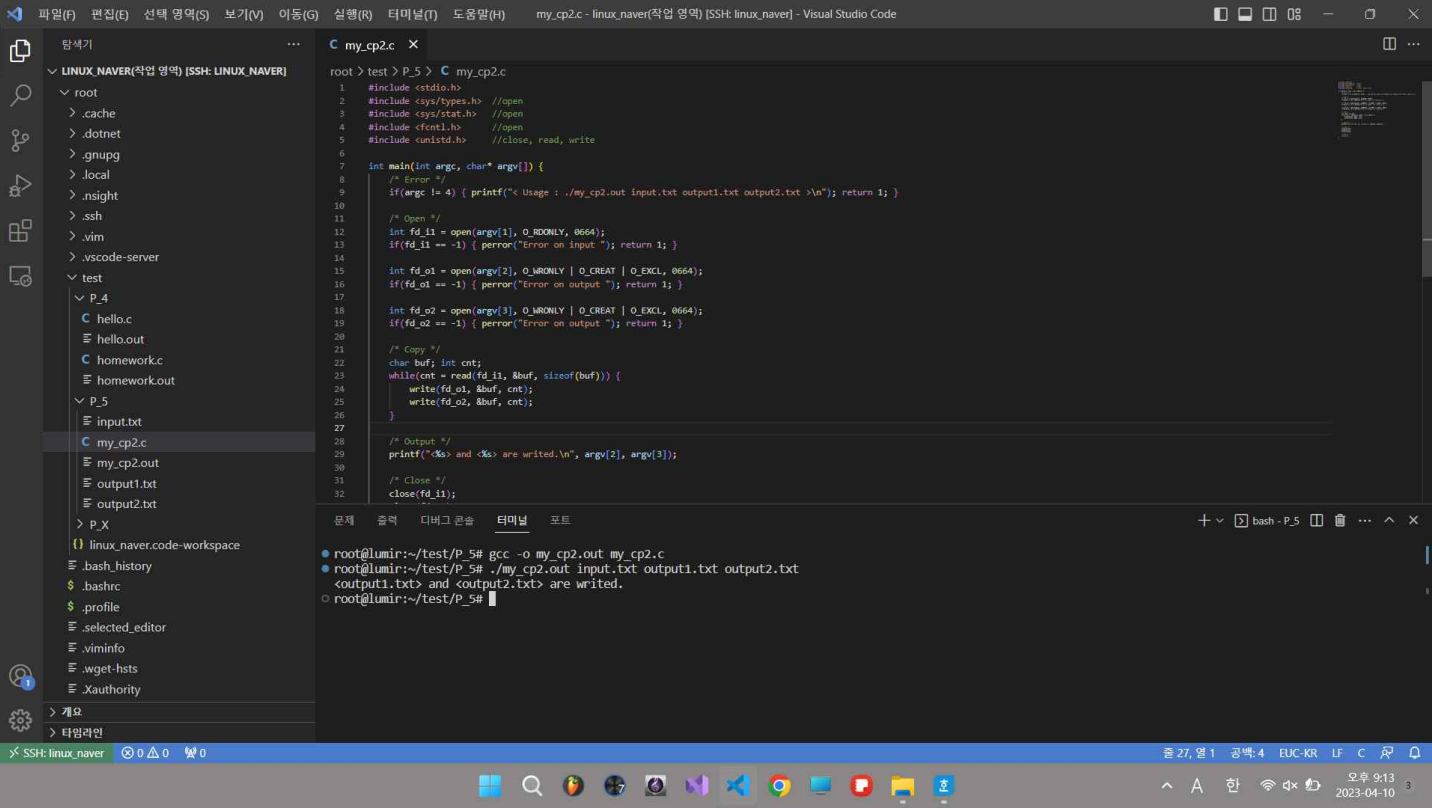
    /* Return */
    return 0;
}

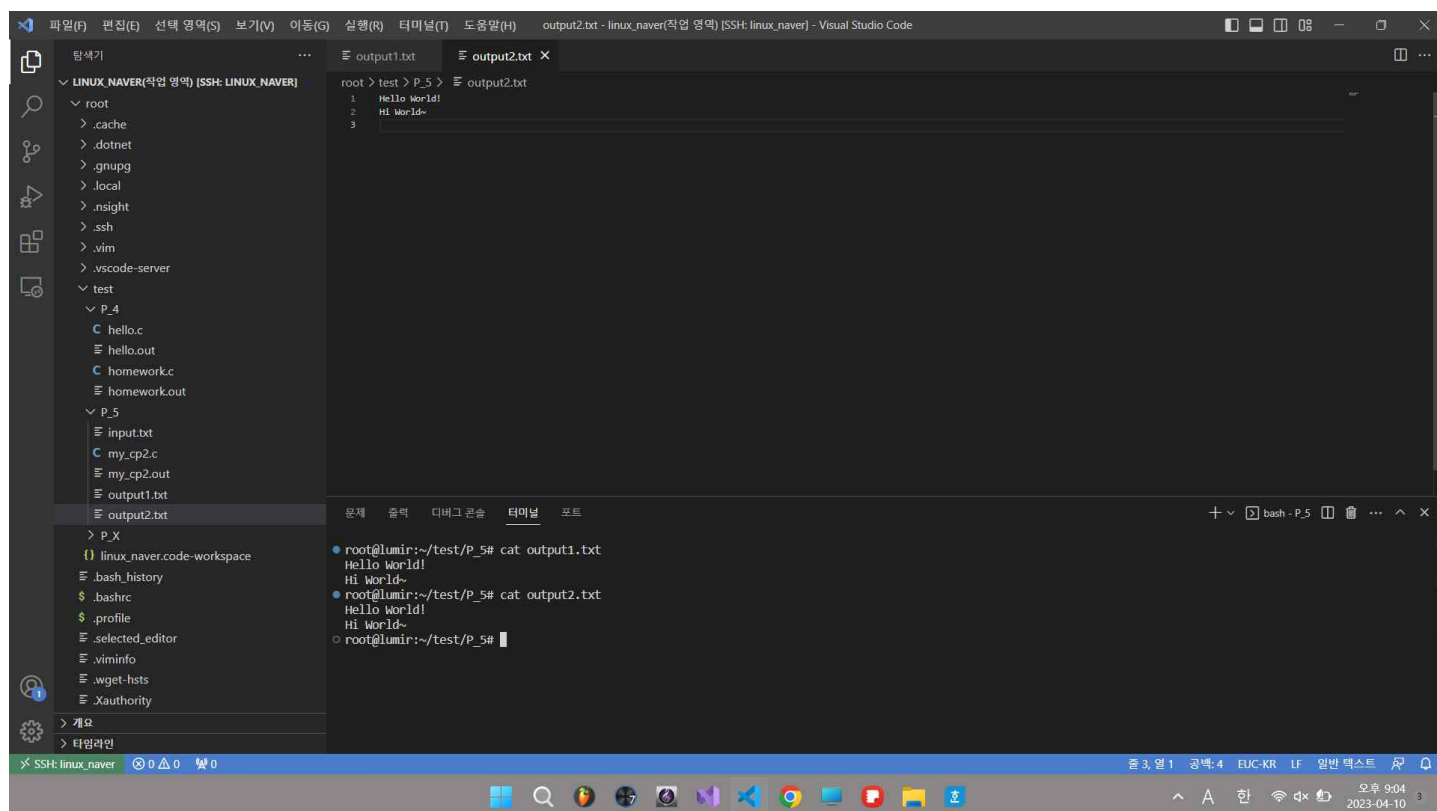
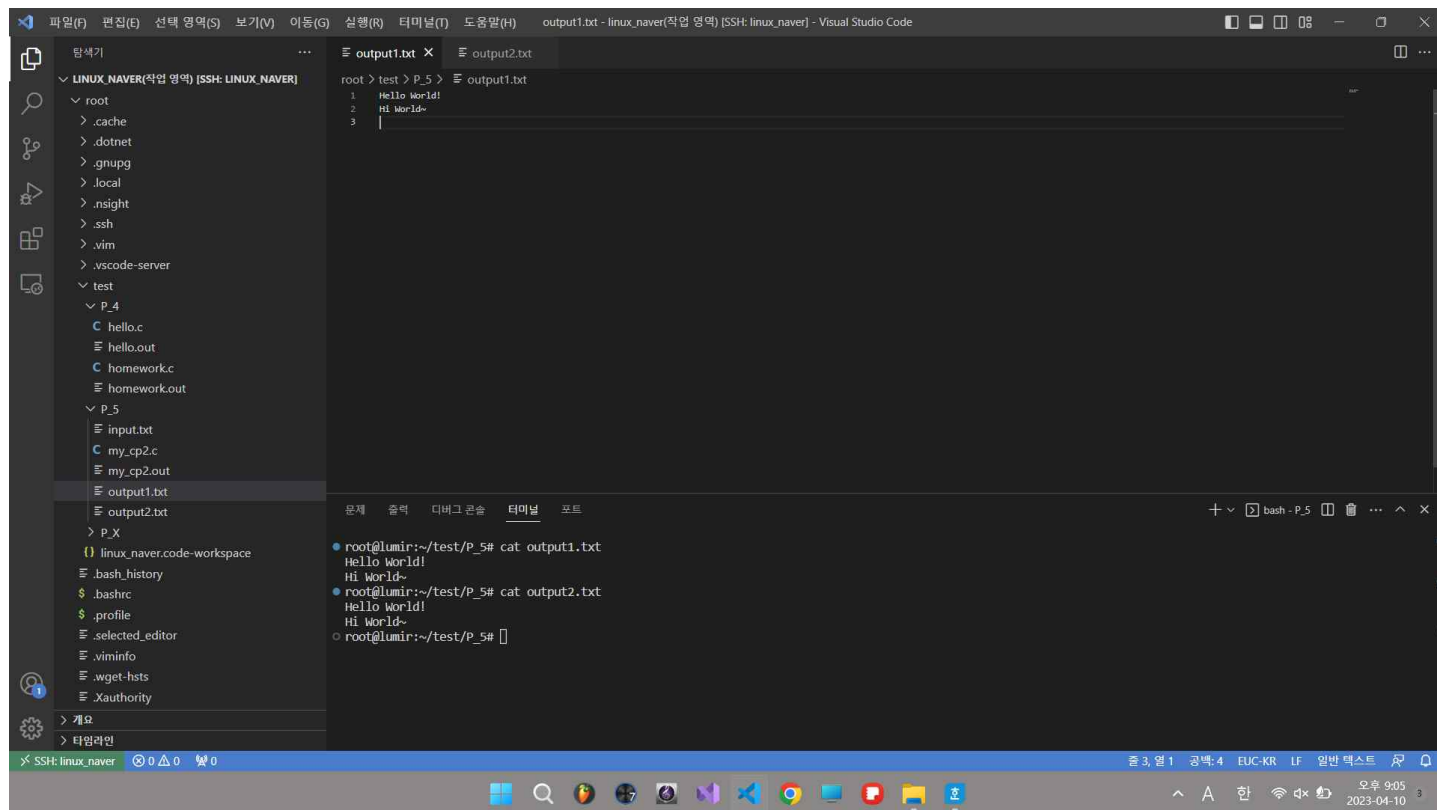
root@lumir:~/test/P_5#
```

[input.txt 파일 내용]

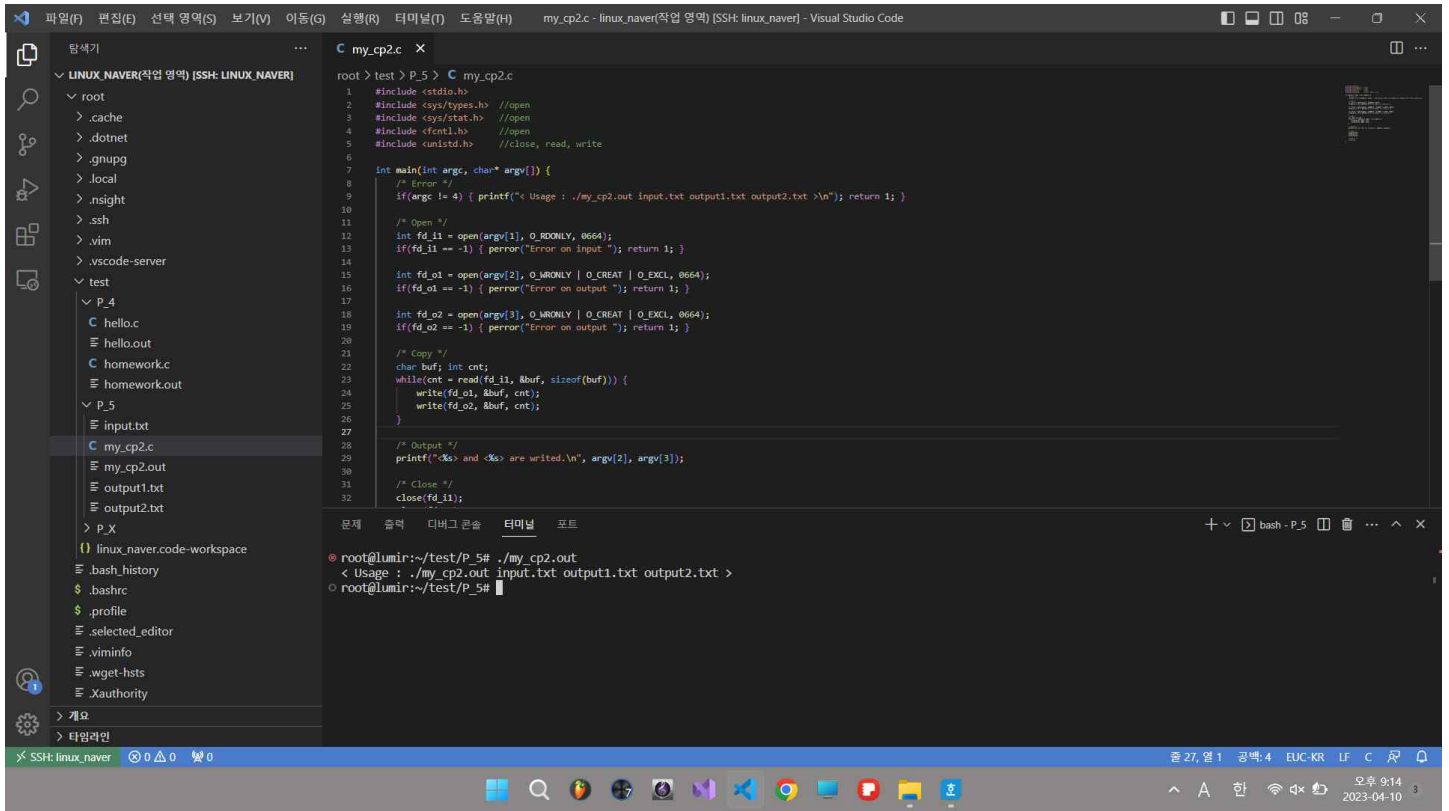


[출력 및 파일 상태]

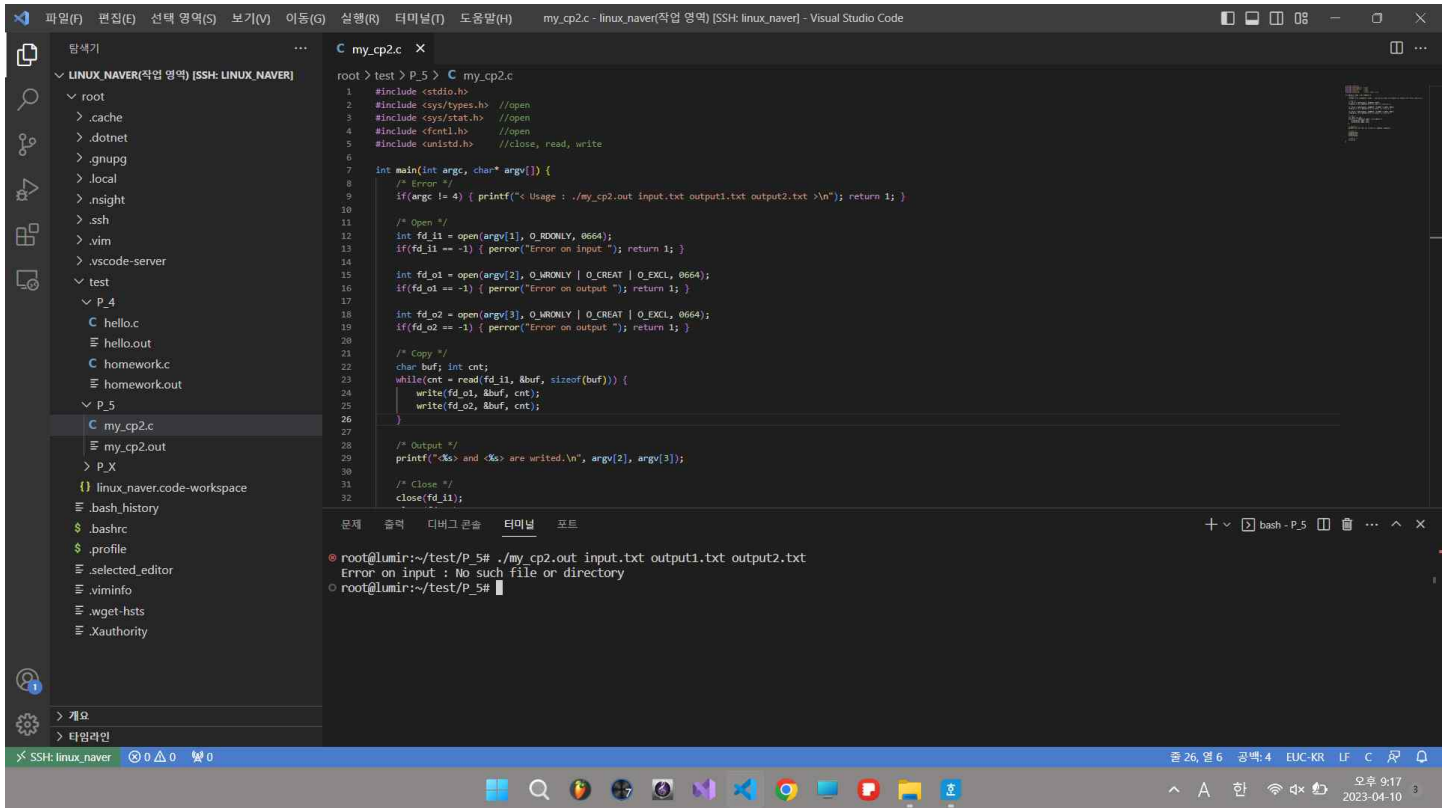




## [인자를 잘못 전달한 경우]



## [input.txt가 존재하지 않는 경우]



## [output1.txt만 존재하는 경우]

The screenshot shows the Visual Studio Code interface with a C program named `my_cp2.c` open. The program is designed to copy data from `input.txt` to `output1.txt` and `output2.txt`. The terminal output shows the following commands and results:

```
root@lumir:~/test/P_5# ls
input.txt  my_cp2.c  my_cp2.out  output1.txt  output2.txt
root@lumir:~/test/P_5# ./my_cp2.out input.txt output1.txt output2.txt
Error on output : File exists
root@lumir:~/test/P_5#
```

The error message "Error on output : File exists" indicates that the file `output2.txt` already exists, which is why the program failed to create or write to it.

## [output2.txt만 존재하는 경우]

The screenshot shows the same Visual Studio Code interface with the `my_cp2.c` program. The terminal output shows the following commands and results:

```
root@lumir:~/test/P_5# ls
input.txt  my_cp2.c  my_cp2.out  output1.txt  output2.txt
root@lumir:~/test/P_5# ./my_cp2.out input.txt output1.txt output2.txt
Error on output : File exists
root@lumir:~/test/P_5#
```

The error message "Error on output : File exists" indicates that the file `output1.txt` already exists, which is why the program failed to create or write to it.



[output1.txt와 output2.txt가 모두 존재하는 경우]

The screenshot shows a Visual Studio Code editor window with a C program named `my_cp2.c` and its execution output in a terminal.

**File Explorer (Left):** Shows the project structure for `linux_naver`. The `test` directory is expanded, showing files like `hello.c`, `homework.c`, `input.txt`, `my_cp2.c`, `my_cp2.out`, `output1.txt`, and `output2.txt`.

**Editor (Center):** Displays the source code of `my_cp2.c`. The code is a C program that takes four arguments: `input.txt`, `my_cp2.out`, `output1.txt`, and `output2.txt`. It opens these files, copies data from `input.txt` to `output1.txt` and `output2.txt`, and prints a message.

```
1 #include <stdio.h>
2 #include <sys/types.h> //open
3 #include <sys/stat.h> //open
4 #include <fcntl.h> //open
5 #include <unistd.h> //close, read, write
6
7 int main(int argc, char* argv[]) {
8     /* Error */
9     if(argc != 4) { printf("< Usage : ./my_cp2.out input.txt output1.txt output2.txt >\n"); return 1; }
10
11     /* Open */
12     int fd_i1 = open(argv[1], O_RDONLY, 0664);
13     if(fd_i1 == -1) { perror("Error on input "); return 1; }
14
15     int fd_o1 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, 0664);
16     if(fd_o1 == -1) { perror("Error on output "); return 1; }
17
18     int fd_o2 = open(argv[3], O_WRONLY | O_CREAT | O_EXCL, 0664);
19     if(fd_o2 == -1) { perror("Error on output "); return 1; }
20
21     /* Copy */
22     char buf; int cnt;
23     while(cnt = read(fd_i1, &buf, sizeof(buf))) {
24         write(fd_o1, &buf, cnt);
25         write(fd_o2, &buf, cnt);
26     }
27
28     /* Output */
29     printf("< %s and %s are writed.\n", argv[2], argv[3]);
30
31     /* Close */
32     close(fd_i1);
33 }
```

**Terminal (Bottom):** Shows the execution of the program. The command `./my_cp2.out input.txt output1.txt output2.txt` is run, and the output is `Error on output : File exists`.

```
root@lumir:~/test/P_5# ls
input.txt my_cp2.c my_cp2.out output1.txt output2.txt
root@lumir:~/test/P_5# ./my_cp2.out input.txt output1.txt output2.txt
Error on output : File exists
root@lumir:~/test/P_5#
```

The status bar at the bottom indicates the current file is `my_cp2.c` and the terminal is running `bash - P_5`.