■ 리눅스 환경설정

리눅스 환경설정

1. 오라클 리눅스 가상환경 이미지(10Gb) file: 11gWS2.ova

2. 오라클 가상환경 설치파일
   VirtualBox-5.0.14-105127-Win.exe
   virtualBox : pc안에 또 다른 pc 환경을 구성

      username : oracle
      pw : oracle

- 오라클 리눅스 서버에 접속하게 도와주는 putty를 이용해서 오라클 리눅스에 접속

■ 리눅스 수업 목차

---

1. 리눅스란 무엇인가?

unix가 너무 고가여서 linux 오픈소스를 핀란드의 리누즈 토발즈가 1991년 11월에 개발
리누즈 토발즈가 리눅스 커널(자동차의 엔진)을 개발하고 소스를 무료로 공개.
전세계 개발자들이 이 오픈소스를 이용하여 더 좋게 개선해서 올리고 개선 작업을 반복하다보니 linux os가 unix보다 더 가볍고 안정적

GNU proj. : 누구든지 배포된 오픈소스를 이용해 개발 할 수 있고 상용화도 가능
소스를 가져다 더 좋게 수정했으면 해당 코드를 인터넷이 올려줘야하는 약속이 있다.

RedHat(상용버전, 유료) ---------- Cent OS

* Linux의 종류

1. Oracle Linux
2. Cent OS
3. Ubuntu

* Linux를 배워야 하는 이유

1. 현업의 대용량 데이터는 대부분 window가 아니라 Linux, Unix 시스템에 저장

2. 하둡이 Linux system에 설치되어져 사용되고 있음
3. 딥러닝을 구현하는 환경의 많은 부분이 linux 환경

---

2. 리눅스 기본명령어

◎ 1. cd 명령어

Change Directory 명령어로 디렉토리를 이동하는 명령어

```
$pwd : 현재 작업 중인 디렉토리 확인
(Print Working Directory)

$ls : 현재 디렉토리에 있는 폴더와 파일 확인
(list)

$cd labs : labs 디렉토리로 이동
```

```
※ 문제1. 다시 /home/oracle 디렉토리로 이동하시오
$cd ..
```

```
※ 문제2. 최상위 디렉토리인 /(root) directory로 바로 이동하시오
$cd /
```

```
※ 문제3. /home으로 이동하시오
$cd /home
```

※ 경로에는 크게 2가지 경로가 존재.

1. 절대경로 : cd '가고자 하는 정확한 위치'

```
$cd /home
```

2. 상대경로 : 나의 현재위치를 상대로 이동

```
$cd .. (현재 위치에서 상위 디렉토리로 이동)
```

```
※ 문제4. 집으로 가시오
$cd
```

※ oracle 유저의 홈 디렉토리로 이동하는 명령어

```
※ 문제5. /home/oracle/labs/demos/fbt_backout로 이동하시오
$cd labs/demos
```

```
※ 문제6. 지금 상태에서 /home/oracle/labs 로 이동하는데 절대경로로 이동하지 말고 상대경로로 이동하시오!
$cd ../..
```

```
※ 문제7. 다시 /home/oracle/labs/demos/fbt_backout 로 가시오. 상대경로로 가시오
$cd -
```

◎ 2. touch 명령어

파일의 용량이 0인 파일을 생성하는 명령어

```
$touch a1.txt # a1.txt 파일을 생성
$ls -l a1.txt # a1.txt 파일의 용량을 확인
```

```
※ 문제8. /home/oracle 밑에 아래의 파일들을 크기를 0으로 생성하시오
a.txt, b.txt, c.txt, d.txt, e.txt, f.txt
$touch a.txt b.txt c.txt d.txt e.txt f.txt
```

◎ 3. mkdir 명령어

디렉토리를 생성하는 명령어

```
$mkdir oracle_data

[orcl:~]$ whatis mkdir
mkdir                (1)  - make directories
mkdir                (1p)  - make directories
mkdir                (2)  - create a directory
mkdir                (3p)  - make a directory

$man mkdir # mkdir 메뉴얼 보는 명령어

:q # manual 빠져나오기
```

```
※ 문제9. 아래와 같이 /home/oracle 밑에 아래의 하위 디렉토리들을 만드시오
/home/oracle/test1/test2/test3/test4/test5
$mkdir -p test1/test2/test3/test4/test5
```

※ -p 옵션은 디렉토리를 만들면서 한 번에 하위 디렉토리를 생성하는 옵션

■ 리눅스 수업 목차

◎ 4. rm 명령어

파일이나 디렉토리를 삭제하는 명령어

※ 주의사항

리눅스나 유닉스는 휴지통이 없기 때문에 삭제할 때 특히 주의해야 한다

```
-rw-r--r--   1     oracle     oinstall    0       Jun  2 10:03 bbb.txt
  권한      링크수   소유자명     그룹명    파일크기    생성일자
```

```
$rm bbb.txt
```

```
※ 문제10. /home/oracle 밑에 확장자가 .txt 인 파일들을 전부 삭제하시오
$rm *.txt
```

※ rm -rf *

현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 다 삭제하겠다

-r 옵션 : 현재 디렉토리 밑에 있는 모든 파일과 디렉토리까지 다 삭제
-f 옵션 : 삭제할 때 확인 메시지 없이 삭제

```
※ 문제11. /home/oracle/test20 이라는 디렉토리를 만드시오
$mkdir test20
```

```
※ 문제12. test20 디렉토리를 지우시오
$rm -rf test20
```

◎ 5. rmdir 명령어

디렉토리를 삭제하는 명령어

```
$rmdir 디렉토리명
$mkdir ddd
$rmdir ddd
```

```
※ 문제13. /home/oracle 밑에 test99 라는 디렉토리를 만들고 test10이라는 디렉토리를 만드시오
$mkdir -p test99/test10
```

```
※ 문제14. /home/oracle 밑에 emp.txt를 올리시오
윈도우OS ------------------> 리눅스 OS
              파일 전송

1. emp.txt를 복사해서 c:\Prgram Files\PuTTY 밑에 둔다.
2. cmd 창에 들어가서 cd c:\Prgram Files\PuTTY 로 들어간다.
3. c:\Program Files\PuTTY >pscp d:\* root@192.168.56.104:/home/oracle
    root@192.168.56.104's password: (비밀번호입력)
```

◎ 6. alias 명령어

자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어 -> **단축키**

```
$alias scott="sqlplus scott/tiger"
$scott -> sql접속
```

* $alias : 만들어진 alias 명령어 확인

```
※ 문제15. p를 입력하면 바로 파이썬으로 접속될 수 있도록 alias를 생성하시오
$alias p="python"
$p
```

```
※ 문제16. p alias를 지우시오
$unalias p
```

◎ 7. cat명령어

파일의 내용을 화면에 출력하는 명령어

```
$cat emp.txt
```

```
※ 문제17. 겨울왕국 대본을 /home/oracle 밑에 winter.txt로 올리시오
$vi winter.txt
i 누르고 입력모드
마우스 우클릭으로 붙여넣기
:wq(저장 후 닫기)
```

```
※ 문제18. 겨울왕국 대본을 cat으로 여시오
$more winter.txt
q로 탈출
```

※설명 : cat은 파일 전체를 한 번에 다 보는 명령어라 긴 텍스트를 볼 때는 적합하지 않으므로 more 명령어를 사용한다.

◎ 8. redirection 명령어

화면에 출력되는 결과를 파일로 저장하는 명령어

```
>> : 없으면 파일을 생성하고 있으면 기존 파일 뒤에 덧붙이기
> : 파일을 생성. 기존에 파일이 있으면 덮어쓰기
$cat emp.txt >> emp_backup.txt
$cat emp_backup.txt
```

```
※ 문제19. /home/oracle 밑으로 가서 확장자가 .txt로 끝나는 파일들이 무엇이 있는지 조회하시오
$ll *.txt
```

```
※ 문제20. 문제19번에 ls -l *.txt 를 했을 때 보이는 화면의 결과를 all.txt라는 이름으로 저장하시오
$ll *.txt > all.txt
```

◎ 9. more 명령어

1페이지가 넘는 문서의 내용을 화면에 출력할 때 페이지 단위로 볼 수 있는 명령어

```
$more winter.txt
전진키 : 스페이스 바
후진키 : b
페이지 단위로 내려갈 때 : f
```

```
※ 문제21. 아나콘다 설치파일을 리눅스 서버에 올리시오
cmd>pscp Anaconda3-2019.07-Linux-x86_64.sh root@192.168.56.104:/home/oracle
```

◎ 10. head 명령어

문서의 처음 몇 줄을 화면에 출력하는 명령어

```
$head 출력줄수 파일명

$head -20 winter.txt
```

```
※ 문제22. 겨울왕국 대본의 100줄까지의 내용을 winter_head.txt로 저장하시오
$head -20 winter.txt >> winter_head.txt
```

◎ 11. wc 명령어

파일안의 단어의 개수 또는 라인수를 출력하는 명령어

```
$wc -l 파일명

$wc -l winter_head.txt

-l : 라인수 출력
-w : 단어수 출력
```

```
-c : 문자수 출력
```

```
※ (점심시간 문제) 아래의 리스트의 숫자의 총 합을 구하시오
b=[1,2,3,4,5,[8,9],10]
```

```
※ 문제23. /home/oracle 밑에 확장자가 .txt로 끝나는 파일들을 ls -l 명령어로 조회하시오
$ll *.txt
```

```
※ 문제24. 화면에 출력된 결과의 건수를 출력하시오
$ls -l *.txt >> tc.txt
$wc -l tc.txt
```

※ 더 간단하게 하는 방법

```
$ls -l *.txt | wc -l
| : pipe 명령어
```

※ pipe 명령어 : 앞의 명령어 | 뒤의 명령어

앞의 명령어의 출력을 뒤의 명령어의 입력으로 보냄으로써 명령어의 실행결과를 다음 명령어로 전달하는 기능

```
※ 문제25. /home/oracle 밑에 있는 파일과 디렉토리가 전부 몇 개가 있는지 조회하시오
$ll | wc -l
```

◎ 12. grep 명령어

파일안에 포함된 특정 단어나 구문을 검색하는 명령어

```
$grep '찾고싶은 단어' 파일명

$grep -i 'scott' emp.txt
-i : 대소문자 구분 X
```

```
※ 문제26. 직업이 salesman인 사원들의 모든 행을 출력하시오
$grep -i 'salesman' emp.txt
```

```
※ 문제27. 직업이 salesman인 사원들이 전부 몇 명인지 출력하시오
$grep -i 'salesman' emp.txt | wc -l
```

```
※ 문제28. 직업이 salesman인 사원들의 이름과 월급을 출력하시오
$grep -i 'salesman' emp.txt | awk '{print $2, $6}'
```

※ awk를 이용해서 열만 선택해서 출력할 수 있음. 열은 공백과 공백으로 구분

```
※ 문제29. 직업이 analyst인 사원들의 이름과 월급과 직업을 출력
$grep -i 'analyst' emp.txt | awk '{print $2, $6, $3}'
```

```
※ 문제30. 부서번호가 10번인 사원들의 이름과 월급과 부서번호를 출력하시오
$grep -i '10' emp.txt
```

※ 설명: ADAMS는 부서번호가 20번인데 출력. ADAMS의 월급 1100에 '10'이 포함되어 있기 때문

```
$awk '{print $2, $6, $8}' emp.txt | grep -iw '10'
```

※ 설명: 옵션 -w는 단어별 검색을 할 때 사용

◎ 13. awk 명령어

특정 컬럼을 출력하고자 할 때 사용하는 명령어

```
$awk '패턴{action}' 대상파일명

$awk '$3=="SALESMAN" {print $2, $3}' emp.txt
$awk '{print $2, $3}' emp.txt
```

```
$awk '$8=="10"{print $2, $6, $8}' emp.txt
```

※ 리눅스의 연산자 3가지

1. 산술 연산자 : +, -, *, /
2. 비교 연산자 : >, <, >=, <=, ==, !=
3. 논리 연산자 : &&, ||, !

```
※ 문제31. 월급이 3000이상인 사원들의 이름과 월급을 출력하시오
$awk '$6>=3000{print $2, $6}' emp.txt
```

```
※ 문제32. 직업이 SALESMAN이 아닌 사원들의 이름과 직업을 출력하시오
$awk '$3!="SALESMAN"{print $2, $3}' emp.txt
$awk '$3!=toupper("salesman"){print $2, $3}' emp.txt
```

```
※ 문제33. 81년도에 입사한 사원들의 이름과 입사일을 출력하시오
$awk 'substr($5,3,2)=="81"{print $2, $5}' emp.txt
```

```
※ 문제34. 이름의 첫 글자가 A로 시작하는 사원들의 이름과 월급을 출력하시오
$awk 'substr($2,1,1)=="A"{print $2, $6}' emp.txt
$awk '{print $2, $6}' emp.txt | grep -i '^a'
```

^ : 시작
$: 끝

```
※ 문제35. 이름의 끝 글자가 T로 끝나는 사원들의 이름을 출력하시오
$awk '{print $2}' emp.txt | grep -i 't$'
```

◎ 14. sort 명령어

data를 특정 컬럼을 기준으로 정렬하는 명령어

```
$sort 옵션 파일명
$sort -k 6 emp.txt
6번째 열을 기준으로 asc 정렬

$sort -rk 6 emp.txt
6번째 열을 기준으로 desc 정렬
정렬기준은 문자열의 ascii 값으로 정렬
```

※ 설명 : sort -옵션 컬럼순서번호

-n : sort를 할 때 숫자형태로 정렬

```
※ 문제36. 직업이 SALESMAN인 사원들의 이름과 월급을 출력하는데 월급이 높은 사원부터 출력
sort -nrk 6 emp.txt | awk '$3=="SALESMAN"{print $2,$3,$6}'
```

```
※ 문제37. 사원 테이블의 월급의 토탈값을 출력하시오
$awk '{print $6}' emp.txt | awk '{sum += $1} END {print sum}'
```

※ 설명 : sum += $1은 sum = sum + $1 과 같음

```
※ 문제38. 현재 디렉토리에 있는 *.txt 파일들의 크기의 총 합을 출력하시오
ll *.txt | awk '{sum += $5} END {print sum}'
```

```
※ 문제39. 위의 스크립트를 따로 파일로 저장해서 수행되게 하시오!
$vi size.sh
    ls -l *.txt | awk '{sum += $5} END {print sum}'
esc -> shift + zz

$sh size.sh
```

vi 편집기에서 저장하고 나오는 단축키 : ZZ

◎ 15. uniq 명령어

중복된 라인을 제거하는 명령어

```
$uniq 옵션 파일명
```

```
※ 문제40. emp.txt에서 직업만 출력하시오
$awk '{print $3}' emp.txt
```

```
※ 문제41. 위의 결과를 abcd 순으로 정렬해서 출력하시오
$awk '{print $3}' emp.txt | sort -k 1
```

```
※ 문제42. 위의 결과를 출력할 때 중복제거를 해서 출력하시오
$awk '{print $3}' emp.txt | sort -k 1 | uniq
```

※ uniq를 사용해서 중복을 제거하려면 먼저 정렬을 해서 같은 데이터끼리 모아놔야 한다

```
※ 문제43. 부서번호를 중복제거해서 출력하시오
awk '{print $8}' emp.txt | sort -nk 1 | uniq
```

*** 다음 실습을 위한 환경구성**

1. emp.txt를 복사해서 emp7.txt로 붙여넣는다

```
$cp emp.txt emp7.txt
$cat emp7.txt
```

2. emp7.txt를 열어서 KING에 대한 행을 복사해서 맨 아래에 붙여넣는다

```
$vi emp7.txt
    yy를 눌러서 행을 복사
    붙여넣을 행으로 이동 후 p
    :wq 또는 ZZ
$cat emp7.txt
```

```
※ 문제44. emp7.txt에서 중복된 라인만 출력하시오
$sort emp7.txt
    옵션을 주지 않고 정렬하면 맨 앞에 있는 데이터를 기준으로 정렬 : 사원번호를 기준으로 정렬
$sort emp7.txt | uniq -d
    -d 옵션은 중복된 행만 출력할 때 사용하는 옵션
```

◎ 16. echo 명령어

출력하고자 하는 글자를 출력할 때 사용하는 명령어

```
$echo 'yahooooooo~~~~~~~~~~~~~~~~~~'
```

```
※ 문제45. 직업을 물어보게하고 직업을 입력하면 해당 직업을 갖는 사원들의 이름과 직업이 출력되게 하시오
```

```
$vi job2.sh
    echo -n "Enter the job : "
    read job # 위에서 입력한 문자를 job변수에 넣는다
    grep -i $job emp.txt | awk'{print $2, $3}' # 변수명 앞에 $를 붙여서 사용
$sh job2.sh
```

```
※ 문제46. 부서번호를 물어보게 하고 부서번호를 입력하면 해당 부서번호에서 근무하는 사원들의 이름과 월급과 부서번호가 출력되게 하시오
$sh deptno2.sh
    Enter deptno : 10
...
$vi deptno2.sh
    echo -n "Enter deptno : "
    read deptno
    awk '$8=='$deptno'{print $2, $6, $8}' emp.txt
$sh deptno2.sh
```

```
awk -v a=$deptno '$8==a {print $2, $6, $8}' emp.txt

-v val=val
    Assign the value val to the variable var. before execution of the program begins.
    such variable values are available to the BEGIN block of an AWK program.
```

```
※ 문제47. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 아래와 같이 출력되게 하는 shell 스크립트를 작성하시오
$sh find_sal.sh
    Enter the ename : scott
    3000

$vi find_sal.sh
    echo -n "Enter the ename : "
    read ename
    awk -v a=$ename  '$2==toupper(a) {print $6}' emp.txt

$sh find_sal.sh
```

```
※ 문제48. 문제47번을 다시 수행하는데 결과가 아래와 같이 한글로 출력되게 하시오
$sh find_sal.sh
    Enter the ename : scott
    SCOTT의 월급은 3000 입니다.

$vi find_sal.sh
    echo -n "Enter the ename : "
    read ename
    awk -v a=$ename '$2==toupper(a) {print $2,"▒~]~X ▒~[~T▒~I▒~]~@"$6,"▒~^~E▒~K~H▒~K▒"}' emp.txt
```

*** linux 서버가 한글을 인식할 수 있도록 설정**

```
$export LANG='ko_KR.UTF-8"
```

LANG 환경변수에 ko_KR.UTF-8을 할당한다.
LANG 환경변수는 이 서버의 언어를 설정하는 예약 변수

```
※ 문제49. dept.txt를 이용해서 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호의 부서위치가 출력되게 하시오
$vi find_loc.sh
    echo -n "부서번호를 입력하세요 : "
    read deptno
    awk -v a=$deptno '$1==a {print $3 $4}' dept.txt
또는
    echo -n "부서번호를 입력하세요 : "
    read deptno
    grep -i "$deptno" dept.txt | awk '{print $3}'
```

```
$sh find_loc.sh
```

dd : 한줄 삭제
u : 방금 했던 작업 취소
x : 철자 한 개 삭제

```
※ 문제50. 이름을 물어보게하고 이름을 입력하면 해당 사원의 부서위치가 출력되게 하시오
$sh find_loc2.sh
    이름을 입력하세요 : scott
    DALLAS에서 근무하는 사원 입니다.

$vi find_loc2.sh
    echo -n "이름을 입력하세요 : "
    read ename
    deptno=`grep -i $ename emp.txt | awk '{print $8}'` # 역 따옴표로 감싼 linux 명령어는 그 실행된 결과가 변
수에 할당
    loc=`grep -i $deptno dept.txt | awk '{print $3}'`
    echo "$loc에서 근무하는 사원입니다."
또는
    echo -n "이름을 입력하세요 : "
    read ename
    deptno=`awk -v a=$ename '$2==toupper(a) {print $8}' emp.txt`
    loc=`grep -i $deptno dept.txt | awk '{print $3}'`
    echo "$loc에서 근무하는 사원입니다."

    echo -n "이름을 입력하세요 : "
    read ename
    deptno=`grep -i $ename emp.txt | awk '{print $8}'` # 역 따옴표로 감싼 linux 명령어는 그 실행된 결과가 변
수에 할당
    loc=`grep -i ${deptno:0:2} dept.txt | awk '{print $3}'` # ${deptno:0:2} sql의 substr과 같은 방식
    echo "$loc에서 근무하는 사원입니다."
```

```
echo ${#deptno} # deptno 변수의 길이를 확인하는 코드
echo ${deptno:0:3} # 20
echo ${deptno:0:2} # 20
echo ${deptno:0:1} # 2
```

```
※ 문제51. 직업을 물어보게 하고 직업을 입력하면 해당 직업을 갖는 사원들의 이름과 월급이 출력되게 하시오
$sh job_sal.sh
    직업을 입력하세요 : SALESMAN
    ALLEN 1600
    MARTIN 1250
    WARD 1250
    TURNER 1500

$vi job_sal.sh
    echo -n '직업을 입력하세요 : '
    read job
    awk -v a=$job '$3==toupper(a) {print $2, $6}' emp.txt
$sh job_sal.sh
```

◎ 17. diff 명령어

두 파일간의 차이점을 찾아서 알려주는 명령어

```
$diff emp.txt emp7.txt
```

```
※ 문제52. 아래와 같이 두 개의 파일 이름을 각각 물어보게하고 결과 파일의 내용의 차이가 출력되게 하시오
$sh diff.sh
    비교할 첫번째 파일명을 입력하세요 emp.txt
    비교할 두번째 파일명을 입력하세요 emp7.txt

$vi diff.sh
    echo -n "비교할 첫번째 파일명을 입력하세요: "
    read c1
    echo -n "비교할 두번째 파일명을 입력하세요: "
    read c2
    diff $c1 $c2
$sh diff.sh
```

◎ 18. find 명령어

검색하고자 하는 파일을 찾을 때 사용하는 명령어

```
$find 디렉토리명  -name 파일명 -print
    검색할 디렉토리   검색할 파일

$find /home/oracle -name 'emp.txt' -print
    #/home/oracle 밑에 emp.txt라는 파일이 있는지 검색
```

```
※ 문제53. /home/oracle 밑에 test100과 test200이라는 디렉토리를 각각 생성하고 그 밑에 emp.txt를 복사하시오
$mkdir test100 test200
$cp emp.txt ./test100/emp.txt
$cp emp.txt ./test200/emp.txt
```

```
※ 문제54. /home/oracle 밑에 있는 emp.txt를 검색하시오
$find -name 'emp.txt' -print
```

```
※ 문제55. 현재 디렉토리 밑에 확장자가 .txt로 끝나는 파일들을 전부 검색하시오(하위 디렉토리까지 전부 다)
$find -name *.txt -print
```

```
※ 문제56. 파일을 검색하는 shell 스크립트를 아래와 같이 생성하시오
$sh find_file.sh
    현재 디렉토리 밑에 검색할 파일명을 입력하세요:
    depth는 어떻게 할까요?

$vi find_file.sh
    echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요: "
    read file
    echo -n "depth를 설정하세요: "
    read depth
    find ./ -name "$file" -maxdepth "$depth" -print
$sh find_file.sh
```

■ shell 스크립트를 실행할 때 마다 계속해서 한글이 안깨지고 잘 보이게 하려면?

```
export LANG=kor_KR.UTF-8
```

Linux를 키고 로그인 할 때 마다 위의 export 명령어가 자동으로 실행되게 하면 매번 수동으로 위의 명령어를 수행하지 않아도 된다.
oracle유저로 로그인 할 때 자동으로 위의 export 명령어가 수행되게 하려면 Linux OS의 환경설정 파일(.bash_profile)에 위의 명령어를 넣으면 된다.

```
$. .bash_profile -> 실행
```

◎ 19. tar 명령어

파일을 압축하고 압축을 해제하는 명령어

1. 압축할 때

```
$tar cvf (압축 파일명) (압축 파일 대상)
```

2. 압축을 해제할 때

```
$tar xvf (압축 파일명) -C (압축을 해제할 위치)
```

c : compress, 여러개의 파일을 하나로 만들어라
v : view, 압축되는 과정을 보여라
f : file, 생성되는 파일명을 지정
x : extract, 묶여있는 파일을 해제

-C 압축이 풀릴 위치를 지정

```
# 예제2
emp.txt를 복사해서 emp2.txt emp3.txt emp4.txt emp5.txt로 복사
```

```
# 예제3
emp로 시작하는 모든 txt 파일을 empall.tar로 압축시키시오
```

```
# 예제4
/home/oracle 밑에 test77이라는 디렉토리를 만들고 그 디렉토리에 empall.tar 파일의 압축을 해제하시오
```

```
※ 문제57. 겨울왕국 대본 winter.txt를 복사해서 winter1.txt ~ winter7.txt로 복사하고 winter.tar라는 이름으로
          이 파일들을 모두 압축하시오
$tar winter.tar winter*.txt
```

◎ 20. ln(link) 명령어

윈도우의 바로가기 기능과 유사함

내가 자주 열어봐야 할 파일이 있다면 바로가기를 생성해 놓으면 편하게 열어볼 수 있다.

```
$ln -s "링크를 걸 파일위치와 파일명" "바로가기 이름"
```

```
$sqlplus scott/tiger

sql> select * from salgrade;
    내용만 copy
sql> exit

$vi salgrade.txt
    내용 붙여넣기
$mkdir -p ./test22/test23/test24/test25/test26/test27
$mv salgrade.txt /test22/test23/test24/test25/test26/test27/

    링크생성
$ln -s ./test22/test23/test24/test25/test26/test27/salgrade.txt
$cat salgrade.txt
```

※ 문제**58. salgrade.txt** 바로가기를 삭제하시오

```
$rm salgrade.txt
```

◎ 21. sed 명령어

grep 명령어는 파일의 특정 내용을 검색하는 기능을 갖는다면
sed는 명령어는 검색 뿐만 아니라 내용을 변경할 수 있다

```
$sed 's/KING/yyy/' emp.txt

    s/변경 전 텍스트/변경 후 텍스트
```

실제로 emp.txt가 변경되는 것이 아니고 변경해서 출력만 해줌

```
※ 문제59. 위의 변경된 내용대로 출력한 내용을 emp9.txt로 저장하시오
sed 's/KING/yyy/' emp.txt >> emp9.txt
```

◎ 22. case문

if문과 유사한 linux의 shell 스크립트 명령어

```
echo "---------------------------------"
echo " 1. 두 파일의 차이를 확인려면 1번
2. 특정 파일을 검색하려면 2번
    "
echo -n "번호를 입력하세요: "
read choice
case $choice in
        1)
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
esac
```

※ 설명 : case문으로 시작했으면 esac로 종료

case와 in 사이에 변수명을 사용하고 변수에 들어가는 숫자를 아래의 스크립트에서 실행

```
※ 문제60. 겨울왕국 스크립트에서 elsa가 포함된 라인이 몇 개 인지 출력하시오
$grep -i 'elsa' winter.txt | wc -l
```

```
※ 문제61. 위의 스크립트를 이용해서 아래와 같이 실행되게 하시오
$sh find_word.sh
    스크립트를 입력하세요 : /home/oracle/winter.txt
    검색할 단어를 입력하세요 : elsa
    318행이 검색되었습니다

$vi find_word.sh
    echo "------------------------------------------------"
    echo -n "스크립트를 입력하세요 : "
    read script
    echo -n "검색할 단어를 입력하세요 : "
    read word
    cnt=`grep -i $word $script | wc -l`
    echo "$cnt행이 검색 되었습니다."
    echo "------------------------------------------------"
$sh find_word.sh
```

```
※ 문제62. 위에서 만든 find_word.sh를 total.sh의 3번으로 실행되게 하시오
$vi total.sh
```

```
    echo "---------------------------------"
    echo " 1. 두 파일의 차이를 확인려면 1번
    2. 특정 파일을 검색하려면 2번
    3. 특정 단어를 검색하려면 3번


         "
    echo -n "번호를 입력하세요: "
    read choice
    case $choice in
        1)
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
        3)
                sh /home/oracle/find_word.sh ;;
    esac
$sh total.sh
```

※ 문제**63. total.sh**의 **4**번에 오라클에 **scott**으로 접속하는 것을 추가하시오

```
$vi total.sh
    echo "---------------------------------"
    echo " 1. 두 파일의 차이를 확인려면 1번
    2. 특정 파일을 검색하려면 2번
    3. 특정 단어를 검색하려면 3번
    4. Oracle 접속(SCOTT으로 접속) 4번
     "
    echo -n "번호를 입력하세요: "
    read choice
    case $choice in
        1)
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
        3)
                sh /home/oracle/find_word.sh ;;
        4)
                sqlplus scott/tiger ;;
    esac

$sh total.sh
```

※ 문제**64. sh total.sh**로 다 써서 실행하게 하지 말고 그냥 **t**라고 하면 **sh total.sh**가 실행되게 **alias**를 만드시오

```
$alias t='sh total.sh'
$t
```

※ 문제**65. Linux**를 켤 때 마다 위의 **alias**를 매번 수행하지 않도록 그냥 **Linux**를 켜고 **oracle** 유저로 로그인 할 때 자동으로 **alias t='sh total.sh'**가 실행되게 하시오

```
$vi .bash_profile
    맨 아래에 alias t='sh total.sh' 추가
$. .bash_profile
$t
```

※ 문제**66. Linux**에서 아래의 파이썬 코드를 실행할 수 있도록 하시오

```
$vi find_ename.py
    import csv
    name=row_input("이름을 입력하세요: ")
    file=open("/home/oracle/emp2.csv")
    emp_csv=csv.reader(file)

    for i in emp_csv:
        if i[1]==name:
            print(i[1],i[5])
$python find_ename.py
```

```
※ 문제67. total.sh 스크립트 5번에 find_ename.py 파이썬 코드를 실행하는 것을 추가하시오
$vi find_ename.sh
    python find_ename.py
$vi total.sh
    5번 추가
    sh find_ename.py
$t
```

■ Oracle DB에서 csv 파일을 추출하기 위한 sqlplus 명령어들

```
$sqlplus scott/tiger

SQL> select * from dept;
SQL> set heading off -- column명 없이 출력
SQL> set colsep ',' -- column 사이에 ,를 붙임
SQL> select * from dept;
        10,ACCOUNTING    ,NEW YORK
        20,RESEARCH      ,DALLAS
        30,SALES         ,CHICAGO
        40,OPERATIONS    ,BOSTON
SQL> set sqlprompt "" -- 'SQL>'을 안보이게 설정
select * from dept;
        10,ACCOUNTING    ,NEW YORK
        20,RESEARCH      ,DALLAS
        30,SALES         ,CHICAGO
        40,OPERATIONS    ,BOSTON
set trimspool on
spool dept.csv
select * from dept;
spool off
exit
$cat dept.csv
```

* 위의 스크립트들을 shell 스크립트에서 수행하기 위한 스크립트 생성

```
$vi create_csv.sh
    echo -n "테이블 명을 입력하세요: "
    read table
    sqlplus scott/tiger << EOF -- 접속 후 자동으로 수행할 스크립트 경계 표시
    set colsep ','
    set sqlprompt ""
    set heading off
    set trimspool on
    spool $table.csv
    select * from $table;
    spool off
    EOF -- 자동 스크립트 종료
$sh create_csv.sh
```

◎ 23. cp 명령어

파일을 복사하는 명령어

```
$cp 파일명 (복사할 파일명)
$cp emp.txt emp40.txt
$cp 위치/파일명 위치/복사할 파일명
$cp ./emp.txt ./labs/emp2.txt
```

```
※ 문제68. /home/oracle 밑에 있는 확장자 .txt 파일들을 전부 /home/oracle/backup 이라는 폴더를 만들고 거기에 전부 복사하시오
$mkdir backup
$cp ./*.txt ./backup
$ll ./backup
```

※ 리눅스는 휴지통이 없으므로 중요한 데이터는 반드시 cp명령어로 백업을 해야한다.

```
※ 문제69. /home/oracle 밑에 total_script라는 디렉토리를 만들고 /home/oracle 밑에 있는 .sh로 끝나는 모든 shell 스크립트를 복사하시오
$mkdir total_script
$cp *.sh ./total_script
```

```
※ 문제70. /home/oracle/total_script 폴더에 있는 모드 shell 스크립트를 d:/script 라는 폴더에 내리시오
cmd)
pscp root@192.168.56.104:/home/oracle/total_script/* d:/script
cmd
pscp root@
```

```
※ 문제71. go.bat 파일처럼 down.bat 파일을 만들어서 리눅스에 있는 파일을 윈도우로 전송하는 것을 자동화 하시오
down.bat
    @echo off
    set /p loc="파일 위치를 입력하세요 : "
    set /p dloc="파일을 저장할 위치를 입력하세요 : "
    pscp root@192.168.56.104:%loc% %dloc%
    pause
```

◎ 24. mv 명령어

파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어

```
$mv 기존파일명 새로운파일명
$mv emp40.txt emp50.txt
$ls emp*.txt
```

```
※ 문제72. /home/oracle 밑에 있는 emp로 시작하는 text 파일들을 /home/oracle/backup 밑으로 이동시키시오
$mv emp*.txt ./backup/
```

※ mv 명령어는 cp와는 다르게 파일 이름이 변경되거나 이동되는 거라서 주의해야 한다.

■ 리눅스 수업 목차

■ 3. vi 편집기 명령어

vi 편집기 : Linux 안에서 사용할 수 있는 문서 편집기

vi : Visual Editor

*** vi 편집기 명령모드 3가지**

command 모드	vi 편집기의 기본 모드이며 vi를 실행하면 바로 보이는 화면. 방향키로 움직일 수 있는 모드
edit 모드	a, i, o, x 등을 누르면서 내용을 입력 또는 삭제하는 명령모드
last line 모드	입력모드에서 저장, 종료, 강제종료 등의 명령어를 입력하는 모드 :wq 또는 ZZ → 저장하고 종료 :q! 또는 ZQ → 저장안하고 종료

※ edit 모드에서 command 모드 또는 edit모드에서 last line 모드로 갈려면 esc(취소)를 눌러야 한다.

■ 편집기를 시작하는 명령어

1. $vi 파일명 : 읽기, 쓰기가 가능한 상태로 파일이 열린다.

2. $view 파일명 : 읽기만 가능한 상태로 파일이 열린다.

※ vi 편집기를 열 때 정상적으로 안열리고 이상한 경고 메시지가 뜨면서 열릴 때 조치방법

- swap파일 삭제

■ command 모드에서 vi 편집기 내에서의 커서 이동

j	아래로 이동
k	위로 이동
h	왼쪽으로 이동
l	오른쪽으로 이동

1G	맨 위로 이동
G	맨 아래로 이동
:set nu	파일 내 텍스트에 번호 표시
:set nonu	파일 내 텍스트에 번호 표시 지움
gg	맨 위로 이동하는 단축키

```
※ 문제73. winter.txt의 1~200번째 줄까지의 내용을 winter200.txt라는 이름으로 저장하시오
$vi winter.txt
    :1,200 w winter200.txt
$cat winter200.txt
```

```
※ 문제74. 겨울왕국 대본의 600번째 라인부터 800번재 라인까지 따로 winter67.txt라는 이름으로 저장하시오
$vi winter.txt
    :600,800 w winter67.txt
$cat winter67.txt
```

■ vi 편집기의 삭제 명령어

x	철자 하나 삭제
dd	한 행 삭제
dw	커서에 있는 단어 삭제
:5, 10 d	5~10번재 행 삭제
D	커서에 있는 행 삭제

```
※ 문제75. winter.txt를 백업하고 열어서 맨 위에 한줄만 남기고 다 삭제하시오
$cp winter.txt winter_backup.txt
$vi winter.txt
    :set nu
    G
    :2,4240 d
```

■ vi 편집기 취소 명령어

u	방금 작업 취소

■ vi 편집기 복사/붙여넣기 명령어

yy	하나의 행 복사
p	붙여넣기
yG	현재행부터 파일 끝까지 복사
:1,2 co 3	1~2행을 3행 다음으로 복사
:1,2 m 3	1~2행을 3행 다음으로 이동

```
※ 문제76. 겨울왕국 대본의 전체 라인수를 먼저 확인하고 겨울왕국 대본 전체를 복사해서 아래에 붙여넣고
          겨울왕국 대본의 라인수가 2배가 되었는지 확인하시오
1. 첫번째 줄에 커서를 대고 yG
```

```
2. 대문자 G를 눌러서 맨 밑으로 이동
3. p를 불러서 붙여넣기
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어

3. vi편집기 명령어
- vi 편집기가 무엇인가
- vi 편집기의 3가지 모드
- vi 편집기 시작하는 명령어
- vi 편집기 내에서의 커서 이동
- vi 편집기의 입력 명령어
- vi 편집기의 삭제 명령어
- vi 편집기의 취소 명령어
- vi 편집기의 복사/붙여넣기 명령어

- vi 편집기 내에서의 문자를 변경하는 명령어

4. 권한 관리 명령어
5. 디스크 관리 명령어
6. 프로세서 관리 명령어
7. 유저 생성 및 관리
8. shell script 작성법
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ vi 편집기 내에서의 문자를 변경하는 명령어

1.

```
$vi winter.txt
    :/문자 ------------> 특정 문자 검색
    n키를 누르면서 계속 검색
    shift+n 누르면서 뒤로 검색
```

```
※ 문제77. emp.txt를 열고 SCOTT 문자가 있는지 검색하시오
$vi emp.txt
    :/SCOTT
```

2.

```
$vi
    :%s/기존문자/변경할 문자
    지금 커서가 있는 현재형의 기존문자를 변경할 문자로 변경하겠다

$vi emp.txt
    :%s/KING/aaa

    문서에 KING이 여러개가 있어서 그 여러개를 전부 변경하겠다고 한다면
    :%s/KING/aaa/g
```

```
※ 문제78. 직업 ANALYST를 전부 kkkkk로 변경하시오
$vi emp.txt
    :%s/ANALYST/kkkkk/g
```

```
※ 문제79. emp.txt에서 직업이 MANAGER인 사원들을 전부 SALESMAN으로 변경하시오
```

```
$vi emp.txt
    :%s/MANAGER/SALESMAN/g
```

```
※ 문제80. emp.txt를 emp1.txt ~ emp20.txt로 복사하시오 (shell 스크립트로 작성)
$vi cp_emp.sh
    i=1                         # i에 1을 할당
    while [ $i -le 20 ]         # i가 20보다 작거나 같을때까지 loop문을 실행
    do                          # 시작
        cp emp.txt emp$i.txt    # 실행할 loop문 작성
        let i = $(($i+1))
    done                        # 종료
$sh ep_emp.sh
```

※ 숫자열 비교

숫자1 -eq 숫자2	두 숫자가 같으면 true
숫자1 -ne 숫자2	두 숫자가 같지 않으면 true
숫자1 -gt 숫자2	숫자1이 숫자2보다 크면 true
숫자1 -ge 숫자2	숫자1이 숫자2보다 크거나 같으면 true
숫자1 -lt 숫자2	숫자 1이 숫자2보다 작으면 true
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면 true
!숫자1	숫자1이 거짓이라면 true

```
※ 문제81. emp1.txt ~ emp.20.txt의 내용을 변경하는데 SALESMAN을 jjj로 전부 변경하시오
$vi emp*.txt
    :argdo %s/SALESMAN/jjj/g | update
```

```
※ 문제82. winter.txt를 winter1.txt ~ winter100.txt로 100개 복사하시오
$vi cp_winter.sh
    i=1
    while [ $i -le 100 ]
    do
        cp winter.txt winter$i.txt
        let i=$(($i+1))
    done
$sh cp_winter.sh
$ll
```

```
※ 문제83. winter1.txt ~ winter100.txt 의 내용중에 by를 bybybybyby로 전부 한 번에 변경하시오
$vi winter*.txt
    :argdo %s/by/bybybyby/g | update
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어

4. 권한 관리 명렁어

5. 디스크 관리 명령어
6. 프로세서 관리 명령어
7. 유저 생성 및 관리
8. shell script 작성법
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ 권한관리 명령어

리눅스에 하둡을 설치하고 운영을 할 때 여러가지 문제들이 발생하는데 그 중의 많은 문제들이 대부분 권한 관련한 오류가 가장 많다.
하둡 운영을 원활하게 하기 위해서는 권한 관리 명령어를 잘 숙지해야 한다.

프로젝트(하둡)

데이터분석가 (예시: 아프리카tv 메시지들을 전부 하둡에 저장을 하고 메시지들을 단어 단위로 쪼개서 DB화 한 후 데이터를 ML로 분석
금지어가 나오면 방 폐쇄를 자동으로 시행할 수 있도록 시스템화)

*** 권한관리 명령어 3가지**

chmod	change mod
chown	change ownership of a file
chattr	change file attributes

번호	권한	대표문자	파일	디렉토리
1	읽기권한	**r**	읽기, 복사	디렉토리에서 ls 가능
2	쓰기권한	**w**	수정	디렉토리에서 파일생성 가능
3	실행권한	**x**	실행	디렉토리에서 cd로 접근 가능

■ 권한 부분을 좀 더 상세하게 분석

* ll emp.txt로 조회했을 때 나오는 권한부분 해석

-	rw-	r--	r--	1	oracle	oinstall
파일 형태확인	소유자 권한	그룹 권한	게스트 권한		소유자	그룹
d: 그룹 - : 파일 l : 바로가기(링크)						

설명 : emp.txt의 소유자는 읽고 쓸 수 있으며 소유자는 oracle
oinstall 그룹에 속한 유저들과 기타 유저들은 emp.txt를 쓸 수는 없고 읽기만 할 수 있다.

```
※예제1. emp.txt의 소유자인 oracle이 emp.txt를 쓰지 못하고 읽기만 하도록 권한을 수정하시오
$chmod u-w emp.txt
```

설명 : 읽기모드라 하더라도 쓰기 작업을 하고 wq!로 저장하면 저장이 된다

※ 권한관리 단축어 정리

권한	r : 읽기 w : 쓰기 x : 실행
u	유저
g	그룹
o	기타 유저

```
※ 문제84. emp.txt의 소유자 권한중에 읽기 권한도 삭제하시오
$chmod u-r emp.txt
```

```
※ 문제85. emp.txt의 소유자의 권한에 읽기와 쓰기 권한을 다시 넣으시오
$chmod u+rw emp.txt
```

```
※ 문제86. emp.txt에 대해서 소유자, 그룹, 기타 유저에 대해서 읽기, 쓰기, 실행권한을 모두 넣으시오
$chmod ugo+rwx emp.txt
또는
$chmod a+rwx emp.txt
```

```
※ 문제87. emp.txt에 대해서 소유자는 읽기만 가능하고 그룹과 기타 유저는 읽기, 쓰기 실행 못하게 하시오
$chmod a-rwx,u=r emp.txt
```

■ chattr 명령어

chattr 명령어를 이용하면 chmod 명령어 수행을 막을 수 있다.

* oracle 유저가 emp.txt에 대해서 chmod 명령어를 수행하지 못하도록 막는 명령어

```
#chattr +i emp.txt
#lsattr emp.txt
----i--------- emp.txt
※ 위와 같이 설정되면 oracle 유저가 emp.txt에 대해서 삭제, 변경, chmod 명령어를 수행하는게 불가능
#exit
$rm emp.txt
    rm: remove write-protected 일반 파일 `emp.txt'? yes
    rm: cannot remove `emp.txt': 명령이 허용되지 않음
```

```
※ 문제88. root유저로 접속해서 /home/oracle 밑에 emp.txt의 권한을 모두 삭제하시오
$su -
#cd /home/oracle
#chattr -i emp.txt
#chmod ugo-rwx emp.txt
```

불가능. 소유자가 oracle이므로 소유자를 root로 변경해줘야 권한 설정이 가능

■ chown 명령어

```
#chown root:root emp.txt # 소유자,그룹 변경
#ll emp.txt
```

```
※ 문제89. 다시 emp.txt의 소유자를 oracle로 변경하고 그룹을 oinstall로 변경하시오
#chown oralce:oinstall emp.txt
```

```
※ 문제90. 다시 oracle 유저로 접속해서 /home/oracle 밑에 ddd라는 디렉토리를 만들고 ddd 디렉토리의 소유자가 누구인지 확인하시오
#exit
$mkdir /home/oracle/ddd
$ll -d ddd
```

```
※ 문제91. /home/oracle/ddd의 소유자를 root로 변경하시오
$su -
#chown -R root:root /home/oracle/ddd
```

-R : 디렉토리 안에 있는 모든 파일들의 소유자를 전부 변경
옵션을 붙이지 않으면 디렉토리 소유권만 변경되고 안의 파일들은 변경되지 않음

```
※ 문제92. oracle 유저로 접속해서 /home/oracle 밑에 ccc라는 디렉토리를 만들고
          winter.txt를 복사해서 붙여넣고 소유자가 누구인지 확인하시오
$mkdir ccc
$cp winter.txt ./ccc/
$ll ./ccc
```

```
※ 문제93. root 유저로 접속해서 /home/oracle/ccc 폴더의 소유자와 /home/oracle/ccc 폴더 밑에 있는
          파일들의 소유자를 전부 root로 변경하시오
$su -
#chwom -R root:root /home/oracle/ccc
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명령어

5. 디스크 관리 명령어

6. 프로세서 관리 명령어
7. 유저 생성 및 관리
8. shell script 작성법
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ 디스크 관리 명령어

데이터를 저장하는 저장 영역 관리 명령어

df 명령어	현재 파일 시스템의 총 사용률을 확인하는 명령어
du 명령어	현재 파일/디렉토리의 디스크 사용량을 표시하는 명령어
sar 명령어	disk의 i/o 성능을 모니터링 하는 명령어

■ 1. df 명령어

현재 파일 시스템의 총 사용률을 확인하는 명령어

```
$df -h
```

■ 2. du 명령어

```
$du -c *.txt
    4       dept.txt
    4       emp.txt
    116     winter.txt
    124     합계
$du -hc *.txt
    4.0K    dept.txt
    4.0K    emp.txt
    116K    winter.txt
    124K    합계
단위까지 표시
```

■ 3. sar 명령어

disk의 I/O 성능을 모니터링 하는 명령어

```
$sar 1 100
`                CPU     %user     %nice   %system   %iowait    %steal     %idle
`                all     13.13      0.00     55.05      0.00      0.00     31.82
`                all     13.13      0.00     55.56      0.00      0.00     31.31
`                all     14.93      0.00     53.23      0.00      0.00     31.84
`                all     20.81      0.00     62.94      0.00      0.00     16.24
`                all     13.43      0.00     60.20      0.00      0.00     26.37
`                all     21.32      0.00     64.47      0.00      0.00     14.21
`                all     11.73      0.00     53.57      0.00      0.00     34.69
`                all     15.50      0.00     55.00      0.00      0.00     29.50
```

```
`                all     16.84      0.00     58.16      0.00      0.00     25.00
`                all     13.64      0.00     56.57      0.00      0.00     29.80
`                all     14.57      0.00     56.78      0.00      0.00     28.64
`                all      7.00      0.00     31.50      0.00      0.00     61.50
`                all      0.50      0.00      2.01      0.00      0.00     97.49
`                all      0.00      0.00      2.00      0.00      0.00     98.00
`                all      0.50      0.00      2.49      0.00      0.00     97.01
`                all      0.50      0.00      1.98      0.50      0.00     97.03
`                all      0.00      0.00      2.55      0.00      0.00     97.45
`                all      0.50      0.00      7.04      0.00      0.00     92.46

$kill -9 pid
    710860
    710861
    710862
    710863
    710864


     LEVEL
----------
    710865
죽었음
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명렁어
5. 디스크 관리 명령어

6. 프로세서 관리 명령어

7. 유저 생성 및 관리
8. shell script 작성법
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ 프로세스 관리 명령어

1. ps 명령어
2. top 명령어
3.
4. jobs 명령어

■ 4. jobs 명령어

동작중인 작업의 상태를 확인하는 명령어

* 상태정보 4가지

running	실행중
stopped	일시 중단중
Done	종료
terminated	강제 종료됨

```
$vi hhh.txt
    ~~~~작성 중 esc+ctrl+z
   [1]       +      Stopped
job 번호  현재job   일시중단됨
$jobs
    동작 중인 작업의 상태를 확인
$fg
    현재 job으로 접속하는 명령어
```

```
※ hhh2.txt ~ hhh4.txt까지 작업 일시 중단 반복
$jobs
[1]   Stopped                 vim hhh.txt
[2]   Stopped                 vim hhh2.txt
[3]-  Stopped                 vim hhh3.txt
[4]+  Stopped                 vim hhh4.txt
$fg 2
    2번 작업을 실행
```

■ 1. ps 명령어

현재 시스템에서 수행되고 있는 프로세서의 정보를 표시하는 명령어

```
$ps 옵션 프로세서 아이디
옵션
    -e : 현재 실행중인 모든 프로세서
    -f : 실제 유저명, 게시 시간을 표시
    -l : 프로세서의 상태, 우선도 등과 같은 상세한 정보 출력
    -p : 프로세서 ID

$ps -ef | grep vim
$kill -9 PID
[orcl:~]$ kill -9 11715
[1]   죽었음               vim hhh.txt
[orcl:~]$ kill -9 11719
시스템 호출
[2]+  죽었음               vim hhh2.txt
[orcl:~]$ kill -9 11726
-bash: echo: write error: 중단된 시스템 호출
[3]-  죽었음               vim hhh3.txt
[orcl:~]$ kill -9 11729
[4]+  죽었음               vim hhh4.txt
[orcl:~]$ jobs
$jobs
$
```

```
※ 문제94. total.sh 스크립트에 6번에 현재 디스크 상태를 모니터링 하는 스크립트를 추가하시오
6. 현재 디스크 상태 모니터링

$vi total.sh
    6번 추가
    sar 1 100
$sh total.sh
```

■ 2. top 명령어

지금 현재 작동중인 프로세서들의 cpu 사용률과 메모리 사용률을 확인하는 명령어

내가 돌린 파이썬 프로그램이 cpu를 많이 사용하고 있는 것은 아닌지 예를 들어 무한루프를 돌렸다거나

프로그램 코딩을 잘못해서 리눅스의 자원을 많이 사용하고 있다면 문제가 발생할 소지가 있으므로 top 명령어를 수행해서 확인할 필요가 있다.

```
※ 문제95. example01.py라는 이름으로 무한루프를 도는 파이썬 프로그램을 만들고 돌리시오
$vi example01.py
    i=0
    while 1:
        i += 1
        print(i)
$python example01.py
```

```
$top -p PID
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
12393 oracle    18   0  7080 2192 1296 S 40.1  0.1   4:28.12 python

$kill -9 12393
```

```
※ 문제96. 프로세서를 kill 시키는 스크립트를 생성하시오
$vi kill_p.sh
    echo -n "죽이고 싶은 PID를 입력하세요 :"
    read pid
    kill -9 $pid
```

```
$sh kill_p.sh
```

```
※ 문제97. total.sh 스크립트에 7번으로 kill_p.sh를 수행하는 명령어를 추가시키시오
$vi total.sh
    7번 항목 추가
$t
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명렁어
5. 디스크 관리 명령어
6. 프로세서 관리 명령어

7. shell script 작성법

8. 유저 생성 및 관리
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ Shell script 작성법

shell

운영체제에서 제공하는 명령을 실행하는 프로그램

Linux 시스템에서 수행하는 모든 프로그램들의 자동화를 구현할 수 있는 기초되는 내용

```
            수작업 ------------------> 자동화 ----------------------> 인공지능
예) 주식  손으로 매매          컴퓨터가 자동으로 트레이딩      손해/이익을 스스로 수학식에 의해 판단(퍼셉트론)
```

◆ Shell 스크립트

인터프리터(통역사) 역할을 하는 것으로 시스템에서 지원하는 명령어들의 집합을 묶어서 프로그램화 한 것

□ Shell 의 종류

1. Bourne shell
2. C shell
3. Korn shell
4. Bash shell

shell 스크립트 작성 시 맨 위에 지금부터 작성하는 shell 스크립트 문법은 bash shell이다 라고 명시하고 프로그래밍

```
#!/bin/bash --> shell 중에서 bash shell을 쓰겠다고 명시
```

□ shell 스크립트 프로그래밍

1. c언어와 유사한 프로그래밍
2. 변수, 반복문(loop문), 제어문(if 문)이 사용 가능
3. 별도의 컴파일을 하지 않고 텍스트 파일 형태로 바로 실행이 가능
4. vi로 작성이 가능
5. Linux의 많은 부분이 shell 스크립트로 작성이 되어있다.

■ shell 스크립트를 작성하고 실행하는 방법

```
#!/bin/bash
$vi a.sh
    echo "yahooooooo"
$sh a.sh
```

```
※ 문제98. a.sh에 실행할 수 있는 권한을 넣으시오
$chmod u+x a.sh
$ll a.sh
$./a.sh
```

※ 설명: ./파일명.sh로 실행하려면 실행권한이 있어야 한다.
sh 파일명.sh는 파일의 실행권한이 없어도 실행이 된다.

■ 변수 사용법

1. 모든 변수는 '문자열(string)으로 취급
2. 변수 이름은 대소문자를 구분
3. 변수에 값을 대입할 때는 '=' 좌우에 공백이 없어야 한다.
4. 변수에 들어간 문자를 출력하려면 변수 앞에 $를 붙이고 echo 명령어로 출력하면 된다.

```
$myval='Hi--------------'
$echo $myval
```

■ 변수의 숫자 계산하는 방법

1. 변수에 대입한 값은 모두 문자열로 취급
2. 변수에 들어있는 값을 숫자로 해서 사칙연산(+ - * /)를 하려면 expr를 사용해야한다.
3. 수식에 괄호 또는 곱하기(*)를 사용하려면 그 앞에 반드시 \ 가 있어야 한다.

```
$n1=100
$n2=200
$echo $n1
$echo $n2
$echo $n1 + $n2 # 100 + 200 출력
$expr $n1 + $n2 # 300 출력
```

```
※ 문제99. n1과 n2의 곱을 구하시오
$expr $n1 \* $n2
```

```
※ 문제100. 아래의 계산식을 구현하시오
($num2 + 200 ) * $num1
$expr $(($n2 + 200)) \* $n1
```

■ 파라미터 변수

1. 파라미터 변수는 $0, $1, $2 ... 의 형태를 가진다

2. 전체 파라미터는 $*로 표현된다

```
$vi b.sh
    echo "실행파일의 이름은 $0 입니다."
    echo "첫번째 파라미터 값은 $1 입니다."
    echo "두번째 파라미터 값은 $2 입니다."
    echo "전체 파라미터 값은 $* 입니다."
$sh b.sh
```

```
※ 문제101. 아래와 같이 두 정수를 파라미터 값으로 입력했을 때 두 정수를 더한 값이 출력되게 하시오
$sh add.sh 24 18
24와 18을 더하면 42 입니다.

$vi add.sh
    echo "$1와 $2를 더하면" $(expr $1 + $2)  "입니다."
$sh add.sh 24 18
```

```
$vi add2.sh
    n1=$1
    n2=$2
    n3=`expr $n1 + $n2`
    echo "$n1와 $n2를 더하면 $n3 입니다."
$sh add2.sh
```

```
※ 문제102. 아래의 두 수를 곱한 결과가 출력되게 하시오!
$sh two_number.sh 12 54
12와 54를 곱하면 648입니다.

$vi two_number.sh
    echo "$1와 $2를 곱하면" $(expr $1 \* $2) "입니다."
$sh two_number.sh 12 54
```

```
$vi two_number.sh
    n1=$1
    n2=$2
    n3=`expr $n1 \* $n2`
    echo "$n1와 $n2를 곱하면 $n3 입니다."
$sh two_number.sh 12 54
```

■ Linux shell 에서 if문 사용방법

```
if[ 조건 ]; then
    실행문
elif [ 조건 ]; then
    실행문
else
    실행문
fi # if문 종료
```

■ if 조건문에 들어가는 비교 연산자

* 문자열 비교

1. "문자열1"="문자열1" : 두 문자열이 같으면 true

2. "문자열1"!="문자열2" : 두 문자열이 같지 않으면 true

*** 숫자열 비교**

숫자1 -eq 숫자2	두 숫자가 같으면 true
숫자1 -ne 숫자2	두 숫자가 같지 않으면 true
숫자1 -gt 숫자2	숫자1이 숫자2보다 크면 true
숫자1 -ge 숫자2	숫자1이 숫자2보다 크거나 같으면 true
숫자1 -lt 숫자2	숫자 1이 숫자2보다 작으면 true
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면 true
!숫자1	숫자1이 거짓이라면 true

```
$vi if1.sh
    if[ 100 -eq 200 ]; then
        echo "100과 200은 같다"
    else
        echo "100과 200은 다르다"
    fi
$sh if1.sh
```

```
※ 문제103. 위의 스크립트에 파라미터 변수를 이용해서 아래와 같이 실행될 수 있게 하시오
$sh if1.sh 100 200

$vi if1.sh
    if [ $1 -eq $2 ];then
            echo "$1과 $2은 같다"
    else
            echo "$1과 $2은 다르다"
    fi
$sh if1.sh 100 200
```

```
※ 문제104. 아래와 같이 실행되게 하는 shell 스크립트를 작성하시오
$sh if2.sh 100 200
100은 200보다 작습니다.

$sh if2.sh 200 100
200은 100보다 큽니다.
$vi if2.sh
    if [ $1 -le $2 ]; then
        echo "$1은 $2보다 작습니다"
    else
        echo "$1은 $2보다 큽니다"
    fi
$sh if2.sh
```

```
※ 문제105. 위의 스크립트를 수정해서 같은 숫자를 넣으면 100과 100은 같은 숫자입니다 라고 나오게 하시오
$sh if2.sh 100 100

$vi if2.sh
    ~
    elif [ $1 -eq $2 ]; then
        echo "$1과 $2는 같습니다"
    ~
$sh if2.sh 100 100
```

■ Linux에서 사용하는 연산자

1. 산술 연산자 : +, -, *, /

2. 비교 연산자 : >, <, >=, <=, ==, !=

3. 논리 연산자 : &&, ||, !

```
if [ $sal -lt 2000 ] && [ &job=="SALESMAN" ]; then
    또는 아래와 같이 수행
if [ $sal -lt 2000 -a &job=="SALESMAN" ]; then
```

```
※ 문제106. 아래와 같이 shell 스크립트를 실행하면 아래와 같은 결과가 출력되게 하시오
$sh find_sal.sh
scott의 월급은 300입니다.

$vi find_sal.sh
    echo -n "Enter the ename : "
    read ename
    awk -v a=$ename '$2==toupper(a) {print $2,"의 월급은"$6,"입니다"}' emp.txt
$sh find_sal.sh
```

```
※ 문제107. (점심시간 문제) 아래와 같이 실행했을 때 해당 사원의 월급이 2000을 기준으로 월급 인상여부를 출력하는 shell을 구현하시오
           (2000 이상이면 월급 인상 안함)
$sh find_sal.sh scott
월급 인상 대상자가 아닙니다.
$sh find_sal.sh martin
월급 인상 대상자 입니다.

$vi find_sal.sh
    echo "Enter the ename : $1 "
    sal=`grep -i $1 emp.txt | awk '{print $6}'`
    if [ $sal -gt 2000 ]; then
        echo "월급 인상 대상자가 아닙니다"
    else
        echo "월급 인상 대상자 입니다"
    fi
$sh find_sal.sh SCOTT
$sh find_sal.sh MARTIN
```

```
※ 문제108. 이름을 입력하고 find_sal.sh을 실행했을 때 부서번호가 30번이고 월급이 2000보다 작다면 "월급 인상 대상자 입니다"를 출력하고
          그렇지 않다면 "월급 인상 대상자가 아닙니다"를 출력하시오
$vi find_sal.sh
    echo "Enter the ename : $1 "
    sal=`grep -i $1 emp.txt | awk '{print $6}'`
    deptno=`grep -i $1 emp.txt | awk '{print $8}'`
    if [ $sal -lt 2000 ] && [ $deptno==30 ]; then
        echo "월급 인상 대상자 입니다"
    else
        echo "월급 인상 대상자가 아닙니다"
    fi
$sh find_sal.sh ALLEN
```

■ loop 문

1. for loop

2. while

```
for 변수 in 값1, 값2, 값3
do
    반복할 문장
done
```

```
$vi for1.sh
    for  i in {1..10}
    do
        echo $i
    done
$sh for1.sh
```

```
※ 문제109. 구구단 2단을 출력하시오
$vi for2.sh
    for i in {1..9}
    do
        let gop=2*$i
        echo "2 x $i = $gop"
    done
$sh for2.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

※ 설명 : let을 사용해서 expr과 \ 를 사용하지 않고 바로 곱할 수 있다.

```
※ 문제110. 구구단 전체를 출력하시오
$vi for2.sh
    for i in {2..9}
    do
        for j in {1..9}
        do
                let gop=$i*$j
                echo "$i x $j = $gop"
        done
    done
$sh for2.sh
```

```
※ 문제111. 아래의 별표를 출력하시오
★
★★
★★★
★★★★
★★★★★
$vi star.sh
    star=""
    for i in {1..5}
    do
        star"$star★"
        echo $star
    done
```

```
$sh star.sh
```

```
※ 문제112. 위의 코드를 수정해서 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오
$sh star.sh
숫자를 입력하세요: 5
★
★★
★★★
★★★★
★★★★★

$vi star.sh
    echo -n "숫자를 입력하세요 : "
    read nb
    for i in `eval echo {1..$nb}`
    do
        star"$star★"
        echo $star
    done
$sh star.sh
```

```
$eval echo {1..10}
eval 명령어는 문자들을 명령문으로 인식하고 실행
```

```
※ 문제113. 위의 코드를 응용해서 아래와 같이 실행되게 하시오
$sh star.sh
    숫자를 입력하세요 : 5
★★★★★
★★★★
★★★
★★
★

$vi star.sh
    echo -n "숫자를 입력하세요>> "
    read num
    for ((i=$num; i>0; i--))
    do
        for ((j=1; j<i+1; j++))
        do
                echo -n "★"
        done
        echo ""
    done
$sh star.sh
```

```
※ 문제114. emp.txt를 복사해서 emp1.txt~emp100.txt로 복사하는 shell 스크립트를 작성하시오
$vi cp_emp.txt
    i=1
    while [ $i -le 100 ]
    do
        cp emp.txt emp$i.txt
        let i=$(($i+1))
    done
$sh cp_emp.txt
```

```
※ 문제115. 짝꿍과 자리를 바꿔서 emp1.txt ~ emp100.txt중 하나를 랜덤으로 골라서 데이터를 변경하고 다시 자리로
돌아오시오
$vi emp34.txt
```

```
※ 문제116. 짝꿍이 어떠한 파일을 변경했는지 shell 스크립트로 한 번에 알아내시오
$vi diff_emp.sh
    for i in {1..100}
    do
```

```
        diff --brief emp.txt emp$i.txt
    done
$sh diff_emp.sh #emp34.txt
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명렁어
5. 디스크 관리 명령어
6. 프로세서 관리 명령어

7. shell script 작성법

8. 유저 생성 및 관리
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ while loop 문

문법 : 조건에 만족할 때만 loop문을 반복수행한다

```
while [ 조건 ]
do
    실행문
done
```

※ 한 칸씩 띄어쓰기를 해야한다.

```
$vi while1.sh
    i=1
    while [ $i -le 10 ]
    do
        echo $i
        i=`expr $i + 1`
    done
$sh while1.sh
```

```
※ 문제117. 구구단 2단을 while loop문으로 출력하시오
$vi  while1.sh
    i=1
    while [ $i -le 9 ]
    do
        let rs=2*$i
        echo "2 x $i = $rs"
        i=`expr $i + 1`
    done
$sh while1.sh
```

■ while loop문의 중첩 loop문

```
while [ 조건 ]
do
    while [ 조건 ]
    do
        실행문
        안쪽 loop문의 카운트 증가 코드
    done
    바깥쪽 loop문의 카운트 증가 코드
done
```

```
※ 문제118. while loop문으로 구구단 전체를 출력하시오
#!/bin/bash
dan=2
while [ $dan -le 9 ]
do
  mul=1
  while [ $mul -le 9 ]
  do
    let ret=$dan*$mul
    echo "$dan * $mul = $ret"
    mul=`expr $mul + 1`
  done
  dan=`expr $dan + 1`
done
```

```
※ 문제119. 아래의 power함수를 while loop문으로 구현하시오
SQL> select power(2,3) from dual;
8
$sh power.sh 2 3
8

$vi power.sh
    #!/bin/bash
    cnt=1
    rs=1
    while [ $cnt -le $2 ]
    do
      rs=`expr $rs \* $1`
      cnt=`expr $cnt + 1`
    done
    echo $rs
$sh power.sh
```

```
※ 문제120. while loop문으로 factorial을 구현하시오
$sh factorial.sh 5
120

$vi factorial.sh
 cnt=1
 rs=1
 while [ $cnt -le $1 ]
 do
        rs=`expr $rs \* $cnt`
        cnt=`expr $cnt + 1`
 done
 echo $rs
$sh factorial.sh
```

■ Linux shell을 현업에서 어떻게 활용하는지 예

라이나 생명에서 데이터 분석을 할 때

1. 고객 데이터를 Linux shell에서 데이터를 필터링(분리)

   ex) 라이나 생명 고객 중에 40대만 따로 분리해서 text파일 생성

2. 데이터를 가지고 군집분석(k-means) (R 또는 파이썬)

3. 데이터를 가지고 연관분석(아프리오리 알고리즘)

새로운 보험 상품이 출시 되었는데 보험 가입을 유도해야 하는데 모든 보험 가입자들에게 전화를 일일이 돌려서 가입을 유도하는 것 보다

가능성 있는 고객들에게만 전화를 하는게 더 효과적

ex) 한 사람의 보험 매니저가 관리하는 고객이 300명이라 하면 그 중 가능성 있는 50명을 추려서 가입 유도를 하도록 데이터 분석가들이 추출

■ 파일을 다루는 shell 스크립트 명령어

-d 파일명	파일이 디렉토리면 true
-e 파일명	파일이 존재하면 true
-f 파일명	파일이 일반 파일이면 true
-r 파일명	파일이 읽기 가능하면 true
-s 파일명	파일의 크기가 0이 아니면 true
-w 파일명	파일이 쓰기 가능하면 true
-x 파일명	파일이 실행 가능하면 true

```
$vi f1.sh
    fname=/home/oracle/emp.txt
    if [ -e %fname ]; then
        cat $fname
    else
        echo "파일이 없습니다."
    fi
$sh f1.sh
```

※ 문제**121.(**점심시간 문제**)** 위의 스크립트를 수정해서 파일명을 물어보게 하고 파일이 존재하면 파일의 내용이 출력되게 하고
파일이 없으면 파일이 없습니다 가 출력되게 하시오

```
$vi f1.sh
    echo -n "파일명을 입력하세요: "
    read file
    fname=/home/oracle/$file
    if [ -e $fname ]; then
        cat $fname
    else
        echo "파일이 없습니다."
    fi
$sh f1.sh
emp.txt
파일명을 입력하세요: emp.txt
7839     KING     PRESIDENT     0     1981-11-17     5000     0     10
7698     BLAKE     MANAGER     7839     1981-05-01     2850     0     30
7782     CLARK     MANAGER     7839     1981-05-09     2450     0     10
7566     JONES     MANAGER     7839     1981-04-01     2975     0     20
7654     MARTIN     SALESMAN     7698     1981-09-10     1250     1400     30
7499     ALLEN     SALESMAN     7698     1981-02-11     1600     300     30
7844     TURNER     SALESMAN     7698     1981-08-21     1500     0     30
7900     JAMES     CLERK     7698     1981-12-11     950     0     30
7521     WARD     SALESMAN     7698     1981-02-23     1250     500     30
7902     FORD     ANALYST     7566     1981-12-11     3000     0     20
7369     SMITH     CLERK     7902     1980-12-09     800     0     20
7788     SCOTT     ANALYST     7566     1982-12-22     3000     0     20
7876     ADAMS     CLERK     7788     1983-01-15     1100     0     20
7934     MILLER     CLERK     7782     1982-01-11     1300     0     10

$sh f1.txt
파일명을 입력하세요: adrian.txt
파일이 없습니다.
```

※ 문제**122. emp.txt**에서 직업이 **salesman**인 사원들의 모든 데이터를 출력하시오

```
$grep -i 'salesman' emp.txt
```

※ 문제**123.** 문제**122**번에서 출력되고 있는 화면을 **salesman.txt**로 저장하시오

```
$grep -i 'salesman' emp.txt >> salesman.txt
```

※ 문제**124.** 위의 스크립트를 이용해서 아래와 같이 직업을 물어보게 하고 직업을 입력하면 해당 직업의 **text** 파일이 생성되게 하시오

```
$sh make_job.sh
    직업을 입력하세요 : analyst
analyst.txt 생성

$vi make_job.sh
    echo -n "직업을 입력하세요 : "
    read job
    grep -i $job emp.txt >> $job.txt
$sh make_job.sh
analyst
$cat analyst.txt
7902     FORD     ANALYST     7566     1981-12-11     3000     0     20
7788     SCOTT     ANALYST     7566     1982-12-22     3000     0     20
```

※ 문제**125.** 문제**124**번 코드를 수정해서 이미 생성된 직업**.txt**가 존재한다면 해당 파일이 이미 있습니다**.** 라는 메세지가 출력되게 하고
없으면 직업**.txt**가 생성되게 하시오

```
$vi make_job.sh
    echo -n "직업을 입력하세요: "
    read job
    if [ -e $job.txt ]; then
        echo "해당 파일은 이미 있습니다"
    else
        grep -i $job emp.txt >> $job.txt
    fi
$sh make_job.sh
직업을 입력하세요: analyst
해당 파일은 이미 있습니다
$sh make_job.sh
```

※ 문제**126.** 직업을 출력하는데 중복을 제거해서 출력하시오

```
$awk '{print $3}' emp.txt | sort -u
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

※ 문제**127.** 위에서 출력되는 직업을 변수에 담고 **for**문을 이용해서 아래와 같이 확장자가 붙어서 출력되게 하시오

```
$sh auto_job.sh

$vi auto_job.sh
    job=`awk '{print $3}' emp.txt | sort -u`
    for i in $job
    do
        echo "$i.txt"
    done
$sh auto_job.sh
ANALYST.txt
CLERK.txt
MANAGER.txt
PRESIDENT.txt
SALESMAN.txt
```

※ 문제**128.** 위의 문제처럼 직업을 일일이 물어보지 않아도 그냥 알아서 모든 직업에 대한 파일이 아래와 같이 자동으로 생성되게 하는 **shell** 스크립트를 작성하시오

```
$vi make_job.sh
    job=`awk '{print $3}' emp.txt | sort -u`
    for i in $job
    do
        if [ -e $i.txt ]; then
                echo "해당 파일은 이미 있습니다"
        else
```

```
                grep -i $i emp.txt >> $i.txt
        fi
    done
$sh make_job.sh

$ cat ANALYST.txt
7902     FORD     ANALYST     7566     1981-12-11     3000     0     20
7788     SCOTT     ANALYST     7566     1982-12-22     3000     0     20
$ cat MANAGER.txt
7698     BLAKE     MANAGER     7839     1981-05-01     2850     0     30
7782     CLARK     MANAGER     7839     1981-05-09     2450     0     10
7566     JONES     MANAGER     7839     1981-04-01     2975     0     20
$ cat CLERK.txt
7900     JAMES     CLERK     7698     1981-12-11     950     0     30
7369     SMITH     CLERK     7902     1980-12-09     800     0     20
7876     ADAMS     CLERK     7788     1983-01-15     1100     0     20
7934     MILLER     CLERK     7782     1982-01-11     1300     0     10
$ cat SALESMAN.txt
7654     MARTIN     SALESMAN     7698     1981-09-10     1250     1400     30
7499     ALLEN     SALESMAN     7698     1981-02-11     1600     300     30
7844     TURNER     SALESMAN     7698     1981-08-21     1500     0     30
7521     WARD     SALESMAN     7698     1981-02-23     1250     500     30
$ cat PRESIDENT.txt
7839     KING     PRESIDENT     0     1981-11-17     5000     0     10
```

```
※ 문제129. 위의 스크립트를 이용해서 부서번호별로 .txt 파일이 아래와 같이 만들어지게 하시오
$sh make_deptno.sh

$vi make_deptno.sh
    deptno=`awk '{print $8}' emp.txt | sort -u`
    for i in $deptno
    do
        if [ -e $i.txt ]; then
                echo "해당 파일이 이미 존재합니다."
        else
                awk -v a=$i '$8==a' emp.txt >> $i.txt
        fi
    done
$sh make_deptno.sh
$cat 10.txt
7839     KING     PRESIDENT     0     1981-11-17     5000     0     10
7782     CLARK     MANAGER     7839     1981-05-09     2450     0     10
7934     MILLER     CLERK     7782     1982-01-11     1300     0     10
$cat 20.txt
7566     JONES     MANAGER     7839     1981-04-01     2975     0     20
7902     FORD     ANALYST     7566     1981-12-11     3000     0     20
7369     SMITH     CLERK     7902     1980-12-09     800     0     20
7788     SCOTT     ANALYST     7566     1982-12-22     3000     0     20
7876     ADAMS     CLERK     7788     1983-01-15     1100     0     20
$cat 30.txt
7698     BLAKE     MANAGER     7839     1981-05-01     2850     0     30
7654     MARTIN     SALESMAN     7698     1981-09-10     1250     1400     30
7499     ALLEN     SALESMAN     7698     1981-02-11     1600     300     30
7844     TURNER     SALESMAN     7698     1981-08-21     1500     0     30
7900     JAMES     CLERK     7698     1981-12-11     950     0     30
7521     WARD     SALESMAN     7698     1981-02-23     1250     500     30
```

■ DB에서 SQL로 데이터를 추출하고 추출된 데이터를 text파일로 생성하는 shell 스크립트 작성

1. SQLPLUS에서 edit를 했을 때 vi 편집기로 연결시키는 방법

```
$cd $ORACLE_HOME/sqlplus/admin
$vi glogin.sql
    G
    i
    define _editor='vi'
    esc
```

```
    ZZ
$cd
$sqlplus scott/tiger
SQL> select * from emp;
SQL> ed
vi편집기로 자동 이동
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명렁어
5. 디스크 관리 명령어
6. 프로세서 관리 명령어

7. shell script 작성법

8. 유저 생성 및 관리
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

```
SQL>ed p.sql
set trimspool on
spool dept_sum.txt
set echo off                -- text file에 SQL>이 안나오게 하는 명령어
set feedback off            -- 몇 건이 출력되었습니다 메시지가 안나오게 하는 명령어
set heading off             -- Column명 안나오게 하는 명령어
select deptno, sum(sal)
        from emp
        group by deptno
        order by deptno
/                           -- / 실행하라는 명령어
spool off                   -- SQLPLUS의 화면에 출력된 내용을 여기까지 text file에 담아라
exit
SQL>@p.sql
```

```
※ 문제130. 직업과 직업별 토탈월급의 결과를 job_sum.txt로 만드시오
$@p2.sql
set trimspool on
spool job_sum.txt
set echo off
set feedback off
set heading off
select job, sum(sal)
        from emp
        group by job
        order by 2
/
spool off
exit
SQL>@p2.sql
```

```
※ 문제131. scott으로 접속하면서 바로 p2.sql이 수행될 수 있도록 하시오
$vi total.sh
########################################
echo "---------------------------------"
echo " 1. 두 파일의 차이 확인
2. 특정 파일을 검색
3. 특정 단어를 검색
4. SQL 접속(SCOTT으로 접속)
5. 사원의 월급 조회
6. 현재 디스크 상태 모니터링
7. 프로세서 kill
8. 직업별 토탈월급에 대한 통계 데이터 추출
    "
echo -n "번호를 입력하세요: "
read choice
case $choice in
        1)
```

```
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
        3)
                sh /home/oracle/find_word.sh ;;
        4)
                sqlplus scott/tiger ;;
        5)
                python find_ename.py ;;
        6)
                sar 1 100 ;;
        7)
                sh /home/oracle/kill_p.sh ;;
        8)
                sqlplus scott/tiger @p2.sql ;;
esac
$t
---------------------------------
1. 두 파일의 차이 확인
2. 특정 파일을 검색
3. 특정 단어를 검색
4. SQL 접속(SCOTT으로 접속)
5. 사원의 월급 조회
6. 현재 디스크 상태 모니터링
7. 프로세서 kill
8. 직업별 토탈월급에 대한 통계 데이터 추출

번호를 입력하세요: 8

SQL*Plus: Release 11.2.0.1.0 Production on Wed Jun 10 10:13:42 2020
Copyright (c) 1982, 2009, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

CLERK           4150
PRESIDENT       5000
SALESMAN        5600
ANALYST         6000
MANAGER         8275
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

■ 리눅스 수업 목차

■ 유저 생성 및 관리

* 유저 생성, 관리 명령어

useradd	유저 생성
usermod	유저 수정
userdel	유저 삭제
passwd	패스워드 변경
groupadd	그룹 추가
groupmod	그룹 수정
groupdel	그룹 삭제
su	유저 변경

■ useradd 명령어

유저를 생성하는 명령어

유저를 생성하려면 유저를 생성할 수 있는 권한이 있는 유저로 접속

```
$su -
암호 :

#useradd 유저명

#useradd oracle2
#cat /etc/passwd
유저들의 목록을 확인
```

oracle2	**x**	**502**	**510**	**/home/oracle2**	**/bin/bash**
유저명	패스워드	유저 id	그룹id	홈디렉토리	로그인 shell

```
※ 문제133. oracle3라는 유저를 생성하고 잘 생성되었는지 확인하시오
#useradd oracle3
#tail -n 3 /etc/passwd
grid:x:501:504::/home/grid:/bin/bash
oracle2:x:502:510::/home/oracle2:/bin/bash
oracle3:x:503:511::/home/oracle3:/bin/bash
```

■ passwd 명령어

사용자의 비밀번호를 지정하거나 변경하는 명령어

```
#passwd 유저명

#passwd oracle2
#exit
$su oracle2
```

```
※ 문제134. oracle2 유저에서 자기 자신의 패스워드를 변경해보시오
$passwd
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

■ 유저로 접속할 때 자동으로 실행되는 환경파일인 .bash_profile을 이용해서 데이터 삭제의 실수를 예방할 수 있게 하시오

```
$vi .bash_profile
```

■ userdel

유저를 삭제하는 명령어

```
#userdel oracle2
#cat /etc/passwd
oracle:x:500:504::/home/oracle:/bin/bash
grid:x:501:504::/home/grid:/bin/bash
oracle3:x:503:511::/home/oracle3:/bin/bash
```

```
※ 문제135. oracle3 유저를 삭제하시오
# userdel oracle3
# tail -4 /etc/passwd
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
vboxadd:x:101:1::/var/run/vboxadd:/bin/false
oracle:x:500:504::/home/oracle:/bin/bash
grid:x:501:504::/home/grid:/bin/bash
```

```
※ 문제136. /home/oracle2 디렉토리를 삭제하시오
#rm -rf /home/oracle2
#ll /home
drwx------  3 grid   oinstall  4096  9월 24  2012 grid
drwx------ 39 oracle oinstall 12288  6월 10 10:51 oracle
drwx------  3    503      511  4096  6월 10 10:24 oracle3
```

■ groupadd 명령어

그룹을 생성하는 명령어

```
#groupadd -g 그룹번호 그룹명

#groupadd -g 516 proj1
#tail -5 /etc/group
oper:x:506:
asmdba:x:507:grid
asmoper:x:508:grid
asmadmin:x:509:grid
proj1:x:516:
```

■ gpasswd 명령어

특정 그룹에 유저를 추가/제거하는 명령어

```
#gpasswd 옵션 유저명 그룹명
          -a : 특정 그룹에 새로운 멤버 추가
          -d : 특정 그룹에 기존 멤버 삭제

#useradd oracle9
#passwd oracle9
#gpasswd -a oracle9 oinstall
※ oracle9 유저를 oinstall 그룹에 포함
#tail /etc/group
oinstall:x:504:oracle9
dba:x:505:oracle
oper:x:506:
asmdba:x:507:grid
asmoper:x:508:grid
asmadmin:x:509:grid
proj1:x:516:
oracle9:x:517:
```

* oracle9 유저를 oinstall 그룹에서 제거

```
#gpasswd -d oracle9 oinstall
사용자 oracle9을(를) 그룹 oinstall로부터 삭제 중
#tail /etc/group
oinstall:x:504:
dba:x:505:oracle
oper:x:506:
asmdba:x:507:grid
asmoper:x:508:grid
asmadmin:x:509:grid
proj1:x:516:
oracle9:x:517:
```

■ usermod 명령어

유저 계정의 정보를 변경하는 명령어

```
#usermod 옵션 유저명
          -d : 사용자의 홈 디렉토리를 지정하는 옵션
```

※ 문제**137. oracle7** 유저를 생성하고 **oracle7** 유저의 홈디렉토리가 생성되었는지 확인하시오
```
#useradd oracle7
#ll /home
drwx------  3 grid    oinstall  4096  9월 24  2012 grid
drwx------ 39 oracle  oinstall 12288  6월 10 10:51 oracle
drwx------  3 oracle7      511  4096  6월 10 10:24 oracle3
drwx------  3 oracle7 oracle7   4096  6월 10 11:15 oracle7
drwx------  3 oracle9 oracle9   4096  6월 10 11:07 oracle9
```

※ 문제**138. oracle7**의 홈 디렉토리를 **/home/oracle777** 로 변경하시오
```
#mkdir -p /home/oracle777
#chown -R oracle7:oinstall /home/oracle777
/home/oracle777의 소유자를 oracle7로 변경

oracle7:x:503:518::/home/oracle7:/bin/bash

*usermod 명령어로 oracle7의 홈 디렉토리를 /home/oracle777로 변경
#usermod -d /home/oracle777 oracle7
#tail /etc/passwd
oracle7:x:503:518::/home/oracle777:/bin/bash
```

※설명 : 홈 디렉토리로 변경할 디렉토리의 소유자를 먼저 변경하고 usermod로 홈 디렉토리를 변경하는 순서로 진행

■ groupmod 명령어

그룹을 변경하는 명령어(group이름, 번호를 변경)

```
#groupmod -n  proj2   proj1
            변경 후  변경 전

#tail /etc/group
proj1:x:516:
#groupmod -n  proj2   proj1
#tail /etc/group
proj2:x:516:
```

■ groupdel 명령어

그룹을 삭제하는 명령어

```
#groupdel proj2
#tail /etc/group
oracle:x:503:
oinstall:x:504:
dba:x:505:oracle
oper:x:506:
asmdba:x:507:grid
asmoper:x:508:grid
asmadmin:x:509:grid
oracle9:x:517:
oracle7:x:518:
```

■ 리눅스 수업 목차

1. 리눅스란 무엇인가?
2. 리눅스 기본 명령어
3. vi편집기 명령어
4. 권한 관리 명렁어
5. 디스크 관리 명령어
6. 프로세서 관리 명령어
7. shell script 작성법
8. 유저 생성 및 관리
9. ubuntu linux 설치
10. Linux에서 파이썬과 R 사용하기

■ Linux에서 R 설치하기

```
1. 리눅스용 R 을 Download  받는다.

wget  https://cran.r-project.org/src/base/R-3/R-3.2.3.tar.gz

2.  Download 받은 파일의 압축을 푼다.

tar  xvf  R-3.2.3.tar.gz

3.  R-3.2.3 디렉토리로 이동해서 그 안에  configure 파일을 아래와 같이 실행한다.

$ cd R-3.2.3
$ ./configure

4. 그 다음 아래의 명령어를 차례대로 수행한다.

   $ make
   $ make  check
   $ make  info
   $ make  pdf
   $ make  install
   $ make  install-info

5. R 로 접속한다.

  $ cd bin
  $ ./R
R>
```

■ Linux의 어느 디렉토리에서든 R에 접속할 수 있도록 설정하시오

```
$vi .bash_profile
export R_HOME=/home/oracle/R-3.2.3
export PATH=$R_HOME/bin:$PATH
```

```
※ 문제139. Linux의 R에 편하게 접속할 수 있도록 total.sh의 9번 항목을 추가하시오
$vi total.sh
echo "---------------------------------"
echo " 1. 두 파일의 차이 확인
2. 특정 파일을 검색
3. 특정 단어를 검색
4. SQL 접속(SCOTT으로 접속)
5. 사원의 월급 조회
```

```
6. 현재 디스크 상태 모니터링
7. 프로세서 kill
8. 직업별 토탈월급에 대한 통계 데이터 추출
9. R 실행
    "
echo -n "번호를 입력하세요: "
read choice
case $choice in
        1)
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
        3)
                sh /home/oracle/find_word.sh ;;
        4)
                sqlplus scott/tiger ;;
        5)
                python find_ename.py ;;
        6)
                sar 1 100 ;;
        7)
                sh /home/oracle/kill_p.sh ;;
        8)
                sqlplus scott/tiger @p2.sql ;;
        9)
                R
esac
```

※ 문제**140. total.sh**에 테이블을 **drop**했을 때 복구할 수 있는 명령어를 **10**번에 추가하시오

```
$vi flash_table.sh
 echo -n "복구할 테이블명을 입력하세요: "
 read tablename
 sqlplus scott/tiger << EOF
 flashback table $tablename to before drop;
 exit;
 EOF

$vi total.sh
echo "----------------------------------"
echo " 1. 두 파일의 차이 확인
2. 특정 파일을 검색
3. 특정 단어를 검색
4. SQL 접속(SCOTT으로 접속)
5. 사원의 월급 조회
6. 현재 디스크 상태 모니터링
7. 프로세서 kill
8. 직업별 토탈월급에 대한 통계 데이터 추출
9. R 실행
10. Table 복구
    "
echo -n "번호를 입력하세요: "
read choice
case $choice in
        1)
                sh /home/oracle/diff.sh ;;
        2)
                sh /home/oracle/find_file.sh ;;
        3)
                sh /home/oracle/find_word.sh ;;
        4)
                sqlplus scott/tiger ;;
        5)
                python find_ename.py ;;
        6)
                sar 1 100 ;;
        7)
                sh /home/oracle/kill_p.sh ;;
        8)
                sqlplus scott/tiger @p2.sql ;;
        9)
                R ;;
```

```
        10)
                sh flash_table.sh ;;
esac

$sqlplus scott/tiger
SQL>drop table emp;
SQL>exit
$sh total.sh
---------------------------------
1. 두 파일의 차이 확인
2. 특정 파일을 검색
3. 특정 단어를 검색
4. SQL 접속(SCOTT으로 접속)
5. 사원의 월급 조회
6. 현재 디스크 상태 모니터링
7. 프로세서 kill
8. 직업별 토탈월급에 대한 통계 데이터 추출
9. R 실행
10. Table 복구

번호를 입력하세요: 10
복구할 테이블명을 입력하세요: emp

SQL*Plus: Release 11.2.0.1.0 Production on Wed Jun 10 14:15:43 2020

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SQL>
Flashback complete.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

※ 문제**150. /home/oracle** 밑에 있는 쉘 스크립트들을 압축해서 윈도우로 내리시오

```
$tar cvf shell_all.zip *.sh
```