

1. What is the path for your user's home directory? How do you change your current working directory to your user's home directory?

```
user@AC4892-Ubuntu:~$ cd /
```

```
user@AC4892-Ubuntu:/$
```

2. How can you find out your current working directory?

```
user@AC4892-Ubuntu:/$ pwd
```

```
user@AC4892-Ubuntu:/$ pwd
```

3. Explain briefly what option -p does when used with mkdir command. In addition, find out what version number of mkdir command you have.

With 'mkdir -p' we can create nested directories using one command

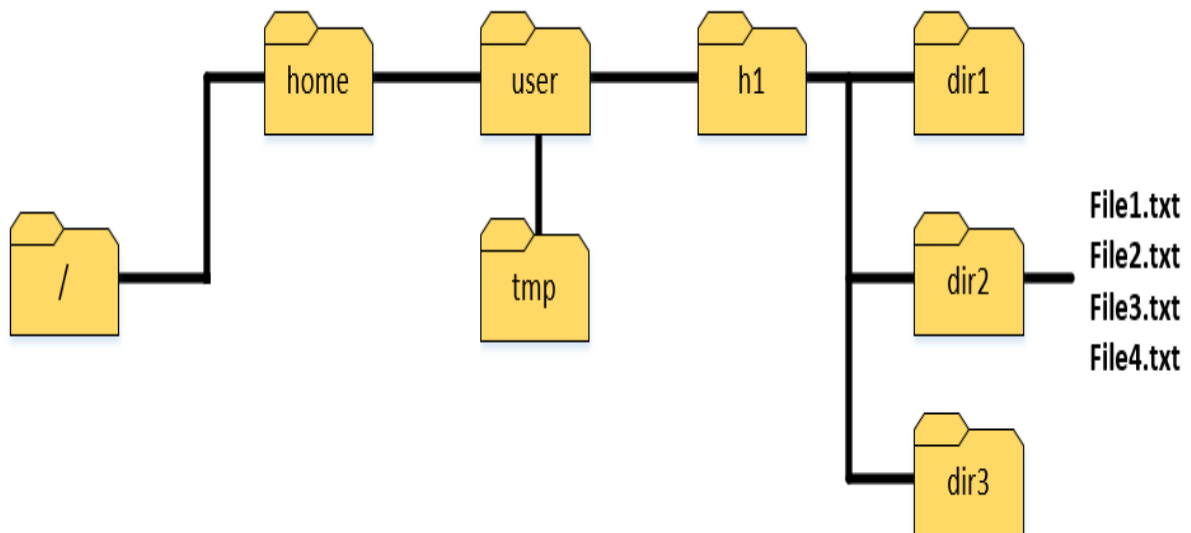
Ex: mkdir -p ex1/ex2/ex3

```
user@AC4892-Ubuntu:/$ mkdir -Z --version
```

```
mkdir (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
```

4. Create the following directory structure including the files to your home directory using your Linux shell. Files can be empty or containing text.



```
user@AC4892-Ubuntu:~$ cd /
user@AC4892-Ubuntu:/$ cd home/user
user@AC4892-Ubuntu:~$ mkdir -p h1/dir1 h1/dir2 h1/dir3 tmp
```

```
user@AC4892-Ubuntu:~$ ls
h1  sand  tmp
user@AC4892-Ubuntu:~$ ls h1
dir1  dir2  dir3
user@AC4892-Ubuntu:~$
```

```
user@AC4892-Ubuntu:~$ cd h1/dir2
user@AC4892-Ubuntu:~/h1/dir2$ touch File1.txt File2.txt
user@AC4892-Ubuntu:~/h1/dir2$ ls
File1.txt  File2.txt
user@AC4892-Ubuntu:~/h1/dir2$ touch File3.txt File4.txt
user@AC4892-Ubuntu:~/h1/dir2$ ls
File1.txt  File2.txt  File3.txt  File4.txt
user@AC4892-Ubuntu:~/h1/dir2$
```

5. Copy all files with .txt extension from dir2 to dir1 using relative path for directories using only one command.

```
user@AC4892-Ubuntu:~/h1$ cp dir2/File1.txt dir2/File2.txt dir2/File3.txt dir2/File4.txt dir1
user@AC4892-Ubuntu:~/h1$ ls dir1
File1.txt File2.txt File3.txt File4.txt
```

6. Move all files starting with string File1 from dir2 to dir3 using absolute path for directories using only one command.

```
user@AC4892-Ubuntu:~/h1$ mv /home/user/h1/dir2/File1.txt /home/user/h1/dir3/
user@AC4892-Ubuntu:~/h1/dir3$ ls
File1.txt
```

7. Create tmp directory inside your user's home directory. Copy directory structure created in fourth part starting from directory h1 into the tmp directory in one command. The final directory tree should then look like this:
/home/your_user/tmp/h1...

```
user@AC4892-Ubuntu:~$ cp -r h1 tmp/
user@AC4892-Ubuntu:~$ ls tmp
h1
user@AC4892-Ubuntu:~$ ls tmp/h1
dir1 dir2 dir3
```

8. Remove tmp/h1 directory structure using only one command.

```
user@AC4892-Ubuntu:~$ rm -r tmp/h1
user@AC4892-Ubuntu:~$ cd tmp
user@AC4892-Ubuntu:~/tmp$ ls
user@AC4892-Ubuntu:~/tmp$
```

9. Rename File1.txt file under dir1 in a way it begins with string NewFileX1
(File1.txt → NewFileX1.txt)

```
user@AC4892-Ubuntu:~/h1/dir1$ mv File1.txt NewFile1.txt
user@AC4892-Ubuntu:~/h1/dir1$ ls
File2.txt  File3.txt  File4.txt  NewFile1.txt
user@AC4892-Ubuntu:~/h1/dir1$
```

10. How do you distinguish the following two paths: relative and absolute? What do these terms mean? Give examples from both paths.

Relative path : in here we determine the path from the current path we are at

Absolute path : when we define the whole path from home directory to the relevant directory it's called the absolute path