

2024.09.26	실습 과제	학번	22312072
과제 4	리눅스 사용 기본 명령어 3	이름	차민경
<p>• 참고사항</p> <p>모든 실습 과제는 각 문항에서 요구하는 문제의 명령어와 그 출력 결과를 동시에 기재하여야 합니다.</p> <p>예: 오늘 날짜를 출력하는 명령어를 쓰시오.</p> <p>답:</p> <p>date</p> <p>2024. 09. 26. (목) 15:00:00 KST</p>			

1. `sleep` 명령어를 통해 3초 동안 대기하는 명령어를 후면 처리로 실행하시오. 3초 후, 출력 결과를 보이시오.

답:

1:

(sleep 3)&

[1] 620491

(3초후 엔터치면)

[1]+ Done (sleep 3)

2:

(sleep 3; echo done)&

[1] 620607

done

2. `sleep` 명령어를 통해 100초 동안 대기하는 명령어를 후면 처리로 실행하시오. 100초가 지나기 전, 해당 작업을 전면으로 전환한 뒤 멈추시오.

답:

(sleep 100; echo done) &

[1] 620675

jobs

[1]+ Running (sleep 100; echo done) &

fg %1

(sleep 100; echo done)

done

3. 현재 시스템에서 실행 중인 모든 프로세스를 출력하는 명령어를 작성하고, 파이프를 통해 해당 명령어 출력 결과의 줄 수를 출력하는 명령어를 작성하시오.

답:

ps -aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

```
root          1  0.0  0.0 169164 13644 ?        Rs   9월12   2:28 /sbin/init
root          2  0.0  0.0    0    0 ?        S    9월12   0:00 [kthreadd]
```

... 생략

ps -aux | wc //표준 출력을 표준 입력으로 넣어줌

```
427    4925    40997
```

4. PID가 `9999`인 프로세스를 강제 종료하기 위한 명령어를 작성하시오.

답:

kill -9 9999

-bash: kill: (9999) - No such process

5. 아래의 내용을 `novel.txt` 파일에 저장하시오. (PuTTY에서 오른쪽 마우스 클릭으로 붙여넣기 가능)

답:

```
cat > novel.txt
If thou couldst answer.
This fair child of mine Shall sum my count
and make my old excuse.
Proving his beauty by succession thine.
This were to be new made when thou art old.
And see thy blood warm when thou feelst it cold.
```

^D

```
If thou couldst answer.
This fair child of mine Shall sum my count
and make my old excuse.
Proving his beauty by succession thine.
This were to be new made when thou art old.
And see thy blood warm when thou feelst it cold.
```

6.`novel.txt`에 대해 다음 과정을 수행하시오.

(1) `thou`가 포함된 문장을 `thou.txt`에 저장하시오.

답:

```
grep thou novel.txt > thou.txt
```

```
cat thou.txt
```

```
If thou couldst answer.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

(2) `thou.txt`에서 `old`가 포함된 문장을 `old.txt`에 저장하시오.

답:

```
grep old thou.txt > old.txt
```

```
cat old.txt
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

(3) `thou.txt`와 `old.txt`를 합친 `thou_old.txt`를 만드시오.

답:

```
cat thou.txt old.txt > thou_old.txt
```

```
cat thou_old.txt
```

```
If thou couldst answer.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

(4) `thou_old.txt`의 내용을 `novel.txt` 내용 끝에 추가하시오.

답:

```
cat thou_old.txt >> novel.txt
```

```
cat novel.txt
```

```
If thou couldst answer.
```

```
This fair child of mine Shall sum my count
```

```
and make my old excuse.
```

```
Proving his beauty by succession thine.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

```
If thou couldst answer.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

(5) `novel.txt`의 문자 개수를 세는 명령어를 실행하시오.

답:

```
wc -c novel.txt //줄수 단어 문자 개수
```

```
434 novel.txt
```

7. `ls` 명령어와 `grep` 명령어를 파이프로 결합하여, 현재 디렉터리에서 확장자가 `txt`인 파일을 모두 출력하시오.

답:

```
ls
```

```
ex3 names.txt novel.txt old.txt snap thou_old.txt thou.txt
```

```
ls | grep '.txt'
```

```
names.txt
```

```
novel.txt
```

```
old.txt
```

```
thou_old.txt
```

```
thou.txt
```

8. `thou.txt`와 `old.txt` 파일의 차이를 `diff`와 `cmp` 명령어를 사용하여 비교하시오.

명령어 결과, 추가적인 해석

```
cat thou.txt
```

```
If thou couldst answer.
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

```
cat old.txt
```

```
This were to be new made when thou art old.
```

```
And see thy blood warm when thou feelst it cold.
```

```
diff thou.txt old.txt
```

```
1d0
```

```
< If thou couldst answer.
```

thou.txt 파일 1번째 줄을 삭제하면 old.txt의 0번째부터(첫 번째 줄 바로 앞=파일 시작점) 두 파일

이 같아진다

```
cmp thou.txt old.txt
```

```
thou.txt old.txt differ: byte 1, line 1
```

1번째 행 1번째 부터 문자가 다르다

두 파일이 서로 다르면 달라지는 위치를 출력해준다

9. `df` 명령어의 결과를 `disk_usage.txt` 파일로 저장한 후, `Available` 필드 기준으로 내림차순 정렬하여 출력하시오.

```
df > disk_usage.txt
```

sort -nrk 4 disk_usage.txt //숫자로 정렬함

```
/dev/nvme0n1p2 490150872 70748788 394430360 16% /
tmpfs          8148128      20  8148108    1% /dev/shm
tmpfs          1629624      64  1629560    1% /run/user/1052
tmpfs          1629624      68  1629556    1% /run/user/1136
tmpfs          1629624      68  1629556    1% /run/user/1098
(생략...)
```

/*sort -rk 4 disk_usage.txt 문자열로 정렬함

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/nvme0n1p1	997456	6228	991228	1%	/boot/efi
tmpfs	8148128	20	8148108	1%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock*

10. `novel.txt` 파일의 하드 링크와 심볼릭 링크를 생성하고, 원본 파일 `novel.txt`을 삭제한 후 두 링크 파일의 내용을 확인하시오.

```
ln novel.txt novel_hard.txt
```

```
ln -s novel.txt novel_sym.txt
```

```
ls -l
```

```
-rw-rw-r-- 2 yu22312072 yu22312072 0 10월 2 18:08 novel_hard.txt
lrwxrwxrwx 1 yu22312072 yu22312072 9 10월 2 18:09 novel_sym.txt-> novel.txt
-rw-rw-r-- 2 yu22312072 yu22312072 0 10월 2 18:08 novel.txt
rm novel.txt
-rw-rw-r-- 1 yu22312072 yu22312072 33 10월 2 18:11 novel_hard.txt
lrwxrwxrwx 1 yu22312072 yu22312072 9 10월 2 18:09 novel_sym.txt-> novel.txt
/*ls -alh | grep novel_hard.txt
-rw-rw-r-- 1 yu22312072 yu22312072 33 10월 2 18:11 novel_hard.txt
ls -alh | grep novel_sym.txt
lrwxrwxrwx 1 yu22312072 yu22312072 9 10월 2 18:09 novel_sym.txt-> novel.txt*/
```