임베디드응용및실습
5주차 과제

과목명	임베디드응용및실습
이름	김민주
학번	2022161023

<파일 압축 해제>

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

```
kmj@kmj:~ $ mv ~/Downloads/IFC181.tar ~/embedded/week5/
kmj@kmj:~ $ []
```

2. tar 명령으로 압축을 해제한다.

```
kmj@kmj:~ $ cd embedded/
kmj@kmj:~/embedded $ cd week5/
kmj@kmj:~/embedded/week5 $ tar -xvf IFC181.tar
IFC181_14.pdf
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
kmj@kmj:~/embedded/week5 $ []
```

3. 2에서 압축해제된 파일들을 IFC181_re.tar로 압축한다.

```
kmj@kmj:~/embedded/week5 $ tar -cvf IFC181_re.tar IFC181_*.pdf
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
kmj@kmj:~/embedded/week5 $ []
```

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

- 이때 zip 명령을 수행할 수 있도록 zip을 설치한다.

: sudo apt-get install zip

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

```
kmj@kmj:~ $ find . -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week6/IFC181_05.pdf
./embedded/week6/IFC181_04.pdf
./embedded/week6/IFC181_08.pdf
./embedded/week6/IFC181_12.pdf
./embedded/week6/IFC181_09.pdf
./embedded/week6/IFC181_07.pdf
./embedded/week6/IFC181_13.pdf
./embedded/week6/IFC181_02.pdf
./embedded/week6/IFC181_01.pdf
./embedded/week6/IFC181_14.pdf
./embedded/week6/IFC181_11.pdf
./embedded/week6/IFC181_10.pdf
./embedded/week6/IFC181_03.pdf
./embedded/week6/IFC181_06.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
kmj@kmj:~ $
```

이 명령어는 현재 디렉토리(.) 내에서 이름이 *.pdf인 파일을 찾으려 합니다. 그러나 *.pdf 는 쉘에 의해 확장되어 현재 디렉토리에 있는 PDF 파일 목록으로 변환됩니다. 따라서, 현재 디렉토리에 PDF 파일이 없다면 아무 것도 찾지 못할 수 있습니다.

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

```
kmj@kmj:~ $ find ./ -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week6/IFC181_05.pdf
./embedded/week6/IFC181_04.pdf
./embedded/week6/IFC181_08.pdf
./embedded/week6/IFC181_12.pdf
./embedded/week6/IFC181_09.pdf
./embedded/week6/IFC181_07.pdf
./embedded/week6/IFC181_13.pdf
./embedded/week6/IFC181_02.pdf
./embedded/week6/IFC181_01.pdf
./embedded/week6/IFC181_14.pdf
./embedded/week6/IFC181_11.pdf
./embedded/week6/IFC181_10.pdf
./embedded/week6/IFC181_03.pdf
./embedded/week6/IFC181_06.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
kmj@kmj:~ $
```

이 명령어도 현재 디렉토리(./) 내에서 이름이 *.pdf인 파일을 찾으려 합니다. ./는 현재 디렉토리와 동일하므로 결과는 첫 번째 명령어와 같습니다.

innosm@innosm:~ $find -name *.pdf

```
kmj@kmj:~ $ find -name *.pdf
./Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
./embedded/week6/IFC181_05.pdf
./embedded/week6/IFC181_04.pdf
./embedded/week6/IFC181_08.pdf
./embedded/week6/IFC181_12.pdf
./embedded/week6/IFC181_09.pdf
./embedded/week6/IFC181_07.pdf
./embedded/week6/IFC181_13.pdf
./embedded/week6/IFC181_02.pdf
./embedded/week6/IFC181_01.pdf
./embedded/week6/IFC181_14.pdf
./embedded/week6/IFC181_11.pdf
./embedded/week6/IFC181_10.pdf
./embedded/week6/IFC181_03.pdf
./embedded/week6/IFC181_06.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
kmj@kmj:~ $
```

이 명령어는 인자가 없을 경우 현재 디렉토리에서 *.pdf로 찾으려 합니다. 마찬가지로 쉘에 의해 *.pdf가 확장되어 현재 디렉토리의 PDF 파일을 찾게 됩니다. 결과는 첫 번째와 동일합니다.

innosm@innosm:~ $find ~ -name *.pdf

```
kmj@kmj:~ $ find ~ -name *.pdf
/home/kmj/Bookshelf/BeginnersGuide-4thEd-Eng_v2.pdf
/home/kmj/embedded/week6/IFC181_05.pdf
/home/kmj/embedded/week6/IFC181_04.pdf
/home/kmj/embedded/week6/IFC181_08.pdf
/home/kmj/embedded/week6/IFC181_12.pdf
/home/kmj/embedded/week6/IFC181_09.pdf
/home/kmj/embedded/week6/IFC181_07.pdf
/home/kmj/embedded/week6/IFC181_13.pdf
/home/kmj/embedded/week6/IFC181_02.pdf
/home/kmj/embedded/week6/IFC181_01.pdf
/home/kmj/embedded/week6/IFC181_14.pdf
/home/kmj/embedded/week6/IFC181_11.pdf
/home/kmj/embedded/week6/IFC181_10.pdf
/home/kmj/embedded/week6/IFC181_03.pdf
/home/kmj/embedded/week6/IFC181_06.pdf
/home/kmj/embedded/week5/IFC181_05.pdf
/home/kmj/embedded/week5/IFC181_04.pdf
/home/kmj/embedded/week5/IFC181_08.pdf
/home/kmj/embedded/week5/IFC181_12.pdf
/home/kmj/embedded/week5/IFC181_09.pdf
/home/kmj/embedded/week5/IFC181_07.pdf
/home/kmj/embedded/week5/IFC181_13.pdf
/home/kmj/embedded/week5/IFC181_02.pdf
/home/kmj/embedded/week5/IFC181_01.pdf
/home/kmj/embedded/week5/IFC181_14.pdf
/home/kmj/embedded/week5/IFC181_11.pdf
/home/kmj/embedded/week5/IFC181_10.pdf
/home/kmj/embedded/week5/IFC181_03.pdf
/home/kmj/embedded/week5/IFC181_06.pdf
kmj@kmj:~ $
```

이 명령어는 홈 디렉토리(~) 내에서 이름이 *.pdf인 파일을 찾으려 합니다. ~도 현재 사용자의 홈 디렉토리를 참조하므로 결과는 홈 디렉토리에서 PDF 파일을 찾는 것이 됩니다. 이 경우, 쉘에서 *.pdf가 확장되므로 홈 디렉토리 내의 PDF 파일을 찾습니다.

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

```
90e3/server/node_modules/proxy-from-env/test.js
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/proxy-from-env/index.js
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/proxy-from-env/package.json
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/tas-client-umd
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/tas-client-umd/LICENSE
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/tas-client-umd/lib
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/tas-client-umd/lib/tas-client-umd.js
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/server/node_modules/tas-client-umd/package.json
/home/kmj/.vscode-server/cli/servers/Stable-4849ca9bdf9666755eb463db297b69e53850
90e3/pid.txt
/home/kmj/.cups
/home/kmj/.cups/lpoptions
/home/kmj/Templates
/home/kmj/.xsession-errors
/home/kmj/.cache
/home/kmj/.cache/lxsession
/home/kmj/.cache/lxsession/LXDE-pi
/home/kmj/.cache/lxsession/LXDE-pi/run.log
/home/kmj/.cache/Microsoft
/home/kmj/.cache/Microsoft/DeveloperTools
/home/kmj/.cache/Microsoft/DeveloperTools/deviceid
/home/kmj/.cache/menus
/home/kmj/.cache/menus/768573c7656bbd74b73d0c82aa5c4e17
/home/kmj/.cache/openbox
/home/kmj/.cache/openbox/openbox.log
/home/kmj/.cache/openbox/sessions
/home/kmj/.Xauthority
/home/kmj/.vnc
/home/kmj/.vnc/config.d
find: '-name': No such file or directory
find: '*.pdf': No such file or directory
kmj@kmj:~ $
```

이 명령어는 /home/kmj 디렉토리 내에서 이름이 *.pdf인 파일을 찾으려 합니다. 여기서 도 마찬가지로 *.pdf가 쉘에 의해 확장되므로 해당 경로 내의 PDF 파일을 찾습니다

- 명령 결과 -
첫 세 명령어(find . -name *.pdf, find ./ -name *.pdf, find -name *.pdf)는 현재 디 렉토리에서 *.pdf로 파일을 찾으려 하며, 사용자의 의도와는 다르게 동작할 수 있습니다. 반면, 마지막 두 명령어(find ~ -name *.pdf, find /home/innosm -name *.pdf)는 각 각 홈 디렉토리와 /home/innosm 경로에서 실제로 .pdf 파일을 찾기 때문에 동작 결과 가 달라질 수 있습니다.

8. ~ (home) 디렉토리로 이동하고, week5 폴더가 있는지 검색하려고 한다.
적절한 명령을 수행하여 week5 폴더를 검색하고 결과를 보이시오.

```
kmj@kmj:~ $ find ./ -name week5 -type d
  ./embedded/week5
kmj@kmj:~ $
```

<파일 용량 확인>
9. 아래 명령을 수행한 결과를 보이시오.
df -h

```
kmj@kmj:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G  3.1G   11G  23% /
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  4.0K  384M   1% /run/user/1000
kmj@kmj:~ $
```

10. 아래 명령을 수행한 결과를 보이시오.
cd ~
cd embedded
du -h

```
kmj@kmj:~/embedded $ du -h
4.0M    ./week6
16M     ./week5
20M     .
kmj@kmj:~/embedded $
```

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

```
kmj@kmj:~ $ du -sh ~/embedded
20M     /home/kmj/embedded
kmj@kmj:~ $
```

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

```
kmj@kmj:~/embedded $ cd
kmj@kmj:~ $ cd embedded/week5
kmj@kmj:~/embedded/week5 $ df .
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/root       14986204  3213940  11110756  23% /
kmj@kmj:~/embedded/week5 $
```

2) 이 폴더에 temp_file 파일 생성하고 파일에 1을 기록

```
kmj@kmj:~/embedded/week5 $ touch temp_file
kmj@kmj:~/embedded/week5 $ echo 1 > temp_file
kmj@kmj:~/embedded/week5 $ cat temp_file
1
kmj@kmj:~/embedded/week5 $
```

3) stat temp_file 명령을 통해 파일의 크기를 확인

```
kmj@kmj:~/embedded/week5 $ stat temp_file
  File: temp_file
  Size: 2           Blocks: 8          IO Block: 4096   regular file
Device: b302h/45826d    Inode: 263684      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/    kmj)  Gid: ( 1000/    kmj)
Access: 2024-10-06 12:12:26.360753534 +0100
Modify: 2024-10-06 12:13:01.180571484 +0100
Change: 2024-10-06 12:13:01.180571484 +0100
 Birth: -
kmj@kmj:~/embedded/week5 $
```

4) df .

```
kmj@kmj:~/embedded/week5 $ df .
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/root       14986204  3213948  11110748  23% /
kmj@kmj:~/embedded/week5 $
```

1)과 4)에서 줄어든 용량과 3)에서 확인한 용량이 다르다면 그 이유는?

df .에서 줄어든 용량은 파일 시스템이 관리하는 블록 단위로 계산됩니다. 대부분의 파일 시스템은 데이터를 블록 단위로 저장하는데, 블록의 크기는 일반적으로 4KB입니다. 그래서 파일이 1바이트인 경우에도 최소 1블록(예: 4KB)이 할당되기 때문에, df에서 용량의 줄어드는 모습을 볼 수 있습니다.

<cpu architecture>

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

```
kmj@kmj:~ $ lscpu
Architecture:        armv7l
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  1
Core(s) per socket:  4
Socket(s):           1
Vendor ID:           ARM
Model:               3
Model name:          Cortex-A72
Stepping:            r0p3
CPU max MHz:         1500.0000
CPU min MHz:         600.0000
BogoMIPS:            108.00
Flags:               half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idi
vt vfpd32 lpae evtstrm crc32
kmj@kmj:~ $ uname -m
armv7l
kmj@kmj:~ $
```

armv7l는 32비트 시스템입니다.