

리눅스 목차

2018년 7월 4일 수요일 오후 1:51

- VM 웨어 설치 및 리눅스 시스템 가동 방법

- 리눅스

- 1. 리눅스가 무엇인가?

- 2. 리눅스 기본 명령어

- 3. vi 편집기 명령어

- 4. 권한 관리

- 5. 디스크 관리

- 1. df 명령어 : 현재 파일 시스템의 총 사용율을 확인하는 명령어

- 2. du 명령어 : 현재 파일/디렉토리의 디스크 사용량을 표시하는 명령어

- 3. sar 명령어 : disk 의 i/o 성능을 모니터링 하는 명령어

- 6. 프로세서 관리

- 1. jobs 명령어

- 2. top 명령어

- 3. ps 명령어

- 4. kill 명령어

- 7. 쉘 스크립트 작성법

- 8. 리눅스 환경에서 R 사용하기

- 9. 리눅스 설치

VM 웨어 설치 및 리눅스 시스템 가동 방법

2018년 6월 21일 목요일 오후 1:55

1. oracle virtual Box 설치
2. oracle 리눅스 환경(ws2.ora)를 불러온다.

오라클 리눅스 로그인 화면

유저 이름 :oracle

패스워드 : oracle

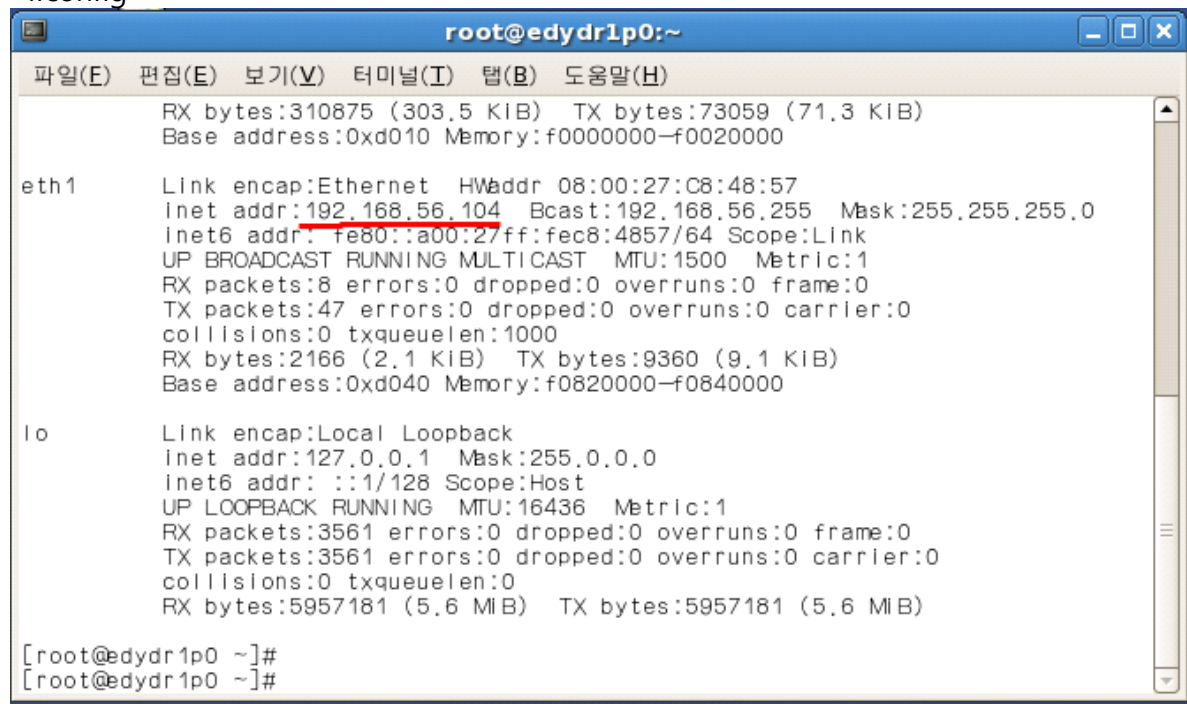
3. ip 주소를 확인하기 위해 root유저로 접속한다.

\$ su -

패스워드 : oracle

4. ip 주소를 확인하는 명령어를 수행한다.

#ifconfig



```
root@edydr1p0:~  
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
RX bytes:310875 (303.5 KIB) TX bytes:73059 (71.3 KIB)  
Base address:0xd010 Memory:f0000000-f0020000  
  
eth1    Link encap:Ethernet HWaddr 08:00:27:C8:48:57  
        inet addr:192.168.56.104 Bcast:192.168.56.255 Mask:255.255.255.0  
        inet6 addr: fe80::a00:27ff:fec8:4857/64 Scope:Link  
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:47 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:2166 (2.1 KIB) TX bytes:9360 (9.1 KIB)  
        Base address:0xd040 Memory:f0820000-f0840000  
  
lo       Link encap:Local Loopback  
        inet addr:127.0.0.1 Mask:255.0.0.0  
        inet6 addr: ::1/128 Scope:Host  
        UP LOOPBACK RUNNING MTU:16436 Metric:1  
        RX packets:3561 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:3561 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:0  
        RX bytes:5957181 (5.6 MiB) TX bytes:5957181 (5.6 MiB)  
  
[root@edydr1p0 ~]#  
[root@edydr1p0 ~]#
```

5. xshell로 리눅스 서버를 접속한다.

- xshell 접속
- 새로만들기
- 이름 지정후 ip번호(호스트)입력
- 이름 : oracle, 비밀번호 : oracle

리눅스

2018년 6월 21일 목요일 오후 4:13

1. 리눅스가 무엇인가?

2. 리눅스 기본 명령어

3. vi편집기 명령어

4. 권한 관리

5. 디스크 관리

6. 프로세스 관리

7. 셸 스크립트 작성법

8. 리눅스 환경에서 R 사용하기

9. 리눅스 설치

- 리눅스를 왜 배워야 하는가?
 - 현업의 대용량 데이터는 대부분 window가 아니라 유닉스 또는 리눅스 os를 사용하는 서버에 저장되어 있다.
큰 기업의 메인 서버 : 유닉스, 리눅스
쿠팡, 인터파크, g마켓 : 리눅스
- 딥러닝 개발/연구, 데이터 분석가도 리눅스를 다루는게 필수.

1. 리눅스가 무엇인가?

유닉스가 너무 고가여서 리눅스 오픈소스를 리누즈 토발즈(핀란드 학생) 1991년 11월에 개발한 os 이다.

리누즈 토발즈가 개발한 것은 리눅스의 커널(ex.자동차 엔진)을 개발하고 소스를 무료로 공개하고 전 세계의 많은 개발자들이 오픈 소스를 가져다가 더 좋게 개선해서 다시 인터넷에 올리고 하는 작업을 반복하다 보니 리눅스 os가 유닉스보다 더 가볍고 안정적이게 되었다.

이러한 프로젝트를 GNU 프로젝트라고 한다.

GNU 프로젝트 = 누구든지 배포된 오픈 소스를 가져다가 개발할 수 있고 돈을 벌 목적으로 상용화 할 수 있는데 한가지 지켜야할 약속은 이 소스를 가져다가

더 좋게 수정했으면 그 코드를 인터넷에 올려줘야함.

레드햇(상용버전, 유료)----->Cent os(무료)

1. Oracle linux
 2. ubuntu
- 가 많이 사용되는데 우리는 Oracle linux를 사용한다.

2. 리눅스 기본 명령어

1. cd 명령어 (Change Directory)

pwd(print working directory): 현재 내가 작업하고있는 디렉토리 확인

```
[orcl:~]$ pwd
/home/oracle
```

2. ls 현재 디렉토리에 있는 폴더와 파일을 확인

```
[orcl:~]$ ls
Adobe Desktop  datamodeler.log  labs  oradiag_oracle
```

ex) cd labs

```
[orcl:~]$ cd labs
[orcl:labs]$ pwd
/home/oracle/labs
[orcl:labs]$
```

문제1. 다시 /home/oracle 밑으로 이동하시오.

답 : cd ..

```
[orcl:~]$ cd ..
```

문제2. 현재 디렉토리에서 /(루트)디렉토리로 바로 이동하시오.

답 : cd /

```
[orcl:~]$ cd /
[orcl:/]$ pwd
/
```

설명 : 경로의 2가지 종류

1. 절대 경로 : cd 내가 가고자 하는 싶은 정확한 위치
ex) cd /home/oracle/labs
2. 상대 경로 : cd .. = 나의 현재 디렉토리에서 상위 디렉토리로 이동한다.
ex) cd ./labs = 현재 디렉토리의 labs 디렉토리로 이동한다.

문제3. 리눅스 서버에 R을 설치하시오.

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>
```

#####xshell에서 sqlplus scott/tiger 로 접속했는데 잠겨있을때(sql 락 푸는 방법)

```
[orcl:~]$ sqlplus / as sysdba
```

```
SQL> show user
USER is "SYS"
SQL> alter user scott
2 account unlock;
```

락을 풀고 비밀번호 변경하기

```
SQL> alter user scott
2 identified by tiger;
```

3. touch 명령어 : 파일의 용량이 0인 파일을 생성하는 명령어

ex) \$ touch a1.txt

```
$ ls -l a1.txt
```

```
[orcl:~]$ touch a1.txt
[orcl:~]$ ls -l a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:11 a1.txt
```

rwx : 권한, 1 oracle : 소유자, oinstall : 소유자가 속한 그룹,

0 : 파일크기, Jun 22 10:11 : 생성날짜

문제4. 현재 디렉토리에 아래의 파일들을 크기가 0으로 생성하시오.

a.txt b.txt c.txt d.txt e.txt f.txt

답)

touch a.txt b.txt c.txt d.txt e.txt f.txt

```
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 d.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 f.txt
```

문제5. 리눅스 서버에 /home/oracle 밑에 emp.txt를 올리시오.



눌러서 왼쪽에서 오른쪽으로 파일 옮긴다.

확인하는 방법(cat)

```
[orcl:~]$ cat emp.txt
7369 SMITH CLERK 7902 17-DEC-80 800 0 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 0 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 0 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 0 10
7788 SCOTT ANALYST 7566 19-APR-87 3000 0 20
7839 KING PRESIDENT 0 17-NOV-81 5000 0 10
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
7876 ADAMS CLERK 7788 23-MAY-87 1100 0 20
7900 JAMES CLERK 7698 03-DEC-81 950 0 30
7902 FORD ANALYST 7566 03-DEC-81 3000 0 20
7934 MILLER CLERK 7782 23-JAN-82 1300 0 10
```

```
[orcl:~]$ cat dept.txt
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON
```

3. mkdir 명령어 : 디렉토리를 생성하는 명령어

ex) \$ mkdir test2

\$ ls -dir test2

```
[orcl:~]$ mkdir test2
[orcl:~]$ ls -dir test2
17072993 test2
```

문제6. /home/oracle 밑에 아래의 하위 디렉토리를 아래와 같이 생성하시오.

보기)

home/oracle/test5/test6/test7/test8/test9

답)

mkdir -p /test5/test6/test7/test8/test9

문제7. 지금 현재 디렉토리가 /home/oracle인 상태에서 한번에
home/oracle/test5/test6/test7/test8/test9 여기로 이동하시오.

답)

```
cd home/oracle/test5/test6/test7/test8/test9
```

```
[orcl:~]$ cd home/oracle/test5/test6/test7/test8/test9
[orcl:test9]$ pwd
/home/oracle/home/oracle/test5/test6/test7/test8/test9
```

문제8. 내가 바로 전에 있었던 디렉토리로 한번에 가게 하시오.

답)

```
[orcl:test9]$ cd -
```

```
[orcl:~]$ pwd
/home/oracle
```

4. rm 명령어 : 파일이나 디렉토리를 삭제하는 명령어

%리눅스나 유닉스는 휴지통이 없기 때문에 삭제할때 특히 주의해야한다.

백업이 없으면 절대 복구가 불가능하다.

ex) \$ rm a.txt b.txt
\$ ls -l a.txt b.txt

```
[orcl:~]$ ls -l a.txt b.txt
ls: a.txt: No such file or directory
ls: b.txt: No such file or directory
```

문제9. 현재 디렉토리 밑에 있는 확장자가 txt인 파일들을 모두 삭제하시오.

답)

```
rm -rf *.txt
```

```
[orcl:~]$ ls -l *.txt
ls: *.txt: No such file or directory
```

```
rm -rf *
```

설명: -r = 현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 전부 다 삭제

-f = 삭제할때 삭제할까요? 라고 물어보지 않고 바로 삭제

5. rmdir 명령어 : 디렉토리를 삭제하는 명령어

ex) \$rmdir 디렉토리명

```
$mkdir ddd
```

```
$ls
```

```
$rmdir ddd
```

```
[orcl:~]$ mkdir ddd
[orcl:~]$ ls
Adobe Desktop  R-3.2.3  R-3.2.3.tar.gz  datamodeler.log  ddd  emp.csv  home  labs  oradiag_oracle  test2
[orcl:~]$ rmdir ddd
[orcl:~]$ ls
Adobe Desktop  R-3.2.3  R-3.2.3.tar.gz  datamodeler.log  emp.csv  home  labs  oradiag_oracle  test2
```

문제10. /home/oracle 밑에 test9 디렉토리를 만들고 test9 디렉토리 밑에 test10 디렉토리를

만드시오.

답)

```
[orcl:~]$ mkdir -p ./test9/test10
[orcl:~]$ cd test10
-bash: cd: test10: No such file or directory
[orcl:~]$ cd test9
[orcl:test9]$ cd test10
[orcl:test10]$ pwd
/home/oracle/test9/test10
[orcl:test10]$
```

ls

```
Adobe Desktop R-3.2.3 R-3.2.3.tar.gz datamodeler.log emp.csv home labs oradiag_oracle test2 test9
[orcl:~]$
```

문제11. /home/oracle 밑에 test9와 test10 디렉토리를 전부 삭제하시오.

```
[orcl:~]$ cd test9
[orcl:test9]$ rmdir test10
[orcl:test9]$ cd ..
[orcl:~]$ rmdir test9
[orcl:~]$ ls
```

```
Adobe Desktop R-3.2.3 R-3.2.3.tar.gz datamodeler.log emp.csv home labs oradiag_oracle test2
[orcl:~]$
```

한번에 삭제 하는 방법

\$rm -r test9

설명 : -r 옵션은 하위 디렉토리까지 다 삭제하겠다.

6. alias 명령어 : 자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어

ex) 오라클 db에 접속

\$sqlplus / as sysdba 원래 이렇게 접속하는데

\$ alias sql='sqlplus / as sysdba' 으로 저장하고

\$ sql을 쓰면 접속이 된다

```
[orcl:~]$ alias sql='sqlplus / as sysdba'
[orcl:~]$ sql

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:14:30 2018

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
```

문제12. scott이라고 아래와 같이 명령어를 수행하면 오라클에 scott으로 접속되게 alias를 만드시오.


```
[orcl:~]$ alias scott='sqlplus scott/tiger'
[orcl:~]$ scott

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:18:05 2018

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
```

alias 삭제하는 방법

unalias alias이름

ex) unalias scott

```
[orcl:~]$ unalias scott
[orcl:~]$ alias
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias sql='sqlplus / as sysdba'
alias vi='vim'
```

문제13. 오라클의 네트워크 설정에 대한 디렉토리가 있는곳으로 이동하시오.

```
[orcl:~]$ cd $ORACLE_HOME/network/admin
[orcl:admin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
[orcl:admin]$
```

```
[orcl:admin]$ ls
samples shrept.lst tnsnames.ora
```

문제14. 아래의 디렉토리로 이동을 쉽게 할 수 있는 alias를 생성하시오.

보기)

/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames.ora

답)

alias oracle_network='cd \$ORACLE_HOME/network/admin'

7. cat 명령어 : 파일의 내용을 화면에 출력하는 명령어

ex \$cat 파일명.txt

문제15. cat으로 본 화면의 결과를 emp40.txt라는 이름으로 저장하시오.

cat emp.txt >> emp40.txt

```
[orcl:~]$ cat emp.txt >> emp40.txt
[orcl:~]$ ls -l emp40.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:33 emp40.txt
```

문제16. 현재 디렉토리에 아래의 파일들을 touch명령어로 생성하시오.

보기)

```
$ touch a.txt b.txt c.txt e.txt
```

```
$ ls -l *.txt
```

답)

```
[orcl:~]$ touch a.txt b.txt c.txt e.txt
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:32 dept.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 e.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:33 emp40.txt
```

문제17. 지금 `ls -l *.txt`로 보고 있는 화면의 결과를 `ls.txt`라는 이름으로 저장하시오.

```
ls -l *.txt >> ls.txt
```

```
cat ls.txt
```

```
10 ACCOUNTING      NEW YORK
20 RESEARCH        DALLAS
30 SALES           CHICAGO
40 OPERATIONS      BOSTON

7369 SMITH         CLERK             7902 17-DEC-80      800      0      20
7499 ALLEN         SALESMAN          7698 20-FEB-81     1600     300     30
7521 WARD          SALESMAN          7698 22-FEB-81     1250     500     30
7566 JONES         MANAGER           7839 02-APR-81     2975      0      20
7654 MARTIN        SALESMAN          7698 28-SEP-81     1250    1400     30
7698 BLAKE         MANAGER           7839 01-MAY-81     2850      0      30
7782 CLARK         MANAGER           7839 09-JUN-81     2450      0      10
7788 SCOTT         ANALYST           7566 19-APR-87     3000      0      20
7839 KING          PRESIDENT         0 17-NOV-81     5000      0      10
7844 TURNER        SALESMAN          7698 08-SEP-81     1500      0      30
7876 ADAMS         CLERK             7788 23-MAY-87     1100      0      20
7900 JAMES         CLERK             7698 03-DEC-81      950      0      30
7902 FORD          ANALYST           7566 03-DEC-81     3000      0      20
7934 MILLER        CLERK             7782 23-JAN-82     1300      0      10
7369 SMITH         CLERK             7902 17-DEC-80      800      0      20
7499 ALLEN         SALESMAN          7698 20-FEB-81     1600     300     30
7521 WARD          SALESMAN          7698 22-FEB-81     1250     500     30
7566 JONES         MANAGER           7839 02-APR-81     2975      0      20
7654 MARTIN        SALESMAN          7698 28-SEP-81     1250    1400     30
7698 BLAKE         MANAGER           7839 01-MAY-81     2850      0      30
7782 CLARK         MANAGER           7839 09-JUN-81     2450      0      10
7788 SCOTT         ANALYST           7566 19-APR-87     3000      0      20
7839 KING          PRESIDENT         0 17-NOV-81     5000      0      10
7844 TURNER        SALESMAN          7698 08-SEP-81     1500      0      30
7876 ADAMS         CLERK             7788 23-MAY-87     1100      0      20
7900 JAMES         CLERK             7698 03-DEC-81      950      0      30
7902 FORD          ANALYST           7566 03-DEC-81     3000      0      20
7934 MILLER        CLERK             7782 23-JAN-82     1300      0      10
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:32 dept.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 e.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:33 emp40.txt
-rw-r--r-- 1 oracle oinstall 2045 Jun 22 11:36 ls.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:32 dept.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:34 e.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:33 emp40.txt
-rw-r--r-- 1 oracle oinstall 2426 Jun 22 11:36 ls.txt
```

8. redirection 명령어 : 화면에 출력되는 결과를 파일로 저장하는 명령어

ex)

>> : 없으면 파일을 생성하고 있으면 기존 파일뒤에 덧붙이겠다.

> : 파일을 생성하겠다. 기존 파일이 있으면 그냥 덮어 쓰겠다.

문제18. emp.txt 를 cat으로 보고 있는 결과를 emp50.txt라는 이름으로 생성하고 다시 emp.txt의 결과를 emp50.txt에 덧붙여서 저장하시오.

```
$ cat emp.txt > emp50.txt
```

```
$ cat emp50.txt
```

```
[orcl:~]$ cat emp50.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81     1600     300     30
7521 WARD        SALESMAN   7698 22-FEB-81     1250     500     30
7566 JONES      MANAGER    7839 02-APR-81     2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81     1250    1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81     2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81     2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87     3000      0      20
7839 KING       PRESIDENT      0 17-NOV-81     5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81     1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87     1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950      0      30
7902 FORD       ANALYST    7566 03-DEC-81     3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82     1300      0      10
```

```
$ cat emp.txt >> emp50.txt
```

```
[orcl:~]$ cat emp50.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81     1600     300     30
7521 WARD        SALESMAN   7698 22-FEB-81     1250     500     30
7566 JONES      MANAGER    7839 02-APR-81     2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81     1250    1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81     2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81     2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87     3000      0      20
7839 KING       PRESIDENT      0 17-NOV-81     5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81     1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87     1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950      0      30
7902 FORD       ANALYST    7566 03-DEC-81     3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82     1300      0      10
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81     1600     300     30
7521 WARD        SALESMAN   7698 22-FEB-81     1250     500     30
7566 JONES      MANAGER    7839 02-APR-81     2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81     1250    1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81     2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81     2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87     3000      0      20
7839 KING       PRESIDENT      0 17-NOV-81     5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81     1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87     1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950      0      30
7902 FORD       ANALYST    7566 03-DEC-81     3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82     1300      0      10
```

9. wc 명령어 : 파일안에 단어의 개수 또는 라인수를 출력하는 명령어

ex) wc -l 파일명

```
wc -l emp.txt
```

```
[orcl:~]$ wc -l emp.txt
14 emp.txt
```

옵션 : -l = 라인수 출력

-w = 단어의 개수 출력

-c = 문자수를 출력

문제19. 겨울왕국 대본의 단어수가 어떻게 되는가?(점심시간)

```
[orcl:~]$ wc -w winter.txt
19909 winter.txt
```

##wc winter.txt

```
[orcl:~]$ wc winter.txt
4239 19909 114548 winter.txt
```

4239 : 라인수, 19909 : 단어수, 114548 : 바이트 크기

문제20. /home/oracle밑에 확장자가 .txt로 끝나는 파일이 몇 개가 있는지 카운트 하시오.

답)

ls -l *.txt | wc -l

```
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall      0 Jun 22 11:34 a.txt
-rw-r--r-- 1 oracle oinstall      0 Jun 22 11:34 c.txt
-rw-r--r-- 1 oracle oinstall  105 Jun 22 11:32 dept.txt
-rw-r--r-- 1 oracle oinstall      0 Jun 22 11:34 e.txt
-rw-r--r-- 1 oracle oinstall   970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall   970 Jun 22 11:33 emp40.txt
-rw-r--r-- 1 oracle oinstall  1940 Jun 22 11:41 emp50.txt
-rw-r--r-- 1 oracle oinstall  2807 Jun 22 11:37 ls.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 22 11:45 winter.txt
[orcl:~]$ ls -l *.txt | wc -l
9
```

설명 : | = 파이프 명령어

앞의 명령어 | 뒤의 명령어 : 앞의 명령어의 표준 출력을 뒤의 명령어의

표준 입력으로 보냄으로써 명령어의 실행 결과를

다음 명령어로 전달하는 기능.

문제20. 아래의 경로로 이동해서 아래의 디렉토리 아래있는 확장자 trc 파일이 총 몇 개가 있는지 확인하시오.

보기)

/u01/app/oracle/diag/rdbms/orcl/orcl/trace

답)

```
[orcl:~]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
```

```
[orcl:trace]$ ls -l *.trc | wc -l
```

```
[orcl:trace]$ ls -l *.trc | wc -l
182
```

10. more 명령어 : 1 페이지가 넘는 문서의 내용을 화면에 출력할때 페이지 단위로 볼 수

있는 명령어

ex) \$ more winter.txt

전진키 : space

후진키 : b

페이지 단위로 내려가는 키 : f

종료키 : q

11. head 명령어 : 문서의 처음 몇줄을 화면에 출력하는 명령어

ex) \$ head -출력줄수 파일명

\$ head -20 winter.txt

```
[orcl:~]$ head -20 winter.txt
"Final Shooting Draft
by
Jennifer Lee
9/23/13
Property of the Walt Disney Animation Studios
OPEN ON: ICE.
We' re underwater looking up at it. A saw cuts through,
heading right for us.
EXT. SNOW-CAPPED MOUNTAINS ? DUSK
ICE HARVESTERS, dressed in traditional Sami clothing, score a
frozen lake. They SING.
" The Frozen Heart (Ice Worker' s Song)"
ICE HARVESTERS
BORN OF COLD AND WINTER AIR
AND MOUNTAIN RAIN COMBINING,
THIS ICY FORCE BOTH FOUL AND FAIR
HAS A FROZEN HEART WORTH MINING.
The men drag giant ice blocks through channels of water.
ICE HARVESTERS (CONT' D)
CUT THROUGH THE HEART, COLD AND CLEAR.
```

문제21. 겨울왕국 대본의 100줄까지 내용을 winter_head.txt라는 이름으로 저장하시오.

답)

head -100 winter.txt >> winter_head.txt

12. tail 명령어 : 문서의 마지막 몇줄을 화면에 출력하는 명령어

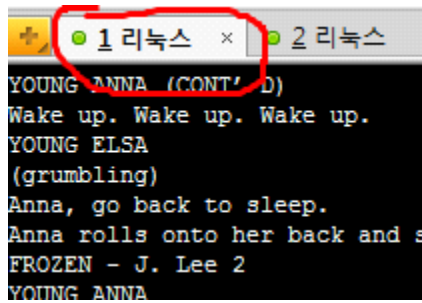
ex) \$ tail -출력줄수 파일명

\$ tail -10 winter.txt

- tail 명령어의 중요한 옵션 : -f 옵션
-f 옵션 = 문서의 끝에 추가되는 새로운 내용을 실시간으로 확인할 수 있는 옵션

ex) 오라클 db가 올라가거나 내려갈때의 그 정보를 적는 파일을 tail -f로 여시오.

cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace 로 가서



파일이름 오른쪽클릭하고 세션 복제 ㄱㄱ

원래 세션으로 와서

tail -f alert_orcl.log

```
[orcl:trace]$ tail -f alert_orcl.log
Thu Jun 21 15:27:05 2018
Starting background process SMC0
Thu Jun 21 15:27:05 2018
SMCO started with pid=22, OS id=5690
Thu Jun 21 17:24:48 2018
Thread 1 advanced to log sequence 16 (LGWR switch)
  Current log# 1 seq# 16 mem# 0: +DATA/orcl/onlinelog/group_1.261.796857743
  Current log# 1 seq# 16 mem# 1: +FRA/orcl/onlinelog/group_1.257.796857749
Fri Jun 22 09:59:40 2018
Starting ORACLE instance (normal)
```

복제 세션으로 가서

[orcl:~]\$ sqlplus / as sysdba

shutdown immediate #디비 종료

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
```

다시 startup

```
Total System Global Area 481259520 bytes
Fixed Size                  1337352 bytes
Variable Size               314574840 bytes
Database Buffers            159383552 bytes
Redo Buffers                 5963776 bytes
Database mounted.
Database opened.
```

ex) 현재 리눅스 시스템의 i/o 성능을 확인 하시오(윈도우의 작업 관리자)

\$ sar 1 100

```
[orcl:~]$ sar 1 100
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/22/18
```

	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:29:00	all	0.00	0.00	3.52	0.00	0.00	96.48
14:29:01	all	0.50	0.00	1.50	0.00	0.00	98.00
14:29:02	all	0.50	0.00	2.49	0.00	0.00	97.01
14:29:03	all	0.00	0.00	0.50	0.00	0.00	99.50
14:29:04	all	0.00	0.00	0.50	0.00	0.00	99.50
14:29:05	all	0.00	0.00	7.37	0.00	0.00	92.63
14:29:06	all	0.50	0.00	0.00	0.00	0.00	99.50
14:29:07	all	0.00	0.00	0.50	0.00	0.00	99.50
14:29:08	all	0.00	0.00	3.50	0.00	0.00	96.50
14:29:09	all	0.50	0.00	0.50	0.00	0.00	99.00
14:29:10	all	0.00	0.00	1.50	0.00	0.00	98.50
14:29:11	all	0.50	0.00	1.49	0.00	0.00	98.01
14:29:12	all	0.00	0.00	4.00	0.00	0.00	96.00
14:29:13	all	0.00	0.00	2.01	0.00	0.00	97.99

나오는 방법 ctrl + z

현재 user가 별로 안바쁘는데, 바쁘게 하는방법

sys세션에서

select count(*)

2 from sh.sales, sh.sales; <-- 끊는 방법 ctrl + c

끊지 않고 다시 원래 세션으로와서보면

```
[orcl:~]$ sar 1 100
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/22/18
```

	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:32:01	all	49.50	0.00	1.50	0.00	0.00	49.00
14:32:02	all	49.75	0.00	0.50	0.00	0.00	49.75
14:32:03	all	49.50	0.00	1.00	0.00	0.00	49.50
14:32:04	all	50.00	0.00	0.00	0.00	0.00	50.00
14:32:05	all	50.00	0.00	0.00	0.00	0.00	50.00
14:32:06	all	50.00	0.00	0.50	0.00	0.00	49.50
14:32:07	all	49.75	0.00	1.00	0.00	0.00	49.25
14:32:08	all	50.00	0.00	0.50	0.00	0.00	49.50
14:32:09	all	49.01	0.00	1.49	0.00	0.00	49.50
14:32:10	all	50.00	0.00	0.00	0.00	0.00	50.00

엄청 바쁘다.

문제22. redirection을 이용해서 sar 1 100 명령어의 결과가 sar_0622.txt 라는 이름으로 저장되게 하시오.

```
[orcl:~]$ sar 1 100 >> sar_0622.txt
```

```
tail -f sar_0622.txt
```

프로세스 죽이는 방법

\$ kill -p 프로세스 번호

13. grep 명령어 : 파일안에 포함된 특정 단어나 구문을 검색하는 명령어

ex) \$ grep '찾고 싶은 단어' 파일명


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

설명 : emp.txt에서 scott 이라는 단어가 포함된 라인을 보여달라.

-i 옵션 : 대소문자를 구분하지 않겠다.

```
[orcl:~]$ grep -i 'scott' emp.txt
7788 SCOTT ANALYST 7566 19-APR-87 3000 0 20
```

문제23. 직업이 salesman인 직원들의 모든 행을 출력하시오.

답)

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

```
[orcl:~]$ grep -i 'salesman' emp.txt
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
```

문제24. 직업이 salesman인 직원들의 이름과 월급을 출력하시오.

답)

```
[orcl:~]$ grep -i 'salesman' emp.txt | awk '{print $2,$6}'
```

```
ALLEN 1600
WARD 1250
MARTIN 1250
TURNER 1500
```

문제25. 부서번호가 10번인 직원들의 이름과 월급을 출력하시오.

```
[orcl:~]$ grep -iw '10' emp.txt | awk '{print $2,$6}'
```

```
CLARK 2450
KING 5000
MILLER 1300
```

그냥 '10'하면 월급의 1100 도 입력된다.

설명 : 옵션 -w는 단어별 검색

14. awk 명령어 : 특정 단어가 들어가 있는 라인에서 특정 컬럼명을 출력하고자 할때 사용하는 명령어(오라클에서 where절)

ex) \$ awk '패턴{action}' 대상 파일명

```
$ awk '{print $2, $6}' emp.txt
```

emp 테이블에서 2번,6번째 컬럼을 볼 수 있다.

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

emp 테이블에서 직업이 SALESMAN인 직원의 2,3번째(이름,직업) 컬럼을 출력

```
ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
```

문제26. 월급이 3000 이상인 직원들의 이름과 월급을 출력하시오.

```
[orcl:~]$ awk '$6>=3000 {print $2,$6}' emp.txt
```

```
SCOTT 3000
KING 5000
```



```
SCOTT 3000
KING 5000
FORD 3000
```

• 리눅스의 연산자 3가지

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

문제27. 직업이 salesman이 아닌 직원들의 이름과 직업을 출력하시오.

```
[orcl:~]$ awk '$3!= "salesman" {print $2,$3}' emp.txt
[orcl:~]$ awk '$3!= toupper("salesman") {print $2,$3}' emp.txt
```

```
[orcl:~]$ awk '$3!="SALESMAN" {print $2,$3}' emp.txt
SMITH CLERK
JONES MANAGER
BLAKE MANAGER
CLARK MANAGER
SCOTT ANALYST
KING PRESIDENT
ADAMS CLERK
JAMES CLERK
FORD ANALYST
MILLER CLERK
```

문제28. 직업이 salesman인 직원들의 이름과 월급을 출력하는데 월급이 1500이상인 직원들만 출력하시오.

```
[orcl:~]$ awk '$3=="SALESMAN" && $6>=1500 {print $2,$6}' emp.txt
```

```
ALLEN 1600
TURNER 1500
```

문제29. 이름이 A로 시작하는 직원들의 이름과 월급을 출력하시오.

힌트 : grep과 정규식을 사용(시작 : ^, 끝 : \$)

```
[orcl:~]$ awk '{print $2,$6}' emp.txt | grep -i '^a'
```

```
ALLEN 1600
ADAMS 1100
```

문제30. 81년도에 입사한 직원들의 이름과 입사일을 출력하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep -i '81$'
```

```
ALLEN 20-FEB-81
WARD 22-FEB-81
JONES 02-APR-81
MARTIN 28-SEP-81
BLAKE 01-MAY-81
CLARK 09-JUN-81
KING 17-NOV-81
TURNER 08-SEP-81
JAMES 03-DEC-81
FORD 03-DEC-81
```

문제31. 81년도에 입사하지 않은 직원들의 이름과 입사일을 출력하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep -v 81
```

설명: -v = 포함되지 않는것을 검색할때 사용.

```
SMITH 17-DEC-80
SCOTT 19-APR-87
ADAMS 23-MAY-87
MILLER 23-JAN-82
```

문제32. 겨울왕국 대본에서 elsa가 포함되어져 있는 라인들의 개수가 전체 몇 개인지 출력하시오.

```
[orcl:~]$ grep -i 'elsa' winter.txt | wc -l
318
```

- R로 접속하는 방법

```
[orcl:~]$ cd R-3.2.3
```

```
[orcl:R-3.2.3]$ cd bin #bin디렉토리:실행파일있는 디렉토리
```

```
[orcl:bin]$ ./R
```

- 빠르게 접속하는 방법

/home/oracle/R-3.2.3/ 복사해두고

/home/oracle 로 와서

```
[orcl:~]$ vi .bash_profile <-- oracle 유저로 접속했을때 자동으로 실행되는 파일
vi = 파일을 여는 명령어
입력모드(o)를 누르고 적어준다.
```

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_SID=orcl
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export PATH=/home/oracle/R-3.2.3/bin:$PATH
```

: 버튼을 누르면 아래에 커서가 나오는데

wq! (저장하고 나가기)를 해준다.

```
[orcl:~]$ . .bash_profile 한번 실행하고
```

15. sort 명령어 : data 를 특정 컬럼을 기준으로 정렬하는 명령어

ex) \$ sort 옵션 파일명

\$ sort -k 6 emp.txt <-- -k : key번호를 써라.

emp.txt 의 모든 데이터를 출력하는데 6번째 컬럼(월급)이 적은 것부터 출력하시오.

문제34. 위의 결과를 다시출력하는데 이름과 월급만 나오게 하시오.

답)

```
[orcl:~]$ sort -k 6 emp.txt | awk '{print $2,$6}'
```

문제35. 이름과 월급을 출력하는데 월급이 높은것 부터 출력하시오.

답)

```
[orcl:~]$ sort -rk 6 emp.txt | awk '{print $2,$6}'
```

문제36. 직업이 salesman인 직원들의 이름과 월급을 출력하는데 월급이 높은것부터 출력하시오.

답)

```
[orcl:~]$ sort -rk 6 emp.txt | awk '$3==toupper("salesman") {print $2, $6}'
```

```
ALLEN 1600
TURNER 1500
MARTIN 1250
WARD 1250
```

문제37. 직원 테이블의 월급의 토탈값을 출력하시오.

답)

```
[orcl:~]$ awk '{print $6}' emp.txt | awk '{sum += $1} END {print sum}'
```

```
29025
```

```
[orcl:~]$ awk '{print $6}' emp.txt | awk '{sum += $1} END {print sum}'
```

문제38. 직업이 SALESMAN인 사원들의 토탈월급을 출력하시오.

문제39. 아래와 같이 결과가 출력되게 하시오.

보기)

KING 의 월급은 5000

ALLEN 의 월급은 3000

답)

```
[orcl:~]$ awk '{print $2 " 의 월급은 " $6 "입니다"}' emp.txt
```

```
SMITH 의 월급은 800입니다
ALLEN 의 월급은 1600입니다
WARD 의 월급은 1250입니다
JONES 의 월급은 2975입니다
MARTIN 의 월급은 1250입니다
BLAKE 의 월급은 2850입니다
CLARK 의 월급은 2450입니다
SCOTT 의 월급은 3000입니다
KING 의 월급은 5000입니다
TURNER 의 월급은 1500입니다
ADAMS 의 월급은 1100입니다
JAMES 의 월급은 950입니다
FORD 의 월급은 3000입니다
MILLER 의 월급은 1300입니다
```

문제40. 아래와 같이 결과를 출력하시오.

보기)

KING 의 직업은 PRESIDENT 입니다

ALLEN 의 직업은 ANALYST 입니다

답)

KING 의 월급은 5000

ALLEN 의 월급은 3000

```
SMITH 의 직업은 CLERK입니다
ALLEN 의 직업은 SALESMAN입니다
WARD 의 직업은 SALESMAN입니다
JONES 의 직업은 MANAGER입니다
MARTIN 의 직업은 SALESMAN입니다
BLAKE 의 직업은 MANAGER입니다
CLARK 의 직업은 MANAGER입니다
SCOTT 의 직업은 ANALYST입니다
KING 의 직업은 PRESIDENT입니다
TURNER 의 직업은 SALESMAN입니다
ADAMS 의 직업은 CLERK입니다
JAMES 의 직업은 CLERK입니다
FORD 의 직업은 ANALYST입니다
MILLER 의 직업은 CLERK입니다
```

문제41. /home/oracle 에서, ls -lrt 라고 조회한 결과를 분석하시오.

```
[orcl:~]$ pwd
```

```
/home/oracle
```

```
[orcl:~]$ ls -lrt
```

설명 : -l = 리스트

-r = reverse

-t = 시간

파일 리스트를 보여주는데 오래된 것부터 보여달라!

문제42. ls -lrt *.txt 했을때 파일 크기에 해당하는 부분만 출력하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{print$5}'
```

```
970
105
970
0
0
0
2807
1940
114548
6774
12364
```

문제43. 위에서 출력된 파일크기를 전부 sum해서 출력하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{sum += $5} END {print sum}'
```

```
140478
```

문제44. 아래와 같이 결과를 출력되게 하시오.

보기)

현재 디렉토리에 있는 txt 파일의 총 크기는 140478 바이트 입니다.

답)

```
[orcl:~]$ ls -lrt *.txt | awk '{sum += $5} END {print "the size of the total file under the current directory is" sum " byte" }'
```

```
the size of the total file under the current directory is140478 byte
```

문제45. 위의 스크립트를 쉘 스크립트로 저장해서 실행되게 하시오.

vi size.sh 입력후 a눌러서 insert모드로 바꾼다음

insert화면에

```
ls -lrt *.txt | awk '{sum += $5} END {print "the size of the total file under the current directory is"
```

```
sum " byte" }' <- 이부분을 붙여 넣고 esc를 누르고 : 을 입력후
```

wq!로 나온다.

16. uniq 명령어 : 중복된 라인을 제거하는 명령어

ex) uniq 옵션 파일명

문제46. emp.txt 에서 직업만 출력하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt
```

```
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK
```

문제47. 문제46번을 abcd순으로 정렬해서 출력하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort
```

```
ANALYST
ANALYST
CLERK
CLERK
CLERK
CLERK
MANAGER
MANAGER
MANAGER
PRESIDENT
SALESMAN
SALESMAN
SALESMAN
SALESMAN
```

문제48. 문제47번을 중복제거해서 출력하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort | uniq
```

```
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제49. 부서번호를 출력하는데 중복제거해서 출력하시오.

```
[orcl:~]$ awk '{print $8}' emp.txt | sort | uniq
```

```
10
20
30
```

문제50. emp.txt를 복사해서 emp7.txt를 생성하고 emp7.txt에 KING에 대한 데이터를 복사해서 맨 아래에 하나를 입력하고 저장하시오.

```
[orcl:~]$ cp emp.txt emp7.txt # cp : 카피
```

vi emp7.txt 로 들어가서 커서를 KING에다가 두고 yy(한줄복사)

커서를 맨아래로가서 p(붙여넣기) 하고 나오면된다.

```
[orcl:~]$ cat emp7.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN    7698 20-FEB-81      1600     300     30
7521 WARD      SALESMAN    7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER     7839 02-APR-81      2975      0      20
7654 MARTIN    SALESMAN    7698 28-SEP-81      1250     1400     30
7698 BLAKE      MANAGER     7839 01-MAY-81      2850      0      30
7782 CLARK      MANAGER     7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST     7566 19-APR-87      3000      0      20
7839 KING      PRESIDENT   0 17-NOV-81      5000      0      10
7844 TURNER    SALESMAN    7698 08-SEP-81      1500      0      30
7876 ADAMS      CLERK       7788 23-MAY-87      1100      0      20
7900 JAMES      CLERK       7698 03-DEC-81      950       0      30
7902 FORD      ANALYST     7566 03-DEC-81      3000      0      20
7934 MILLER    CLERK       7782 23-JAN-82      1300      0      10
7839 KING      PRESIDENT   0 17-NOV-81      5000      0      10
```

문제51. emp7.txt에서 중복된 라인만 출력하시오.

```
[orcl:~]$ sort emp7.txt | uniq -d #일단 sort(정렬)을 먼저해준다.
```

```
7839 KING      PRESIDENT   0 17-NOV-81      5000      0      10
```

문제52. emp7.txt에서 중복된 라인을 제거하고 출력하시오.

```
[orcl:~]$ sort emp7.txt | uniq -u
```

```
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN    7698 20-FEB-81      1600     300     30
7521 WARD      SALESMAN    7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER     7839 02-APR-81      2975      0      20
7654 MARTIN    SALESMAN    7698 28-SEP-81      1250     1400     30
7698 BLAKE      MANAGER     7839 01-MAY-81      2850      0      30
7782 CLARK      MANAGER     7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST     7566 19-APR-87      3000      0      20
7844 TURNER    SALESMAN    7698 08-SEP-81      1500      0      30
7876 ADAMS      CLERK       7788 23-MAY-87      1100      0      20
7900 JAMES      CLERK       7698 03-DEC-81      950       0      30
7902 FORD      ANALYST     7566 03-DEC-81      3000      0      20
7934 MILLER    CLERK       7782 23-JAN-82      1300      0      10
```

17. echo 명령어 : 출력하고자 하는 글자를 출력할 때 사용

ex) echo '나는 나다'

```
[orcl:~]$ echo '나는 나다'
나는 나다
```

문제53. 직업을 물어보게하고 직업을 입력하면 해당 직업을 갖는 직원들의 이름과 직업이 출력되게하시오.

\$ vi job2.sh <- 편집기를 열고 아래 코드를 붙여 넣는다

```
#!/bin/bash    <- 스크립트를 bash 셸을 쓰겠다.
$ echo "      "    <- 공백입력
$ echo -n "input jobs"    <- -n : enter가 수행된것.
$ read job
echo "      "
$ grep -i $job emp.txt | awk '{print $2, $3}'
저장하고 나와서
sh job2.sh
```

```
[orcl:~]$ vi job2.sh
[orcl:~]$ sh job2.sh

input jobs salesman

ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
```

문제54. 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호에서 근무하는 직원들의 이름과 월급과 부서번호가 출력되게하시오.

vi deptno2.sh

```
#!/bin/bash
echo "      "
echo -n "input deptno"
read deptno
echo "      "
awk '{print $2,$6,$8}' emp.txt | grep -iw $deptno
```

```
input deptno 20

SMITH 800 20
JONES 2975 20
SCOTT 3000 20
ADAMS 1100 20
FORD 3000 20
```

문제55. 이름을 물어보게하고 이름을 입력하면 해당사원의 월급이 아래와 같이 출력되게 하시오.

답)

vi find_sal.sh

```
#!/bin/bash
echo "      "
echo -n "input ename"
read ename
echo "      "
grep -iw $ename emp.txt | awk '{print $2,"의 월급은", $6, "입니다"}'
```

```
input ename KING
KING 의 월급은 5000 입니다
```

문제56. 직업을 물어보게하고 직업을 입력하면 해당 직업의 토탈월급이 아래와 같이 출력되게 하시오.

```
보기)
sh job_sumsal.sh
직업을 입력하세요~ salesman
토탈 월급은 5600 입니다.
```

```
답)
vi job_sumsal.sh
```

```
#!/bin/home
echo "      "
echo -n "input jobs"
read job
echo "      "
grep -i $job emp.txt | awk '{sum += $6} END {print "해당 직업의 토탈 월급은", sum, "입니다"}'
```

```
input jobs salesman
해당 직업의 토탈 월급은 5600 입니다
```

문제57. 입사일을 출력하는데 뒤에 년도만 출력하시오.

```
[orcl:~]$ awk '{print substr($5,8,10)}' emp.txt
```

```
80
81
81
81
81
81
81
87
81
81
87
81
81
82
```


문제57. 년도를 물어보게하고 년도를 입력하면 해당 년도에 입사한 직원들의 이름과 입사일이 출력되게 하시오.

보기)

sh year.sh

년도를 입력하십시오 ~ 81

답)

vi year.sh

```
#!/bin/home
```

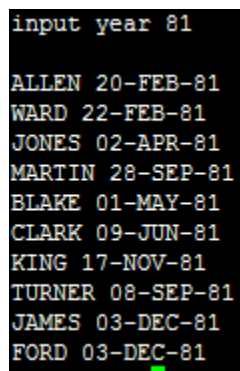
```
echo "      "
```

```
echo -n "input year"
```

```
read year
```

```
echo "      "
```

```
grep -iw $year emp.txt | awk '{print $2, $5}'
```



```
input year 81

ALLEN 20-FEB-81
WARD 22-FEB-81
JONES 02-APR-81
MARTIN 28-SEP-81
BLAKE 01-MAY-81
CLARK 09-JUN-81
KING 17-NOV-81
TURNER 08-SEP-81
JAMES 03-DEC-81
FORD 03-DEC-81
```

문제58. (점심시간)년도를 물어보게하고 년도를 입력하면 해당 년도에 토탈월급이 출력되게 하시오.

보기)

sh year_sum.sh

년도를 입력하세요 ~ 81

해당 년도의 토탈 월급은 22825 입니다

답)

vi year_sum.sh

```
#!/bin/home
```

```
echo "      "
```

```
echo -n "input year"
```

```
read year
```

```
echo "      "
```

```
grep -iw $year emp.txt | awk '{sum += $6} END {print "해당 년도의 토탈 월급은", sum, "입니다"}'
```

```
input year 81
해당 년도의 토탈 월급은 22825 입니다
```

18. diff 명령어 : 두 파일간의 차이점을 찾아서 알려주는 명령어

ex) diff emp.txt emp7.txt

```
[orcl:~]$ diff emp.txt emp7.txt
14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```

문제59. 아래와 같이 두개의 파일 이름을 각각 물어보게하고, 결과로 파일의 내용의 차이가 출력되게 하시오.

보기)

sh diff.sh

비교할 첫 번째 파일명을 입력하세요 ~ emp.txt

비교할 두 번째 파일명을 입력하세요 ~ emp7.txt

```
[orcl:~]$ diff emp.txt emp7.txt
14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```

답)

vi diff.sh

```
#!/bin/home
```

```
echo "      "
```

```
echo -n "input file1"
```

```
read f1_t
```

```
echo "      "
```

```
echo -n "input file2"
```

```
read f2_t
```

```
diff $f1_t $f2_t
```

```
[orcl:~]$ diff emp.txt emp7.txt
14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```

19. find 명령어 : 검색하고 하는 파일을 찾을때 사용하는 명령어

ex) \$find 디렉토리 -name 파일명 -print

```
$find /home/oracle -name 'emp.txt' -print
```

설명 : /home/oracle 디렉토리 밑에 emp.txt라는 파일을 있는지 찾아보시오.

```
-bash: /home/oracle: is a directory
```

문제60. /home/oracle 밑에 test100 과 test200이라는 디렉토리를 생성하고 그 밑에 emp.txt 를 각각 복사하시오.

```
[oracle:~]$ mkdir test100 test200
```

```
[oracle:~]$ cp emp.txt ./test100/emp.txt
```

```
[oracle:~]$ cp emp.txt ./test200/emp.txt
```

./ : 현재 디렉토리

문제61. /home/oracle 밑에 있는 emp.txt를 검색하시오.

답1)

```
[oracle:~]$ find /home/oracle -name emp.txt -print
```

```
/home/oracle/test100/emp.txt
/home/oracle/test200/emp.txt
/home/oracle/emp.txt
```

답2)

```
[oracle:~]$ find . -maxdepth 1 -name 'emp.txt'
```

```
./emp.txt
```

```
[oracle:~]$ find . -maxdepth 2 -name 'emp.txt'
```

```
./test100/emp.txt
./test200/emp.txt
./emp.txt
```

문제62. 파일을 검색하는 쉘 스크립트를 아래와 같이 생성하시오.

보기)

```
sh find_file.sh
```

현재 디렉토리 밑에 검색할 파일명을 입력하세요~ emp.txt

depth는 어떻게 할까요~ 1

```
./emp.txt
```

답)

```
#!/bin/home
```

```
echo "      "
```

```
echo -n "input find file"
```

```
read file1
```

```
echo "      "
```

```
echo -n "depth ?"
```

```
read depth1
```

```
find . maxdepth $depth1 -name $file1
```

```
input find file :emp.txt

depth ? :1
./test100/emp.txt
./test200/emp.txt
./emp.txt
```

20. tar 명령어 : 파일을 압축하고 압축을 해제하는 명령어

ex) 1. 압축할때

tar cvf 압축파일명 압축파일대상

2. 압축을 해제할때

tar xvf 압축파일명 압축해제할 위치

옵션 : -c = compress, 여러 개의 파일을 하나로 만들어라

-v = view, 압축되는 과정을 보여준다.

-f = file, 생성되는 파일명을 지정

-x = extract, 묶여있는 파일을 풀어줘라

-C = 압축이 풀릴 위치를 지정

문제63. /home/oracle 밑에 test63이라는 디렉토리를 생성하고, 그 밑에 emp.txt를 복사해서 아래의 파일들을 생성하시오.

보기)

a.txt b.txt c.txt d.txt e.txt

답)

```
[orcl:~]$ mkdir test63
```

```
[orcl:~]$ cp emp.txt ./test63/a.txt
```

```
[orcl:~]$ cp emp.txt ./test63/b.txt
```

```
[orcl:~]$ cp emp.txt ./test63/c.txt
```

```
[orcl:~]$ cp emp.txt ./test63/d.txt
```

```
[orcl:~]$ cp emp.txt ./test63/e.txt
```

문제64. test63 디렉토리 밑에있는 확장자.txt 파일들을 all.tar라는 이름으로 압축하시오.

답)

```
[orcl:~]$ tar cvf all.tar ./test63/*.txt
```

```
./test63/a.txt
```

```
./test63/b.txt
```

```
./test63/c.txt
```

```
./test63/d.txt
```

```
./test63/e.txt
```

문제65. /home/oracle 밑에 test65라는 디렉토리를 생성하고 그 밑에 all.tar 압축파일의 압축을 해제하시오.

답)

```
[orcl:~]$ tar xvf all.tar -C /home/oracle/test65
```

21. ln(link) 명령어 : 윈도우의 바로가기 기능과 유사함

내가 자주 열어 봐야 하는 파일이 있다면 바로가기를 생성해 놓으면 편하게 열 수 있음.

ex) ln -s "링크를 걸 파일위치와 파일명" "바로가기이름"

ln -s /u01/app/oracle/product/11.2.0/dbhome_1/sqlplus/admin/glogin.sql glogin.sql

scott으로 접속후

vi glogin.sql <-- sqlplus 툴의 환경설정 파일

define_editor='gedit' <-- 마지막줄에 입력 에디터로 리눅스의 메모장인 gedit을 사용

```
SQL> select ename, sal
  2  from emp
  3  where empno = 7788;

ENAME          SAL
-----
SCOTT           3000
```

ed 하면

gedit 창이 뜬다

22. .bash_profile 설정하는 방법

.bash_profile 이란? oracle 유저를 위한 리눅스 환경설정 파일

ex) vi .bash_profile

맨 아래쪽에 아래의 내용을 적고 저장하고 나온다.

```
unset LANG
echo -e "여기는 운영서버 입니다."
```

```
[orcl:~]$ su - oracle
Password:
여기는 운영서버 입니다.
```

su - oracle : 오라클로 접속

문제66. oracle 유저로 접속할때 마다 아래의 메시지가 출력되게 하시오.

보기)

저한테 왜 그러시는 건데요!!

답)

[orcl:~]\$ su - oracle

```
echo -e "여기는 운영서버 입니다. 조심히 작업하세요~"
echo -e "저한테 왜 그러시는 건데요!!"
```

23. sed 명령어 : grep 명령어는 파일의 특정 내용을 검색하는 기능을 갖는다면

sed 명령어는 검색뿐만 아니라 내용을 변경을 할 수 있다.

ex) sed 's/KING/yyy/' emp.txt

설명 : emp.txt를 출력할때 KING을 yyy로 변경해서 출력한다.

```
[orcl:~]$ sed 's/KING/yyy/' emp.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81      1600     300     30
7521 WARD       SALESMAN   7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER    7839 02-APR-81      2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81      1250     1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81      2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87      3000      0      20
7839 yyy        PRESIDENT  0 17-NOV-81      5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81      1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87      1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950       0      30
7902 FORD       ANALYST    7566 03-DEC-81      3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82      1300      0      10
```

여기 화면에서만 바뀐거고, 파일이 변한게 아니다.

문제67. 위의 변경된 내용으로 출력한 내용을 emp9.txt로 저장하시오.

```
[orcl:~]$ sed 's/KING/yyy/' emp.txt >> emp9.txt
```

```
[orcl:~]$ cat emp9.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81      1600     300     30
7521 WARD       SALESMAN   7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER    7839 02-APR-81      2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81      1250     1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81      2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87      3000      0      20
7839 yyy        PRESIDENT  0 17-NOV-81      5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81      1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87      1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950       0      30
7902 FORD       ANALYST    7566 03-DEC-81      3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82      1300      0      10
```

24. cp 명령어 : 파일을 복사하는 명령어(백업할때 사용)

ex) cp 위치/원본 파일명 위치/복사할 파일명

```
cp emp.txt emp400.txt
```

문제68. 리눅스의 oracle 계정을 위한 환경설정 파일인 .bash_profile을 백업하시오.

답)

```
[orcl:~]$ cp .bash_profile .bash_profile2
```

문제69. /home/oracle 밑에 있는 확장자 .txt파일들 전부를 /home/oracle/backup 이라는 폴더를 만들고 거기에 복사 하시오.

답)

```
[orcl:~]$ cp /home/oracle/*.txt /home/oracle/backup/
```

```
[orcl:~]$ cd backup
```

```
[orcl:backup]$ ls -a
```

```
[orcl:backup]$ ls -a
.      a.txt  dept.txt  emp.txt    emp400.txt  emp7.txt  ls.txt    winter.txt
..     c.txt  e.txt    emp40.txt  emp50.txt  emp9.txt  sar_0622.txt  winter_head.txt
```

25. mv명령어 : 파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어

ex) mv 기존 파일명 새로운 파일명
mv emp400.txt emp500.txt <- 이름바꾸기
mv emp.txt ./backup/ <- emp.txt를 backup 폴더로 이동

문제70. 다시 /home/oracle/backup 밑에 있는 emp.txt를 지금 현재 디렉토리인 /home/oracle 밑으로 이동 시키시오.

답)
mv ./backup/emp.txt . <- . : 현재 디렉토리

3. vi 편집기 명령어

- vi 편집기란? 리눅스 안에서 사용할 수 있는 문서 편집기
vi(Visual Editor) 의 뜻에서 유래가 됨.

- vi 편집기의 명령모드 3가지

- 1. command 모드

- : vi의 기본 모드이며 vi를 실행하면 바로 보이는 기본 화면
방향키로 움직일 수 있는 화면

- 2. Edit 모드

- : a, i, o 등을 누르면서 내용을 입력 또는 삭제하는 명령모드

- 3. last line 모드

- : 입력모드에서 저장 종료, 강제 종료등의 명령어를 입력하는 모드

ex) vi aaa.txt

- 편집기를 시작하는 명령어

- 1. vi 명령어 : 읽기, 쓰기가 가능한 상태로 파일이 열린다.
 - 2. view 파일명 : 읽기만 가능한 상태로 파일이 열린다.
 - 3. vi -r 파일명 : 손상된 파일을 복구하는 명령어

vi aaa.txt 들어간 상태에서 ctrl + z로 나오면 이상한 화면이나오는데 없애주는 방법

[orcl:~]\$ ls -al .aaa.txt.swp <-- ls -a : 숨김 파일까지 보여주기

[orcl:~]\$ rm .aaa.txt.swp

- vi 편집기 내에서 커서이동

ex) vi winter.txt

h, j, k, l : 왼, 아래, 위, 오른쪽

G : 화면 맨 아래로 이동

1G : 화면 첫줄로 이동 (가고싶은행)G

: set nu : 파일내의 텍스트에 번호 표시

: set nonu : 번호 안보이게 하기

문제71. 겨울왕국 대본의 1~200줄까지 내용을 winter500.txt라는 이름으로 저장하시오.

vi winter.txt

: 1,200 w winter500.txt <--를 입력

- **vi 편집기의 입력 명령어**

1. i : 커서의 왼쪽부터 입력하겠다
2. a : 커서의 오른쪽부터 입력하겠다.
3. o : 커서의 다음 라인부터 입력하겠다.
4. O : 커서의 윗줄부터 입력하겠다.

- **vi 편집기의 삭제 명령어**

1. x : 철자 하나 삭제
2. dd : 한줄 삭제
3. dw : 커서에 있는 단어 삭제
4. :5,10 d : 5~10번째 줄 삭제
5. D : 커서 오른쪽 행 삭제

- **취소 명령어**

u : 방금 작업한거 취소 (ctrl + z)

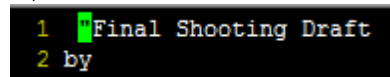
- **복사/붙여넣기 명령어**

1. yy : 하나의 행을 복사
2. p : 붙여넣기
3. yG : 현재행 부터 파일끝까지 복사
4. :1,2 co 3 : 1~2행을 3행 다음으로 복사
5. :1,2 m 3 : 1~2행을 3행 다음으로 이동

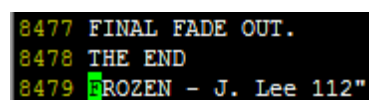
문제72.(마지막문제) 겨울왕국 대본의 전체 라인수를 먼저 확인하고 겨울왕국 대본 전체를 복사해서 아래에 붙여넣고 겨울왕국 대본의 라인수가 2배가 되었는지 확인하시오.

답)

:1,4239 co 4240



```
1 Final Shooting Draft
2 by
```



```
8477 FINAL FADE OUT.
8478 THE END
8479 FROZEN - J. Lee 112"
```

- **문자 변경하는 명령어**

1. ./문자 = 문자 검색
ex) vi winter.txt

:/ELSA 하고 n을누르면 다음으로 검색, shift+n은 이전으로 검색

2. :s/기존문자/변경할 문자 = 지금 커서가 있는 현재행의 기존문자를 변경할 문자로 변경 하겠다.

ex) s/ELSA/anna

3. :%s/기존문자/변경할 문자/g = 문서 전체의 기존문자를 변경할 문자로 변경하겠다.

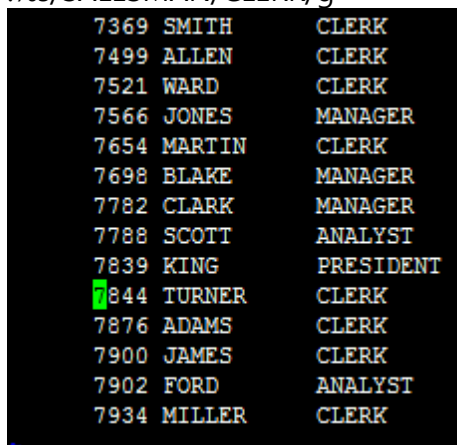
문제73. emp.txt 를 emp750.txt로 복사하고 emp750.txt를 열어서 SALESMAN을 전부다 CLERK으로 변경하시오.

답)

```
cp emp.txt emp750.txt
```

```
vi emp750.txt
```

```
:%s/SALESMAN/CLERK/g
```



7369	SMITH	CLERK
7499	ALLEN	CLERK
7521	WARD	CLERK
7566	JONES	MANAGER
7654	MARTIN	CLERK
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	CLERK
7876	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

문제74. emp.txt를 emp1.txt ~ emp20.txt로 복사하시오.

답)

```
vi cp_emp.sh
```

```
#!/bin/bash
```

```
i=1
```

```
while [ $i -le 20 ] <--공백 처리 잘 해줘야함
```

```
do
```

```
    cp emp.txt emp$i.txt
```

```
    let i=i+1
```

```
done
```

```
[orcl:~]$ sh cp_emp.sh
```

```
[orcl:~]$ ls -l emp*.txt
```

```
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp1.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp10.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp11.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp12.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp13.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp14.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp15.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp16.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp17.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp18.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp19.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp2.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp20.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp3.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp4.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:33 emp40.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 25 15:26 emp400.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp5.txt
-rw-r--r-- 1 oracle oinstall 1940 Jun 22 11:41 emp50.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp6.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp7.txt
-rw-r--r-- 1 oracle oinstall 958 Jun 26 09:58 emp750.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp8.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:03 emp9.txt
```

문제75. emp1.txt ~ emp20.txt의 내용을 변경하는데 ANALYST를 jjj로 변경하시오.

vi emp*.txt

:argdo %s/ANALYST/jjj/g |update

ZZ : vi에서 저장하고 빠져나오기

문제76. winter.txt 대본을 100개로 복사하고 winter.txt ~ winter100.txt로 복사한다.

winter.txt ~ winter100.txt 내용 중에 by를 bybybyby로 변경하시오.

답)

vi cp_winter.sh

```
#!/bin/bash
```

```
i=1
```

```
while [ $i -le 100 ]
```

```
do
```

```
    cp winter.txt winter$i.txt
```

```
    let i=i+1
```

```
done
```

```
sh cp_winter.sh
```

```
:argdo %s/by/bybybyby/g |update
```

- 숫자열 비교

1. 숫자1 -eq 숫자2 : 두 숫자가 같으면 true

2. 숫자1 -ne 숫자2 : 두 숫자가 같지 않으면 true

3. 숫자1 -gt 숫자2 : 숫자1이 숫자2보다 크다면 true
4. 숫자1 -ge 숫자2 : 숫자1이 숫자 2보다 크거나 같다면 true
5. 숫자1 -lt 숫자2 : 숫자1이 숫자 2보다 작다면 true
6. 숫자1 -le 숫자2 : 숫자1이 숫자2보다 작거나 같으면 true
7. !숫자1 : 숫자 1이 거짓이면 true

문제77. winter1.txt ~ winter100.txt 를 전부 삭제하시오.

답)

```
vi rm_winter.sh
```

```
#!/bin/bash
```

```
i=1
```

```
while [ $i -le 100 ]
```

```
do
```

```
rm winter$i.txt
```

```
let i=i+1
```

```
done
```

```
[oracle:~]$ ls -l winter*.txt
```

```
-rw-r--r-- 1 oracle oinstall 229637 Jun 26 10:16 winter.txt
```

```
-rw-r--r-- 1 oracle oinstall 6524 Jun 25 16:39 winter500.txt
```

```
-rw-r--r-- 1 oracle oinstall 6774 Jun 22 14:17 winter_head.txt
```

4. 권한 관리

: 리눅스에서 파이썬 배치파일을 실행하거나 R 배치 파일을 실행하거나 하둡을 설치하고 운영을 할때 여러가지 문제들이 발생하는데 그중에 가장 많은 문제들이 권한에 관련한 오류들이 많다. 그래서 권한관리를 잘 알아둬야한다.

- 권한관리에 대한 명령어 3가지

1. chmod
2. chown
3. chattr

- 권한의 종류 3가지

1. 읽기 권한
2. 쓰기 권한
3. 실행 권한

번호	권한	대표문자	파일	디렉토리
1	읽기권한	r	읽고,copy	디렉토리에서 ls 실행가능
2	쓰기권한	w	수정	디렉토리에서 파일을 생성

3	실행권한	x	실행	디렉토리에서 파일을 실행
---	------	---	----	---------------

- ls -l로 특정 파일을 조회했을때 나오는 권한부분을 해석
\$ ls -l emp.txt

-rw-r--r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt

- : 디렉토리인지 파일인지 나타냄

rw- : 파일의 소유자의 권한

r-- : 그룹의 권한

r-- : 기타유저의 권한

1. chmod

문제78. emp.txt의 소유자의 권한을 읽기만 가능하도록 수정하시오.

[orcl:~]\$ chmod u-w emp.txt

[orcl:~]\$ ls -l emp.txt

```
-r--r--r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

- 참고 사항 : 1. u : 소유자
2. g : 그룹
3. o : 기타 유저

설명 : read only 파일도 ! 붙이고 저장하면 강제 저장 and 종료가 가능하다

문제79. emp.txt의 소유자의 권한에 읽기 권한도 없애시오.

[orcl:~]\$ chmod u-r emp.txt

```
-----r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

문제80. emp.txt의 소유자의 권한에 읽기, 쓰기, 실행 권한을 추가하시오.

[orcl:~]\$ chmod u+rwX emp.txt

```
-rwxr--r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

문제81. emp.txt에 대해서 소유자, 그룹, 기타 유저에 대해서 읽기,쓰기,실행 권한을 모두 추가.

답1)

[orcl:~]\$ chmod 777 emp.txt

```
-rwxrwxrwx 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

답2)

[orcl:~]\$ chmod ugo+rwX emp.txt

```
-rwxrwxrwx 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

답3)

```
[orcl:~]$ chmod a+rwX emp.txt
```

문제82. oracle 유저는 emp.txt에 대해서 무조건 읽기만 가능하게 하고 쓰지는 못하게 하며 그룹과 다른 기타 유저에 대해서는 읽기, 쓰기, 실행 다 못하게 하시오.

답)

```
[orcl:~]$ chmod u-wX,go-rwX emp.txt
```

```
-r----- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

chattr

문제83. oracle 유저가 emp.txt에 대해서 chmod 명령어 자체를 수행 못하게 하시오.

- root 유저로 접속한다(최상위 유저)

su -

패스워드 : oracle

```
[orcl:~]$ su -  
Password:  
[root@edydr1p0 ~]# whoami  
root
```

```
[root@edydr1p0 ~]# cd /home/oracle  
[root@edydr1p0 oracle]# ls -l emp.txt  
-r----- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

chattr +i emp.txt

- 다시 oracle 유저로 접속해서 chmod 명령어를 emp.txt 수행 해본다.

su - oracle

whoami

chmod u+rwX emp.txt

```
[orcl:~]$ chmod u+rwX emp.txt  
chmod: changing permissions of `emp.txt': Operation not permitted
```

다시 풀어준다

su -

[root@edydr1p0 ~]# cd /home/oracle

[root@edydr1p0 oracle]# chattr -i emp.txt

[root@edydr1p0 oracle]# su - oracle

[orcl:~]\$ chmod u+rwX emp.txt

• 권한을 설정하는 숫자

2진법을 생각하면 된다.

1 : 실행(x)

2 : 쓰기(w)

4 : 읽기(r)

ex)
chmod u+rw, g+rw, o+r emp.txt
chmod 764 emp.txt

문제84. emp.txt에 대해서 아래와 같이 권한 설정이 될 수 있도록 하시오.

```
chmod 444 emp.txt  
-r--r--r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

문제85. 겨울 왕국 대본의 권한을 rwx r-- r--으로 변경하시오.

답)
[orcl:~]\$ chmod 744 winter.txt
-rwxr--r-- 1 oracle oinstall 229637 Jun 26 10:16 winter.txt

- 점심시간 문제를 위한 명령어
겨울왕국 대본에 off라는 단어가 몇번 나오는지 카운트 하시오.

```
[orcl:~]$ grep -w -o -i 'off' winter.txt | wc -l  
[orcl:~]$ grep -w -o -i 'off' winter.txt | wc -l  
130
```

문제86. 겨울왕국에는 elsa가 많이나오는지 anna가 많이 나오는지?

```
[orcl:~]$ grep -w -o -i 'elsa' winter.txt | wc -l  
329  
[orcl:~]$ grep -w -o -i 'anna' winter.txt | wc -l  
678
```

문제87. (점심시간)R로 접속해서 elsa가 329개 이고 anna가 678개인지 확인하시오.

답)
R
> length(grep("elsa", winter, ignore.case = TRUE))
[1] 318
> length(grep("anna", winter, ignore.case = TRUE))
[1] 677

2. chown : 파일이나 디렉토리의 소유자를 변경하는 명령어

ex) \$ ls -l emp.txt
[orcl:~]\$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 644 Jun 26 10:22 emp.txt

```
[orcl:~]$ chmod 777 emp.txt  
[orcl:~]$ ls -l emp.txt  
-rwxrwxrwx 1 oracle oinstall 644 Jun 26 10:22 emp.txt  
[orcl:~]$ su -  
Password:  
[root@edydr1p0 ~]# cd /home/oracle  
[root@edydr1p0 oracle]# chown root:root emp.txt  
유저명 : 그룹명  
exit
```

```
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 root root 644 Jun 26 10:22 emp.txt
```

문제88. 그럼 다시 emp.txt의 소유자를 oracle로 변경하고 그룹을 oinstall로 변경하시오.

```
[root@edydr1p0 ~]# cd /home/oracle
[root@edydr1p0 oracle]# chown oracle:oinstall emp.txt
[root@edydr1p0 oracle]# exit
```

```
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 644 Jun 26 10:22 emp.txt
```

문제89. /home/oracle 밑에 bbb라는 디렉토리를 만들고 bbb 디렉토리밑에 /home/oracle 밑에 있는 확장자 txt 파일들을 전부 copy하시오.

```
[orcl:~]$ mkdir bbb
[orcl:~]$ cd bbb
[orcl:bbb]$ cp /home/oracle/*.txt .
```

```
[orcl:bbb]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 26 14:07 a.txt
-rw-r--r-- 1 oracle oinstall 101 Jun 26 14:07 aaa.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 26 14:07 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 26 14:07 dept.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 26 14:07 e.txt
-rwxr-xr-x 1 oracle oinstall 644 Jun 26 14:07 emp.txt
-rw-r--r-- 1 oracle oinstall 2807 Jun 26 14:07 ls.txt
-rw-r--r-- 1 oracle oinstall 76 Jun 26 14:07 rm_emp.txt
-rw-r--r-- 1 oracle oinstall 12364 Jun 26 14:07 sar_0622.txt
-rwxr--r-- 1 oracle oinstall 114548 Jun 26 14:07 winter.txt
-rw-r--r-- 1 oracle oinstall 6524 Jun 26 14:07 winter500.txt
-rw-r--r-- 1 oracle oinstall 6774 Jun 26 14:07 winter_head.txt
```

문제90. /home/oracle 밑에 bbb 디렉토리의 소유자를 root유저로 변경하시오.

보기)

```
[orcl:~]$ ls -ld /home/oracle/bbb
```

```
[orcl:~]$ ls -ld /home/oracle/bbb
drwxr-xr-x 2 oracle oinstall 4096 Jun 26 14:07 /home/oracle/bbb
```

답)

```
[orcl:~]$ su -
```

Password:

```
[root@edydr1p0 ~]# cd /home/oracle
```

```
[root@edydr1p0 oracle]# chown -R root:root /home/oracle/bbb
```

```
[root@edydr1p0 oracle]# ls -ld /home/oracle/bbb
```

```
drwxr-xr-x 2 root root 4096 Jun 26 14:07 /home/oracle/bbb
```

설명 : -R 옵션 = 그 디렉토리 안에 있는 모든 파일들의 소유자를 전부 변경하겠다.
, -R이 없으면 디렉토리 소유자만 변경되고 파일들은 변경 안된다.

문제91. 다시 /home/oracle/bbb 디렉토리의 소유자를 oracle 유저로 변경하고, 그룹도 oinstall로 변경하시오.

답)

```
[root@edydr1p0 oracle]# chown -R oracle:oinstall /home/oracle/bbb
```

```
drwxr-xr-x 2 oracle oinstall 4096 Jun 26 14:07 /home/oracle/bbb
```

5. 디스크 관리

• 디스크 관리 명령어 3가지

1. df 명령어 : 현재 파일 시스템의 총 사용율을 확인하는 명령어

```
[orcl:~]$ df -hP
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol100 190G  45G 136G  25% /
/dev/sda1                99M   18M   76M  20% /boot
tmpfs                   1014M  442M  572M  44% /dev/shm
/dev/hdc                 56M    56M    0 100% /media/VBox_GAs_5.2.12
```

2. du 명령어 : 현재 파일/디렉토리의 디스크 사용량을 표시하는 명령어

ex) du -sh

```
[orcl:~]$ du -sh
583M .
```

```
[orcl:~]$ du -s *.txt
0      a.txt
4      aaa.txt
0      c.txt
4      dept.txt
0      e.txt
4      emp.txt
4      ls.txt
4      rm_emp.txt
16     sar_0622.txt
116    winter.txt
8      winter500.txt
8      winter_head.txt
```

문제92. 위의 txt 파일의 총 사용량 크기를 출력하시오.

답)1

```
[orcl:~]$ du -s *.txt | awk '{sum=sum+$1} END { print sum}'
```

```
168
```

답)2

```
[orcl:~]$ du -c *.txt
```

```
[orcl:~]$ du -c *.txt
0      a.txt
4      aaa.txt
0      c.txt
4      dept.txt
0      e.txt
4      emp.txt
4      ls.txt
4      rm_emp.txt
16     sar_0622.txt
116    winter.txt
8      winter500.txt
8      winter_head.txt
168    total
```


3. sar 명령어 : disk 의 i/o 성능을 모니터링 하는 명령어

ex) [orcl:~]\$ sar 1 50

14:29:49	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:29:50	all	0.00	0.00	1.00	0.00	0.00	99.00
14:29:51	all	0.00	0.00	0.50	0.00	0.00	99.50
14:29:52	all	0.00	0.00	3.48	0.00	0.00	96.52
14:29:53	all	0.00	0.00	3.02	0.00	0.00	96.98
14:29:54	all	0.50	0.00	2.99	0.00	0.00	96.52
14:29:55	all	0.00	0.00	1.01	0.00	0.00	98.99
14:29:56	all	0.00	0.00	1.00	0.00	0.00	99.00
14:29:57	all	0.99	0.00	2.97	0.00	0.00	96.04
14:29:58	all	0.00	0.00	7.54	0.00	0.00	92.46
14:29:59	all	0.50	0.00	1.99	0.00	0.00	97.51

컬럼 소개

%user : oracle 유저와 같은 일반 유저가 사용하는 disk i/o

%nice : cpu를 양보하는 친절도

%system : system을 사용하는 disk i/o

%iowait : i/o를 일으키면서 대기하는 비율

%steal :

%idle : 작업을 안하고 있는 idle한 상태

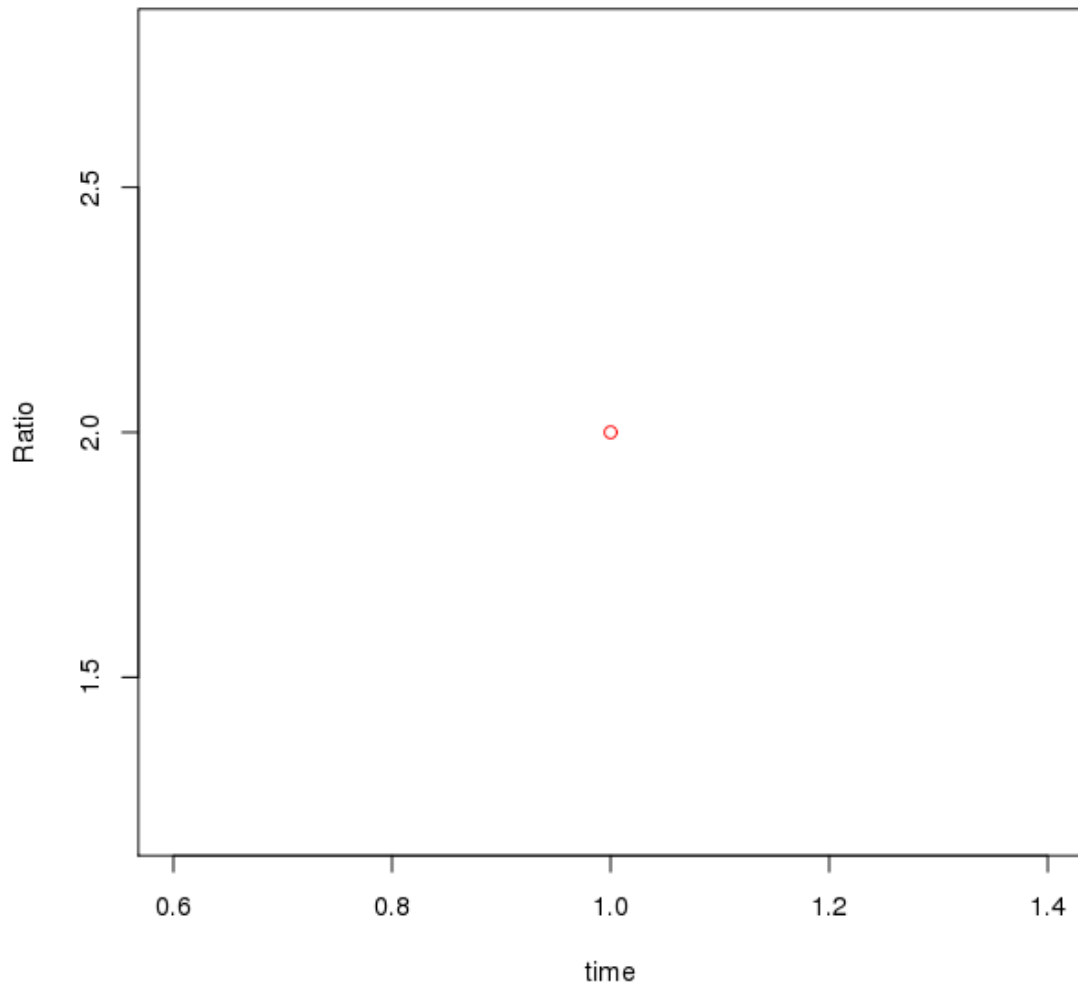
문제93. sar의 user i/o를 R로 시각화 하시오.

1. R로 접속한다.

\$ R

2. 아래의 R코드의 plot 그래프를 확인하시오.

plot(2, type='o', col='red', xlab='time', ylab='Ratio')



3. 리눅스 sar명령어에서 아래와 같이 %user 컬럼 부분만 가져오게 awk명령어를 작성

[orcl:~]\$ sar 1 1 | awk '{print \$3}'

```
%user
0.50
0.50
```

4. R에서 리눅스 명령어를 수행하시오.

> system(" sar 1 1 | awk '{print \$3}'", intern=TRUE)

```
[1] "(edydr1p0.us.oracle.com)" ""
[3] "%user" "0.00"
[5] "0.00"
```

5. 위의 결과를 re라는 변수에 넣고 5번째 요소만 가져오시오.

> re <- system(" sar 1 1 | awk '{print \$3}'", intern=TRUE)

> re[5]

```
> re[5]
[1] "0.50"
```

6. 아래와 같이 NA를 반복하게 하고 m이라는 변수에 담는다.

> m <- rep(NA,10)

> m

```
[1] NA NA NA NA NA NA NA NA NA NA
```

7. re[5]의 값을 m이라는 변수에 첫번째 요소로 담으시오.

```
> m[1] <- re[5]
```

```
> m
```

```
> m
[1] "0.50" NA NA NA NA NA NA NA NA
```

8. 위의 m데이터를 가지로 plot그래프를 그리시오.

```
plot(m, type='o', col='red', xlab='time', ylab='Ratio')
```

9. 실시간으로 그래프가 이어질수 있도록 m의 데이터를 담고 plot 그래프를 그리시오.

```
> m[2] <- re[5]
```

```
> m
```

```
> m[2] <- re[5]
> m
[1] "0.50" "0.50" NA NA NA NA NA NA NA
```

10. 위의 R 코드에 for loop문을 입혀서 user i/o 데이터가 실시간 그래프로 그려지게 하시오.

```
> sar <- function(n) {
```

```
+ m <- rep(NA,n)
```

```
+ for ( i in 1:n) {
```

```
+   re <- system("sar 1 1 | awk '{print $3}'", intern=TRUE)
```

```
+   re <- re[5]
```

```
+   m[i] <- re
```

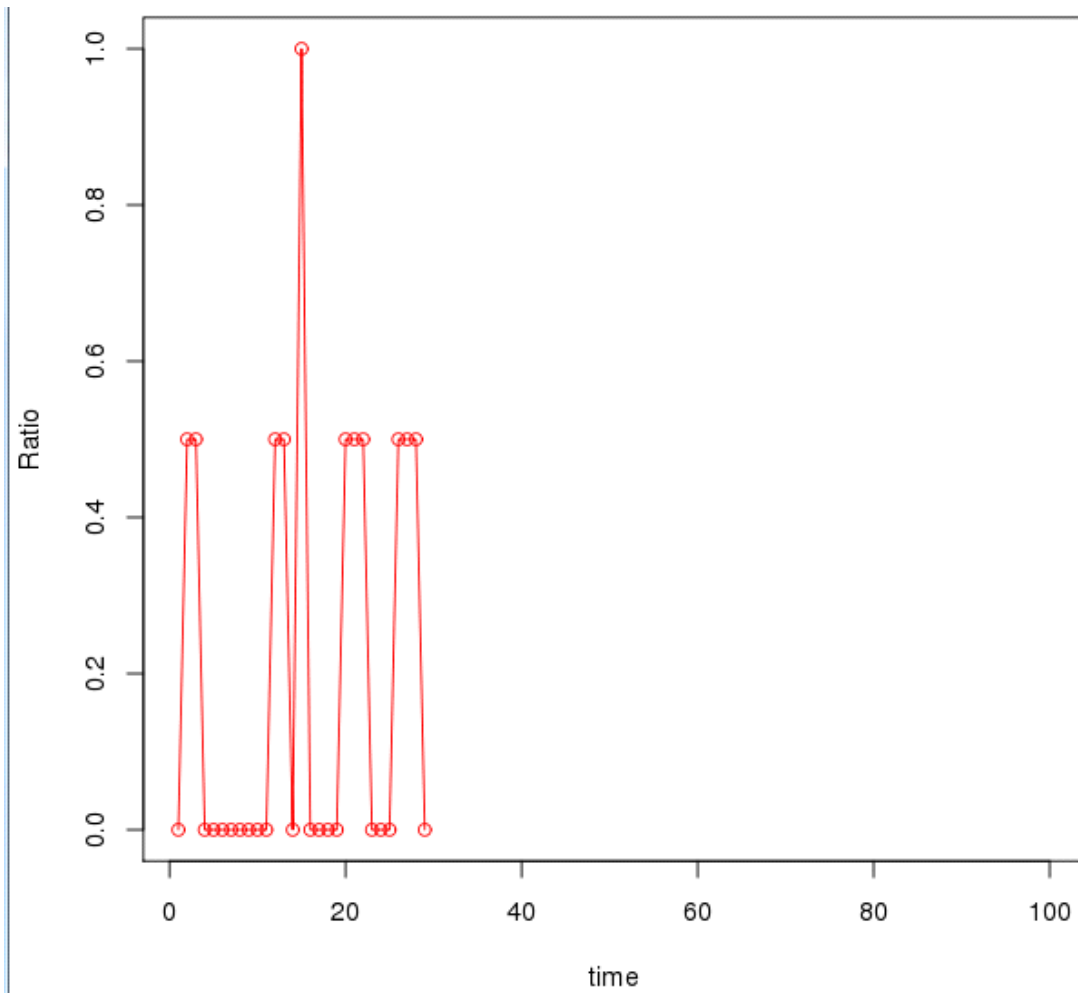
```
+   plot(m, type='o', col='red', xlab='time', ylab='Ratio')
```

```
+ }
```

```
+ rm(m)
```

```
+ }
```

```
> sar(100)
```



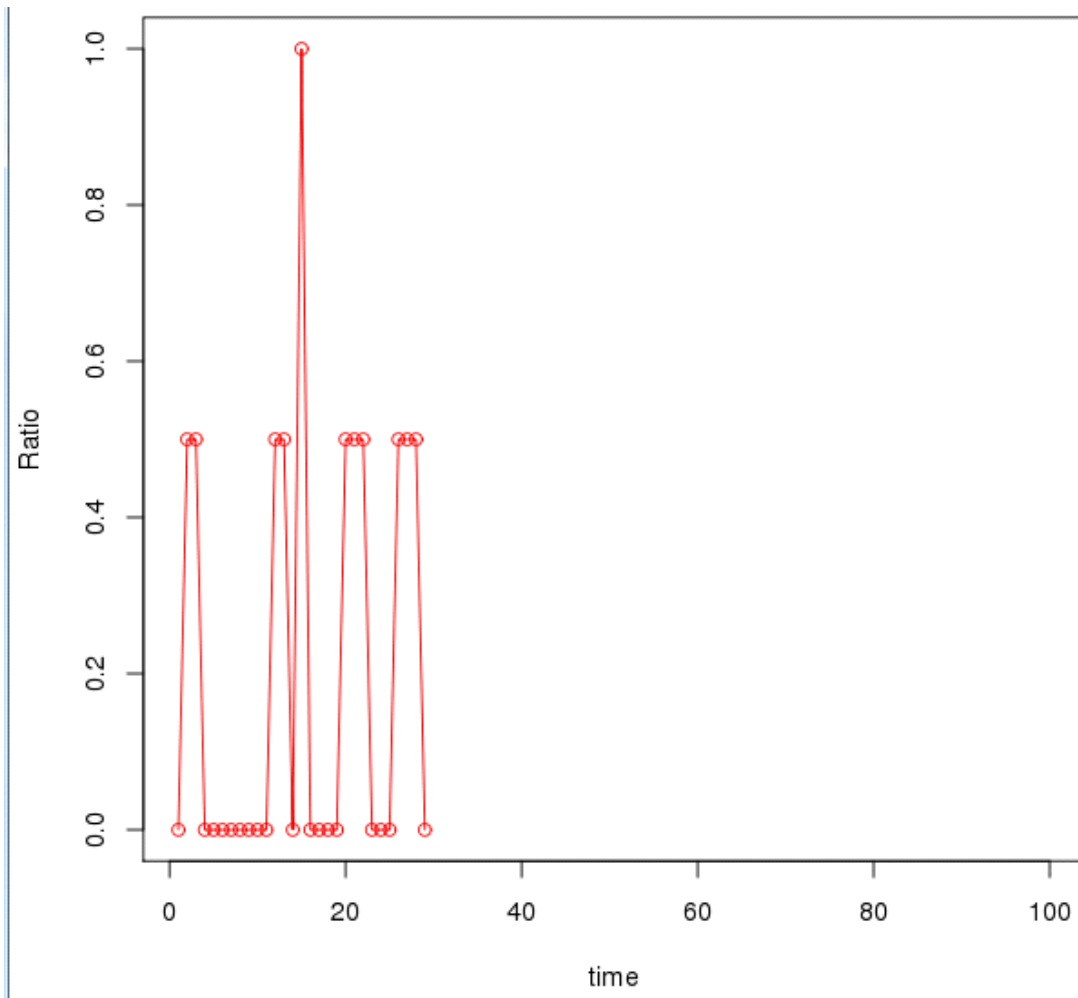
문제94. 위의 스크립트를 리눅스에서 별도로 저장해서 평상시에 쉽게 수행될 수 있도록 하시오.

vi sar.R

```
x11()
sar <- function(n) {
  m <- rep(NA,n)
  for ( i in 1:n) {
    re <- system("sar 1 1 | awk '{print $3}'", intern=TRUE)
    re <- re[5]
    m[i] <- re
    plot(m, type='o', col='red', xlab='time', ylab='Ratio')
  }
  rm(m)
}
sar(100)
```

쓰고

[orcl:~]\$ Rscript sar.R



문제95. 위의 함수를 응용해서 emp.csv의 deptno의 데이터가 실시간으로 plot그래프로 그려지게 하시오.

위의 소스에서 초록색 부분을 변경한다.

```
x11()
sar <- function(n) {
  m <- rep(NA,n)
  emp <- read.csv("emp.csv")

  for ( i in 1:n) {
    re <- emp[i,8]
    m[i] <- re
    Sys.sleep(1)
    graphics.off()
    plot(m, type='o', col='red', xlab='time', ylab='Ratio')
  }
  rm(m)
}
sar(100)
```

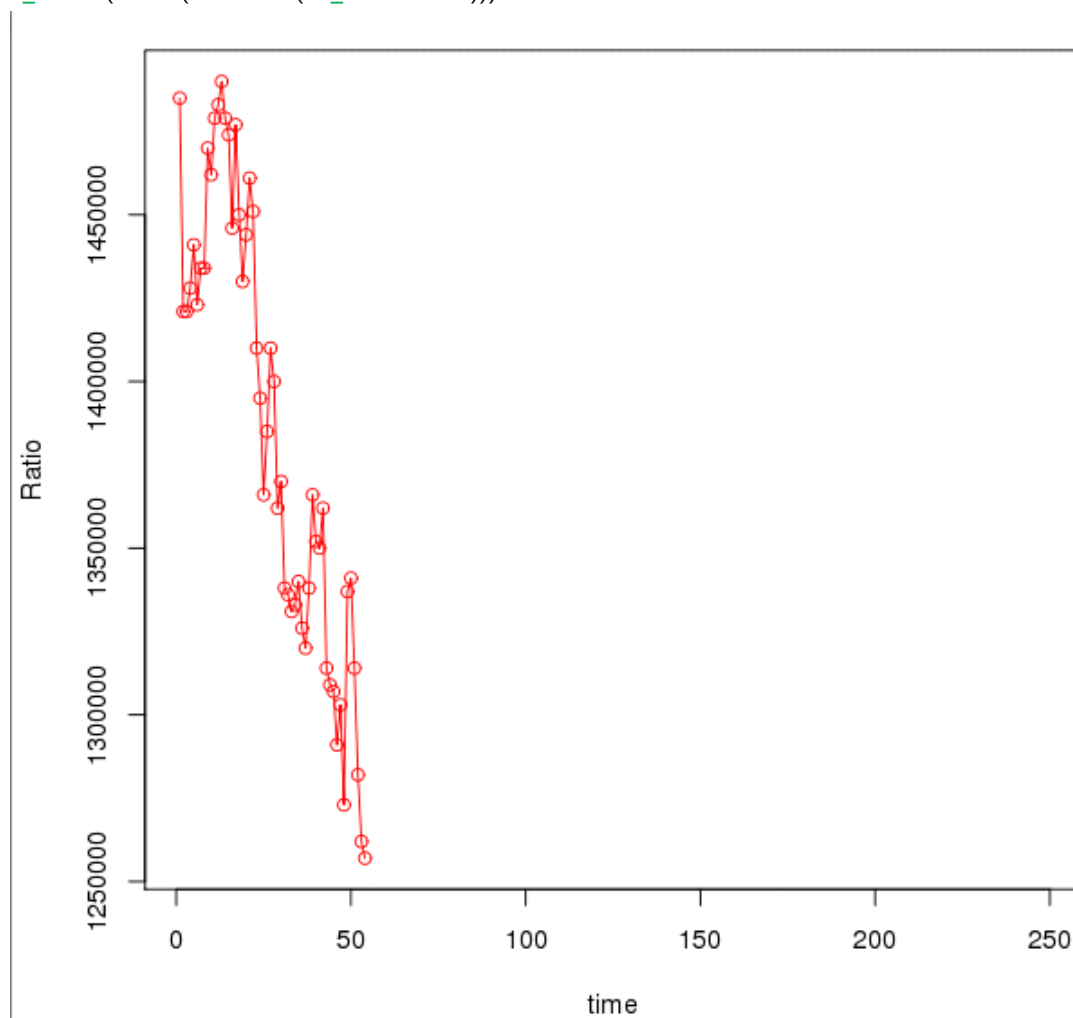
문제96. 삼성 주식 등락율 1년치 데이터를 가지고 실시간 plot 그래프를 그려보시오.

1. /home/oracle 밑에 S_stock.csv 파일을 올린다.

2. S_stock 함수를 생성한다.

x11()

```
s_stock <- function(n) {  
  m <- rep(NA,n)  
  s_stock <- read.csv("S_stock.csv")  
  
  for ( i in 1:n) {  
    re <- s_stock$s_price[i]  
    m[i] <- re  
    Sys.sleep(0.1)  
    plot(m, type='o', col='red', xlab='time', ylab='Ratio')  
  }  
  rm(m)  
}  
s_stock(nrow(read.csv("S_stock.csv")))
```



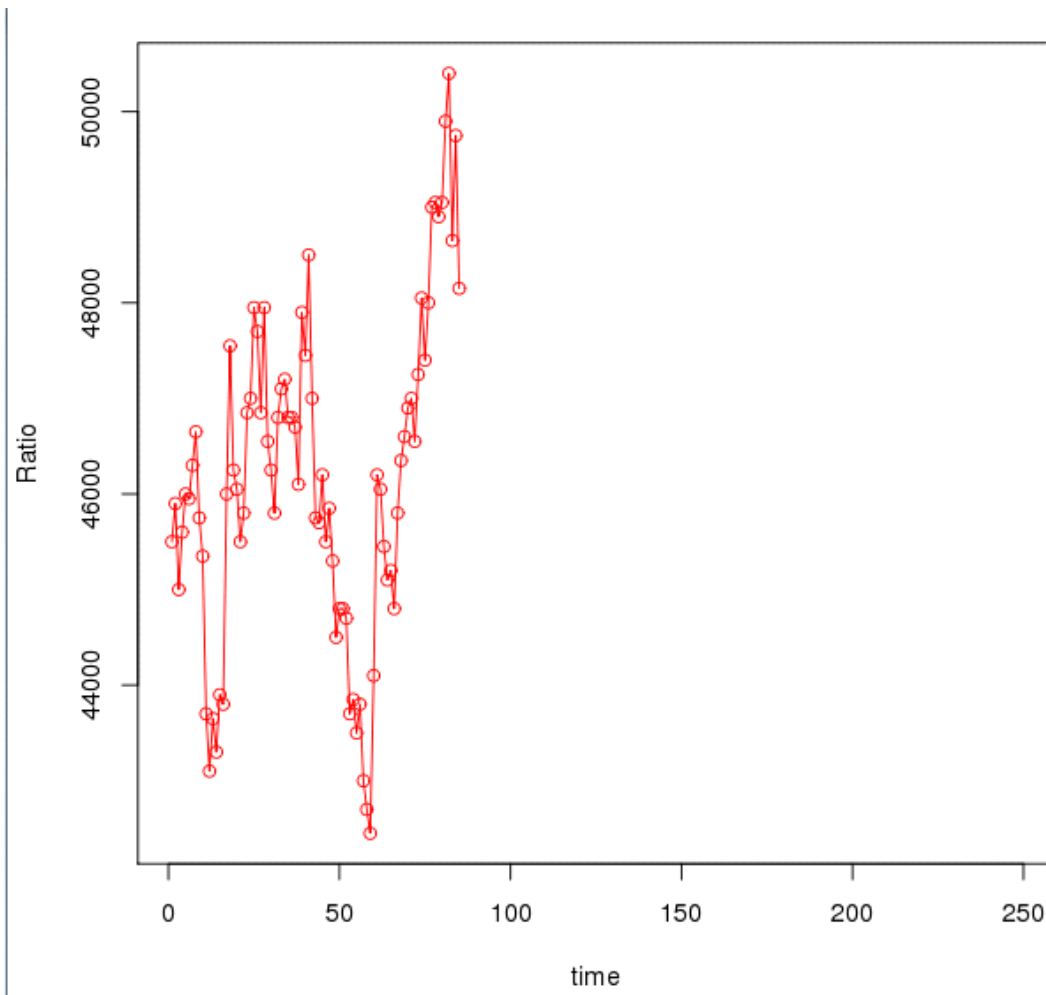
문제97. 현대 자동차의 주식 등락폭을 실시간 그래프로 그려보시오.

```

x11()
h_stock <- function(n) {
  m <- rep(NA,n)
  data <- read.csv("H_stock.csv")

  for ( i in 1:n) {
    re <- data[i,2]
    m[i] <- re
    Sys.sleep(0.1)
    plot(m, type='o', col='red', xlab='time', ylab='Ratio')
  }
  rm(m)
}
h_stock(nrow(read.csv("H_stock")))

```



문제98. 방금 작성한 스크립트를 이용해서 함수를 생성하는데, 아래와 같이 함수를 실행하면 csv파일을 물어보게하고 실시간 그래프가 그려지게 하시오.

보기)

> stock_fun()

입력파일을 로드하시오 ~ H_stock.csv

```

x11()
stock_fun <- function() {

    data <- read.csv(file.choose())
    data_nrow <- nrow(data)
    col <- menu(colnames(data), title = '주가 컬럼을 선택하세요~')

    m <- rep(NA,data_nrow)

    for ( i in 1:data_nrow) {
        re <- data[i,col]
        m[i] <- re
        Sys.sleep(0.1)
        plot(m, type='o', col='red', xlab='time', ylab='Ratio')
    }
    rm(m)

}
stock_fun()

```

문제99. 리눅스에서 Rscript 명령어로 아래와 같이 수해오디어서 실시간 그래프가 수행되게 하시오.

```

보기)
$ sh s.sh
파일을 입력하세요~ S_stock.csv

```

- 리눅스의 변수를 받아서 Rscriptd의 변수에 입력하는 방법

1. s.sh 을 만든다

```
#!/bin/bash
```

```
echo -n " 파일을 입력하세요 ~ "
read filecsv
```

```
Rscript /home/oracle/s.R $filecsv
```

2. s.R을 만든다

```
stock_fun()
```

```
##x11()
```



```
args <- commandArgs()
```

```
print(args[6])
```

```
stock_fun <- function(){
```

```
data <- read.csv(args[6])
```

```
n <- nrow(data)
```

```
m <- rep(NA,n)
```

```
for (i in 1:n){
```

```
  re <- data[i,2]
```

```
  m[i] <- re
```

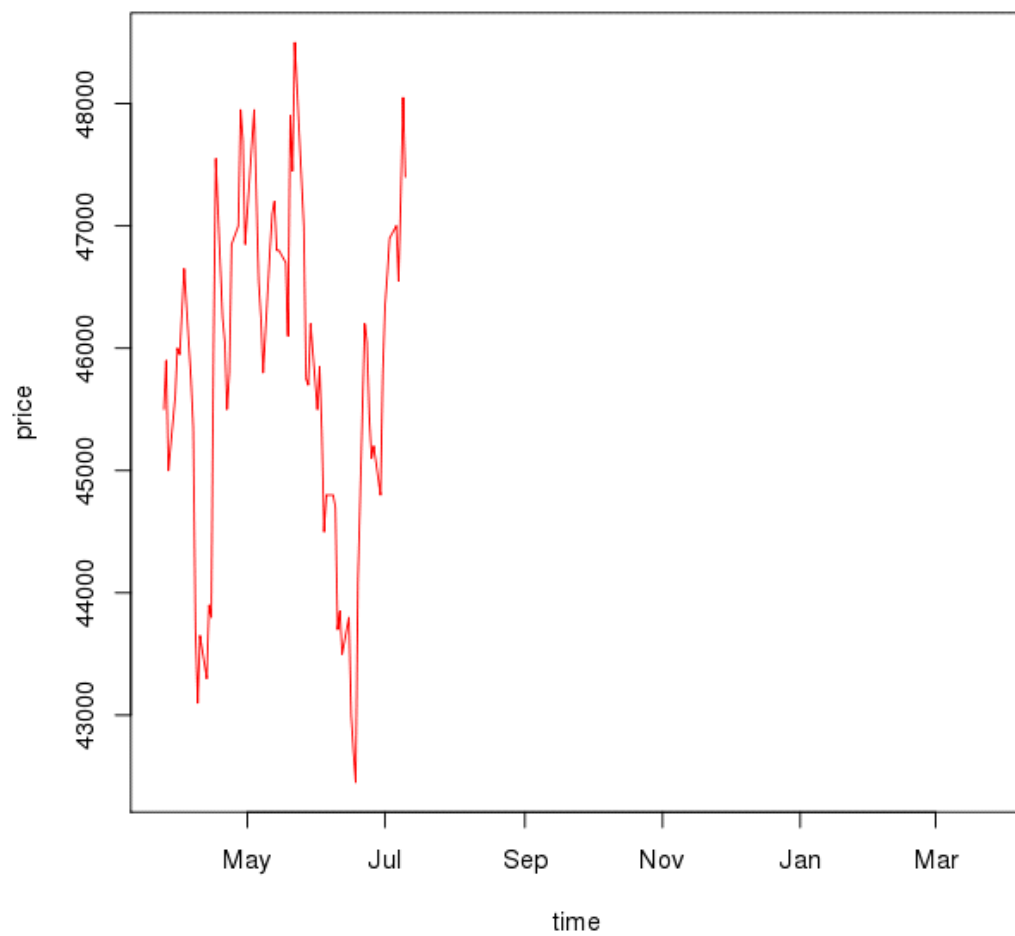
```
  plot(m,x=as.Date(data[,1]), type = 'l',col = 'red', xlab='time', ylab='price')
```

```
}
```

```
rm(m)
```

```
stock_fun()
```

```
[orcl:~]$ sh s.sh
```



6. 프로세서 관리

- 프로세서 관리 명령어

- 1. jobs 명령어

: 동작중인 작업의 상태를 확인하는 명령어

ex) \$ jobs 옵션

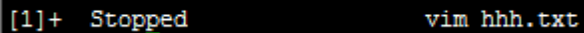
- 상태 정보 4가지
 1. running : 실행중
 2. stopped : 일시중단중
 3. Done : 종료
 4. terminated : 강제종료

vi hhh.txt

```
select ename, sal, job, deptno
from emp
where
```

치고 esc누르고

ctrl + z(작업중지) 를 친다.



[1] : job 번호

+ : 현재 job

Stopped : 일시정지

vim hhh.txt : 동작중인 job

그 다음에

\$ jobs 를 치고

\$ fg(포 그라운드)를 친다

그러면 다시 들어갈 수 있다.

문제100. jobs 명령어를 수행하면 아래와 같이 결과가 출력되게 상황을 만드시오.

보기)

\$jobs

[1] stopped vim hhh.txt

[2] - stopped vim hhh2.txt

[3] + stopped vim hhh3.txt

답)

vi hhh.txt 들어가서 아무거나쓰고 ctrl +z를 반복해준다

문제102. 위의 일시중단된 3가지 job중에서 2번 job으로 fg하려면 어떻게 해야하는가?

답)

fg 2

2. top 명령어

: 지금 현재 작동중인 프로세서들의 cpu 사용율과 메모리 사용율을 확인하는 명령

ex) top

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
24613	oracle	-2	0	612m	13m	11m	S	1.3	0.6	4:12.02	oracle
5016	oracle	18	0	198m	19m	7500	S	1.0	1.0	4:59.09	ocssd.bin
5413	oracle	-2	0	409m	11m	10m	S	1.0	0.6	5:21.03	oracle
4327	oracle	15	0	334m	29m	18m	S	0.7	1.4	5:33.64	ohasd.bin
5082	oracle	15	0	8976	1332	888	S	0.7	0.1	1:41.56	VBoxClient
875	oracle	15	0	10104	1812	1064	S	0.3	0.1	0:19.84	sshd
2067	root	18	0	1976	768	684	S	0.3	0.0	0:23.22	hald-addon-stor
2412	root	18	0	5300	1112	572	S	0.3	0.1	0:00.18	crond
4960	oracle	15	0	400m	23m	11m	S	0.3	1.2	5:42.47	oraagent.bin
5090	oracle	15	0	97884	11m	5672	S	0.3	0.6	1:37.24	diskmon.bin
5162	oracle	15	0	39460	5088	3188	S	0.3	0.2	0:54.19	scim-panel-gtk
5315	oracle	15	0	50552	6564	4952	S	0.3	0.3	0:27.52	gnome-power-man
5411	oracle	15	0	410m	13m	12m	S	0.3	0.7	0:08.12	oracle
5437	oracle	16	0	412m	15m	13m	S	0.3	0.8	0:13.20	oracle
5485	oracle	16	0	410m	14m	13m	S	0.3	0.7	0:08.10	oracle
24625	oracle	18	0	612m	16m	14m	S	0.3	0.8	0:21.65	oracle
24651	oracle	15	0	619m	20m	12m	S	0.3	1.0	0:04.06	oracle
24723	oracle	15	0	616m	46m	40m	S	0.3	2.3	0:10.06	oracle
1	root	15	0	2072	624	532	S	0.0	0.0	0:01.32	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.34	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0

3. ps 명령어

: 현재 시스템에서 수행되고 있는 프로세서의 정보를 표시하는 명령어

ex) \$ps 옵션 프로세서 아이디

옵션 : -e : 현재 실행중인 모든 프로세서

-f : 실제 유저명, 개시 시간등을 표시

-l : 프로세서의 상태, 우선도등과 같은 상세한 정보

-p : 프로세서 아이디

문제 102. 현재 리눅스 시스템에서 실행중인 모든 프로세서를 출력하시오.

ps -ef

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	Jun26	?	00:00:01	init [5]
root	2	1	0	Jun26	?	00:00:00	[migration/0]
root	3	1	0	Jun26	?	00:00:00	[ksoftirqd/0]
root	4	1	0	Jun26	?	00:00:00	[watchdog/0]
root	5	1	0	Jun26	?	00:00:00	[migration/1]
root	6	1	0	Jun26	?	00:00:00	[ksoftirqd/1]
root	7	1	0	Jun26	?	00:00:00	[watchdog/1]
root	8	1	0	Jun26	?	00:00:01	[events/0]
root	9	1	0	Jun26	?	00:00:26	[events/1]
root	10	1	0	Jun26	?	00:00:00	[khelper]
root	11	1	0	Jun26	?	00:00:00	[kthread]
root	15	11	0	Jun26	?	00:00:00	[kblockd/0]
root	16	11	0	Jun26	?	00:00:01	[kblockd/1]
root	17	11	0	Jun26	?	00:00:00	[kacpid]

4. kill 명령어

kill -9 프로세스번호

문제103. 지금 작업을 vim(vi)으로 수행하고 있는 프로세서들을 출력하시오.

답)

복제한 세션에서 vi aaaa2.txt를 들어가서 아무거나 입력하고 원래 세션으로와서
ps -ef | grep vim 을 입력.

```
oracle  8986  8904  0 09:54 pts/3    00:00:00 vim aaaa2.txt
oracle  9013  8759  0 09:55 pts/2    00:00:00 grep vim
```

```
[orcl:~]$ kill -9 8986
```

```
[orcl:~]$ ps -ef | grep vim
oracle  9057  8759  0 09:57 pts/2    00:00:00 grep vim
```

- kill 명령어의 옵션

옵션

-9 : 프로세서를 죽인다.

-6 : 프로세서를 중단시킨다.

-2 : 프로세서에게 종료하도록 지시한다.

문제104. kill 옵션중에 -2옵션을 테스트 하시오.

답)

터미널창1 에서

sar 1 10000

터미널창2 에서

ps -ef | grep sar

kill -2 프로세서 번호

하면 그냥 프로세서가 죽는다...

7. 셸 스크립트 작성법

셸(shell)이란 무엇인가?

: 운영체제에서 제공하는 명령을 실행하는 프로그램이다.

셸(shell) 스크립트란?

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

- 셸의 종류

1. bourne shell
2. C shell
3. korn shell
4. bash shell

ex) #!/bin/bash --> 쉘 줄에 bash 쉘을 쓰겠다.

- 쉘 스크립트 프로그래밍이란?

1. C 언어와 유사하게 프로그래밍이 가능
2. 변수, 반복문(loop문), 제어문(if문)을 사용가능
3. 별도로 컴파일 하지 않고 텍스트 파일형태로 바로 실행이 가능하다.
4. vi나 gedit으로 작성이 가능
5. 리눅스의 많은 부분이 쉘 스크립트로 작성이 되어있다.

- 쉘 스크립트를 작성하고 실행하는 방법

vi a.sh

echo "호스트 이름: " \$HOSTNAME <-- 리눅스 컴퓨터의 이름
입력후 저장하고 나와서

[orcl:~]\$ sh a.sh

```
[orcl:~]$ sh a.sh
호스트 이름: edydrip0.us.oracle.com
```

./a.sh 를치면 실행할 수 가없다(권한이없어서)

```
[orcl:~]$ ./a.sh
-bash: ./a.sh: Permission denied
[orcl:~]$ ls -l a.sh
-rw-r--r-- 1 oracle oinstall 31 Jun 27 10:11 a.sh
```

문제105. a.sh에 소유자가 실행할 수 있도록 실행권한을 추가하시오.

답)

[orcl:~]\$ chmod u+x a.sh

```
[orcl:~]$ chmod u+x a.sh
[orcl:~]$ ls -l a.sh
-rwxr--r-- 1 oracle oinstall 31 Jun 27 10:11 a.sh
```

```
[orcl:~]$ ./a.sh
호스트 이름: edydrip0.us.oracle.com
```

- 변수 사용법

1. 모든 변수는 '문자열(string)'로 취급된다.
2. 변수이름은 대소문자로 구분한다.
3. 변수에 값을 대입할때는 '=' 좌우에 공백이 없어야한다.
4. 변수에 들어간 문자를 출력하려면 변수앞에 \$를 붙이고 echo명령어로 출력한다.

ex) \$ myvar="Hi ~~~"

\$ echo \$myvar

```
[orcl:~]$ myvar="Hi ~~~"
[orcl:~]$ echo $myvar
Hi ~~~
```

- 변수의 숫자 계산하는 방법

1. 변수에 대입한 값은 모두 문자열로 취급이 된다.
2. 변수에 들어있는 값을 숫자로 해서 +, *, /, - 등의 연산을 하려면 expr을 사용해야한다.
3. 수식에 괄호 또는 곱하기(*)를 사용하려면 수식 앞에 꼭 w(역슬래쉬)를 붙여야 한다.

ex)

```
[orcl:~]$ num1=100
```

```
[orcl:~]$ num2=200
```

```
[orcl:~]$ echo $num1
```

```
100
```

```
[orcl:~]$ echo $num2
```

```
200
```

```
[orcl:~]$ echo $num1 + $num2
```

```
[orcl:~]$ echo $num1 + $num2
100 + 200
```

이러한 결과가 나오는데 아래처럼 expr을 붙여주면

```
[orcl:~]$ expr $num1 + $num2
```

```
[orcl:~]$ expr $num1 + $num2
300
```

문제106. num1과 num2의 곱을 구하시오.

```
[orcl:~]$ expr $num1 w* $num2
```

```
[orcl:~]$ expr $num1 \* $num2
20000
```

문제107. 아래의 계산식을 구현하시오.

보기)

$(\$num2 + 200) * \$num1$

답)

```
[orcl:~]$ expr w( $num2 + 200 w) w* $num1
```

```
[orcl:~]$ expr \( $num2 + 200 \) \* $num1
40000
```

- 파라미터 변수

1. 파라미터 변수는 \$0, \$1, \$2 .. 의 형태를 가진다.
2. 전체 파라미터는 \$* 로 표현한다.

ex) vi b.sh

```
echo "실행파일 이름은 $0 입니다"
```

```
echo "첫 번째 파라미터 값은 $1 입니다"
```

```
echo "두 번째 파라미터 값은 $2 입니다"
```

echo "전체 파라미터 값은 \$* 입니다"

입력후

[orcl:~]\$ sh b.sh 10 20

```
[orcl:~]$ sh b.sh 10 20
실행파일 이름은 b.sh 입니다
첫 번째 파라미터 값은 10 입니다
두 번째 파라미터 값은 20 입니다
전체 파라미터 값은 10 20 입니다
```

문제108. 아래와 같이 두 정수를 파라미터 값으로 입력 했을때 나눈 나머지 값이 출력되게 하시오.

보기)

sh f.sh 24 18

나눈 나머지값은 6입니다.

답)

num1=\$1

num2=\$2

num3=`expr \$num1 % \$num2` <-- `(역따옴표)

echo "나눈 나머지값은 \$num3 입니다"

```
[orcl:~]$ sh f.sh 24 18
나눈 나머지값은 6 입니다
```

설명 : 변수 = `리눅스 명령어` <-- 리눅스 명령어에 의해서 수행된 결과가 변수에 입력 되어야 한다면 역따옴표를 사용해야한다.

문제109. 이름이 scott인 사원의 부서번호가 출력되게 하는 쉘을 작성하시오.

답)

vi a2.sh

#!/bin/bash

deptno=`grep -i 'scott' emp.txt | awk '{print \$8}'`

echo "\$deptno 번에서 근무합니다."

문제110. 위의 스크립트를 수정해서 이름을 물어보게 하고 이름을 입력하면 해당 사원의 부서번호가 출력되게 하시오.

보기)

sh a2.sh

이름을 입력하세요 ~ smith

10번에서 근무합니다.

답)

```
#!/bin/bash
```

```
echo -n "이름을 입력하세요~"
```

```
read ename
```

```
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
```

```
echo "$deptno 번에서 근무합니다."
```

문제111. 위의 코드를 수정해서 결과가 아래와 같이 출력되게 하시오.

보기)

이름을 입력하세요 ~ allen

allen은 30번에서 근무합니다.

답)

```
#!/bin/bash
```

```
echo -n "이름을 입력하세요~"
```

```
read ename
```

```
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
```

```
echo "$ename 은 $deptno 번에서 근무합니다."
```

문제112. dept.txt를 /home/oracle에 올리고 부서번호를 물어보게하고 부서번호를 입력하면 해당 부서번호의 부서위치가 출력되게 하시오.

보기)

```
sh a3.sh
```

부서번호를 입력하세요 ~ 20

DALLAS 입니다.

답)

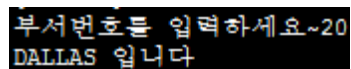
```
#1/bin/bash
```

```
echo -n "부서번호를 입력하세요~"
```

```
read deptno
```

```
loc=`grep -iw $deptno dept.txt | awk '{print $3}'`
```

```
echo "$loc 입니다"
```



```
부서번호를 입력하세요~20
DALLAS 입니다
```

문제113. 이름을 물어보게하고 이름을 입력하면 해당 사원의 부서위치가 출력되게 하시오.

보기)

```
sh a99.sh
```

이름을 입력하세요 ~~ scott

scott은 DALLAS에서 근무합니다.

답)

```
#1/bin/bash
```

```
echo -n "이름을 입력하세요~"
```

```
read ename
```

```
deptno=`grep -i $ename emp.txt | awk '{print $8}`
```

```
loc=`grep -w $deptno dept.txt | awk '{print $3}`
```

```
echo "$ename은 $loc 에서 근무합니다."
```

문제114. 직업을 물어보게하고 직업을 입력하면 해당 직업인 직원들의 월급이 출력되게 하시오.

보기)

```
sh job_sal.sh
```

```
직업을 입력하세요~ salesman
```

```
1600
```

```
1250
```

```
1250
```

```
1500
```

답)

```
#1/bin/bash
```

```
echo -n "직업을 입력하세요"
```

```
read job
```

```
sal=`grep -i $job emp.txt | awk '{print $6}`
```

```
echo "$sal"
```

문제115. 직업을 물어보게 하고 직업을 입력하면 해당 직업의 토털월급이 출력되게 하시오.

보기)

```
sh job_sumsal.sh
```

```
직업을 입력하세요~ salesman
```

```
29020입니다.
```

답)

```
#!/bin/home
```

```
echo " "
```

```
echo -n "input jobs "
```

```
read job
```

```
echo " "
```

```
grep -i $job emp.txt | awk '{sum += $6} END {print "해당 직업의 토털 월급은", sum, "입니다"}'
```

```
[orcl:~]$ sh job_sumsal.sh
input jobs salesman
해당 직업의 토탈 월급은 5600 입니다
```

문제116. 직업을 물어보게하고 해당 직업의 평균 월급이 출력되게 하시오.

보기)

sh job_avsal.sh

직업을 입력하세요~ salesman

평균 월급은 xxxx입니다.

답)

#!/bin/bash

echo -n "직업을 입력하시오"

read job

sal=`grep -iw \$job emp.txt | awk '{sum += \$6} END {print sum}'`

count=`grep -w -i \$job emp.txt | wc -l`

result=`expr \$sal / \$count`

echo "\$job의 평균월급은 \$result입니다."

```
[orcl:~]$ sh job_avsal.sh
직업을 입력하시오salesman
salesman의 평균월급은 1400입니다.
```

문제117. 컬럼번호를 물어보게하고 컬럼번호를 입력하면 해당 컬럼의 데이터의 토탈값과 평균값이 아래와 같이 출력되게 하시오.

보기)

sh total.sh

컬럼번호를 입력하세요 ~ 6

토탈값 : xxxx

평균값 : xxxx

답)

#!/bin/home

echo " "

echo -n "컬럼번호를 입력하세요~ "

read colnum

total=`awk '{sum += '\$colnum'} END {print sum}' emp.txt`

count=`awk '{print \$colnum}' emp.txt | wc -l`

avg=`expr \$total / \$count`

echo "토탈값 : \$total"

echo "평균값 : \$avg"

- 리눅스 셸에서 if문 사용방법

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

% 주의사항 : 공백 처리를 잘해줘야함!!!

- if 문 조건문에 들어가는 비교 연산자

-문자열 비교

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

-숫자열 비교

1. 숫자1 -eq 숫자2 : 두 숫자가 같으면 true
2. 숫자1 -ne 숫자2 : 두 숫자가 같지 않으면 true
3. 숫자1 -gt 숫자2 : 숫자1이 숫자2보다 크다면 true
4. 숫자1 -ge 숫자2 : 숫자1이 숫자 2보다 크거나 같다면 true
5. 숫자1 -lt 숫자2 : 숫자1이 숫자 2보다 작다면 true
6. 숫자1 -le 숫자2 : 숫자1이 숫자2보다 작거나 같으면 true
7. !숫자1 : 숫자 1이 거짓이면 true

ex) vi if1.sh

```
#!/bin/bash
if [ 100 -eq 200 ]; then
    echo "100과 200은 같다"
else
    echo "100과 200은 다르다"
fi
```

```
[orcl:~]$ sh if1.sh
100과 200은 다르다
```

문제118. 위의 스크립트를 활용해서 아래 처럼 숫자를 물어보게하고, 아래와 같이 결과가 출력되게 하시오.

보기)

\$ sh if2.sh

첫 번째 숫자를 입력하세요~ 200

두 번째 숫자를 입력하세요~ 250

250은 200보다 큽니다.

답)

```
#!/bin/bash
```

```
echo -n "첫번째 숫자를 입력하세요~ "
```

```
read num1
```

```
echo -n "두번째 숫자를 입력하세요~ "
```

```
read num2
```

```
if [ $num1 -gt $num2 ]; then
```

```
    echo "$num1이 $num2보다 큽니다"
```

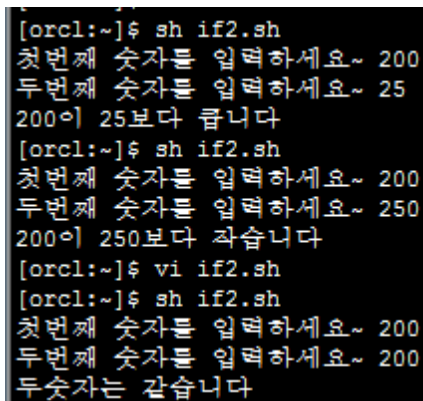
```
elif [ $num1 -lt $num2 ]; then
```

```
    echo "$num1이 $num2보다 작습니다"
```

```
else
```

```
    echo "두숫자는 같습니다"
```

```
fi
```



```
[orcl:~]$ sh if2.sh
첫번째 숫자를 입력하세요~ 200
두번째 숫자를 입력하세요~ 25
200이 25보다 큽니다
[orcl:~]$ sh if2.sh
첫번째 숫자를 입력하세요~ 200
두번째 숫자를 입력하세요~ 250
200이 250보다 작습니다
[orcl:~]$ vi if2.sh
[orcl:~]$ sh if2.sh
첫번째 숫자를 입력하세요~ 200
두번째 숫자를 입력하세요~ 200
두숫자는 같습니다
```

- 리눅스에서 and와 or 연산자
 1. and 는 && 또는 -a를 사용한다.
 2. or 는 || 또는 -o를 사용한다.

```
ex) if [ $sal -lt 2000 ] && [ $job="SALESMAN" ]; then
```

```
    또는
```

```
    if [ $sal -lt 2000 -a $job="SALESMAN" ]; then
```

문제119. 사원 이름을 물어보게하고, 사원이름을 입력했을때 해당 사원의 부서번호가 30번 이고 월급이 2000보다 작다면 월급 인상 대상자 입니다. 라는 메시지가 출력되게 하고, 그렇지 않다면 월급 인상 대상자가 아닙니다. 라는 메시지 출력되게 하시오.

보기)

```
sh a199.sh
```

```
이름을 입력하세요~ scott
```

월급 인상 대상자가 아닙니다.

답)

```
echo -n "이름을 입력하세요~ "
```

```
read ename
```

```
sal=`grep -i $ename emp.txt | awk '{print $6}'`
```

```
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
```

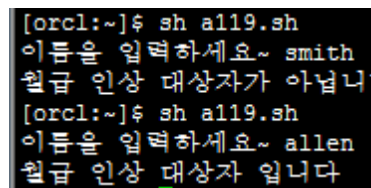
```
if [ $sal -lt 2000 -a $deptno -eq 30 ]; then
```

```
    echo "월급 인상 대상자 입니다"
```

```
else
```

```
    echo "월급 인상 대상자가 아닙니다"
```

```
fi
```



```
[orcl:~]$ sh a119.sh
이름을 입력하세요~ smith
월급 인상 대상자가 아닙니다
[orcl:~]$ sh a119.sh
이름을 입력하세요~ allen
월급 인상 대상자 입니다
```

- **loop 문**

for loop 문 문법

```
for 변수 in 값1,값2,값3
```

```
do
```

```
    반복할 문장
```

```
done
```

ex)

```
#!/bin/bash
```

```
hap=0
```

```
for i in {1..10}
```

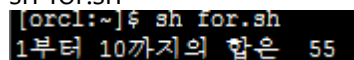
```
do
```

```
    hap=`expr $hap + $i`
```

```
done
```

```
    echo "1부터 10까지의 합은 " $hap
```

```
sh for.sh
```



```
[orcl:~]$ sh for.sh
1부터 10까지의 합은 55
```

문제120. 위의 코드를 수정해서 구구단 2단을 출력하시오.

보기)

```
2 x 1 = 2
```

```
2 x 2 = 4
```

```
.
```

2 x 9 = 18

답1)

```
#!/bin/bash
```

```
num1=2
```

```
for i in {1..9}
do
  gop=`expr $num1 \* $i`
  echo "      "
  echo " | 2 x $i = " $gop |
  echo "      "
done
```

답2)

```
for i in {1..9}
do
  let hap=2*i
  echo "2 x " $i " = " $hap
done
```

설명 : expr 대신에 let을 쓰면 역슬래시를 안써도 된다.

문제121. 구구단 2단부터 9단까지 출력하시오.

```
#!/bin/bash
```

```
for i in {2..9}
do
  for j in {1..9}
  do
    let hap=i*j
    echo $i " x " $j " = " $hap
  done
  echo "-----"
done
```

```

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

```

```

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27

```

```

8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72

```

```

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81

```

문제122. 아래의 별표를 출력하시오.

보기) sh star.sh

```

★
★★
★★★
★★★★
★★★★★

```

답)

```

star=""
for i in {1..5}
do
    star="$star★"
    echo $star
done

```

done

문제123. 위의 코드를 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오.

보기)

sh star.sh

숫자를 입력하세요~ 6

★

★★

★★★

★★★★

★★★★★

★★★★★★

답)

#!/bin/bash

echo -n "숫자를 입력하세요"

read num1

star=""

for i in `eval echo {1..\$num1}`

do

star="\$star★"

echo \$star

done

설명 : eval 명령어는 문자열을 명령문으로 인식하고 실행하는 명령어

ex) \$ str="ls -l emp.txt"

\$ echo \$str

```
[orcl:~]$ str="ls -l emp.txt"
[orcl:~]$ echo $str
ls -l emp.txt
```

\$ eval \$str

```
[orcl:~]$ eval $str
-rwxrwxrwx 1 oracle oinstall 970 Jun 27 11:50 emp.txt
```

문제124. 숫자를 입력하면 아래와 같이 ★가 출력되게 하시오.

보기)

\$ sh star.sh

숫자를 입력하세요~ 5

★★★★★


```
★★★★
★★★
★★
★
```

답)

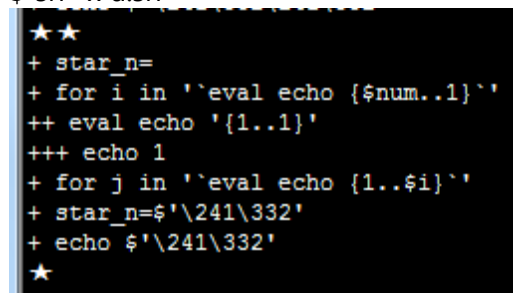
```
#!/bin/bash
```

```
echo -n "숫자를 입력하세요"
read num
```

```
star="★"
star_n=""
for i in `eval echo {$num..1}`
do
  for j in `eval echo {1..$i}`
  do
    star_n="$star_n"$star
  done
  echo "$star_n"
  star_n="" <- 초기화!!!
done
```

문제125. 위의 스크립트를 디버깅하시오.

```
$ vi a.sh
$ sh -x a.sh
```



```
★★
+ star_n=
+ for i in `eval echo {$num..1}`
++ eval echo '{1..1}'
+++ echo 1
+ for j in `eval echo {1..$i}`
+ star_n=$'\241\332'
+ echo $'\241\332'
★
```

문제126. emp.txt를 for loop문을 이용해서 emp1.txt ~ emp100.txt로 복사하시오~

```
#!/bin/bash
```

```
for i in {1..100}
do
  cp emp.txt emp$i.txt
done
```

문제127. 짝공하고 자리를 바꿔서 emp1.txt ~ emp100.txt 중 하나의 파일을 랜덤으로 골라서 데이터를 변경하고 자리로 돌아오시오.

문제128. emp3.txt를 열어서 3000을 3900으로 변경하시오.

답)
\$ vi emp3.txt
:%s/3000/3900/g

문제129. emp.txt와 emp3.txt파일 간의 데이터의 차이가 있는지 확인하시오.

```
$ diff emp.txt emp3.txt
[orcl:~]$ diff emp.txt emp3.txt
8c8
<      7788 SCOTT      ANALYST      7566 19-APR-87      3000      0      20
---
>      7788 SCOTT      ANALYST      7566 19-APR-87      3900      0      20
13c13
<      7902 FORD      ANALYST      7566 03-DEC-81      3000      0      20
---
>      7902 FORD      ANALYST      7566 03-DEC-81      3900      0      20
```

```
$ diff --brief emp.txt emp3.txt
[orcl:~]$ diff --brief emp.txt emp3.txt
Files emp.txt and emp3.txt differ
```

문제130. 짝공이 어떠한 .txt 파일을 변경했는지 한번에 알아내시오.

```
#!/bin/bash

for i in {1..100}
do
diff --brief emp.txt emp$i.txt
done
Files emp.txt and emp3.txt differ
Files emp.txt and emp77.txt differ
```

- sed 명령어로 공백라인을 지우는 방법(데이터 정제 작업)

1. emp.txt를 열어서 o를 사용해서 중간에 공백라인을 하나 넣으시오.

```
7654 MARTIN      SALESMAN      7698 28-SEP-81      1250      1400      30
7698 BLAKE      MANAGER      7839 01-MAY-81      2850      0      30
```

2. emp.txt 안에 공백라인이 있는지 검색한다.

```
$ sed -n /^$/p emp.txt | wc -l    <--p : print
```

3. emp.txt 안에 공백라인을 삭제한다.

```
$ sed '/^$/d' emp.txt >> emp_backup.txt    <-- d : delete
$ rm emp.txt
$ mv emp_backup.txt emp.txt
```

```
sed -n /^$/p emp.txt | wc -l
[orcl:~]$ sed -n /^$/p emp.txt | wc -l
0
```

- 엔터로 넣은 공백까지 다 지우는 방법

```
$ sed '/^ *$/d' emp.txt
```

```
sed -n /^$/p emp.txt | wc -l
[orcl:~]$ sed -n /^$/p emp.txt | wc -l
13
```

```
$ sed '/^ *$/d' emp.txt >> emp_backup.txt
$ rm emp.txt
$ mv emp_backup.txt emp.txt
```

```
[orcl:~]$ sed -n /^$/p emp.txt | wc -l
0
```

문제131. 짝공과 자리를 바꿔서 emp1.txt ~ emp100.txt 중 몇 개를 열어서 공백을 막 넣어주고 자리로 돌아오시오~

문제132. 자리로 돌아와서 짝공이 어떠한 파일에 공백을 넣었는지 확인하시오.

```
답1)
for ((i=1;i<=100;i++))
do
    diff --brief emp.txt emp$i.txt
done
```

```
답2)
for ((i=1; i<=100; i++))
do
    a=`sed -n '/^ $/p' emp$i.txt | wc -l`
    if [ $a -ne 0 ]; then
        echo emp$i
    fi
done
```

문제133. 짝공이 공백을 입력한 파일들의 공백을 지우시오.

```
답)
for in (1..100)
do
    sed -n '/^ $/d' emp$i.txt >> emp_backup$i.txt
    rm emp$i.txt
```

```
mv emp_backup$i.txt emp$i.txt
done
```

```
[orcl:~]$ sh find_gongback.sh
find_gongback.sh: line 1: syntax error near unexpected token `('
find_gongback.sh: line 1: `for in (1..100)\'
[orcl:~]$ sed -n /^$/p emp.txt | wc -l
0
```

문제134. (마지막문제, 알고리즘) bc카드 공채문제 아래의 쉘을 생성하시오.

보기)

```
$ sh factorial.sh 5
120
```

답)

```
num=$1
```

```
gop=1
```

```
for ((i=1;i<=$1;i++))
```

```
do
```

```
    gop=`expr $gop W* $i`
```

```
done
```

```
echo "$gop"
```

```
[orcl:~]$ sh factorial.sh 5
120
```

문제135. (알고리즘) 아래의 power함수를 쉘로 구현하시오.

보기)

```
select power(2,3) from dual;
```

```
$ sh power.sh 2 3
```

답)

문제136. (알고리즘) 최대 공약수를 출력하시오.

보기)

```
$ sh maxyaksu.sh 16 24
```

답)

```
#!/bin/bash
```

```
num1=$1
```

```
num2=$2
```

```

let mod=$num1%$num2
for ((i=$mod;i>0;))
do
    num1=$num2
    num2=$i
    i=`expr $num1 % $num2`
done
echo $num2

```

```

[orcl:~]$ sh maxyaksu.sh 16 24
8

```

문제137. 직업을 중복제거해서 출력하시오.

답)

```

awk '{print $3}' emp.txt | sort | uniq

```

```

ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN

```

문제138. 위의 결과를 변수에 담아서 for loop문으로 변수의 내용을 출력하시오.

답)

```

job=`awk '{print $3}' emp.txt | sort | uniq`
for i in $job
do
    echo $i
done

```

```

ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN

```

문제139. 직업이 SALESMAN인 직원들의 토탈월급을 출력하시오.

답)

```

[orcl:~]$ grep -i 'salesman' emp.txt | awk '{sum += $6} END {print sum}'
5600

```

문제140. 직업과 직업별 토탈월급을 출력하시오.

보기)

```

$sh 140.sh

```

```

#!/bin/bash

```

```

job=`awk '{print $3}' emp.txt | sort | uniq`

```

```
for i in $job
do
    grep -i $i emp.txt | awk '{sum += $6} END {print $3 " " sum}'
done
```

```
ANALYST 6000
CLERK 4150
MANAGER 8275
PRESIDENT 5000
SALESMAN 5600
```

문제141. 부서번호, 부서번호별 토탈월급, 평균월급을 출력하시오.

답)

```
#!/bin/bash
```

```
deptno=`awk '{print $8}' emp.txt | sort | uniq`
```

```
for i in $deptno
```

```
do
```

```
totalsal=`grep -i $i emp.txt | awk '{sum += $6} END {print sum}'`
```

```
count=`grep -iw $i emp.txt | wc -l`
```

```
result=`expr $totalsal / $count`
```

```
echo $i $result
```

```
done
```

```
10 3283
20 2079
30 2783
```

%%LG 전자가 고객이 사수에게 던진 질문

1. 라이나 생명에서 어떠한 분석을 했는가?
2. 데이터가 많았을때 R로 어떻게 분석을 했는가?
 - 2-1. 하둡, 오라클, 쉘에서 데이터를 필터링을 하고 40대만 따로 분리해서 별도의 test 파일을 생성
 - 2-2. 40대의 데이터를 가지고 군집분석 (k-means)
 - 2-3. 40대의 데이터를 가지고 연관분석 (아프리오리)

문제142. 직업이 salesman인 직원들의 데이터만 추출해서 SALESMAN.txt로 생성하시오.

답)

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

```
[orcl:~]$ cat salesman.txt
7499 ALLEN      SALESMAN      7698 20-FEB-81      1600      300      30
7521 WARD       SALESMAN      7698 22-FEB-81      1250      500      30
7654 MARTIN    SALESMAN      7698 28-SEP-81      1250     1400      30
7844 TURNER    SALESMAN      7698 08-SEP-81      1500       0      30
```

문제143. (점심시간)emp.txt를 직업별로 데이터를 분리해서 아래와 같이 텍스트 파일들이 생성

되게 하시오.

보기)

```
$ sh 143.sh
```

```
$ ls *.txt
```

SALESMAN.txt

ANALYST.txt

CLERK.txt

MANAGER.txt

PRESIDENT.txt

와 같이 한번에 생성되게 하시오.

답)

```
#!/bin/bash
```

```
job=`awk '{print $3}' emp.txt | sort | uniq`
```

```
for i in $job
```

```
do
```

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

```
done
```

```
ANALYST.txt
CLERK.txt
MANAGER.txt
PRESIDENT.txt
SALESMAN.txt
```

• 데이터를 db에서 추출하는 방법 2가지

1. dump file

```
$ sqlplus scott/tiger
```

```
SQL> select count(*) from emp;
```

```
SQL> exit;
```

```
[orcl:~]$ exp scott/tiger file=emp.dmp tables=emp
```

```
About to export specified tables via Conventional Path ..
. . exporting table                                EMP      14 rows exported
```

```
ls -l emp.dmp
```

```
orcl:~]$ ls -l emp.dmp
```

```
rw-r--r-- 1 oracle oinstall 16384 Jun 28 13:59 emp.dmp
```

```
orcl:~]$ █
```

dmp파일로 살려내는 방법

```
[orcl:~]$ sqlplus scott/tiger
```

```
SQL> drop table emp;
```

```
SQL> drop table emp;
```

```
Table dropped.
```

다시 db에 덤프 파일을 넣는 작업

```
$ imp scott/tiger file=emp.dmp tables=emp
```

```
[orcl:~]$ sqlplus scott/tiger
```

```
SQL> select count(*) from emp;
```

```

COUNT(*)
-----
         14
```

2. csv 또는 text file

```
$ sqlplus scott/tiger
```

```
SQL> set colsep ','
```

```
SQL> set pagesize 0
```

```
SQL> set linesize 120
```

```
SQL> spool emp222.csv      <-- spool 을쓰면 off 할때까지 여기에 저장된다.
```

```
SQL> select * from emp;
```

7369,SMITH	,CLERK	,7902,17-DEC-80,	800,	,	20
7499,ALLEN	,SALESMAN	,7698,20-FEB-81,	1600,	300,	30
7521,WARD	,SALESMAN	,7698,22-FEB-81,	1250,	500,	30
7566,JONES	,MANAGER	,7839,02-APR-81,	2975,	,	20
7654,MARTIN	,SALESMAN	,7698,28-SEP-81,	1250,	1400,	30
7698,BLAKE	,MANAGER	,7839,01-MAY-81,	2850,	,	30
7782,CLARK	,MANAGER	,7839,09-JUN-81,	2450,	,	10
7788,SCOTT	,ANALYST	,7566,19-APR-87,	3000,	,	20
7839,KING	,PRESIDENT	,17-NOV-81,	5000,	,	10
7844,TURNER	,SALESMAN	,7698,08-SEP-81,	1500,	0,	30
7876,ADAMS	,CLERK	,7788,23-MAY-87,	1100,	,	20
7900,JAMES	,CLERK	,7698,03-DEC-81,	950,	,	30
7902,FORD	,ANALYST	,7566,03-DEC-81,	3000,	,	20
7934,MILLER	,CLERK	,7782,23-JAN-82,	1300,	,	10

```
SQL> spool off
```

```
SQL> exit;
```

```
[orcl:~]$ cat emp222.csv
```

```
[orcl:~]$ cat emp222.csv
```

```
SQL> select * from emp;
```

7369,SMITH	,CLERK	,7902,17-DEC-80,	800,	,	20
7499,ALLEN	,SALESMAN	,7698,20-FEB-81,	1600,	300,	30
7521,WARD	,SALESMAN	,7698,22-FEB-81,	1250,	500,	30
7566,JONES	,MANAGER	,7839,02-APR-81,	2975,	,	20
7654,MARTIN	,SALESMAN	,7698,28-SEP-81,	1250,	1400,	30
7698,BLAKE	,MANAGER	,7839,01-MAY-81,	2850,	,	30
7782,CLARK	,MANAGER	,7839,09-JUN-81,	2450,	,	10
7788,SCOTT	,ANALYST	,7566,19-APR-87,	3000,	,	20
7839,KING	,PRESIDENT	,17-NOV-81,	5000,	,	10
7844,TURNER	,SALESMAN	,7698,08-SEP-81,	1500,	0,	30
7876,ADAMS	,CLERK	,7788,23-MAY-87,	1100,	,	20
7900,JAMES	,CLERK	,7698,03-DEC-81,	950,	,	30
7902,FORD	,ANALYST	,7566,03-DEC-81,	3000,	,	20
7934,MILLER	,CLERK	,7782,23-JAN-82,	1300,	,	10

위의 빨간 부분 없애는 방법

sql 다시 접속해서 show all을 한다.

```
sqlpluscompatibility 11.2.0
sqlprefix "&" (hex 23)
sqlprompt "SQL> "
sqlterminator ";" (hex 3b)
suffix "sql"
```

이부분이 만나와야 한다.

```
SQL> set colsep ' '
```

```
SQL> set pagesize 0
```

```
SQL> set linesize 120
```

```
SQL> set sqlprompt "" <-- SQL> 지우기
```

```
select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

emp223으로 다시 저장하기.

SQL을 나갔다가 들어오면 set이 초기화 되서 다시 설정을 해줘야한다.

```
SQL> set colsep ' '
```

```
SQL> set pagesize 20
```

```
SQL> set linesize 120
```

```
SQL> feedback off <-- 14 row selected 안나오게 하기
```

```
SQL> set echo off <-- select * from emp; 안나오게 하기
```

```
SQL> set sqlprompt ""
```

```
SQL> underline off <-- 밑줄 제거
```

```
spool emp227.csv
```

```
select * from emp;
```

```
spool off
```

```
exit
```

```
cat emp227.csv
```

```

select * from emp;

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```

spool off

```

이제 이두개만 지우면 된다.

• while loop문

```

ex)
i=1
while [ $i -le 10 ]
do
    echo $i
    i=`expr $i + 1`
done
[orcl:~]$ sh exa.sh
1
2
3
4
5
6
7
8
9
10

```

문제144. while loop문으로 factorial을 구현하시오.

```

보기)
$ sh 144a.sh 5
120

```

답)

```

#!/bin/bash

```

```

i=1
num=1
while [ $i -le $1 ]

```

```
do
    let num=$num*$i
    let i=$i+1
done
echo $num
```

문제145. 최대 공약수를 while loop문으로 구현하시오.

보기)
\$ sh 145a 16 24

답1)
#!/bin/bash

```
num1=$1
num2=$2
```

```
let mod=$num1%$num2
while [ $mod -gt 0 ]
do
    num1=$num2
    num2=$mod
    mod=`expr $num1 % $num2`
done
echo $num2
```

답2)
#!/bin/bash

```
max=$2
min=$1
```

```
i=1
while [ $i -gt 0 ]
do
    i=`expr $max % $min`
    max=$min
    min=$i
done
echo $max
```

문제146. emp 테이블의 데이터를 csv로 생성하고 R 에서 로드하시오.

```
$vi aaa.sql
```

```
set colsep ','
```

```
set pagesize 20
```

```
set linesize 120
```

```
set sqlprompt ""
```

```
set feedback off
```

```
set echo off
```

```
set underline off
```

```
set 입력해주고
```

```
spool emp.csv
```

```
select * from emp;
```

```
spool off
```

```
exit
```

-----여기까지 스크립트

```
cat emp.csv
```

하면

```
select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
spool off
```

이런 결과가나오는데 다시 오라클로 돌아와서

vi emp227.csv 로 빨간 부분을 지워주고

R로 접속해서

```
$ sqlplus scott/tiger @sqlsetting.sql
```

SQL 접속과 동시에 sqlsetting 스크립트를 돌리겠다.

```
emp<-read.csv('emp.csv',strip.white=T)
```

```
emp
```

을 보면 깔~꿈

```
> emp<-read.csv('emp227.csv',strip.white=T)
> emp
  EMPNO  ENAME      JOB  MGR  HIREDATE   SAL  COMM  DEPTNO
1   7369  SMITH    CLERK 7902 17-DEC-80   800    NA     20
2   7499  ALLEN  SALESMAN 7698 20-FEB-81  1600   300     30
3   7521   WARD  SALESMAN 7698 22-FEB-81  1250   500     30
4   7566  JONES   MANAGER 7839 02-APR-81  2975    NA     20
5   7654 MARTIN  SALESMAN 7698 28-SEP-81  1250  1400     30
6   7698  BLAKE   MANAGER 7839 01-MAY-81  2850    NA     30
7   7782  CLARK   MANAGER 7839 09-JUN-81  2450    NA     10
8   7788  SCOTT   ANALYST 7566 19-APR-87  3000    NA     20
9   7839   KING  PRESIDENT   NA 17-NOV-81  5000    NA     10
10  7844 TURNER  SALESMAN 7698 08-SEP-81  1500    0     30
11  7876 ADAMS    CLERK 7788 23-MAY-87  1100    NA     20
12  7900 JAMES    CLERK 7698 03-DEC-81   950    NA     30
13  7902  FORD    ANALYST 7566 03-DEC-81  3000    NA     20
14  7934 MILLER   CLERK 7782 23-JAN-82  1300    NA     10
```

문제147. sed명령어로 emp.sql의 emp를 dept로 변경하시오.

```
$ sed 's/emp/dept/' emp.sql > dept.sql
```

```
$ sqlplus scott/tiger @dept.sql
```

```
DEPTNO,DNAME      ,LOC
    10,ACCOUNTING ,NEW YORK
    20,RESEARCH    ,DALLAS
    30,SALES        ,CHICAGO
    40,OPERATIONS  ,BOSTON
```

문제148. 위의 스크립트를 이용해서 아래와 같이 테이블명을 물어보게하고, 테이블명을 입력하면 csv파일이 생성되게 하시오.

보기)

```
$ create_csv.sh
```

테이블명을 입력하세요 ~ salgrade

답)

```
#!/bin/bash
```

```
echo -n "테이블명을 입력하세요~ "
```

```
read table_name
```

```
sed 's/emp/'$table_name/' emp.sql > $table_name.sql
```

```
sqlplus scott/tiger @$table_name.sql
```

```
GRADE,      LOSAL,      HISAL
    1,         700,        1200
    2,        1201,        1400
    3,        1401,        2000
    4,        2001,        3000
    5,        3001,        9999
```

case문

" if 문과 유사한 문법인데 특정 셸 스크립트를 실행할 수 있도록 도와주는 문법 "

R의 menu 함수와 같다

ex)

```
[orcl:~]$ vi m2.sh
```

```
echo "
```

```
    1. Sar 그래프
```

```
    2. csv 파일 생성
```

```
"
```

```
echo " "
```

```
echo -n "번호를 입력하세요"
```

```
read choice
```

```
case $choice in
```

```
    1)
```

```
        /home/oracle/Sar.sh ;;
```

```
    2)
```

```
        /home/orcale/create_csv.sh ;;
```

```
esac
```

```
[orcl:~]$ vi sar.sh
```

```
Rscript /home/oracle/Sar.R
```

```
[orcl:~]$ vi sar.R
```

```
x11()
```

```
sar <- function(n) {
```

```
    m <- rep(NA,n)
```

```
    for ( i in 1:n) {
```

```
        re <- system("sar 1 1 | awk '{print $3}'", intern=TRUE)
```

```
        re <- re[5]
```

```
        m[i] <- re
```

```
        plot(m, type='o', col='red', xlab='time', ylab='Ratio')
```

```
    }
```

```
    rm(m)
```

```
}
```

sar(100)

문제149. 이전의 만들었던 find_file.sh가 잘 실행 되는지 확인하여라.

답)

```
[orcl:~]$ vi find_file.sh
```

```
#!/bin/home
```

```
echo "      "
```

```
echo -n "input find file :"
```

```
read file1
```

```
echo "      "
```

```
echo -n "depth ? :"
```

```
read depth1
```

```
find . maxdepth $depth1 -name $file1
```

```
[orcl:~]$ sh find_file.sh
```

```
input find file :emp.txt
```

```
depth ? :3
```

```
./test100/emp.txt
```

```
./test200/emp.txt
```

```
./emp.txt
```

```
./backup/emp.txt
```

```
./bbb/emp.txt
```

```
find: maxdepth: No such file or directory
```

```
find: 3: No such file or directory
```

문제150. 리눅스 자동화 스크립트인 m2.sh의 세번째로 find_file.sh를 추가하여라.

답)

```
[orcl:~]$ vi m2.sh
```

```
echo "
```

```
1. Sar 그래프
```

```
2. csv 파일 생성
```

```
3. file 찾기
```

```
"
```

```
echo " "
```

```
echo -n "번호를 입력하세요"
```

```
read choice
```

```
case $choice in
```

```
1)
```

```
    /home/oracle/Sar.sh ;;
```

```
2)
```

```

        /home/orcale/create_csv.sh ;;
3)
        /home/oracle/find_file.sh ;;
esac

```

문제151. 특정 파일 안에 특정 단어가 몇 건이 있는지 검색하는 쉘을 만들기 위해서 겨울왕국의 elsa라는 단어가 몇 건이 나오는지 검색하여라.

답)

```

[orcl:~]$ grep -woi 'elsa' winter.txt |wc -l

[orcl:~]$ grep -woi 'elsa' winter.txt |wc -l
329

```

문제152. 위의 스크립트로 아래와 같이 쉘 스크립트를 작성하여라.

답)

```

[orcl:~]$ vi find_word.sh
#!/bin/bash
read -p "검색 대상 파일명을 입력하세요" file
read -p "검색할 단어를 입력하세요" word

grep -iow $word $file | wc -l

```

문제153. 자동화 스크립트 4번에 fine_word.sh를 추가하여라

답)

```

[orcl:~]$ vi m2.sh
echo "
    1. Sar 그래프
    2. csv 파일 생성
    3. file 찾기
    4. 파일의 특정 단어 수 찾기
"
echo " "
echo -n "번호를 입력하세요"
read choice

case $choice in
    1)
        /home/oracle/Sar.sh ;;
    2)
        /home/orcale/create_csv.sh ;;
    3)

```



```
/home/oracle/find_file.sh ;;
```

4)

```
/home/oracle/fine_word.sh ;;
```

esac

##sar.sh 실행하던 도중에 ctrl+z을 누르면 그래프 그려지는 것이 중단된다. 하지만 그 때 켜졌던 x11()창은 열리지 않는다. 이거 어떻게 꺼야할까?

```
[orcl:~]$ ps -ef | grep sar.R
oracle 10716 10715 0 11:05 pts/1 00:00:00 /home/oracle/R-3.2.3/bin/exec/R --slave --no-restore --
file=/home/oracle/sar.R
oracle 10866 10126 0 11:07 pts/1 00:00:00 grep sar.R
[orcl:~]$ kill -9 10716
```

또는,

ctrl + C : kill 로 끝수 있다

다음 오류 수정하는 스크립트

```
E325: ATTENTION
Found a swap file by the name ".aaa.sh.swp"
    owned by: oracle    dated: Fri Jun 29 11:12:47 2018
    file name: ~oracle/aaa.sh
    modified: YES
    user name: oracle   host name: edydr1p0.us.oracle.com
    process ID: 10925 (still running)
While opening file "aaa.sh"
    dated: Thu Jun 28 16:30:02 2018

(1) Another program may be editing the same file.
    If this is the case, be careful not to end up with two
    different instances of the same file when making changes.
    Quit, or continue with caution.

(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r aaa.sh"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".aaa.sh.swp"
    to avoid this message.

Swap file ".aaa.sh.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (Q)uit, (A)bort: 
```

```
[orcl:~]$ rm *.swp
```

: 이러면 swp 파일 다 remove할 수 있음

**문제154. 아래의 명령어를 rmswp.sh 스크립트에 저장하고 수행하여라. 파일이 없을 경우, '파일이 없습
니다'라는 메시지가 출력되게 하여라.**

힌트)

```
[orcl:~]$ vi rmswp.sh
```

```
#!/bin/bash
```

```
rm *.swp
```

결과)

```
[orcl:~]$ sh rmswp.sh
```

```
rm: cannot remove `*.swp': No such file or directory
```

=> 중단된 파일이 없으면 이렇게 나옴.

답)

```
#!/bin/bash
```

```
num=`ls -al *.swp|wc -l`
```

```
if [ $num -eq 0 ]; then
```

```
    echo "파일이 없습니다"
```

```
else
```

```
    rm *.swp
```

```
    echo "총 $num 개의 파일이 제거되었습니다"
```

```
fi
```

결과)

```
[orcl:~]$ sh rmswp.sh
ls: *.swp: No such file or directory
파일이 없습니다
```

문제155. 위의 rmswp.sh를 자동화 스크립트에 추가하여라.

답)

```
echo "
```

```
    1. Sar 그래프
```

```
    2. csv 파일 생성
```

```
    3. file 찾기
```

```
    4. 파일의 특정 단어 수 찾기
```

```
    5. swp파일 모두 삭제
```

```
"
```

```
echo " "
```

```
echo -n "번호를 입력하세요"
```

```
read choice
```

```
case $choice in
```

```
    1)
```

```
        /home/oracle/Sar.sh ;;
```

```
    2)
```

```
        /home/orcale/create_csv.sh ;;
```

```
    3)
```

```

        /home/oracle/find_file.sh ;;
4)
        /home/oracle/fine_word.sh ;;
5)
        /home/oracle/rmswp.sh ;;
esac

```

문제156. 두개의 파일의 차이를 확인하는 diff명령어를 아래와 같이 수행하고 자동화 스크립트에 6번에 추가할 쉘을 만드시오.

답)

```

#!/bin/home
echo "      "
echo -n "input file1 : "
read f1_t
echo "      "
echo -n "input file2 : "
read f2_t

diff $f1_t $f2_t

```

문제157. diff.sh를 m2.sh 6번에 추가하여라.

답)

```

echo "
1. Sar 그래프
2. csv 파일 생성
3. file 찾기
4. 파일의 특정 단어 수 찾기
5. swp파일 모두 삭제
6. 두개 파일의 차이 찾기
"

echo " "
echo -n "번호를 입력하세요"
read choice

case $choice in
1)
        /home/oracle/Sar.sh ;;
2)
        /home/orcale/create_csv.sh ;;
3)

```

```

/home/oracle/find_file.sh ;;
4)
/home/oracle/fine_word.sh ;;
5)
/home/oracle/rmswp.sh ;;
6)
/home/oracle/diff.sh ;;
esac

```

문제 158. 오라클의 hr 계정의 락을 해제하고 hr 계정이 가지고 있는 테이블 리스트를 조회하여라.

답)

```
[orcl:~]$ sqlplus / as sysdba
```

#hr 계정의 락을 해제하는 명령어

```
SQL> alter user hr
2 account unlock;
```

User altered.

#hr 계정의 패스워드를 hr로 변경하는 명령어

```
SQL> alter user hr
2 identified by hr;
```

#hr 계정으로 접속

```
SQL> connect hr/hr
```

#hr 계정이 가지고 있는 테이블 리스트 조회

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
COUNTRIES	TABLE	
DEPARTMENTS	TABLE	
EMPLOYEES	TABLE	
EMP_DETAILS_VIEW	VIEW	
JOBS	TABLE	
JOB_HISTORY	TABLE	
LOCATIONS	TABLE	
REGIONS	TABLE	

8 rows selected.

문제 159. (점심시간) hr 계정이 가지고 있는 테이블 리스트를 전부 CSV 파일로 생성하시오.

답)

```
[orcl:~]$ vi emp.sql
```

```

set colsep ','
set pagesize 20
set linesize 120
set sqlprompt ""
set feedback off
set echo off
set underline off

spool alltable.txt
select * from tab;      <-- 테이블 목록이 나온다.

spool off

exit
-----

echo "sql 계정에 있는 모든 테이블을 csv 파일로 불러옵니다."
read -p "sql id:" id
read -p "pw:" pw
sqlplus $id/$pw @emp.sql
num=`cat alltable.txt|wc -l`
for i in `eval echo {1..$num}`
do
    name=`awk '{print $1}' alltable.txt | head -n $i | tail -n 1`
    sed 's/emp/'$name'/ ' emp.sql > $name.sql
    sqlplus $id/$pw @$name.sql
done

```

문제160. emp.csv에서 위의 1번과 2번 라인을 삭제하고 emp2.csv로 저장하여라.

답)

```

[orcl:~]$ sh create_csv.sh
테이블명을 입력하세요~ emp

```

```

[orcl:~]$ sed '1d' emp.csv > emp2.csv
[orcl:~]$ sed '1d' emp.csv

```

문제161. spool off 를 지워라.

```

[orcl:~]$ sed '/spool/d' emp2.csv > emp3.csv

```

문제162. 공백라인이 있을 수 있으니 공백라인을 제거하고 emp4.csv로 저장하여라.

```
[orcl:~]$ sed '/^%/d' emp3.csv > emp4.csv
```

문제163. emp4.csv가 R에서 잘 로드되는지 확인하여라.

```
[orcl:~]$ R
> emp<-read.csv("/home/oracle/emp4.csv",strip.white=T)
> emp
  > emp
    X7839   KING PRESIDENT    X X1981.11.17 X5000   X.1 X10
1    7698  BLAKE   MANAGER 7839   1981-05-01   2850   NA   30
2    7782  CLARK   MANAGER 7839   1981-05-09   2450   NA   10
3    7566  JONES   MANAGER 7839   1981-04-01   2975   NA   20
4    7654 MARTIN  SALESMAN 7698   1981-09-10   1250 1400   30
5    7499  ALLEN  SALESMAN 7698   1981-02-11   1600   300   30
6    7844 TURNER  SALESMAN 7698   1981-08-21   1500    0   30
7    7900  JAMES    CLERK 7698   1981-12-11    950   NA   30
8    7521  WARD   SALESMAN 7698   1981-02-23   1250   500   30
9    7902  FORD   ANALYST 7566   1981-12-11   3000   NA   20
10   7369  SMITH    CLERK 7902   1980-12-09    800   NA   20
11   7788  SCOTT   ANALYST 7566   1982-12-22   3000   NA   20
12   7876  ADAMS    CLERK 7788   1983-01-15   1100   NA   20
13   7934 MILLER    CLERK 7782   1982-01-11   1300   NA   10
```

문제164. 위에서 사용한 아래의 5개의 스크립트를 쉘로 저장해서 R로 깔끔하게 로드되는 csv로 생성하여라.

```
echo "sql 계정에 있는 모든 테이블을 csv 파일로 불러옵니다."
read -p "sql id:" id
read -p "pw:" pw
sqlplus $id/$pw @emp.sql
num=`cat alltable.txt|wc -l`
for i in `eval echo {1..$num}`
do
    name=`awk '{print $1}' alltable.txt | head -n $i | tail -n 1`
    sed 's/emp/'$name/' emp.sql > $name.sql
    sqlplus $id/$pw @$name.sql
    sed '1d' $name.csv > $name1.csv
    rm $name.csv
    mv $name1.csv $name.csv
done
```

문제165. hr_.sh를 더 발전 시켜라.

```
echo "sql 계정에 있는 모든 테이블을 csv 파일로 불러옵니다."
read -p "sql id:" id
read -p "pw:" pw
read -p "모든 테이블을 불러올래? (y/n) (n 선택시 특정 테이블 선택 가능)" ans

if [ $ans = "n" ]; then
```

```

read -p "어떤 테이블 불러올래?" name
sed 's/emp/'$name/' emp.sql > $name.sql
sqlplus $id/$pw @$name.sql
sed '1d' $name.csv > $name1.csv
rm $name.csv
mv $name1.csv $name.csv

else
sqlplus $id/$pw @emp.sql
num=`cat alltable.txt|wc -l`
for i in `eval echo {1..$num}`
do
    name=`awk'{print $1}' alltable.txt | head -n $i | tail -n 1`
    sed 's/emp/'$name/' emp.sql > $name.sql
    sqlplus $id/$pw @$name.sql
    sed '1d' $name.csv > $name1.csv
    rm $name.csv
    mv $name1.csv $name.csv
done
fi

```

문제166. 자동화 스크립트에 hr_sh를 추가하여라.

```

echo "
1. Sar 그래프
2. csv 파일 생성
3. 파일의 특정 단어 수 찾기
4. swp파일 모두 삭제
5. 두개 파일의 차이 찾기
6. SQL의 테이블 csv로 생성
7. 하둡 파일 시스템을 중단 시키려면 7번을
8. 하둡 파일 시스템을 시작 시키려면 8번을
9. 하둡 파일 시스템의 상태를 확인하려면 9번을
"

echo " "
echo -n "번호를 입력하세요"
read choice

case $choice in
    1)

```

```

/home/oracle/Sar.sh ;;
2)
/home/oracle/find_file.sh ;;
3)
/home/oracle/fine_word.sh ;;
4)
/home/oracle/rmswp.sh ;;
5)
/home/oracle/diff.sh ;;
6)
/home/oracle/hr_.sh ;;
7)
stop-all.sh ;;
8)
start-all.sh ;;
9)
jps ;;
esac

```

8. 리눅스 환경에서 R 사용하기

문제167. 리눅스 환경에서 R과 오라클을 연동하여라.

1. 리눅스 오라클 리스터상태를 확인한다.

```

[orcl:~]$ lsnrctl status
[orcl:~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 03-JUL-2018 17:52:46

Copyright (c) 1991, 2009, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1521))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                21-JUN-2018 15:20:57
Uptime                    12 days 2 hr. 31 min. 49 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/product/11.2.0/grid/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/edydr1p0/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=edydr1p0.us.oracle.com) (PORT=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM", status READY, has 1 handler(s) for this service...
Service "orcl.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
Service "orclXDB.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
The command completed successfully

```

2. listener.ora 파일을 열어서 아이피 주소를 192.168.56.104로 세팅하고 포트를 1521로 변경한다.

```

[orcl:~]$ cd $ORACLE_HOME/network/admin

```

```

[orcl:admin]$ ls

```



```
samples shrept.lst tnsnames.ora
```

```
[orcl:admin]$ vi listener.ora
```

```
>> 없다. 만들어 줘야함
```

```
sqlplus scott/tiger@edydr1p0.us.oracle.com:1521/orcl.us.oracle.com
```

3. 오라클과 R의 연동을 위해 필요한 패키지를 설치한다.

```
install.packages("rJava")
```

```
install.packages("DBI")
```

```
install.packages("RJDBC")
```

4. 자바 홈을 설정한다.

```
Sys.setenv(JAVA_HOME='/path/to/java_home')
```

```
options(java.parameters="-Xmx2g")
```

```
library(rJava)
```

5. 자바 버전 확인.

```
.jinit()
```

```
print(jcall("java/lang/System", "S", "getProperty", "java.version"))
```

a. RJDBC 라이브러리 불러오기

```
library(DBI)
```

```
library(RJDBC)
```

b. 커넥션 드라이버 생성

```
jdbcDriver <- JDBC(driverClass="oracle.jdbc.OracleDriver",  
classPath="/u01/app/oracle/product/11.2.0/dbhome_1/jdbc/lib/ojdbc6.jar")
```

```
> jdbcDriver <- JDBC(driverClass="oracle.jdbc.OracleDriver",
+ classPath="/u01/app/oracle/product/11.2.0/dbhome_1/jdbc/lib/ojdbc6.jar")
#
# A fatal error has been detected by the Java Runtime Environment:
#
# Internal Error (threadLocalStorage.cpp:56), pid=9112, tid=3086431984
# guarantee(get_thread() == thread) failed: must be the same thread, quickly
#
# JRE version: (7.0_60-b19) (build )
# Java VM: Java HotSpot(TM) Server VM (24.60-b09 mixed mode linux-x86 )
# Failed to write core dump. Core dumps have been disabled. To enable core dums
re starting Java again
#
# An error report file with more information is saved as:
# /home/oracle/hs_err_pid9112.log
#
# If you would like to submit a bug report, please visit:
# http://bugreport.sun.com/bugreport/crash.jsp
#
Aborted
```

c. 오라클로 연결한다. (자바 버전이 1.6만된다, 1.7x)

```
scott <-dbConnect(jdbcDriver, "jdbc:oracle:thin:
@//edydr1p0.us.oracle.com:1521/orcl.us.oracle.com", "scott", "tiger")
emp <- dbGetQuery(scott, "SELECT * from emp")
print(emp)
```

```
> print(emp)
  EMPNO  ENAME      JOB   MGR      HIREDATE    SAL  COMM DEPTNO
1   7369  SMITH      CLERK  7902 1980-12-17 00:00:00   800    NA     20
2   7499  ALLEN    SALESMAN  7698 1981-02-20 00:00:00  1600   300     30
3   7521   WARD    SALESMAN  7698 1981-02-22 00:00:00  1250   500     30
4   7566  JONES    MANAGER  7839 1981-04-02 00:00:00  2975    NA     20
5   7654 MARTIN    SALESMAN  7698 1981-09-28 00:00:00  1250  1400     30
6   7698  BLAKE    MANAGER  7839 1981-05-01 00:00:00  2850    NA     30
7   7782  CLARK    MANAGER  7839 1981-06-09 00:00:00  2450    NA     10
8   7788  SCOTT    ANALYST  7566 1987-04-19 00:00:00  3000    NA     20
9   7839   KING  PRESIDENT    NA 1981-11-17 00:00:00  5000    NA     10
10  7844 TURNER    SALESMAN  7698 1981-09-08 00:00:00  1500     0     30
11  7876 ADAMS      CLERK  7788 1987-05-23 00:00:00  1100    NA     20
12  7900 JAMES      CLERK  7698 1981-12-03 00:00:00   950    NA     30
13  7902  FORD    ANALYST  7566 1981-12-03 00:00:00  3000    NA     20
14  7934 MILLER    CLERK  7782 1982-01-23 00:00:00  1300    NA     10
```

문제168 .emp 변수의 내용을 emp900.csv로 생성하여라.

```
R
>write.csv(emp, 'emp900.csv')
```

문제169. oracle에서 dept 테이블을 가져와서 dept900.csv를 생성하여라.

```
답)
> dept<-dbGetQuery(scott,"select * from dept")
> write.csv(dept, 'dept900.csv')
> dept900<-read.csv('dept900.csv')
```

```
> dept900
  X DEPTNO      DNAME      LOC
1 1      10 ACCOUNTING NEW YORK
2 2      20  RESEARCH  DALLAS
3 3      30    SALES   CHICAGO
4 4      40 OPERATIONS  BOSTON
```

문제170. scott이 가지고 있는 테이블 리스트를 가져와 변수에 저장하여라.

답)

```
> scott_table <- dbGetQuery(scott, "select * from tab")
> scott_table
```

```
> for (i in 1:num){
+ x<-scott_table[i,]
+ x_name<-paste('select * from',x)
+ y<-dbGetQuery(scott,x_name)
+ print(y)}
```

이렇게하면

	TNAME	TABTYPE	CLUSTERID
1	BIN\$b6ygp+aAsSXgUAB/AQBLKA==\$0	TABLE	NA
2	BONUS	TABLE	NA
3	DEPT	TABLE	NA
4	EMP	TABLE	NA
5	SALGRADE	TABLE	NA

이상한 테이블이 나와서 안된다.

그래서 아래와 같은 방법으로 해야함.

```
> scott_table<-dbGetQuery(scott, "SELECT table_name from user_tables")
> scott_table
> scott_table
  TABLE_NAME
1      DEPT
2      BONUS
3  SALGRADE
4        EMP
```

9. 리눅스 설치

1. 오라클 리눅스 설치



完.OEL5
설치_통합...

2. 우분투 리눅스 설치



우분투3

3. Cent OS 리눅스 설치



센토7 (1)

1. 호스트(컴퓨터 이름) 이름

hostname

2. 어떤 os이고 버전이 어떻게 되는지 확인

uname -an