

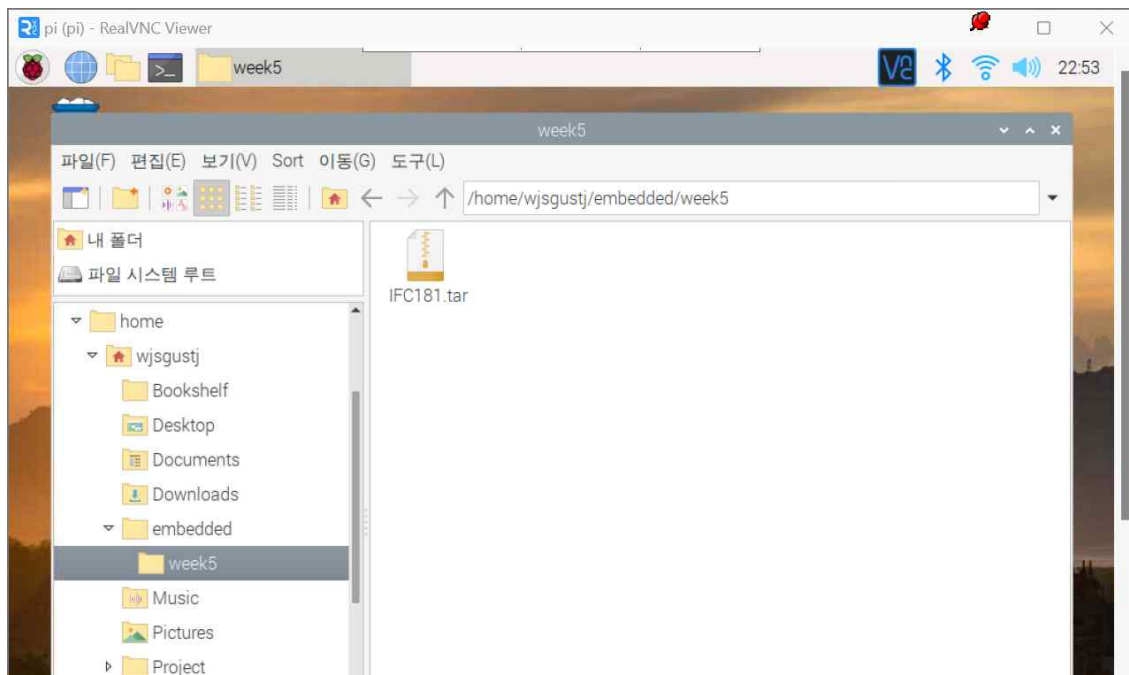
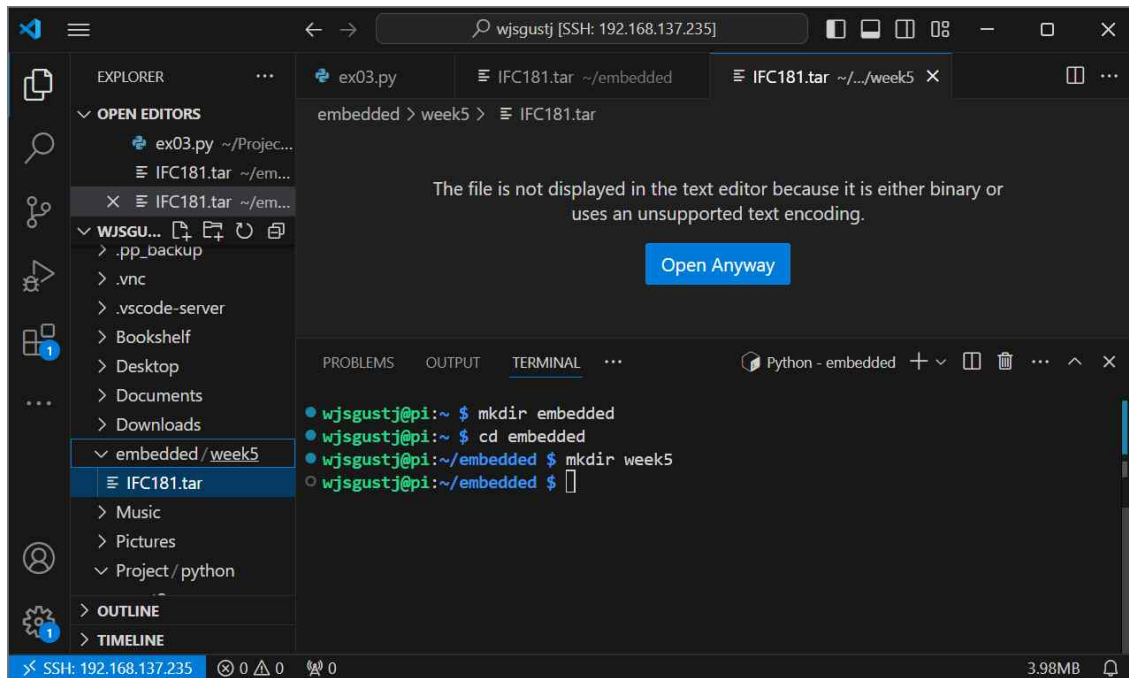
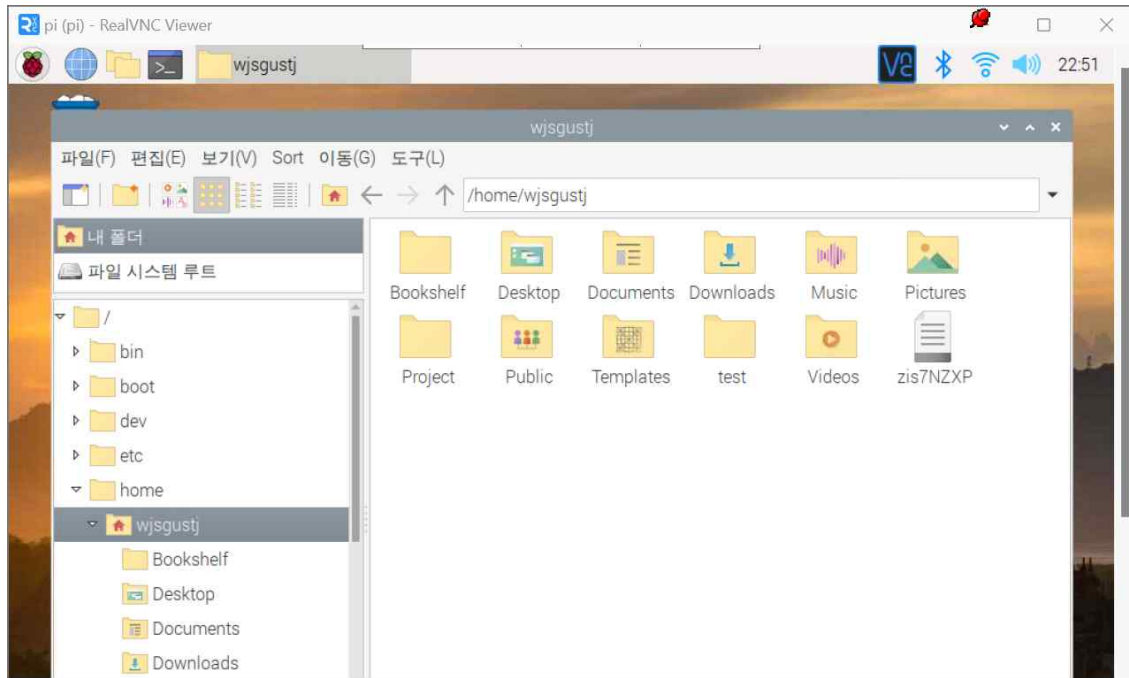
임베디드 응용 및 실습 과제1

- 1. 리눅스 간단 명령 수행하기

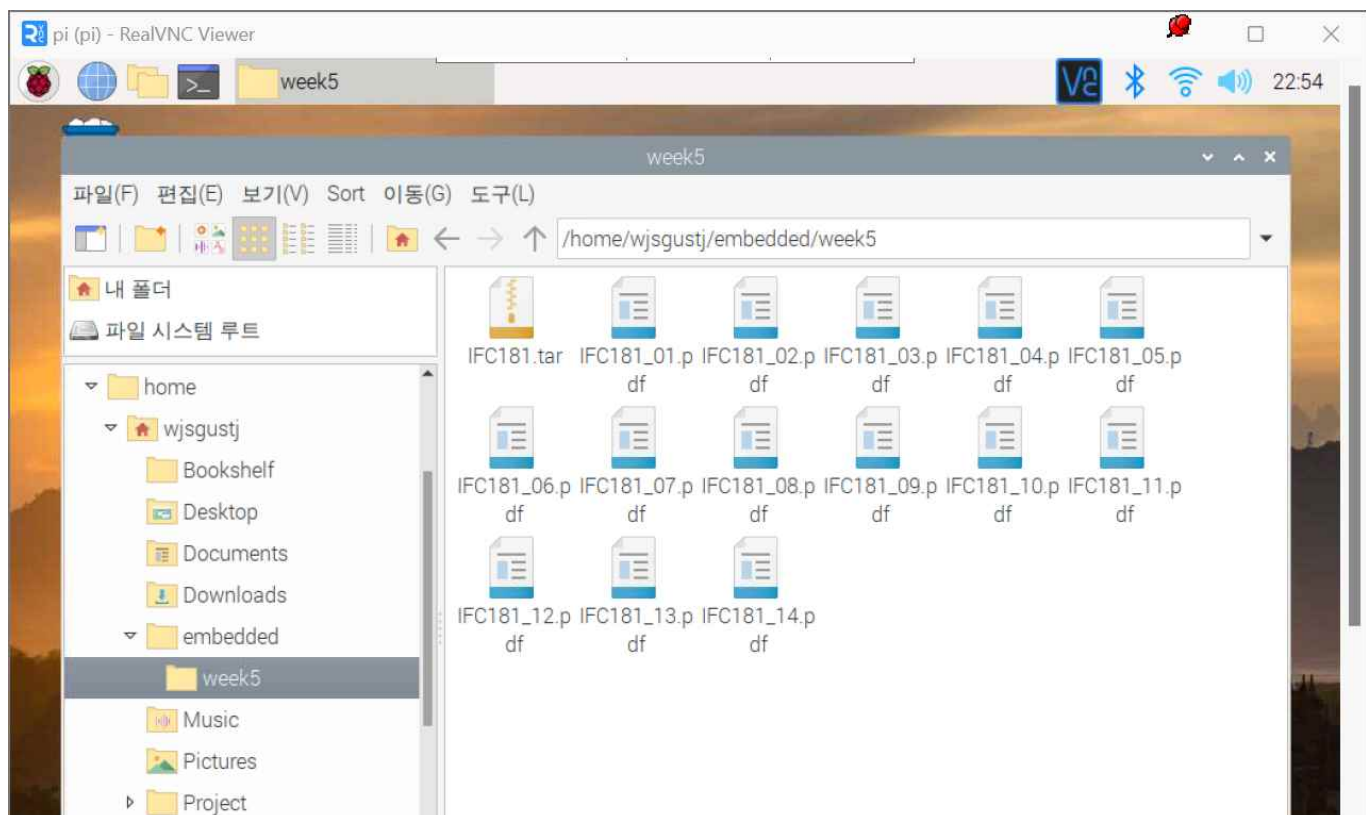
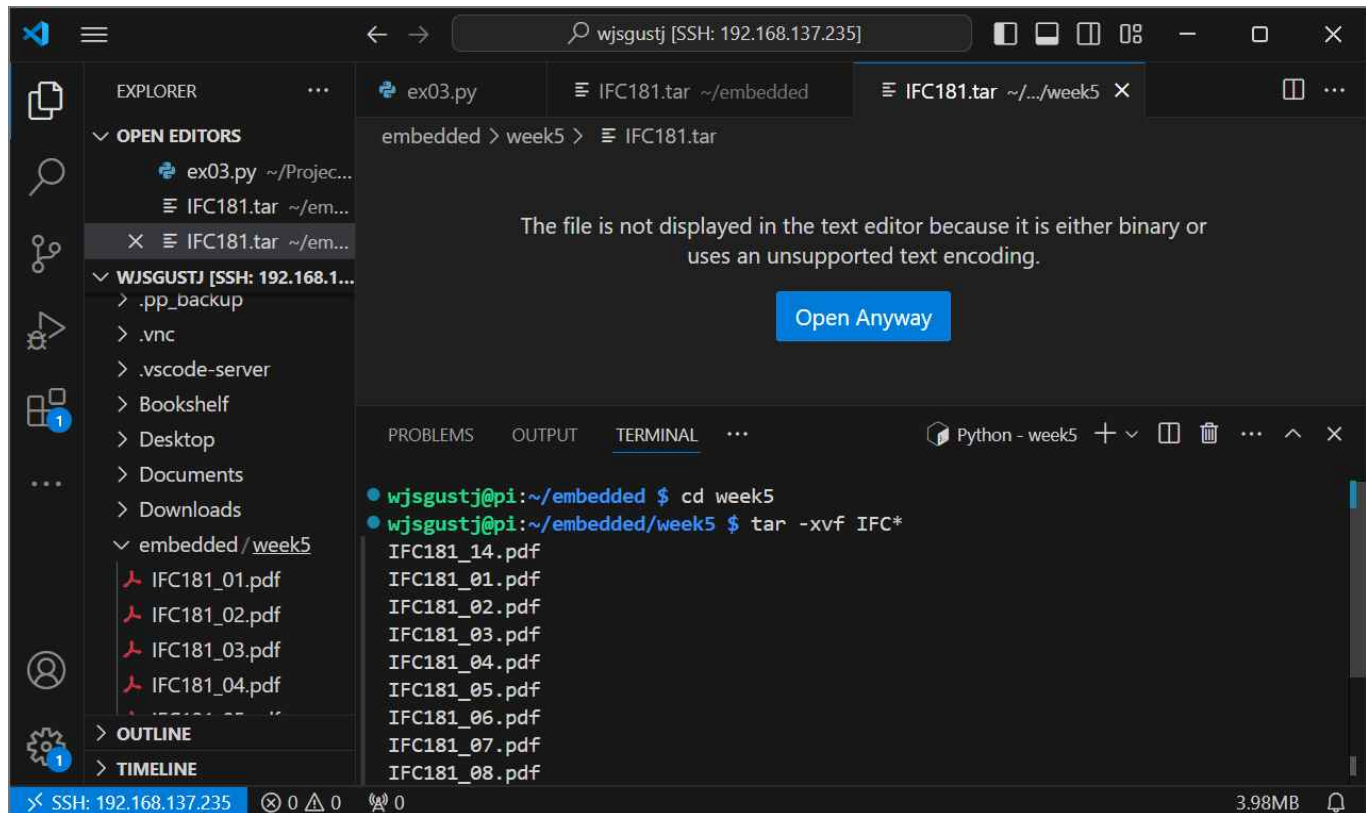
학과	전기공학과
학년	3학년
학번	2020161110
이름	전현서
과목명	임베디드 응용 및 실습
분반	01분반
담당교수님	우성민 교수님

<파일 압축 해제>

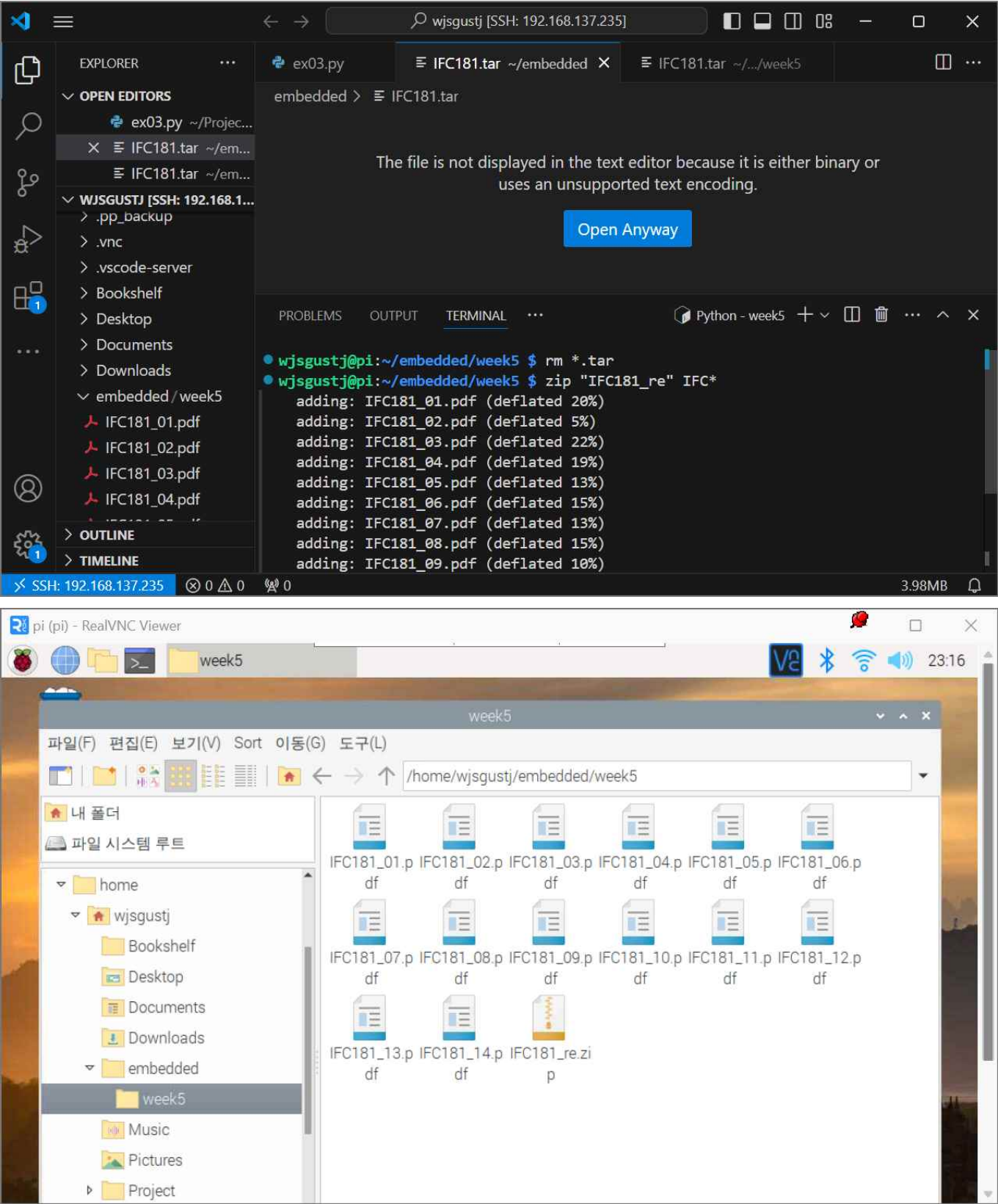
1. IFC181.tar 파일을 받아 ~/embedded/week5 로 옮긴다.



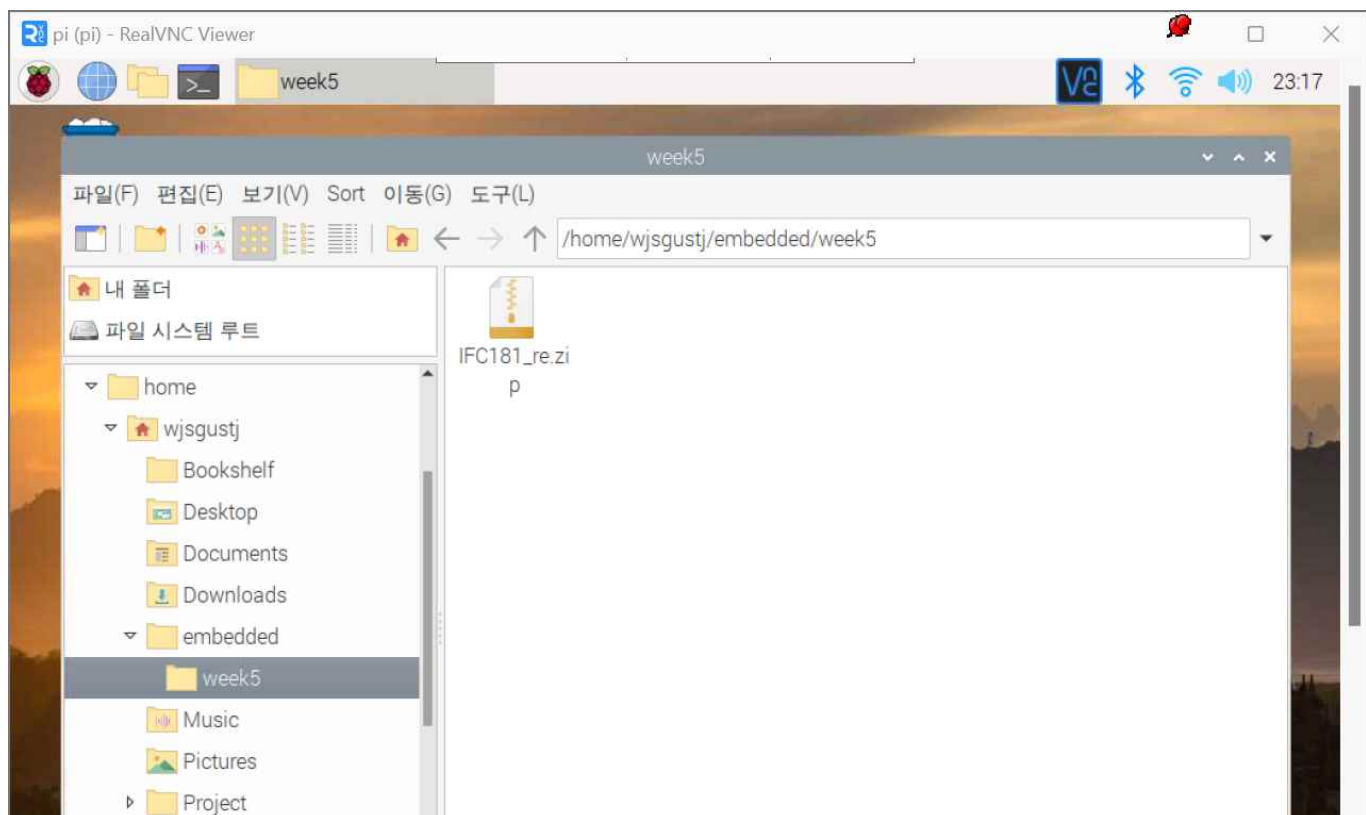
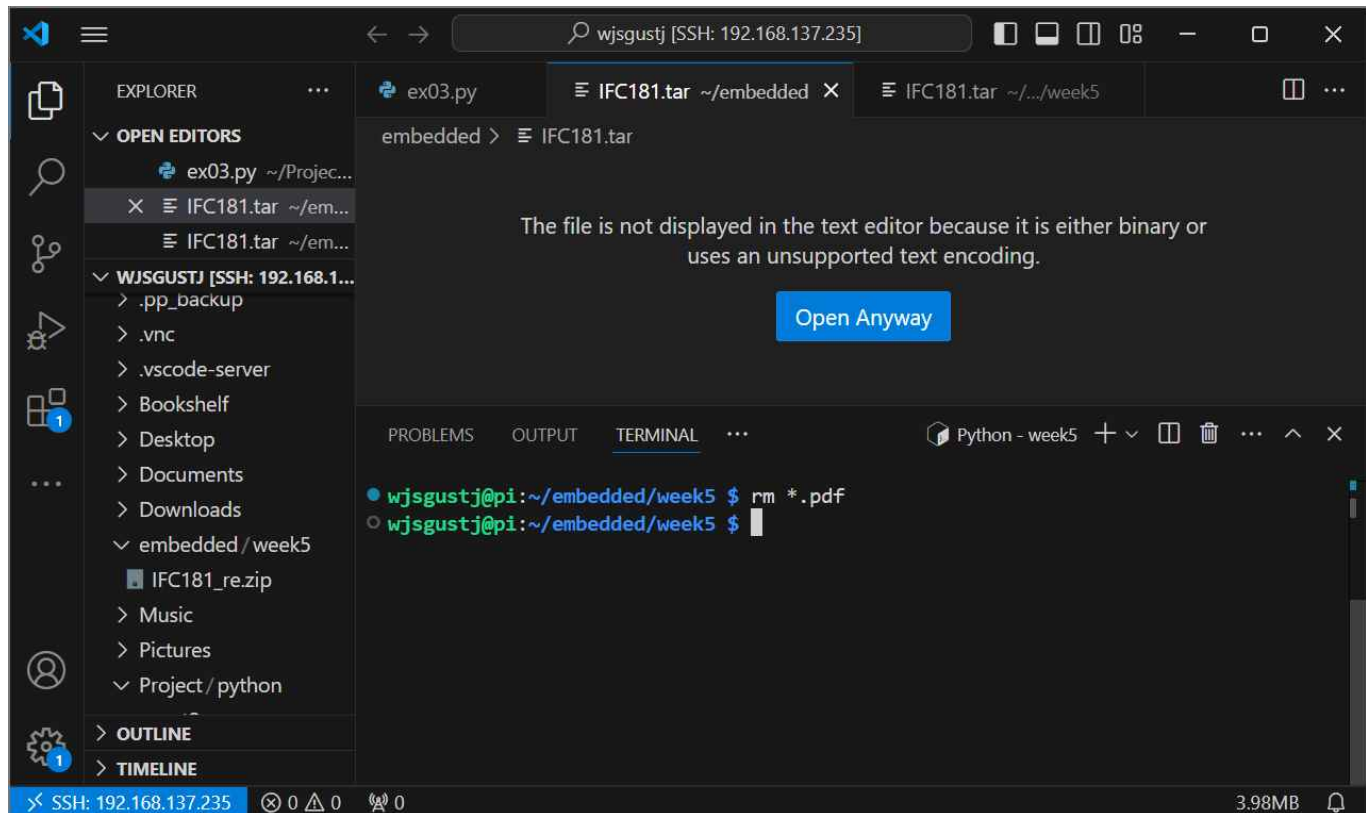
2. tar 명령으로 압축을 해제한다.

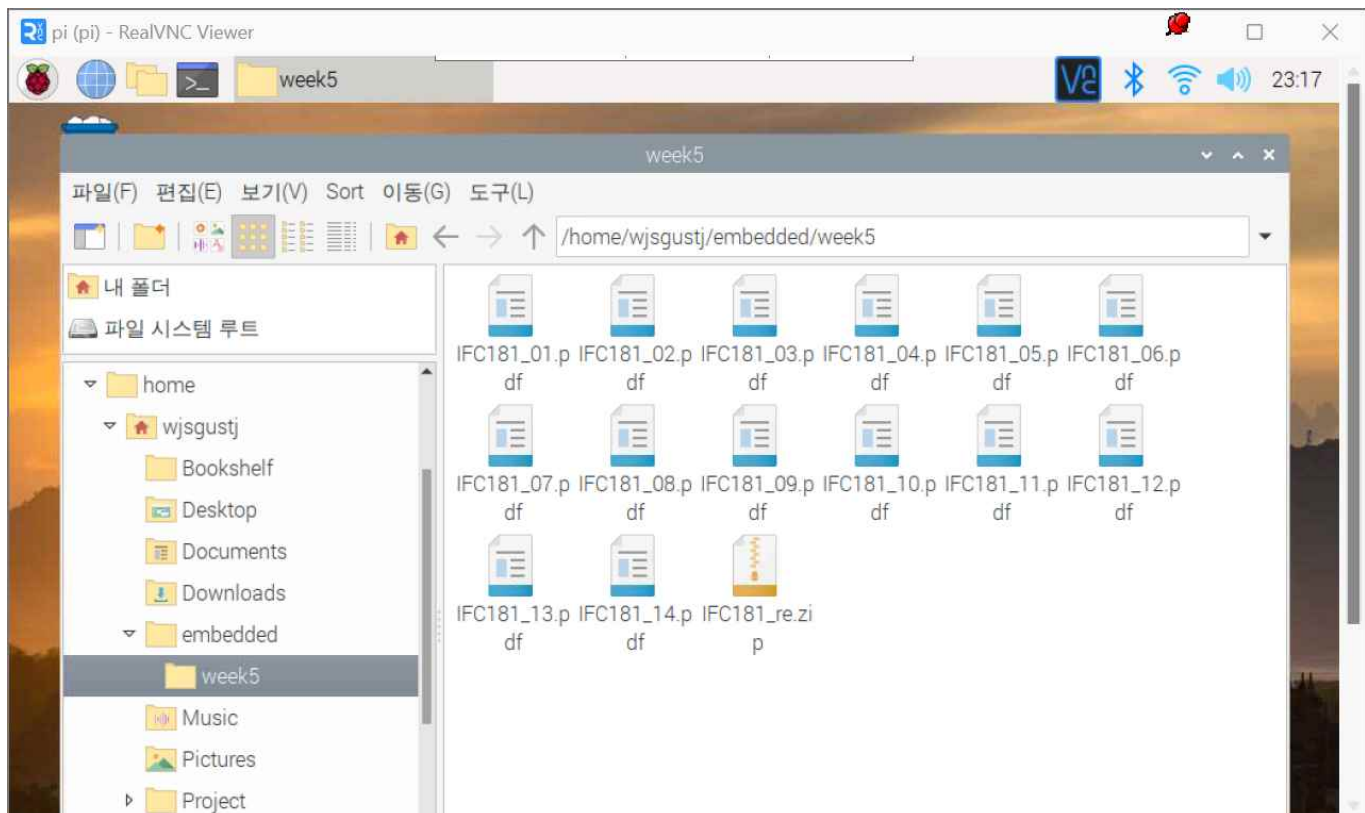
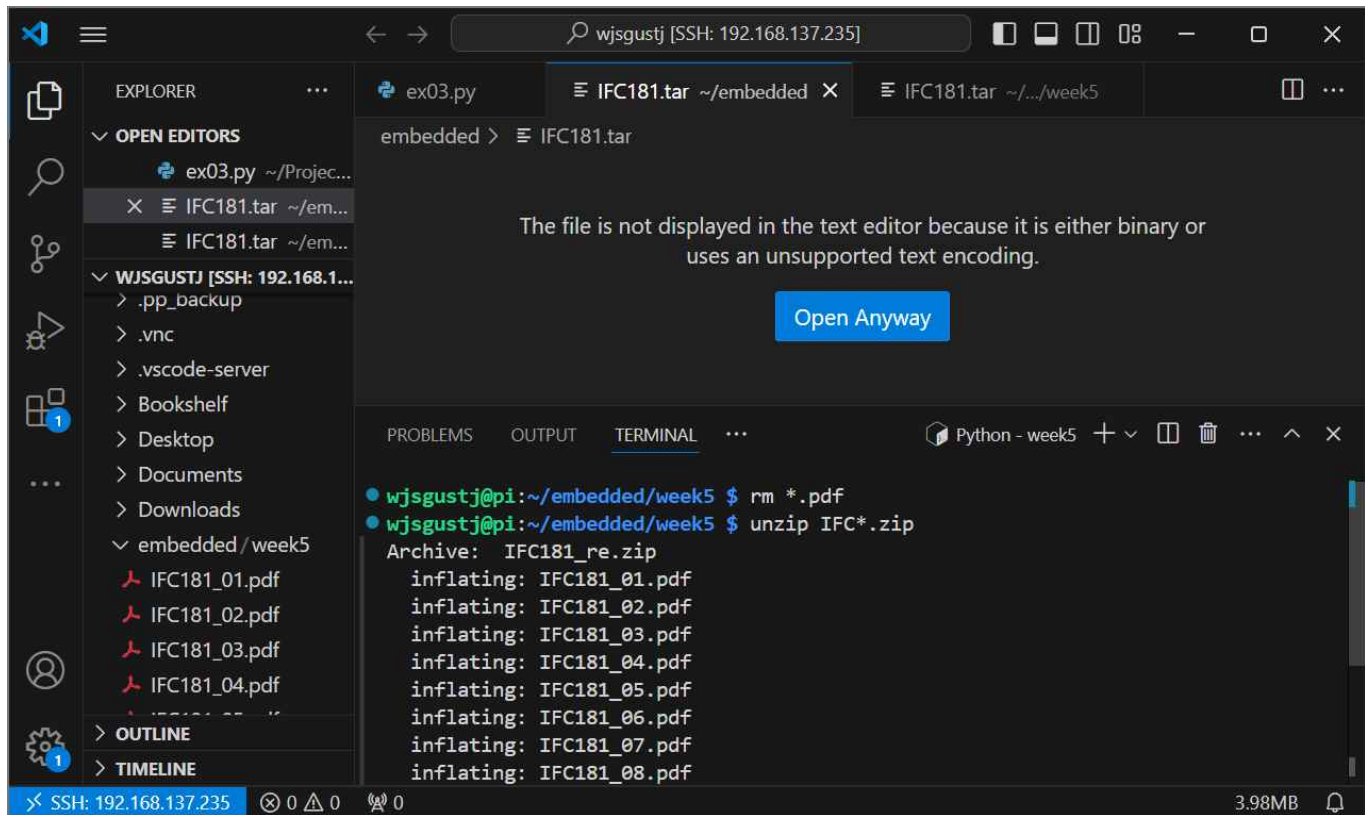


4. 2에서 압축해제된 파일들을 zip명령으로 IFC181_re.zip으로 압축한다.



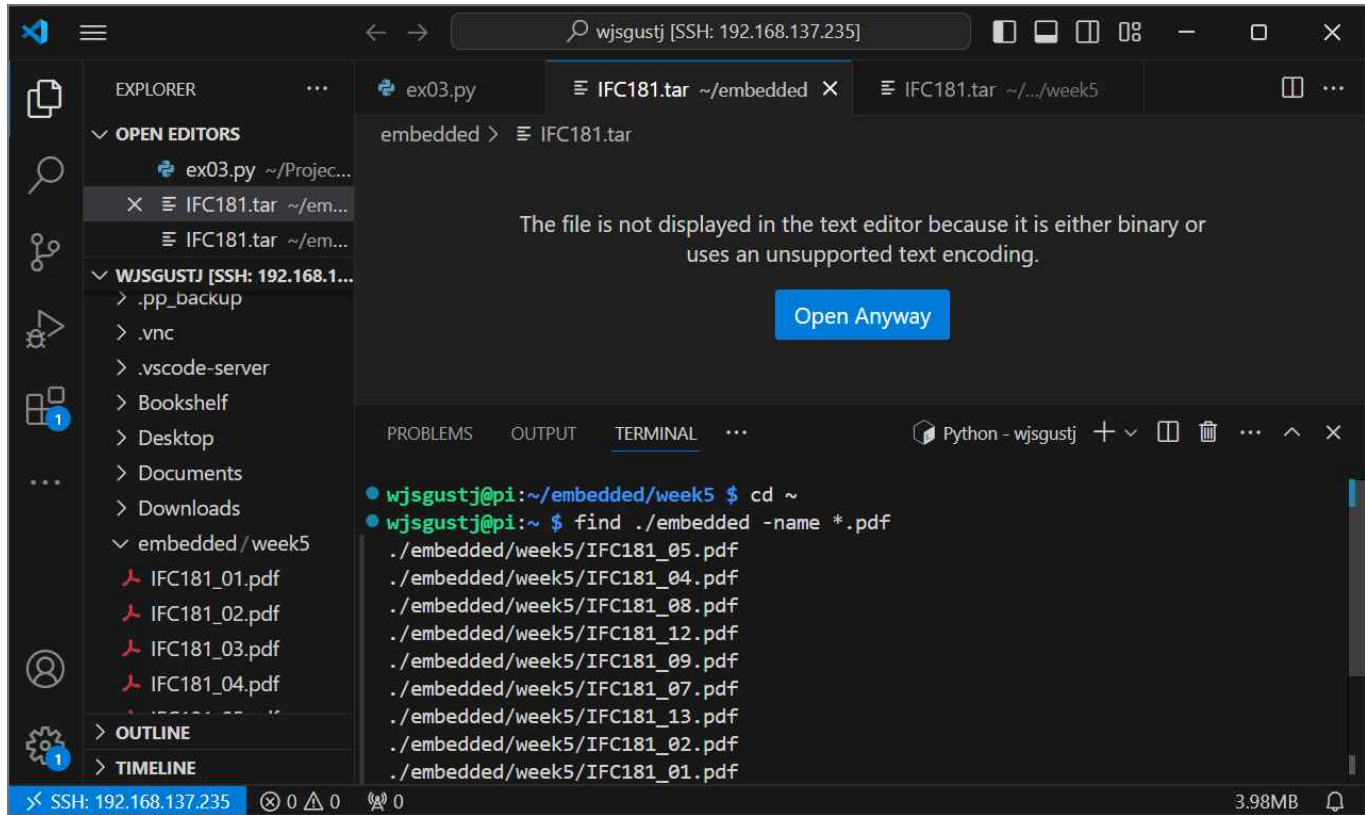
5. 4에서 나온 IFC181_re.zip 파일을 unzip 명령을 통해 압축해제 한다.





<find 명령어 연습>

6. ~ (home) 디렉토리로 이동하고 find 명령을 통해 embedded폴더에서(하위폴더 포함) .pdf 파일을 찾는 명령을 수행해본다.



7. 아래의 명령 결과가 동일한지 아닌지 결과를 보이고 동작 결과를 설명하시오

```
innosm@innosm:~ $find . -name *.pdf
innosm@innosm:~ $find ./ -name *.pdf
innosm@innosm:~ $find -name *.pdf
innosm@innosm:~ $find ~ -name *.pdf
innosm@innosm:~ $find /home/innosm -name *.pdf
```

```
wjsgustj [SSH: 192.168.137.235]
EXPLORER
OPEN EDITORS
ex03.py ~/Projec...
IFC181.tar ~/em...
IFC181.tar ~/em...
WJSGUSTJ [SSH: 192.168.1...
  .pp_backup
  .vnc
  .vscode-server
  Bookshelf
  Desktop
  Documents
  Downloads
  embedded / week5
    IFC181_01.pdf
    IFC181_02.pdf
    IFC181_03.pdf
    IFC181_04.pdf
    IFC181_05.pdf
    IFC181_06.pdf
    IFC181_07.pdf
    IFC181_08.pdf
    IFC181_09.pdf
    IFC181_10.pdf
  IFC181.tar
  IFC181.tar ~/../week5
embedded > IFC181.tar
The file is not displayed in the text editor because it is either binary or
uses an unsupported text encoding.
Open Anyway
PROBLEMS OUTPUT TERMINAL
Python - wjsgustj
wjsgustj@pi:~ $ find . -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week5/IFC181_05.pdf
./embedded/week5/IFC181_04.pdf
./embedded/week5/IFC181_08.pdf
./embedded/week5/IFC181_12.pdf
./embedded/week5/IFC181_09.pdf
./embedded/week5/IFC181_07.pdf
./embedded/week5/IFC181_13.pdf
./embedded/week5/IFC181_02.pdf
./embedded/week5/IFC181_01.pdf
./embedded/week5/IFC181_14.pdf
./embedded/week5/IFC181_11.pdf
./embedded/week5/IFC181_10.pdf
./embedded/week5/IFC181_03.pdf
./embedded/week5/IFC181_06.pdf
wjsgustj@pi:~ $
```

```
wjsgustj [SSH: 192.168.137.235]
EXPLORER
OPEN EDITORS
ex03.py ~/Projec...
IFC181.tar ~/em...
IFC181.tar ~/em...
WJSGUSTJ [SSH: 192.168.1...
  .pp_backup
  .vnc
  .vscode-server
  Bookshelf
  Desktop
  Documents
  Downloads
  embedded / week5
    IFC181_01.pdf
    IFC181_02.pdf
    IFC181_03.pdf
    IFC181_04.pdf
    IFC181_05.pdf
    IFC181_06.pdf
    IFC181_07.pdf
    IFC181_08.pdf
    IFC181_09.pdf
    IFC181_10.pdf
  IFC181.tar
  IFC181.tar ~/../week5
embedded > IFC181.tar
The file is not displayed in the text editor because it is either binary or
uses an unsupported text encoding.
Open Anyway
PROBLEMS OUTPUT TERMINAL
Python - wjsgustj
wjsgustj@pi:~ $ find ./ -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week5/IFC181_05.pdf
./embedded/week5/IFC181_04.pdf
./embedded/week5/IFC181_08.pdf
./embedded/week5/IFC181_12.pdf
./embedded/week5/IFC181_09.pdf
./embedded/week5/IFC181_07.pdf
./embedded/week5/IFC181_13.pdf
./embedded/week5/IFC181_02.pdf
./embedded/week5/IFC181_01.pdf
./embedded/week5/IFC181_14.pdf
./embedded/week5/IFC181_11.pdf
./embedded/week5/IFC181_10.pdf
./embedded/week5/IFC181_03.pdf
./embedded/week5/IFC181_06.pdf
wjsgustj@pi:~ $
```


VS Code interface showing a remote connection to `wjsgustj [SSH: 192.168.137.235]`. The Explorer sidebar shows the file structure, including `embedded > IFC181.tar`. The main editor displays a message: "The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding." with an **Open Anyway** button.

The Terminal panel shows the command `find ~ -name *.pdf` and its output:

```
wjsgustj@pi:~ $ find ~ -name *.pdf
/home/wjsgustj/Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
/home/wjsgustj/embedded/week5/IFC181_05.pdf
/home/wjsgustj/embedded/week5/IFC181_04.pdf
/home/wjsgustj/embedded/week5/IFC181_08.pdf
/home/wjsgustj/embedded/week5/IFC181_12.pdf
/home/wjsgustj/embedded/week5/IFC181_09.pdf
/home/wjsgustj/embedded/week5/IFC181_07.pdf
/home/wjsgustj/embedded/week5/IFC181_13.pdf
/home/wjsgustj/embedded/week5/IFC181_02.pdf
/home/wjsgustj/embedded/week5/IFC181_01.pdf
/home/wjsgustj/embedded/week5/IFC181_14.pdf
/home/wjsgustj/embedded/week5/IFC181_11.pdf
/home/wjsgustj/embedded/week5/IFC181_10.pdf
/home/wjsgustj/embedded/week5/IFC181_03.pdf
/home/wjsgustj/embedded/week5/IFC181_06.pdf
wjsgustj@pi:~ $
```

VS Code interface showing a remote connection to `wjsgustj [SSH: 192.168.137.235]`. The Explorer sidebar shows the file structure, including `embedded > IFC181.tar`. The main editor displays a message: "The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding." with an **Open Anyway** button.

The Terminal panel shows the command `find -name *.pdf` and its output:

```
wjsgustj@pi:~ $ find -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week5/IFC181_05.pdf
./embedded/week5/IFC181_04.pdf
./embedded/week5/IFC181_08.pdf
./embedded/week5/IFC181_12.pdf
./embedded/week5/IFC181_09.pdf
./embedded/week5/IFC181_07.pdf
./embedded/week5/IFC181_13.pdf
./embedded/week5/IFC181_02.pdf
./embedded/week5/IFC181_01.pdf
./embedded/week5/IFC181_14.pdf
./embedded/week5/IFC181_11.pdf
./embedded/week5/IFC181_10.pdf
./embedded/week5/IFC181_03.pdf
./embedded/week5/IFC181_06.pdf
wjsgustj@pi:~ $
```

VS Code interface showing a remote connection to wjsgustj [SSH: 192.168.137.235].

EXPLORER

- OPEN EDITORS (1 unsaved)
 - ex03.py ~/Projec...
- WJSGUSTJ [SSH: 192.168.1...]
 - .pp_backup
 - .vnc
 - .vscode-server
 - Bookshelf
 - Desktop
 - Documents
 - Downloads
 - embedded/week5
 - IFC181_01.pdf
 - IFC181_02.pdf
 - IFC181_03.pdf
 - IFC181_04.pdf
 - IFC181_05.pdf
 - IFC181_06.pdf
 - IFC181_07.pdf
 - IFC181_08.pdf
 - IFC181_09.pdf
 - IFC181_10.pdf
 - IFC181_11.pdf
 - IFC181_12.pdf
 - IFC181_13.pdf
 - IFC181_14.pdf
 - IFC181_re.zip
 - temp_file
 - Music
 - Pictures
 - Project/python
 - part2
 - ex03.py
 - temporary.zip

ex03.py

```
Project > python > part2 > ex03.py
1 import ctypes
2 print(ctypes.sizeof(ctypes.c_void_p), "bytes")
```

TERMINAL

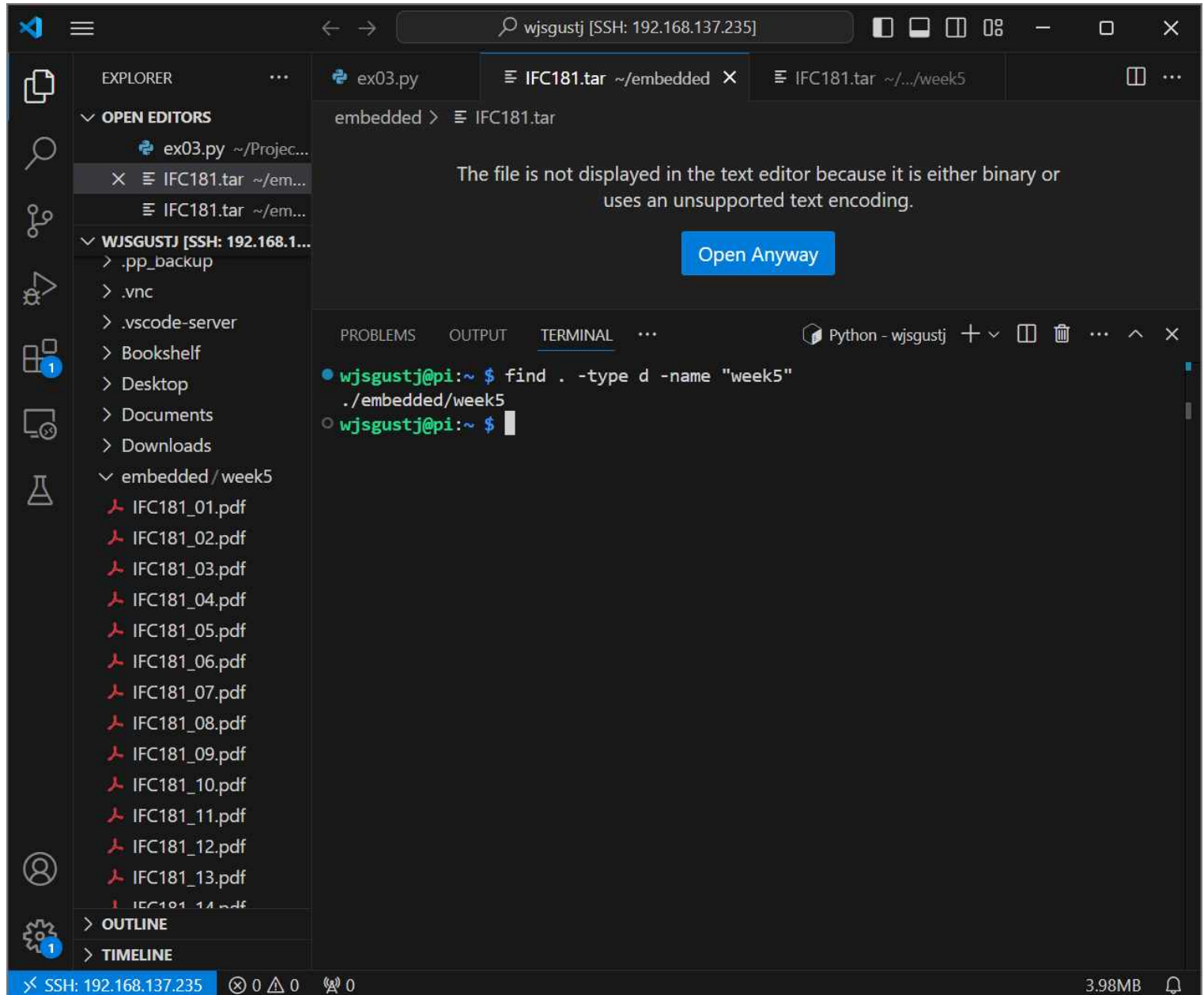
```
bash - wjsgustj
wjsgustj@pi:~$ find /home/wjsgustj -name *.pdf
/home/wjsgustj/Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
/home/wjsgustj/embedded/week5/IFC181_05.pdf
/home/wjsgustj/embedded/week5/IFC181_04.pdf
/home/wjsgustj/embedded/week5/IFC181_08.pdf
/home/wjsgustj/embedded/week5/IFC181_12.pdf
/home/wjsgustj/embedded/week5/IFC181_09.pdf
/home/wjsgustj/embedded/week5/IFC181_07.pdf
/home/wjsgustj/embedded/week5/IFC181_13.pdf
/home/wjsgustj/embedded/week5/IFC181_02.pdf
/home/wjsgustj/embedded/week5/IFC181_01.pdf
/home/wjsgustj/embedded/week5/IFC181_14.pdf
/home/wjsgustj/embedded/week5/IFC181_11.pdf
/home/wjsgustj/embedded/week5/IFC181_10.pdf
/home/wjsgustj/embedded/week5/IFC181_03.pdf
/home/wjsgustj/embedded/week5/IFC181_06.pdf
wjsgustj@pi:~$
```

1. innosm@innosm:~ \$find . -name *.pdf
2. innosm@innosm:~ \$find ./ -name *.pdf
3. innosm@innosm:~ \$find -name *.pdf
4. innosm@innosm:~ \$find ~ -name *.pdf
5. innosm@innosm:~ \$find /home/innosm -name *.pdf

번호	현재 폴더	경로 명령어	명령어 설명	검색 대상
1	Home directory	.	현재 디렉토리	Home directory 로 모두 동일
2	Home directory	./	현재 디렉토리 하 경로를 지정하지 않음	
3	Home directory		경로를 지정하지 않았으므로 현재 디렉토리	
4	Home directory	~	Home directory	
5	Home directory	/home/wjsgustj	Home directory를 직접 주소 입력	

8. ~ (home) 디렉토리로 이동하고, week5 폴더가 있는지 검색하려고 한다.

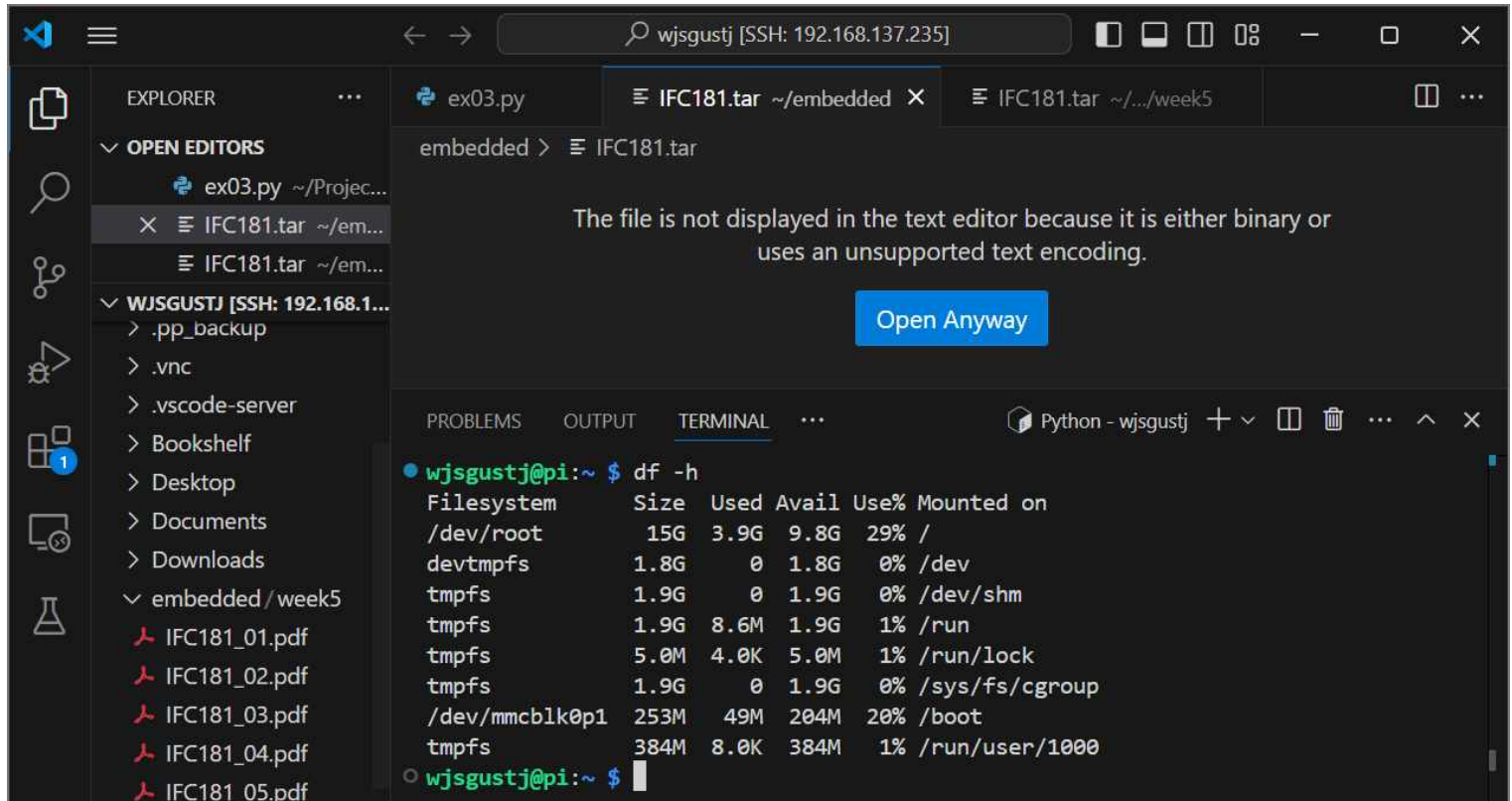
적절한 명령을 수행하여 week5 폴더를 검색하고 결과를 보이시오.



<파일 용량 확인>

9. 아래 명령을 수행한 결과를 보이시오.

```
df -h
```



The screenshot shows the VS Code interface with a terminal window open. The terminal displays the output of the 'df -h' command, showing disk usage for various filesystems.

```
wjsgustj [SSH: 192.168.137.235]
```

EXPLORER: OPEN EDITORS: ex03.py, IFC181.tar ~/em..., IFC181.tar ~/em... WJSGUSTJ [SSH: 192.168.1...]: .pp_backup, .vnc, .vscode-server, Bookshelf, Desktop, Documents, Downloads, embedded/week5, IFC181_01.pdf, IFC181_02.pdf, IFC181_03.pdf, IFC181_04.pdf, IFC181_05.pdf

embedded > IFC181.tar

The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding.

Open Anyway

PROBLEMS OUTPUT TERMINAL Python - wjsgustj

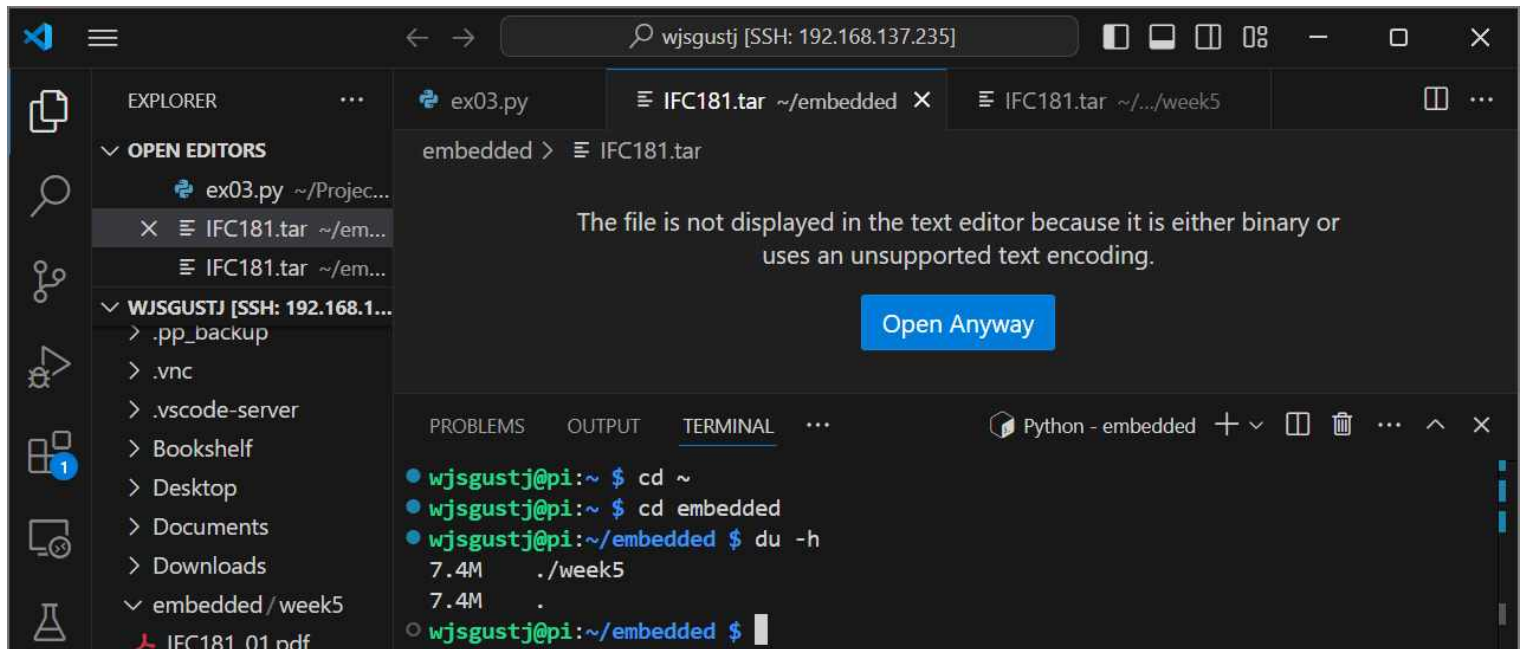
```
wjsgustj@pi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G   3.9G   9.8G  29% /
devtmpfs         1.8G     0   1.8G   0% /dev
tmpfs            1.9G     0   1.9G   0% /dev/shm
tmpfs            1.9G   8.6M   1.9G   1% /run
tmpfs            5.0M   4.0K   5.0M   1% /run/lock
tmpfs            1.9G     0   1.9G   0% /sys/fs/cgroup
/dev/mmcblk0p1  253M   49M  204M  20% /boot
tmpfs            384M   8.0K   384M   1% /run/user/1000
```

10. 아래 명령을 수행한 결과를 보이시오.

```
cd ~
```

```
cd embedded
```

```
du -h
```



The screenshot shows the VS Code interface with a terminal window open. The terminal displays the output of the 'cd ~', 'cd embedded', and 'du -h' commands.

```
wjsgustj [SSH: 192.168.137.235]
```

EXPLORER: OPEN EDITORS: ex03.py, IFC181.tar ~/em..., IFC181.tar ~/em... WJSGUSTJ [SSH: 192.168.1...]: .pp_backup, .vnc, .vscode-server, Bookshelf, Desktop, Documents, Downloads, embedded/week5, IFC181_01.pdf

embedded > IFC181.tar

The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding.

Open Anyway

PROBLEMS OUTPUT TERMINAL Python - embedded

```
wjsgustj@pi:~$ cd ~
wjsgustj@pi:~$ cd embedded
wjsgustj@pi:~/embedded$ du -h
7.4M  ./week5
7.4M  .
wjsgustj@pi:~/embedded$
```

11. 10번에서 현재 embedded 폴더의 총 사용량은(하위폴더 포함, 단위 표시)?

The screenshot shows a VS Code editor window with a dark theme. The Explorer sidebar on the left displays a file tree with folders like `.cache`, `.config`, `.cups`, `.dotnet`, `.gnupg`, `.local`, `.pki`, `.pp_backup`, `.vnc`, `.vscode-server`, `Bookshelf`, `Desktop`, `Documents`, `Downloads`, and `embedded/week5`. The `embedded/week5` folder is expanded, showing a list of PDF files and a ZIP file. The Open Editors sidebar shows `ex03.py` as the active file. The main editor area displays the content of `ex03.py`:

```
Project > python > part2 > ex03.py
1 import ctypes
2 print(ctypes.sizeof(ctypes.c_void_p), "bytes")
```

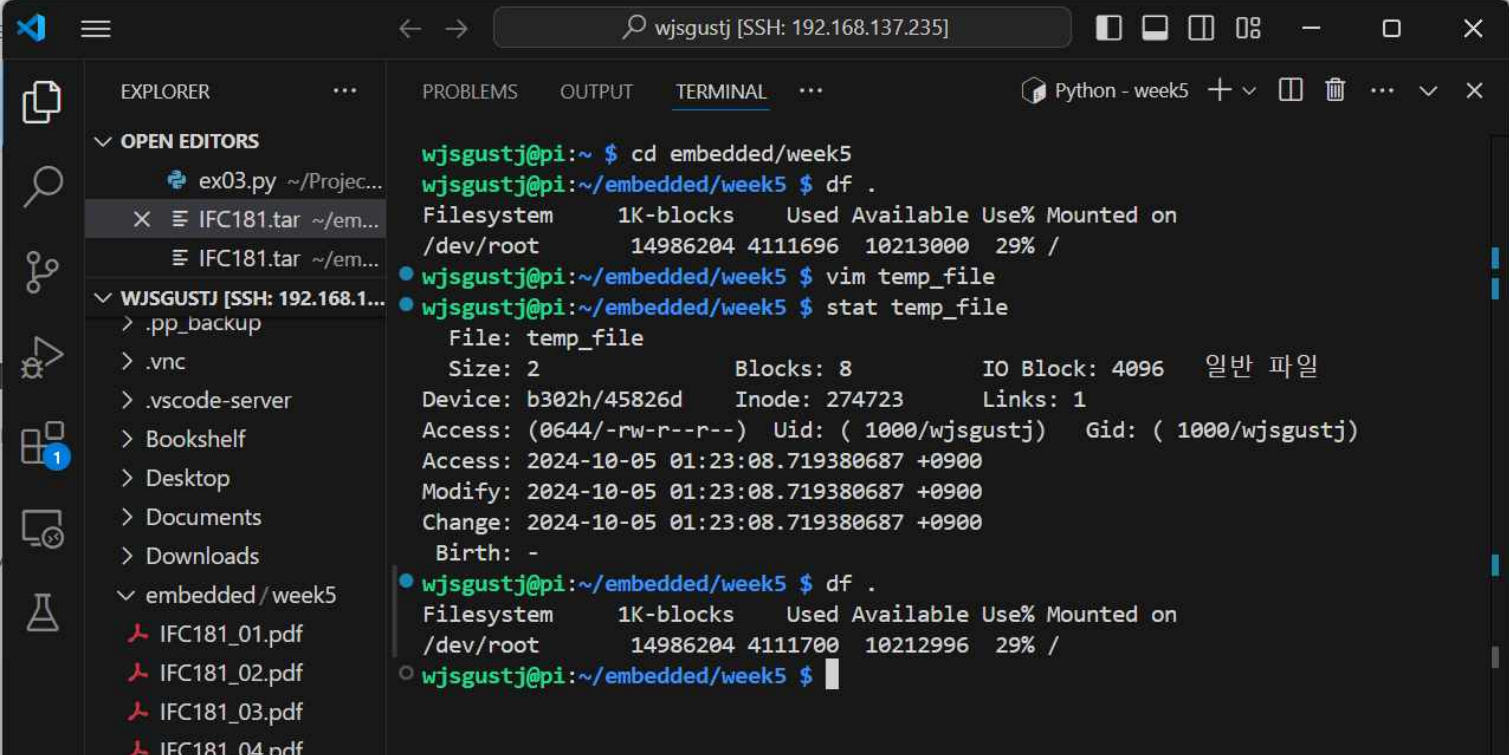
Below the editor, the TERMINAL panel is active, showing a shell prompt `wjsgustj@pi:~/embedded` and the output of the `du -ha` command:

```
wjsgustj@pi:~/embedded $ du -ha
128K    ./week5/IFC181_05.pdf
72K     ./week5/IFC181_04.pdf
116K    ./week5/IFC181_08.pdf
232K    ./week5/IFC181_12.pdf
3.4M    ./week5/IFC181_re.zip
252K    ./week5/IFC181_09.pdf
248K    ./week5/IFC181_07.pdf
200K    ./week5/IFC181_13.pdf
584K    ./week5/IFC181_02.pdf
124K    ./week5/IFC181_01.pdf
112K    ./week5/IFC181_14.pdf
280K    ./week5/IFC181_11.pdf
160K    ./week5/IFC181_10.pdf
1.5M    ./week5/IFC181_03.pdf
136K    ./week5/IFC181_06.pdf
4.0K    ./week5/temp_file
7.4M    ./week5
7.4M    .
wjsgustj@pi:~/embedded $
```

The status bar at the bottom indicates the current file is `ex03.py`, the encoding is UTF-8, and the Python version is 3.7.3 32-bit.

12. cd embedded/week5 를 수행하여 위치를 이동하고, 아래 명령을 차례로 수행하시오

- 1) df .
 - 2) 이 폴더에 temp_file 파일 생성하고 파일에 1을 기록
 - 3) stat temp_file 명령을 통해 파일의 크기를 확인
 - 4) df .
- 1)과 4)에서 줄어든 용량과 3)에서 확인한 용량이 다르다면 그 이유는?



The screenshot shows a terminal window with the following commands and output:

```
wjsgustj@pi:~ $ cd embedded/week5
wjsgustj@pi:~/embedded/week5 $ df .
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        14986204 4111696  10213000  29% /

wjsgustj@pi:~/embedded/week5 $ vim temp_file
wjsgustj@pi:~/embedded/week5 $ stat temp_file
File: temp_file
Size: 2          Blocks: 8          IO Block: 4096   일반 파일
Device: b302h/45826d Inode: 274723       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/wjsgustj)   Gid: ( 1000/wjsgustj)
Access: 2024-10-05 01:23:08.719380687 +0900
Modify: 2024-10-05 01:23:08.719380687 +0900
Change: 2024-10-05 01:23:08.719380687 +0900
Birth: -

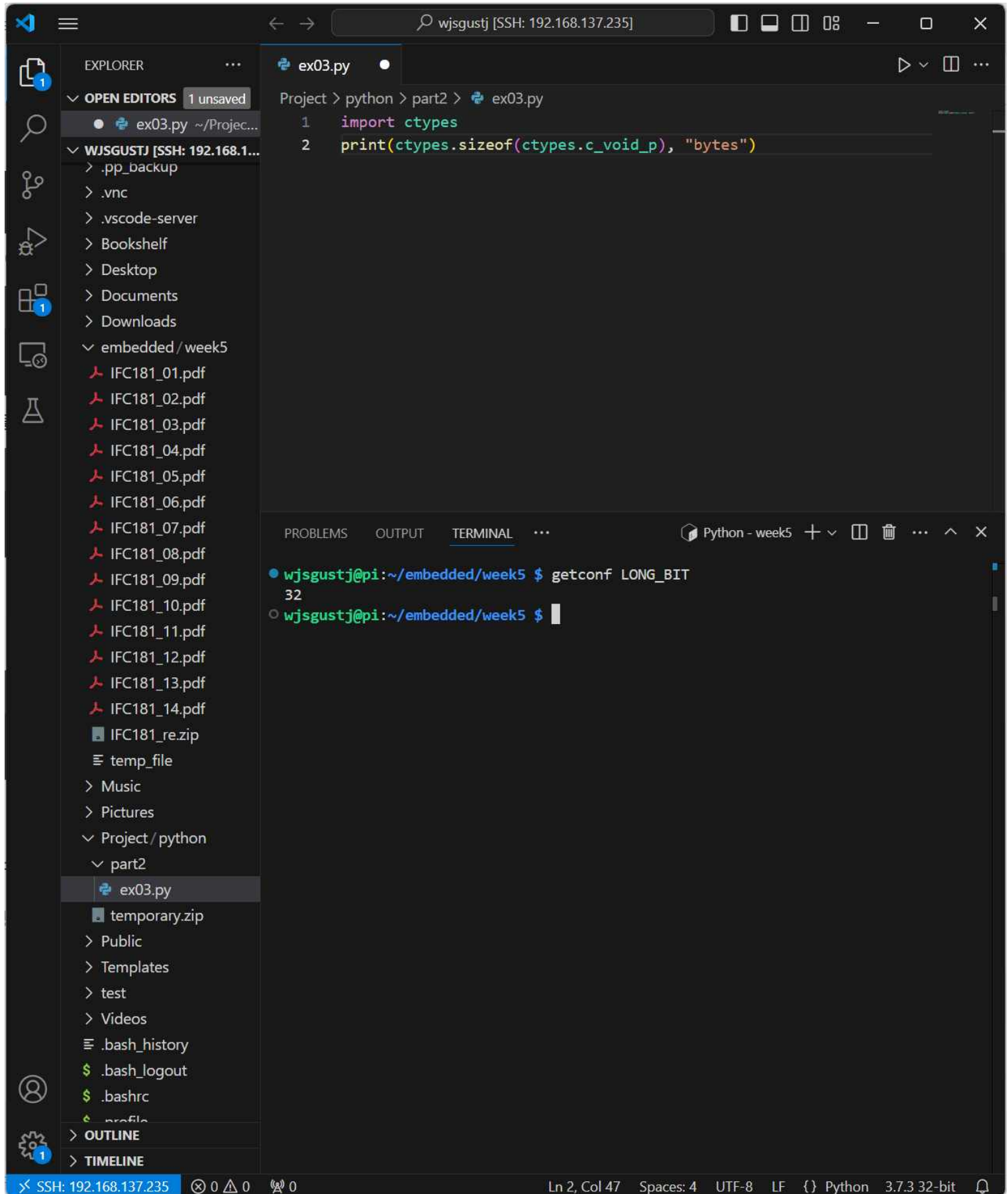
wjsgustj@pi:~/embedded/week5 $ df .
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        14986204 4111700  10212996  29% /

wjsgustj@pi:~/embedded/week5 $
```

vim 명령어를 통해 생성한 temp_file은 2 byte 크기의 파일이나, Linux는 파일을 저장하기 위해 파티션을 block으로 나눔. 최소 block의 크기가 4K이므로, temp_file은 4K byte 크기를 차지함. 따라서 사용 가능한 용량은 4K byte가 감소했으나, 파일 크기는 2byte임.

<cpu architecture>

13. 현재 사용하는 라즈비언 OS가 몇비트 시스템인지 확인하고 결과를 첨부하시오.



The screenshot shows a Visual Studio Code editor window connected to a Raspberry Pi via SSH. The Explorer sidebar on the left shows the file structure, including a directory named 'part2' containing 'ex03.py'. The main editor area displays the contents of 'ex03.py':

```
Project > python > part2 > ex03.py
1 import ctypes
2 print(ctypes.sizeof(ctypes.c_void_p), "bytes")
```

Below the editor, the TERMINAL panel shows the command prompt on the Raspberry Pi. The user has run the command 'getconf LONG_BIT', which returned the value '32', indicating a 32-bit system.

```
wjsgustj@pi:~/embedded/week5 $ getconf LONG_BIT
32
wjsgustj@pi:~/embedded/week5 $
```

The status bar at the bottom of the window indicates the current file is 'ex03.py' at line 2, column 47, using Python 3.7.3 32-bit.