

201818716

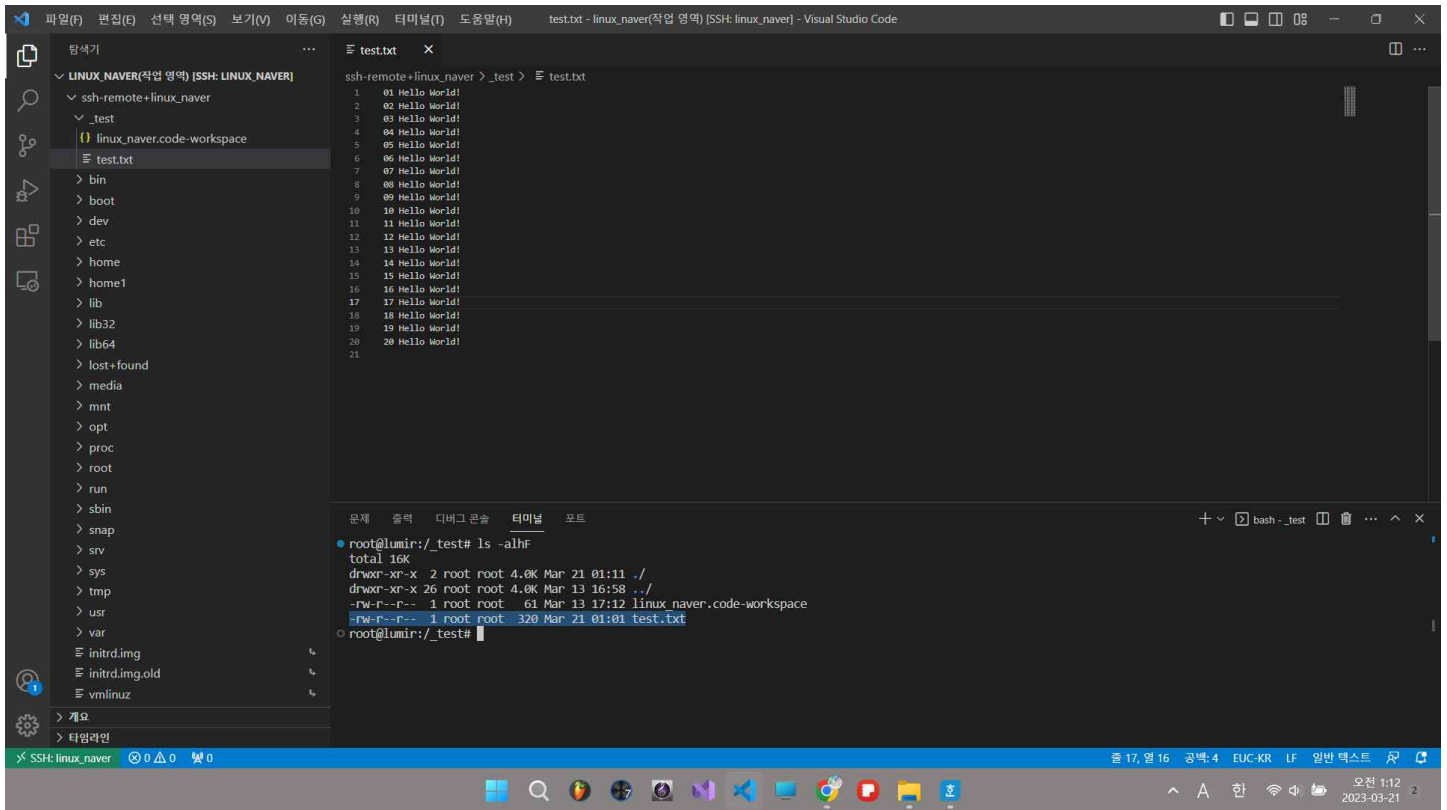
컴퓨터공학부

김용현

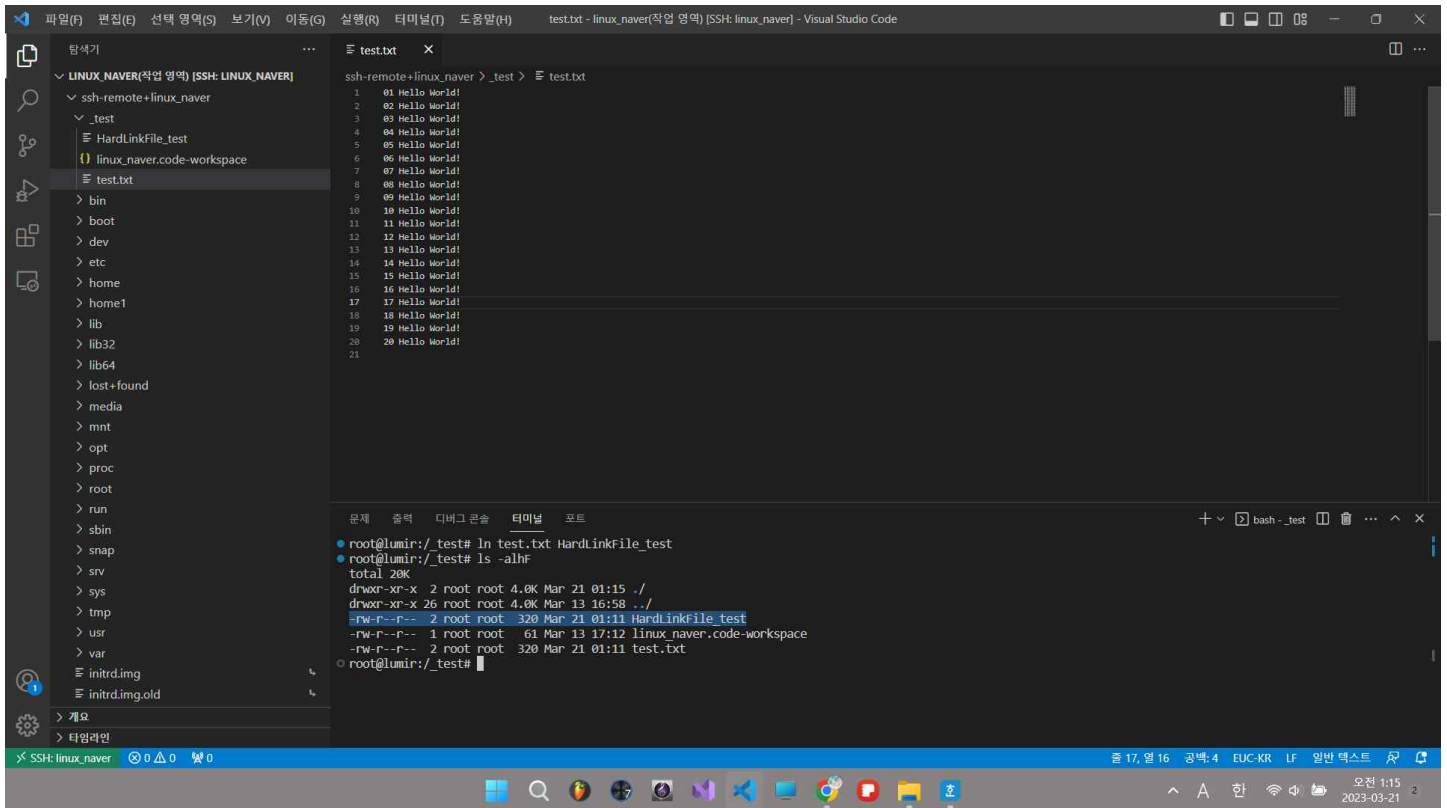
※ 실습 간 Putty · Xshell 대신, VScode상의 SSH 확장프로그램을 이용하여 실습 서버에 접근하였습니다.

[하드링크 실습]

1. 임의의 텍스트 파일을 만든다. : test.txt 파일을 만들었다.

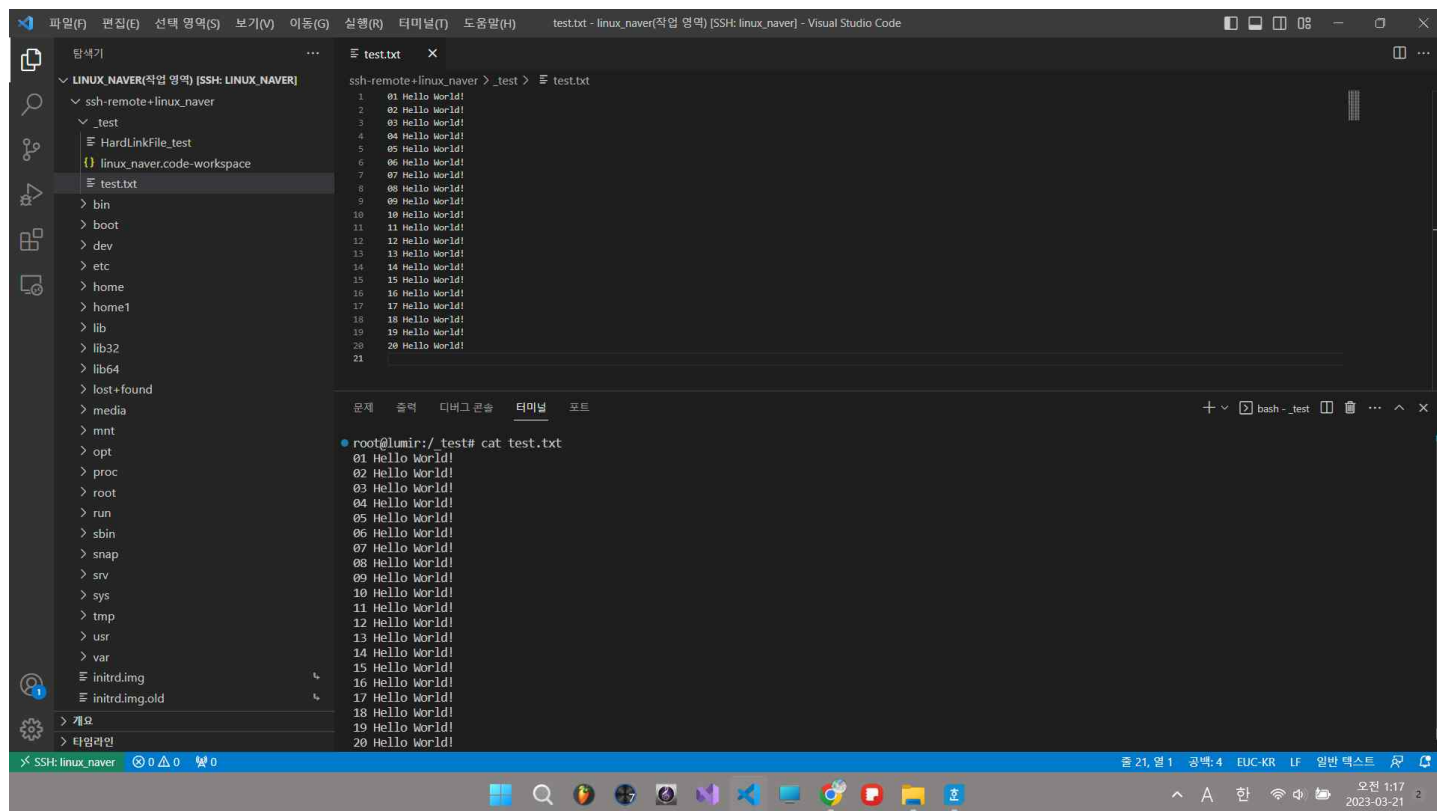


2. 해당 파일에 대한 하드링크를 작성한다. : HardLinkFile_test 란 이름의 Hard Link를 만들었다.



3. 하드링크 파일의 내용을 수정한다.

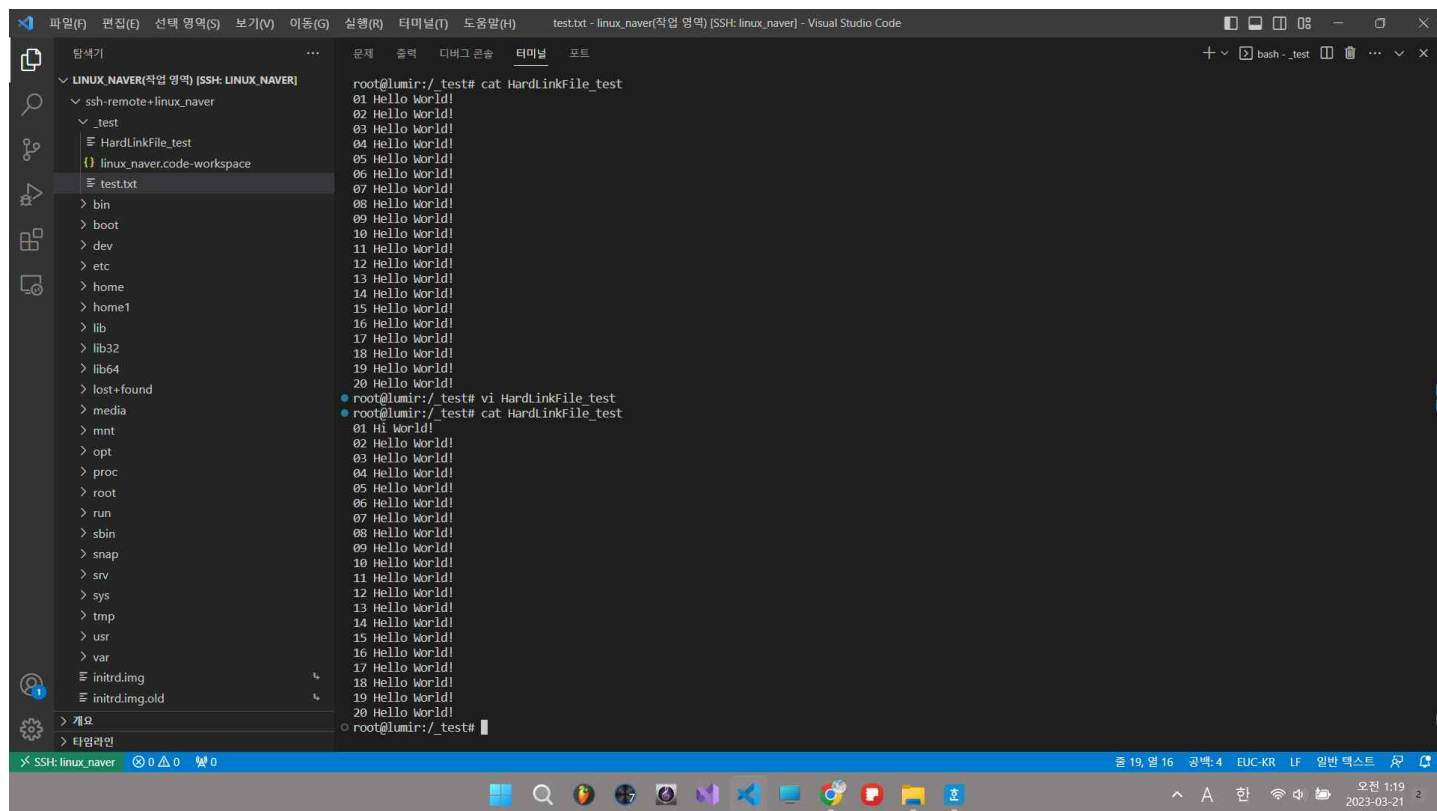
[변경 전]



```
test.txt
1 01 Hello World!
2 02 Hello World!
3 03 Hello World!
4 04 Hello World!
5 05 Hello World!
6 06 Hello World!
7 07 Hello World!
8 08 Hello World!
9 09 Hello World!
10 10 Hello World!
11 11 Hello World!
12 12 Hello World!
13 13 Hello World!
14 14 Hello World!
15 15 Hello World!
16 16 Hello World!
17 17 Hello World!
18 18 Hello World!
19 19 Hello World!
20 20 Hello World!
```

```
root@lumi:/ test# cat test.txt
01 Hello World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

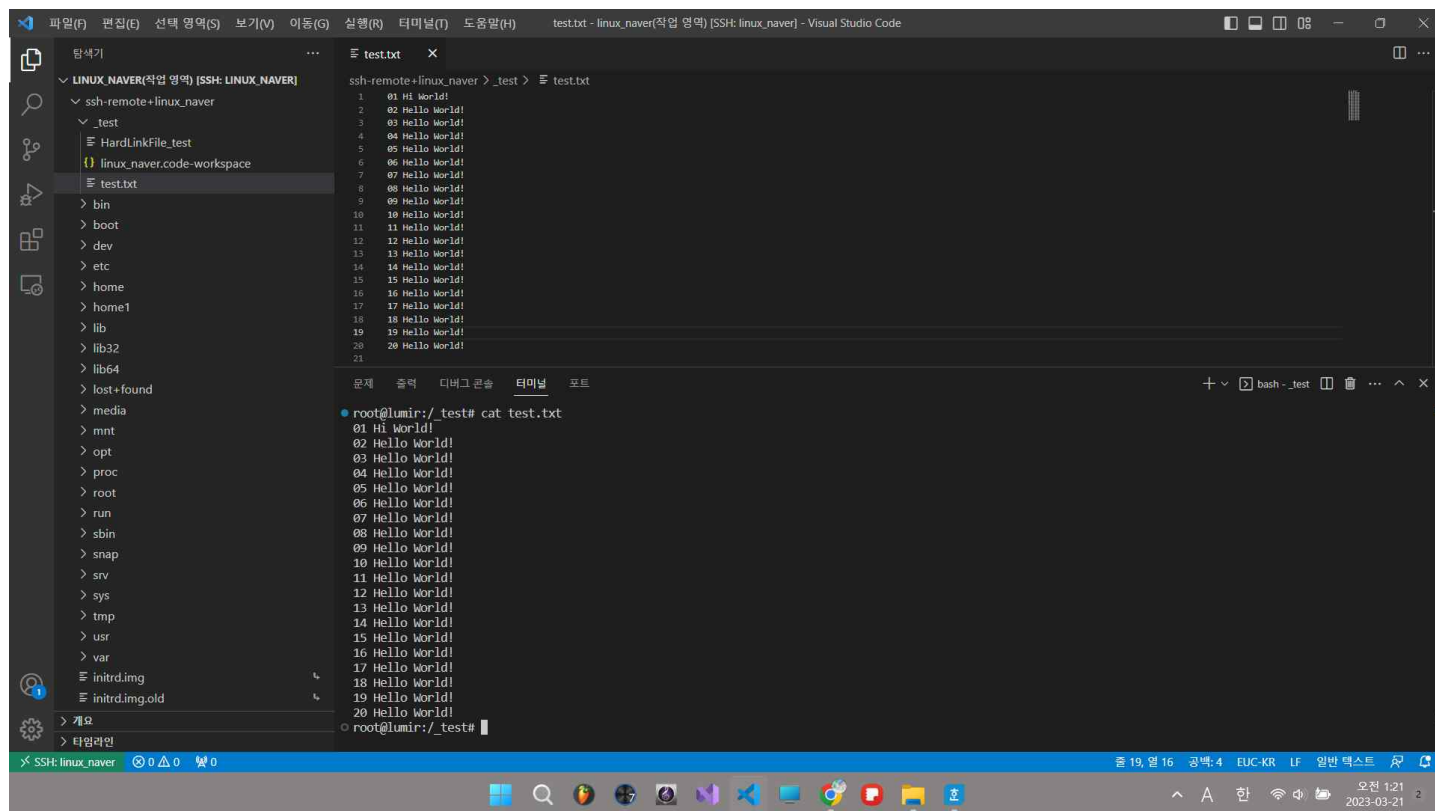
[변경 후] : 01번 줄의 '01 Hello World!'를 '01 Hi World!'로 변경하였다.



```
test.txt
1 01 Hi World!
2 02 Hello World!
3 03 Hello World!
4 04 Hello World!
5 05 Hello World!
6 06 Hello World!
7 07 Hello World!
8 08 Hello World!
9 09 Hello World!
10 10 Hello World!
11 11 Hello World!
12 12 Hello World!
13 13 Hello World!
14 14 Hello World!
15 15 Hello World!
16 16 Hello World!
17 17 Hello World!
18 18 Hello World!
19 19 Hello World!
20 20 Hello World!
```

```
root@lumi:/ test# vi HardLinkFile_test
root@lumi:/ test# cat HardLinkFile_test
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

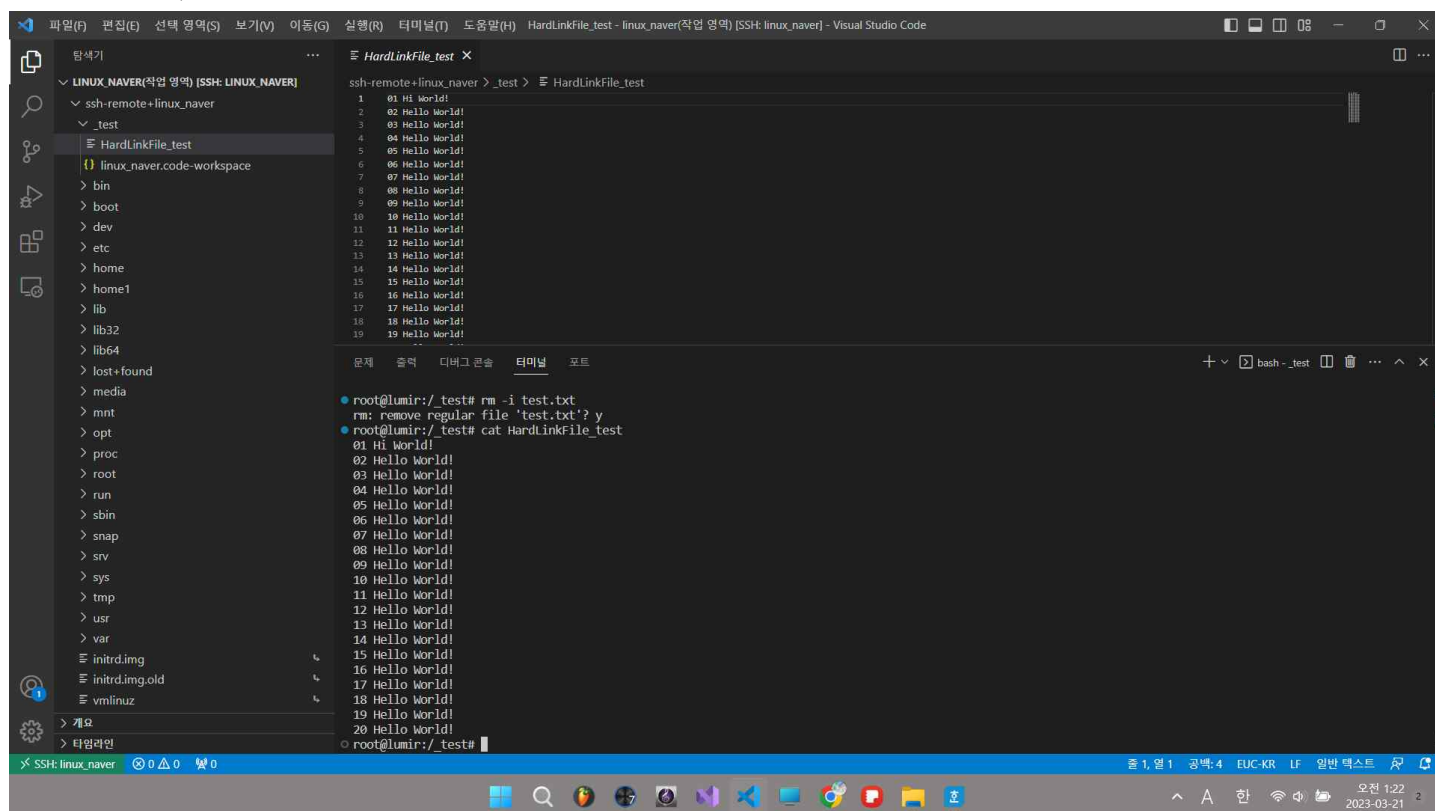
4. 원본 파일 확인 : 원본 파일 역시 이번 줄의 '01 Hello World!'가 '01 Hi World!'로 변경되었다.



```
test.txt
1 01 Hi World!
2 02 Hello World!
3 03 Hello World!
4 04 Hello World!
5 05 Hello World!
6 06 Hello World!
7 07 Hello World!
8 08 Hello World!
9 09 Hello World!
10 10 Hello World!
11 11 Hello World!
12 12 Hello World!
13 13 Hello World!
14 14 Hello World!
15 15 Hello World!
16 16 Hello World!
17 17 Hello World!
18 18 Hello World!
19 19 Hello World!
20 20 Hello World!
```

```
root@lumir:/test# cat test.txt
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

5. 원본 파일 삭제 후, 하드링크 파일 확인 : test.txt 파일의 출력결과와 동일한 결과가 출력된다.

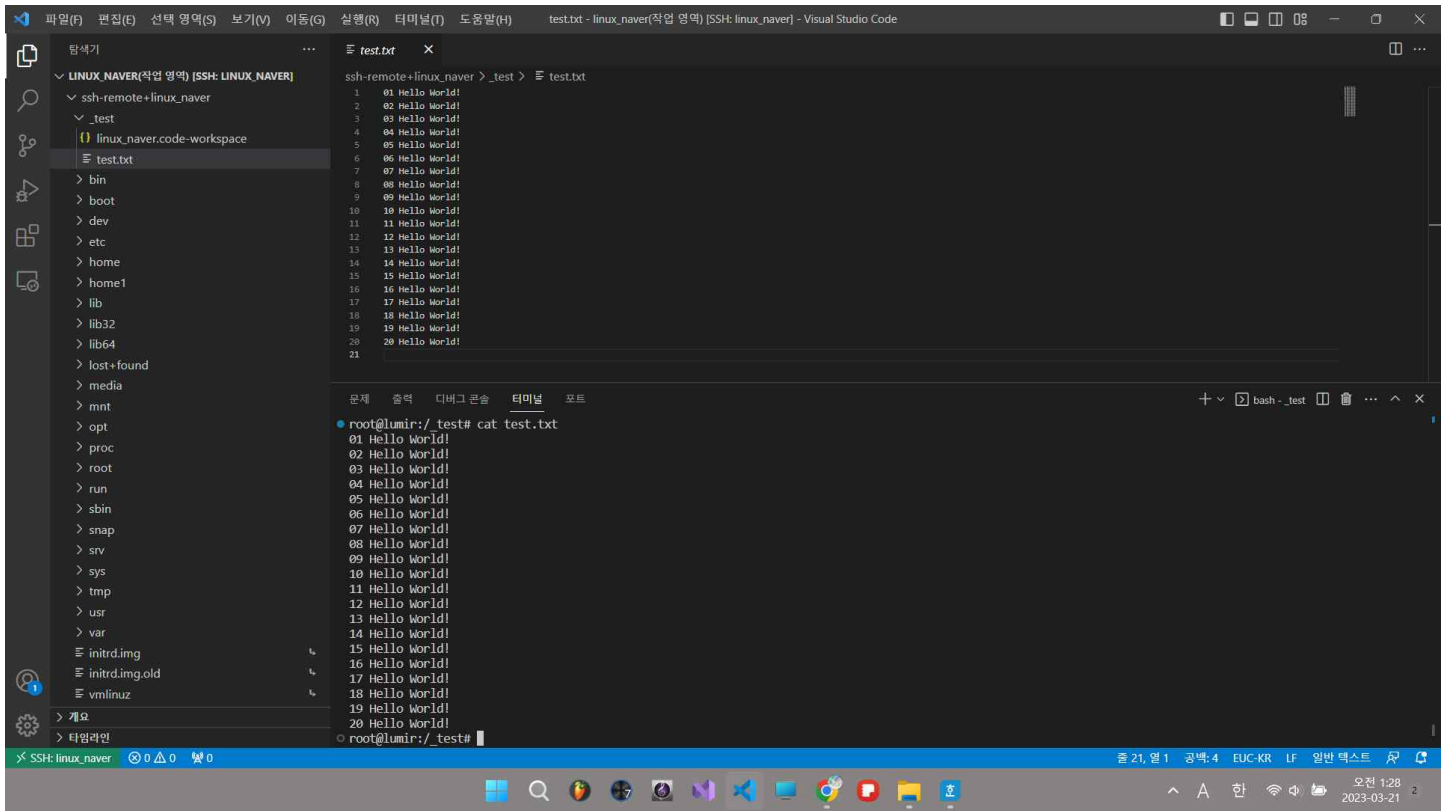


```
HardLinkFile_test
1 01 Hi World!
2 02 Hello World!
3 03 Hello World!
4 04 Hello World!
5 05 Hello World!
6 06 Hello World!
7 07 Hello World!
8 08 Hello World!
9 09 Hello World!
10 10 Hello World!
11 11 Hello World!
12 12 Hello World!
13 13 Hello World!
14 14 Hello World!
15 15 Hello World!
16 16 Hello World!
17 17 Hello World!
18 18 Hello World!
19 19 Hello World!
20 20 Hello World!
```

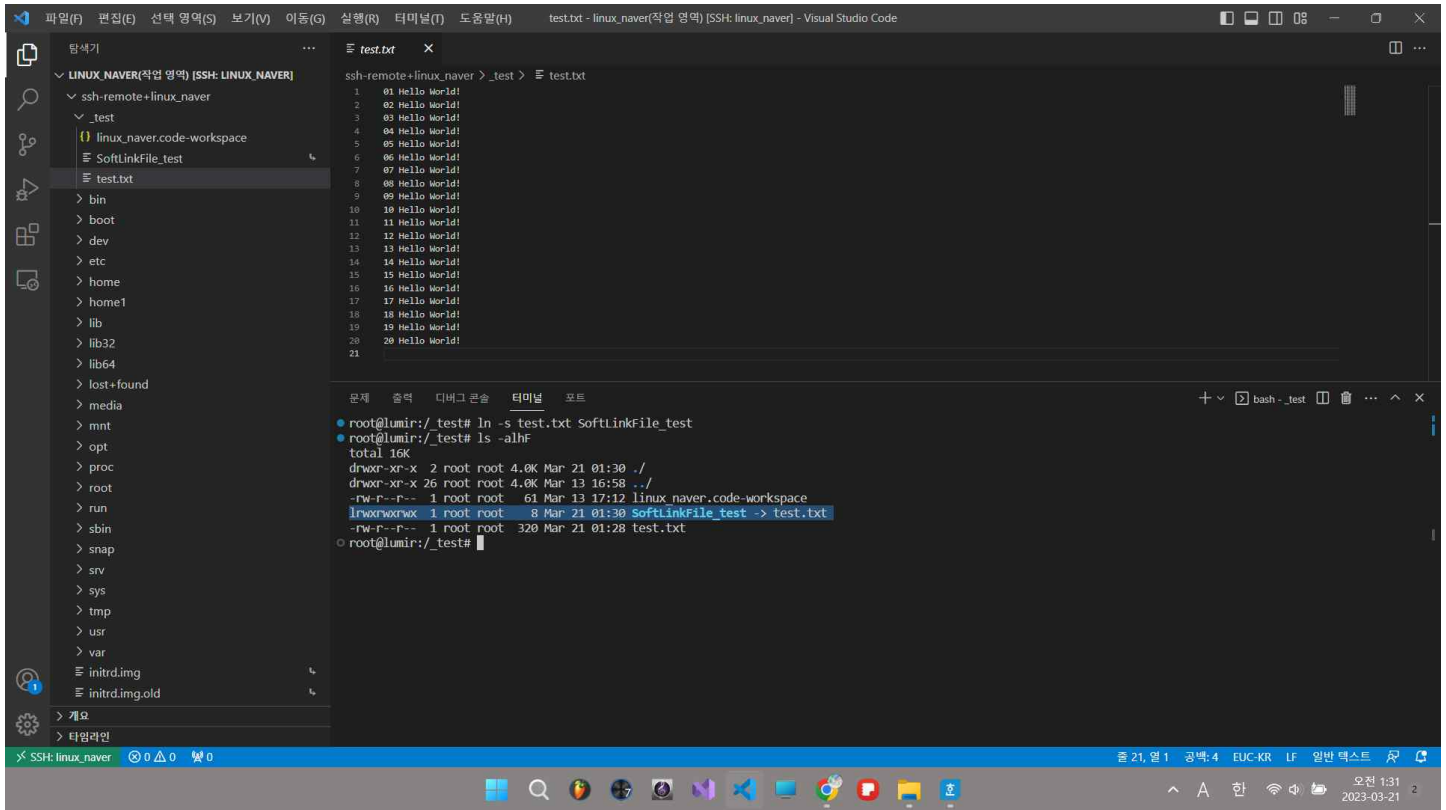
```
root@lumir:/test# rm -i test.txt
rm: remove regular file 'test.txt'? y
root@lumir:/test# cat HardLinkFile_test
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

[심볼릭링크 = 소프트링크 실습]

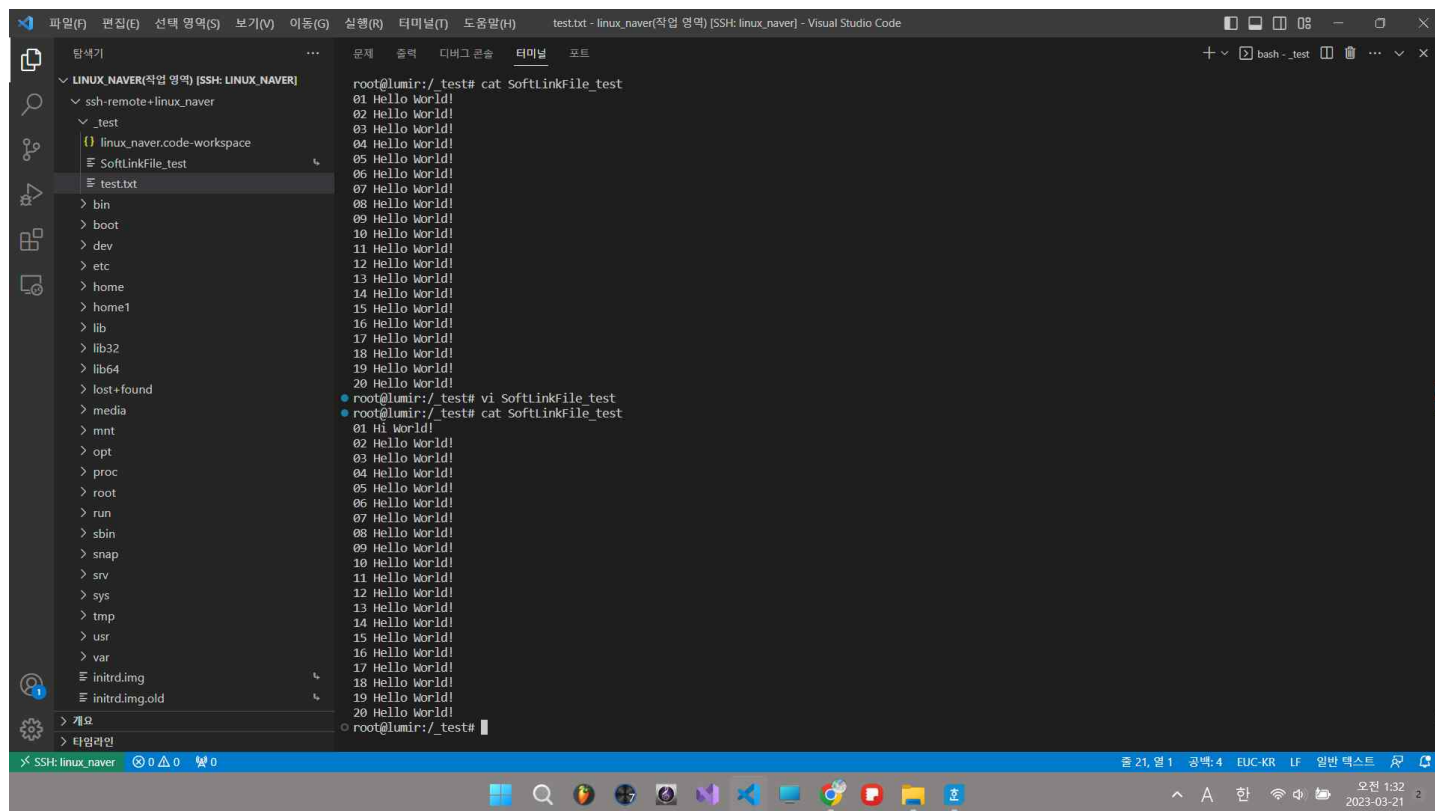
1-1. 임의의 텍스트 파일을 만든다. : test.txt 파일을 만들었다. (하드링크 실습 때 사용한 파일과 동일한 내용.)



1-2. 심볼릭 링크 파일을 만든다. : SoftLinkFile_test란 이름의 Symbolic Link(=Soft Link)를 만들었다.



2. 작성한 심볼릭링크 파일의 내용을 수정해 보자. : 01번 줄의 '01 Hello World!'를 '01 Hi World!'로 변경하였다.



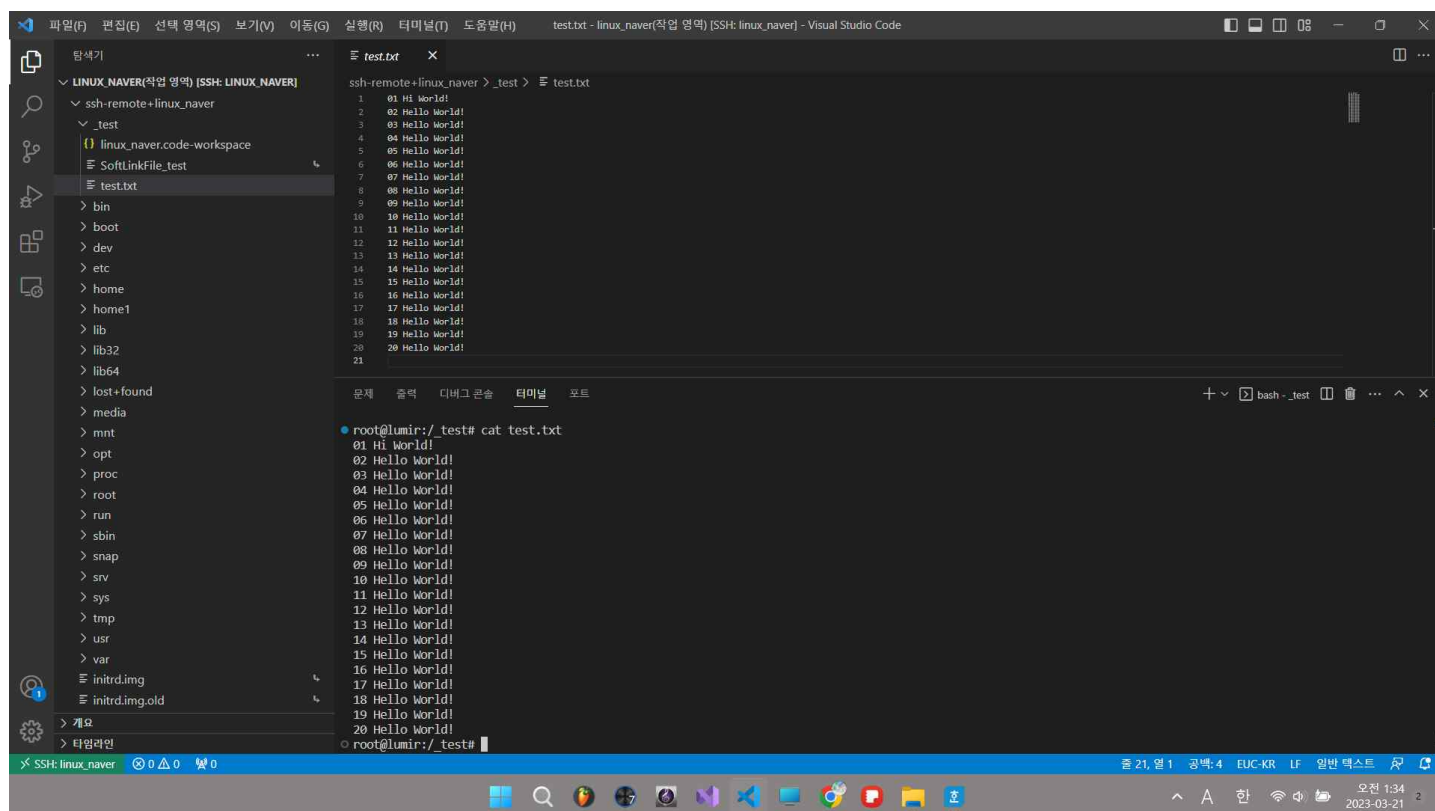
The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
root@lumir:/ test# cat SoftLinkFile_test
01 Hello World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!

root@lumir:/ test# vi SoftLinkFile_test
root@lumir:/ test# cat SoftLinkFile_test
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

The file explorer on the left shows the project structure, including the `SoftLinkFile_test` directory and the `test.txt` file.

3. 원본 파일을 확인해보자. : 원본 파일 역시 동일하게 수정되었다. (원본 파일 역시 01번 줄의 '01 Hello World!'가 '01 Hi World!'로 변경되었다.)



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
ssh-remote+linux_naver > _test > cat test.txt
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!

root@lumir:/ test# cat test.txt
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

The file explorer on the left shows the project structure, including the `test.txt` file.

[실습과제 #2]

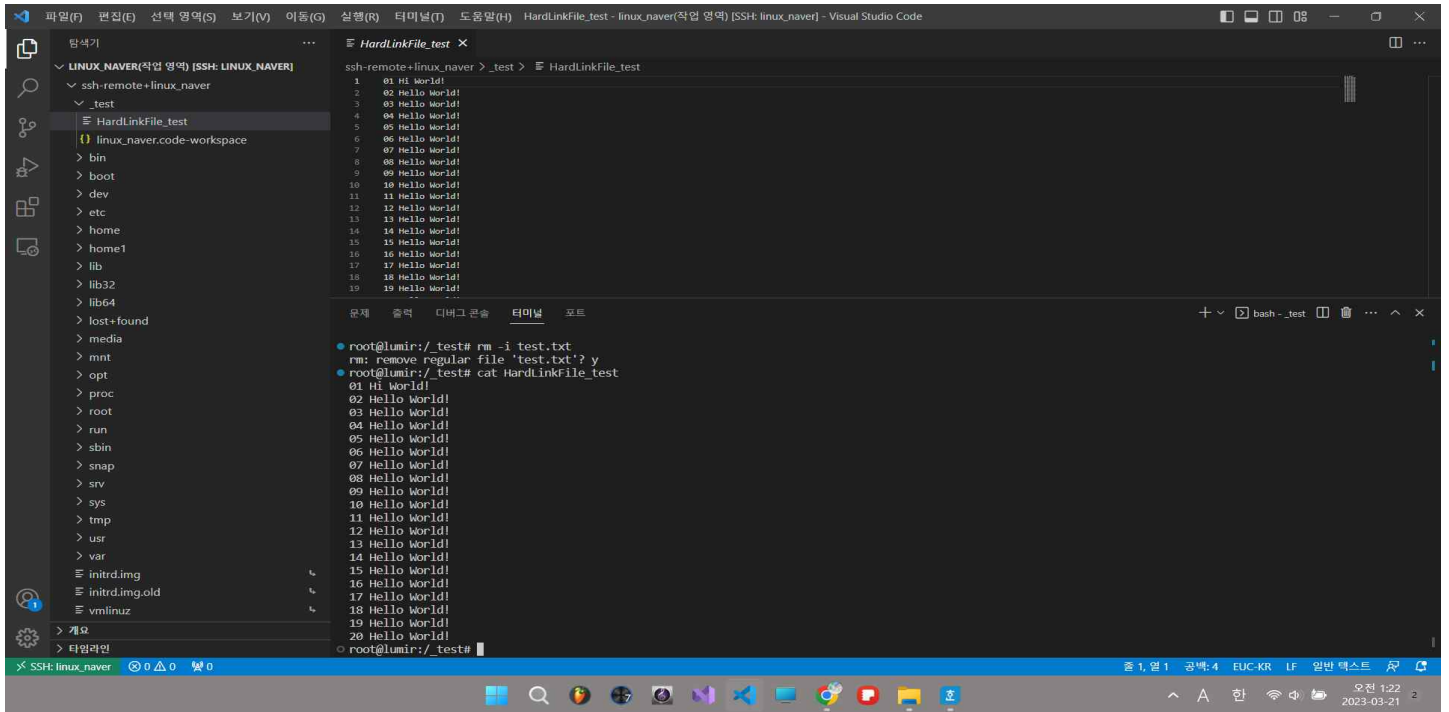
1. 원본파일과의 내용 대조 : 보고서의 1번 내용에 해당하는 나머지 부분들은 모두 앞선 페이지들에 서술하였다.

HardLinkFile_test(하드링크 파일)와 test.txt(원본 파일) : 두 파일 모두 동일하게 수정되었다.

SoftLinkFile_test(심볼릭링크 파일)와 test.txt(원본 파일) : 두 파일 모두 동일하게 수정되었다.

2. 원본파일 삭제.

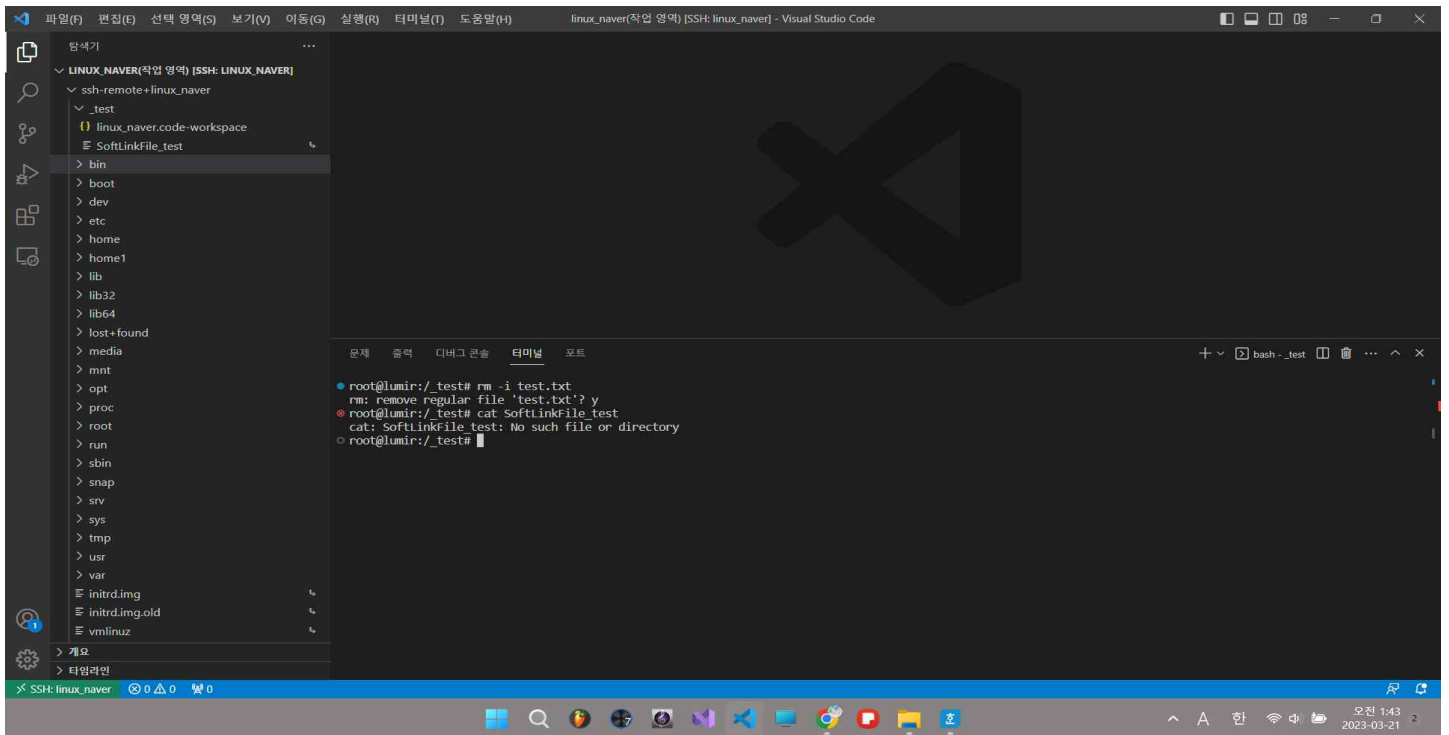
<Hard Link> : test.txt 파일의 출력결과와 동일한 결과가 출력된다.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
root@lumir:/test# rm -i test.txt
rm: remove regular file 'test.txt'? y
root@lumir:/test# cat HardLinkFile_test
01 Hi World!
02 Hello World!
03 Hello World!
04 Hello World!
05 Hello World!
06 Hello World!
07 Hello World!
08 Hello World!
09 Hello World!
10 Hello World!
11 Hello World!
12 Hello World!
13 Hello World!
14 Hello World!
15 Hello World!
16 Hello World!
17 Hello World!
18 Hello World!
19 Hello World!
20 Hello World!
```

<Symbolic Link> : 'No such file or directory'라고 표시되며, 파일이 열리지 않는다.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
root@lumir:/test# rm -i test.txt
rm: remove regular file 'test.txt'? y
root@lumir:/test# cat SoftLinkFile_test
cat: SoftLinkFile test: No such file or directory
root@lumir:/test#
```

3. 실습 수행 결과를 토대로, 링크 파일들을 사용할 때 예상되는 주의사항에 대해 서술하시오.

하드링크 파일은 원본을 삭제해도, 원본 파일의 내용에 접근하는 것에 아무런 문제가 없다. 하지만, 심볼릭링크 파일은 원본을 삭제하게 될 경우, 원본 파일의 내용에 접근할 수 없게 되므로 원본 파일 삭제 시 주의를 요한다. 또한, 하드링크 파일과 심볼릭링크 파일 모두 링크파일의 내용을 수정할 경우 원본 파일의 내용 역시도 수정되므로 이 부분도 주의해야 한다. 이외의 주의사항으로는, 하드링크 파일의 경우는 원본파일의 경로가 변경되더라도 링크가 깨지지 않는 반면, 심볼릭링크 파일의 경우는 원본파일의 경로가 바뀌게 되면 링크가 깨져버린다는 점이다. 또, 링크파일을 다수 사용하게 될 경우, 링크 수가 늘어나며 또 다른 링크를 찾기 어려워질 수 있기 때문에 파일 관리가 어려워진다.