

Step1

```
mjswindells@ubuntu:~$ mkdir 2018603020_KHC
mjswindells@ubuntu:~$ cd 2018603020_KHC/
mjswindells@ubuntu:~/2018603020_KHC$ cat > Seoul.txt
mjswindells@ubuntu:~/2018603020_KHC$ cat > cs.kw
mjswindells@ubuntu:~/2018603020_KHC$ cat > busan.txt
mjswindells@ubuntu:~/2018603020_KHC$ cat > wmail.kw
mjswindells@ubuntu:~/2018603020_KHC$ cat > NY.txt
mjswindells@ubuntu:~/2018603020_KHC$ cat > kupis.kw
mjswindells@ubuntu:~/2018603020_KHC$ cat > lect1.doc
mjswindells@ubuntu:~/2018603020_KHC$ cat > lect2.txt
mjswindells@ubuntu:~/2018603020_KHC$ cat > lect3.doc
mjswindells@ubuntu:~/2018603020_KHC$ cat > lect4.doc
mjswindells@ubuntu:~/2018603020_KHC$ cat > middle.test
mjswindells@ubuntu:~/2018603020_KHC$ cat > final.test
mjswindells@ubuntu:~/2018603020_KHC$ ls
busan.txt  final.test  lect1.doc  lect3.doc  middle.test  Seoul.txt
cs.kw      kupis.kw   lect2.txt  lect4.doc  NY.txt       wmail.kw
mjswindells@ubuntu:~/2018603020_KHC$
```

[과제1]의 복습

Step2-1

```
mjswindells@ubuntu:~/2018603020_KHC$ find . -name "*.txt"
./Seoul.txt
./lect2.txt
./NY.txt
./busan.txt
mjswindells@ubuntu:~/2018603020_KHC$
```

find [경로] [-옵션] [파일이름]

- 1) 위 예제에서의 경로는 . 으로 현재 디렉토리 및 하위 디렉토리에서 파일을 검색한다
- 2) 옵션 : -name : 찾고자 하는 파일의 이름 지정
-perm : 권한(permission)이 일치되는 파일 찾기 ex) find . -perm -777
- 3) 파일이름에서 *의 의미는 와일드카드로서 길이가 0이상인 임의의 문자열에 대응한다
즉, .txt로 끝나는 파일들을 찾을 수 있다.

Step2-2

```
mjswindells@ubuntu:~/2018603020_KHC$ find lect*
lect1.doc
lect2.txt
lect3.doc
lect4.doc
mjswindells@ubuntu:~/2018603020_KHC$
```

Step2-1에서 했던 방식과 동일하게 [find . -name 'lect*'] 로 찾을 수도 있지만

Find [파일이름]으로 단순히 대상 디렉토리에 있는 파일 및 디렉토리 리스트를 표시할 수도 있다.

```
mjswindells@ubuntu:~/2018603020_KHC$ find lect*
lect1.doc
lect2.txt
lect3.doc
lect4.doc
lectExample
lectExample/bbbb
lectExample/aaaa
mjswindells@ubuntu:~/2018603020_KHC$
```

경로를 지정하지 않을 경우 lectExample 디렉토리의 하위 폴더는 전부 다 출력을 하게된다

Find . -name 'lect*' 로 입력을 할 경우 디렉토리까지만 출력이되고 내부 파일에는 lect로 시작하는 파일이 존재하지 않으므로 출력되지 않는다

```
mjswindells@ubuntu:~/2018603020_KHC$ find . -name 'lect*'
./lectExample
./lect1.doc
./lect2.txt
./lect3.doc
./lect4.doc
mjswindells@ubuntu:~/2018603020_KHC$
```

옵션 :

- 1) -user [name] : 파일 사용자의 name에서 검색

Ex) 루트계정 setuid 찾기 : find . -user root -perm -4000

- 2) -group [name] : 그룹소유주가 name인 파일 검색

- 3) -print : 찾는 파일을 화면에 출력

Step 2-3

```
mjswindells@ubuntu:~/2018603020_KHC$ find . -name 'lect*' | grep .doc
./lect1.doc
./lect3.doc
./lect4.doc
mjswindells@ubuntu:~/2018603020_KHC$
```

grep [pattern] [파일이름] : 조건에 맞는 문자열을 찾아서 출력

find나 ls를 사용할 때 파이프라인(|)을 이용하여 검색범위를 줄일 수 있다. (And연산자라 생각)

```
mjswindells@ubuntu:~/2018603020_KHC$ find . -name "Seoul.txt" | xargs grep -n "s"
1:This is Seoul.txt
2:seoul
3:busan
4:abcdefgssss
mjswindells@ubuntu:~/2018603020_KHC$
```

Grep과 find로파일 내부에 특정 글자가 포함된 라인도 찾을 수 있다

xargs의 의미 : 파이프를 통해 넘어온 결과물을 매개변수로 넘겨주는 역할

Seoul.txt 파일 중에서 s가 들어가는 라인을 출력해준다

Step 3

```
mjswindells@ubuntu:~/2018603020_KHC$ ls -l Seoul.txt
-rw-rw-r-- 1 mjswindells mjswindells 0 Mar 17 23:17 Seoul.txt
mjswindells@ubuntu:~/2018603020_KHC$ chmod g-r Seoul.txt
mjswindells@ubuntu:~/2018603020_KHC$ ls -l Seoul.txt
-rw--w-r-- 1 mjswindells mjswindells 0 Mar 17 23:17 Seoul.txt
mjswindells@ubuntu:~/2018603020_KHC$
```

chmod [ugoa][+ -=][rwx] [파일또는 디렉토리] : change mode

- 1) ugoa : 유저 그룹 기타 all 4개중 하나를 선택해 바꾼다
- 2) + -= : 첨가 삭제 대입
- 3) rwx : 읽기 쓰기 실행허가 rwx 3개를 다 입력할 수 있다

chmod g-r Seoul.txt

chmod 700 Seoul.txt 둘 다 그룹의 읽기권한을 없앤다 700은 유저를 제외한 사용자의 권한을 없앤다

느낀점 : 각 명령어에 다양한 옵션이 있는데 현재 자주 사용할법한 옵션 위주로만 계속 연습을 하고 있다. '그 외의 옵션들 까지 연습을 자주해야할까?'란 의문이 든다.