

리눅스시스템 Lab09

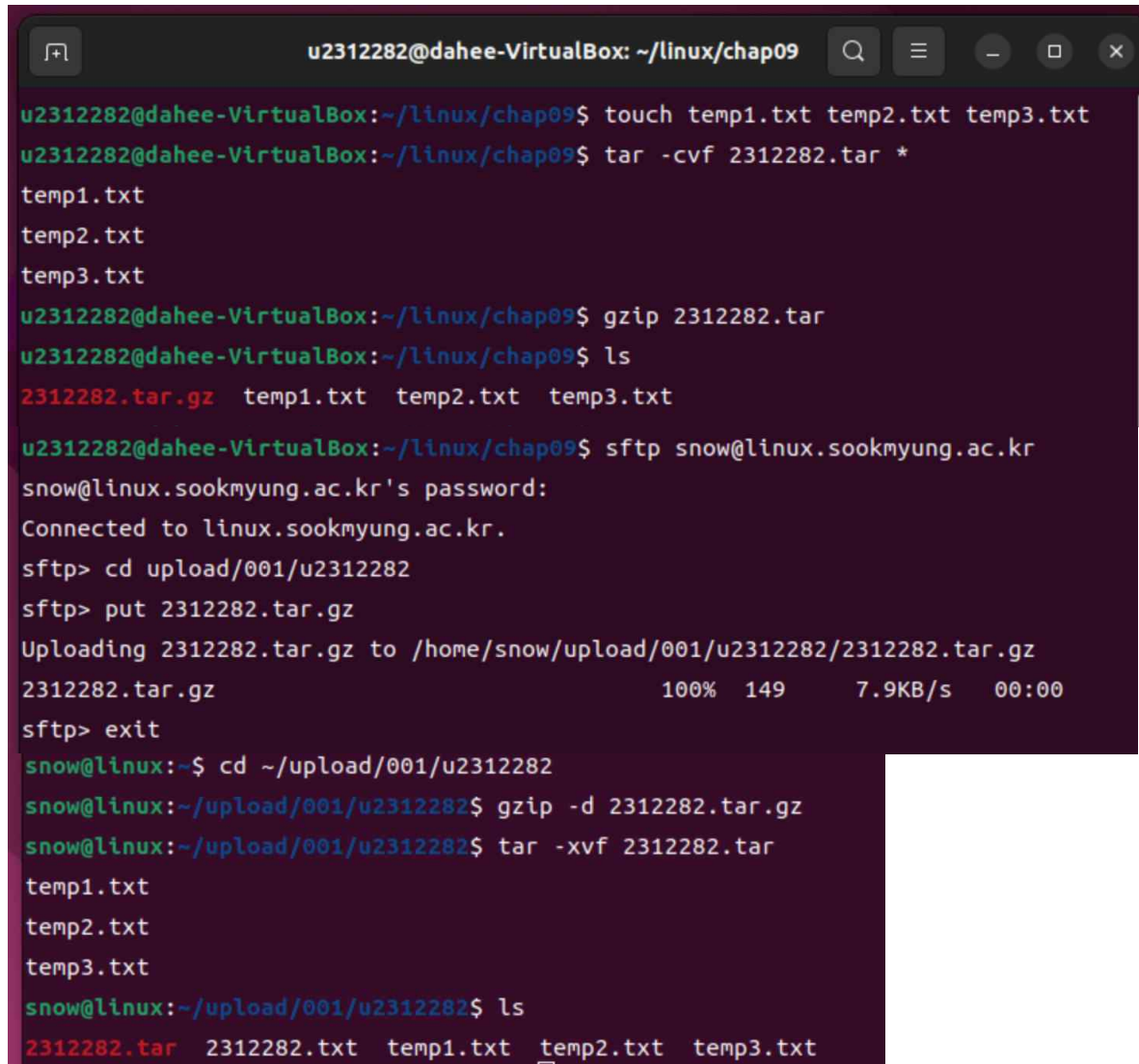
분반:001 학과:컴퓨터과학전공

학번:2312282

이름:임다희

1. 파일 압축 실습

1) (1)~(5) 과정을 모두 수행한 터미널 창을 캡처한다.



```
u2312282@dahee-VirtualBox: ~/linux/chap09
u2312282@dahee-VirtualBox:~/linux/chap09$ touch temp1.txt temp2.txt temp3.txt
u2312282@dahee-VirtualBox:~/linux/chap09$ tar -cvf 2312282.tar *
temp1.txt
temp2.txt
temp3.txt
u2312282@dahee-VirtualBox:~/linux/chap09$ gzip 2312282.tar
u2312282@dahee-VirtualBox:~/linux/chap09$ ls
2312282.tar.gz temp1.txt temp2.txt temp3.txt
u2312282@dahee-VirtualBox:~/linux/chap09$ sftp snow@linux.sookmyung.ac.kr
snow@linux.sookmyung.ac.kr's password:
Connected to linux.sookmyung.ac.kr.
sftp> cd upload/001/u2312282
sftp> put 2312282.tar.gz
Uploading 2312282.tar.gz to /home/snow/upload/001/u2312282/2312282.tar.gz
2312282.tar.gz                               100% 149    7.9KB/s   00:00
sftp> exit
snow@linux:~$ cd ~/upload/001/u2312282
snow@linux:~/upload/001/u2312282$ gzip -d 2312282.tar.gz
snow@linux:~/upload/001/u2312282$ tar -xvf 2312282.tar
temp1.txt
temp2.txt
temp3.txt
snow@linux:~/upload/001/u2312282$ ls
2312282.tar 2312282.txt temp1.txt temp2.txt temp3.txt
```

(2312282.txt는 이전 실습에서 업로드한 파일)

2) tar 파일로 먼저 묶고 압축하여 다시 원상복귀하는 방법에 대해 어떤 명령어 및 옵션을 써야 하는지 명령 순서대로 설명한다. ((2),(3),(5),(6) 과정 참고)

tar 명령어의 -cvf 옵션을 사용해 여러 개의 파일들을 하나의 타르파일로 묶는다. \$ tar -cvf 2312282.tar * 와 같이 사용하면 현재 작업 디렉터리 내에 위치한 모든 파일을 2312282.tar라는 이름의 새로운 타르파일로 묶을 수 있다.

gzip 명령어를 사용해 파일(들)을 압축해 .gz 파일을 만들 수 있다. gzip 2312282.tar 와 같이 사용하면 파일 2312282.tar를 압축해 .gz 파일을 만든다.

gzip 명령어의 -d 옵션을 사용하면 gzip으로 압축되어 있는 파일들을 복원할 수 있다. \$ gzip -d 2312282.tar.gz와 같이 사용하면 .gz 파일로 압축된 파일이 원상복귀되어 2312282.tar의 형태로 돌아온다.


tar 명령어의 -xvf 옵션을 사용해 하나의 타르파일을 풀고 원래 파일들을 복원할 수 있다. \$ tar -xvf 2312282.tar 와 같이 사용하면 2312282.tar 타르파일 내부에 있는 3개의 파일 temp1.txt,temp2.txt,temp3.txt가 다시 개별의 파일로 존재하도록 한다.

2. 파일 정렬 실습

1) members.txt 파일을 이름 순으로 정렬하여 members_name 파일에 저장하기 위한 명령을 쓰고, 실행 결과를 캡처한다. (명령 실행 후, gedit으로 members_name 파일을 연 뒤 캡처)

```
$ sort -k 2 -o members_name members.txt
```

```
u2312282@dahee-VirtualBox:~/linux/chap09$ sort -k 2 -o members_name members.txt
u2312282@dahee-VirtualBox:~/linux/chap09$ gedit members_name
```




| | 이름 | 직업 | 부서 | 급여 | 이메일 | |
|----|-----|--------|-----------|------------|------|--------------------|
| 1 | 007 | Aiden | Developer | Technology | 6000 | aiden@gmail.com |
| 2 | 009 | Emma | Manager | Technology | 9000 | emma@gmail.com |
| 3 | 002 | Jason | Developer | Technology | 5500 | jason@yahoo.com |
| 4 | 006 | John | Staff | Sales | 4000 | john@macys.com |
| 5 | 004 | Nisha | Manager | Marketing | 9500 | nisha@gmail.com |
| 6 | 010 | Olivia | DBA | Technology | 6500 | olivia@gmail.com |
| 7 | 005 | Randy | DBA | Technology | 6000 | randy@oracle.com |
| 8 | 003 | Sanjay | Sysadmin | Technology | 7000 | sanj@sys.admin.com |
| 9 | 001 | Thomas | Manager | Sales | 5000 | thomas@gmail.com |
| 10 | 008 | Zoe | Staff | Sales | 4500 | zoe@macys.com |

2) members 파일을 급여 순으로 정렬하여 members_salary 파일에 저장하기 위한 명령을 쓰고, 실행 결과를 캡처한다. (명령 실행 후, gedit으로 members_salary 파일을 연 뒤 캡처)

```
$ sort -k 5 -o members_salary members.txt
```

```
u2312282@dahee-VirtualBox: ~/linux/chap09
u2312282@dahee-VirtualBox:~/linux/chap09$ sort -k 5 -o members_salary members.txt
u2312282@dahee-VirtualBox:~/linux/chap09$ gedit members_salary
```



| | 이름 | 직업 | 부서 | 급여 | 이메일 | |
|----|-----|--------|-----------|------------|------|--------------------|
| 1 | 006 | John | Staff | Sales | 4000 | john@macys.com |
| 2 | 008 | Zoe | Staff | Sales | 4500 | zoe@macys.com |
| 3 | 001 | Thomas | Manager | Sales | 5000 | thomas@gmail.com |
| 4 | 002 | Jason | Developer | Technology | 5500 | jason@yahoo.com |
| 5 | 007 | Aiden | Developer | Technology | 6000 | aiden@gmail.com |
| 6 | 005 | Randy | DBA | Technology | 6000 | randy@oracle.com |
| 7 | 010 | Olivia | DBA | Technology | 6500 | olivia@gmail.com |
| 8 | 003 | Sanjay | Sysadmin | Technology | 7000 | sanj@sys.admin.com |
| 9 | 009 | Emma | Manager | Technology | 9000 | emma@gmail.com |
| 10 | 004 | Nisha | Manager | Marketing | 9500 | nisha@gmail.com |

3. AWK 프로그램 작성

awk 프로그램 5개 각각에 대하여 다음과 같이 명령을 실행한 뒤 터미널 창을 캡처한다. (소스코드 전체가 다 보이게 캡처)

```
$ cat [awk 파일명]
$ awk -f [awk 파일명] [입력 파일명] // p11-12 참고
```

awk 프로그램 5개 각각에 대하여 p10의 기능을 구현하기 위해 어떻게 프로그래밍 하였는지 설명한다. (기능 구현이 100% 안되었더라도 구현한 범위까지 설명 작성)

1) 이름과 이메일만을 출력한다(hw1.awk)

```
u2312282@dahee-VirtualBox:~/linux/chap09$ cat hw1.awk
{print $2, $6}
u2312282@dahee-VirtualBox:~/linux/chap09$ awk -f hw1.awk members.txt
Thomas thomas@gmail.com
Jason jason@yahoo.com
Sanjay sanj@sys.admin.com
Nisha nisha@gmail.com
Randy randy@oracle.com
John john@macys.com
Aiden aiden@gmail.com
Zoe zoe@macys.com
Emma emma@gmail.com
Olivia olivia@gmail.com
```

파일의 모든 줄에 대하여 두 번째 필드 \$2(이름), 여섯 번째 필드 \$6(이메일)을 출력하도록 한다.

2) 가장 많은 급여를 받는 직원의 이름과 급여만을 출력한다(hw2.awk)

```
u2312282@dahee-VirtualBox:~/linux/chap09$ cat hw2.awk
{if($5>most){
    most=$5
    name=$2
}
}
END{print "Name:",name, "Salary:",most}
u2312282@dahee-VirtualBox:~/linux/chap09$ awk -f hw2.awk members.txt
Name: Nisha, Salary: 9500
```

파일의 모든 줄에 대하여 다섯 번째 필드 \$5(급여)가 급여의 최대값을 저장하는 변수 most 보다 크면 해당 급여값을 most로 설정하고, 해당 줄의 두 번째 필드(이름)을 가장 많은 급여를 받는 직원 이름 name 변수에 저장한다. 파일 끝을 만나면 name 값과 most값을 출력해 가장 많은 급여를 받는 직원의 이름과 급여 값을 표시한다.

3) 각 부처별 급여액을 계산하여 출력하고, 전체 급여액을 계산하여 출력한다.(hw3. awk)

```
u2312282@dahee-VirtualBox: ~/linux/chap09
u2312282@dahee-VirtualBox:~/linux/chap09$ cat hw3.awk

{
if($4=="Technology") salarys[$4]+=$5
else if($4=="Marketing") salarys[$4]+=$5
else if($4=="Sales") salarys[$4]+=$5
total+=$5
}
END{
for(s in salarys)
    print "Dept: " s " , Salary: " salarys[s]
    print "Total Salary: "total
}

u2312282@dahee-VirtualBox:~/linux/chap09$ awk -f hw3.awk members.txt
Dept: Technology, Salary: 40000
Dept: Marketing, Salary: 9500
Dept: Sales, Salary: 13500
Total Salary: 63000
```

파일의 각 줄에 대해 4번째 필드(부처)가 “Technology”일 경우, “Marketing”일 경우, “Sales”일 경우의 if-else if 문을 작성한다. 부처 이름을 나타내는 단어를 인덱스로 사용해 부서별 전체 급여액을 저장하는 연관 배열 salarys를 만든다. Technology 부처에 해당하는 줄의 경우 해당 줄의 5번째 필드(급여)를 salarys[\$4]에 더한다. 다른 부처들에 대해서도 똑같은 작업을 실행하고, 전체 급여액의 총합을 구하기 위해 전체 급여 총합 변수 total에 모든 줄의 급여 정보를 더한다.

파일의 끝을 만날 경우 salarys 배열의 모든 인덱스와 그에 대응하는 값을 출력해 부서 이름과 부서별 총 급여액을 출력하고, total 값을 출력해 전체 급여액 총합을 출력한다.

4. 급여가 6000 이상인 직원들의 이름과 급여를 출력한다.(hw4.awk)

```
u2312282@dahee-VirtualBox:~/linux/chap09$ cat hw4.awk
{if($5>=6000) print $2,$5}
u2312282@dahee-VirtualBox:~/linux/chap09$ awk -f hw4.awk members.txt
Sanjay 7000
Nisha 9500
Randy 6000
Aiden 6000
Emma 9000
Olivia 6500
```

이 파일의 모든 줄에 대하여 각 줄의 5번째 필드(급여)가 6000 이상인지 if문을 통해 확인하고, 확인 결과가 참일 경우 해당 줄의 2번째 필드(이름)과 5번째 필드(급여)를 출력한다.

5. 이 파일의 각 줄을 역순으로 출력한다. (hw5.awk)

```
u2312282@dahee-VirtualBox:~/linux/chap09$ cat hw5.awk
{a[NR]=$0}
END{for(i=NR; i>=1; i--) print a[i]}
u2312282@dahee-VirtualBox:~/linux/chap09$ awk -f hw5.awk members.txt
010 Olivia DBA Technology 6500 olivia@gmail.com
009 Emma Manager Technology 9000 emma@gmail.com
008 Zoe Staff Sales 4500 zoe@macys.com
007 Aiden Developer Technology 6000 aiden@gmail.com
006 John Staff Sales 4000 john@macys.com
005 Randy DBA Technology 6000 randy@oracle.com
004 Nisha Manager Marketing 9500 nisha@gmail.com
003 Sanjay Sysadmin Technology 7000 sanj@sys.admin.com
002 Jason Developer Technology 5500 jason@yahoo.com
001 Thomas Manager Sales 5000 thomas@gmail.com
```

이 파일의 모든 줄에 대해 배열 a에 자신의 줄 내용 전체를 자신의 줄 번호 번째 인덱스를 가진 원소로 저장한다. 파일의 끝을 만나면 for문을 통해 줄번호의 최대값 10에서 1까지 번호를 역으로 짚으며 배열 a의 원소를 출력해 최종적으로 각 줄이 역순으로 출력되도록 한다.