

리눅스

2018년 6월 21일 목요일 오후 4:16

[01 설치](#)

[02 리눅스란?](#)

[03 리눅스 기본 명령어](#)

[04 vi 편집기 명령어](#)

[05 권한 관리](#)

[06 디스크 관리](#)

[07 프로세서 관리](#)

[08 유저 생성 및 관리](#)

[09 쉘 스크립트 작성법](#)

[10 리눅스 환경에서 R과 오라클 연동시키기](#)

01 설치

2018년 6월 21일 목요일 오후 2:02

1. VM웨어 설치 및 리눅스 시스템 가동 방법

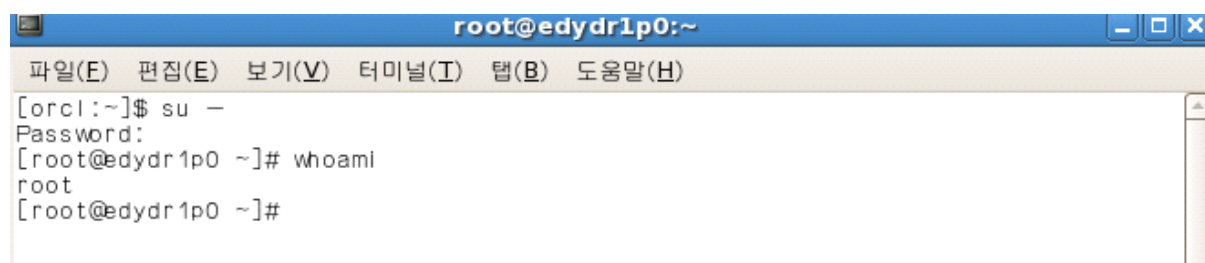
1. oracle virtual box 설치
2. oracle 리눅스 환경(ws2.ora)를 불러온다

2. 오라클 리눅스 로그인 화면

유저이름: oracle
패스워드: oracle

3. ip 주소를 확인하기 위해 root 유저로 접속한다

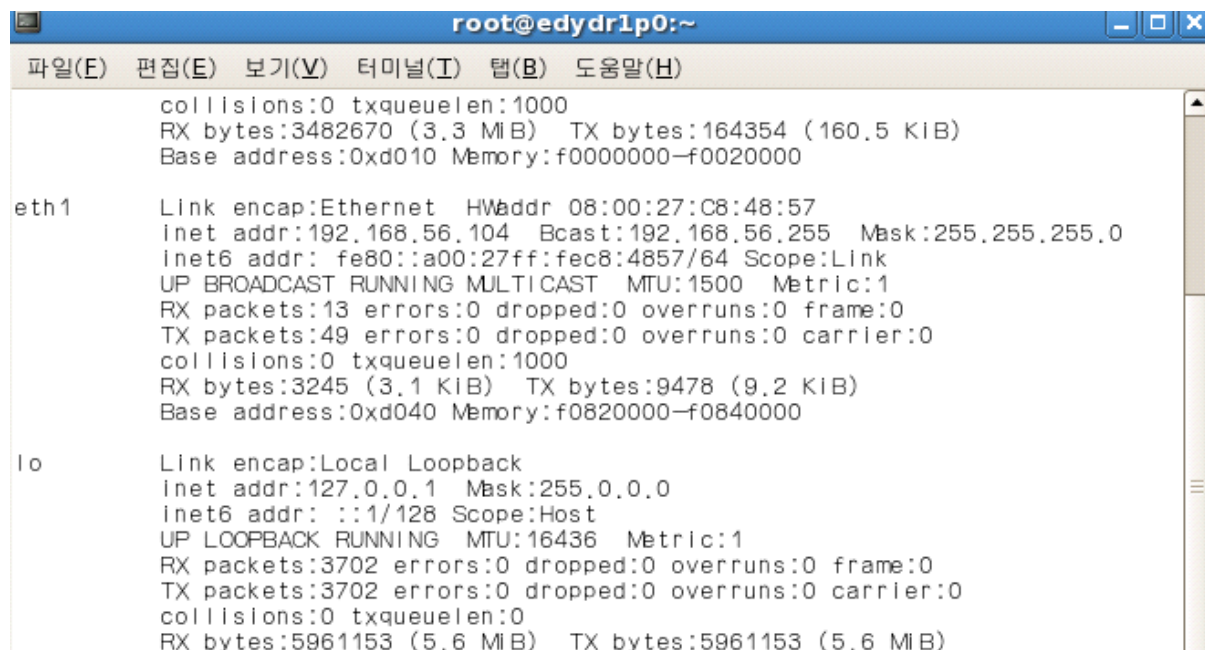
```
$ su -  
패스워드: oracle  
  
#whoami
```



```
root@edydr1p0:~  
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
[orcl:~]$ su -  
Password:  
[root@edydr1p0 ~]# whoami  
root  
[root@edydr1p0 ~]#
```

4. ip 주소 확인 명령어

```
# ifconfig
```



```
root@edydr1p0:~  
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
collisions:0 txqueuelen:1000  
RX bytes:3482670 (3.3 MiB) TX bytes:164354 (160.5 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:13 errors:0 dropped:0 overruns:0 frame:0  
       TX packets:49 errors:0 dropped:0 overruns:0 carrier:0  
       collisions:0 txqueuelen:1000  
       RX bytes:3245 (3.1 KiB) TX bytes:9478 (9.2 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:3702 errors:0 dropped:0 overruns:0 frame:0  
       TX packets:3702 errors:0 dropped:0 overruns:0 carrier:0  
       collisions:0 txqueuelen:0  
       RX bytes:5961153 (5.6 MiB) TX bytes:5961153 (5.6 MiB)
```

5. xshell로 리눅스 서버를 접속한다

xshell 접속

새로만들기

이름 저장 후 IP번호(호스트)입력

이름: oracle 비밀번호: oracle

■ 리눅스 설치 3가지

1. Vmware - 오라클 리눅스 설치

환영합니다

사용 계약서

방화벽

SELinux


Kdump

날짜 및 시간

▶ 사용자 생성

사운드 카드

추가 CD



사용자 생성

시스템의 일반 사용을 위한 (관리자 계정이 아닌) '사용자 계정'을 생성하시기 바랍니다. 시스템 '사용자 계정'을 만드시려면, 아래에서 요구하는 정보를 기입해 주십시오.

사용자명(U):

이름(E):

암호(P):

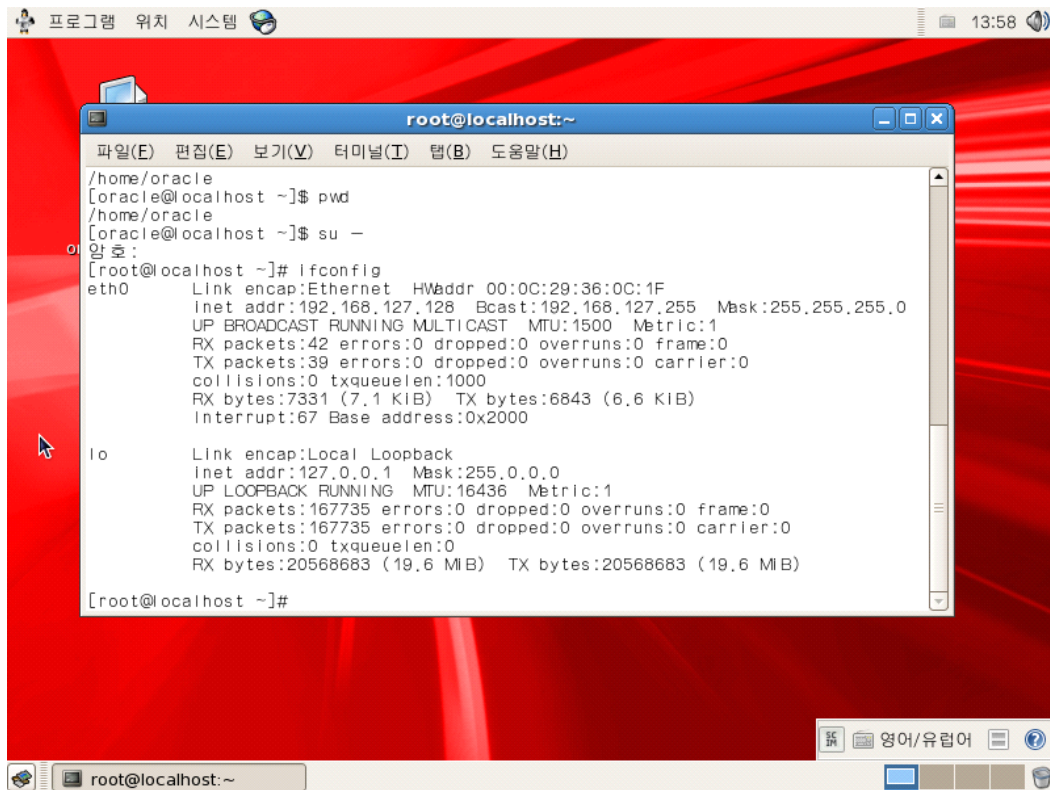
암호 확인(M):

커베로스나 NIS와 같은 네트워크 인증을 사용하시려면, 네트워크 로그인 버튼을 클릭해 주십시오.

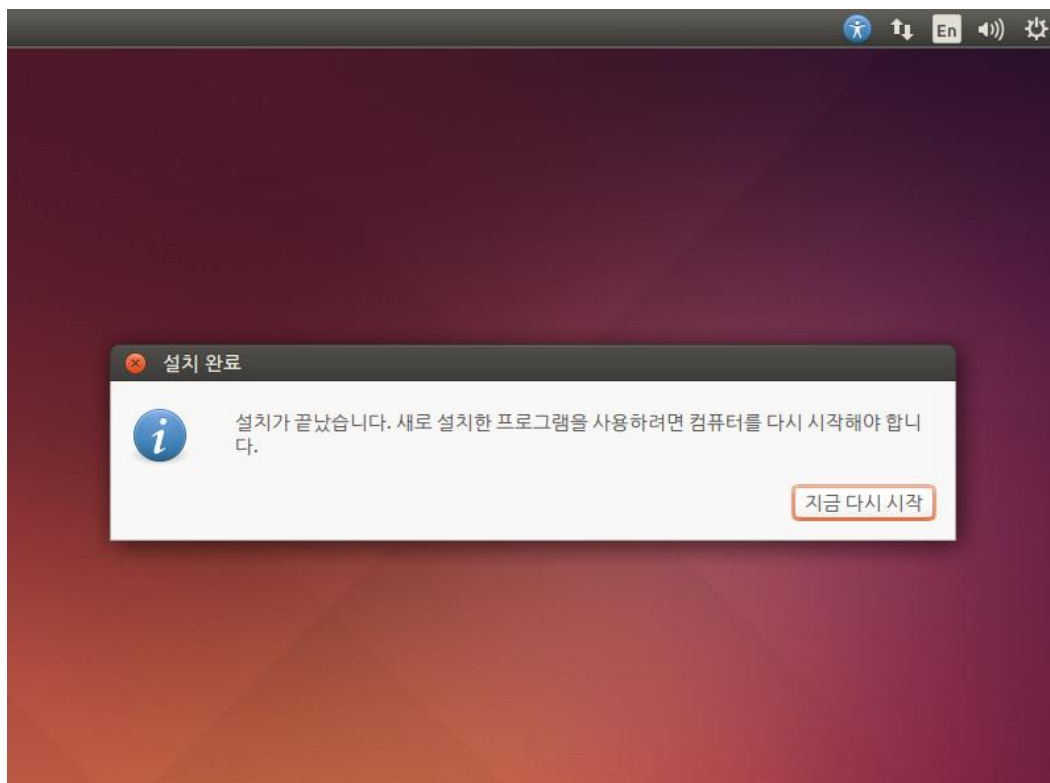
네트워크 로그인 사용(L)...

◀ 뒤로(B)

▶ 앞으로(E)



2. 우분투 리눅스 설치



3. Cent OS 리눅스 설치

1. 호스트 이름

```
# hostname
```

2. 어떤 OS이고 버전이 어떻게 되는지 확인

```
# uname -an
```

02 리눅스란?

2018년 6월 21일 목요일 오후 4:21

1. 리눅스란

유닉스가 너무 고가여서 리눅스 오픈소스를 리누즈 토발즈 핀란드 학생이 1991년 11월에 개발한 OS이다.

리누즈 토발즈가 개발한 것은 리눅스의 커널(자동차 엔진)을 개발하고 소스를 무료로 공개하고 전 세계의 많은 개발자들이 이 오픈소스를 가져다가 더 좋게 개선해서 다시 인터넷에 올리고 하는 작업을 반복하다보니 리눅스 OS가 유닉스보다 더 가볍고 안정적이게 되었다.

이 프로젝트를 GNU 프로젝트라고 한다.

누구든지 배포된 오픈 소스를 가져다가 개발할 수 있고 돈을 벌 목적으로 상용화할 수도 있는데 한가지 지켜야할 약속이 있다. 그것은 이 소스를 가져다가 더 좋게 수정했으면 수정한 코드를 인터넷에 올려줘야 한다.

유료	무료
레드햇(상용 버전)	CentOS

종류

1. Oracle linux (우리가 사용할 것)
2. Ubuntu (우분투)

2. 리눅스를 왜 배워야 하는가?

1. 현업의 대용량 데이터는 대부분 window가 아니라 유닉스 또는 리눅스 OS를 사용하는 서버에 저장되어 있다

큰 기업의 메인 서버 유닉스, 리눅스
쿠팡, 인터파크, G마켓: 리눅스

2. 딥러닝 개발/연구, 데이터 분석가(하둡이 리눅스 기반)도 리눅스를 다루는게 필수
centOS 우분투

03 리눅스 기본 명령어

2018년 6월 21일 목요일 오후 4:21

목차

03 리눅스 기본 명령어

- [1. cd 명령어](#)
- [2. touch 명령어](#)
- [3. mkdir 명령어](#)
- [4. rm 명령어](#)
- [5. rmdir 명령어](#)
- [6. alias 명령어](#)
- [7. cat 명령어](#)
- [8. redirection 명령어](#)
- [10. more 명령어](#)
- [11. head 명령어](#)
- [12. tail 명령어](#)
- [13. grep 명령어](#)
- [14. awk 명령어](#)
- [15. sort 명령어](#)
- [16. uniq 명령어](#)
- [17. echo 명령어](#)
- [18. diff 명령어](#)
- [19. find 명령어](#)
- [20. tar 명령어](#)
- [21. ln \(link\) 명령어](#)
- [22. .bash_profile 설정하는 방법](#)
- [23. sed 명령어](#)
- [24. cp 명령어](#)
- [25. mv 명령어](#)

1. cd 명령어

"Change Directory 명령어로 디렉토리를 이동하는 명령어"

예제)

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

현재 내가 작업하고 있는 디렉토리 확인

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

```
Adobe Desktop datamodeler.log labs oradiag_oracle
```

현재 디렉토리에 있는 폴더와 파일을 확인

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

문제 1) 다시 /home/oracle 로 이동하시오

답)

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

문제 2) 현재 디렉토리에서 /(루트) 디렉토리로 바로 이동하시오

답)

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

설명)

경로의 2가지 종류

1. 절대 경로: cd 내가 가고자하는 정확한 위치

예: cd /home/oracle/labs

2. 상대 경로

\$ cd .. : 현 위치에서 상위 디렉토리로 이동

\$ cd ./labs : 현재 디렉토리의 labs 디렉토리로 이동

아래와 같이 scott/tiger가 lock 걸려 있는 문제 해결 방법

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


SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 09:59:07 2018

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

ERROR:

ORA-28000: the account is locked

해결 방법)

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

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 10:04:01 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

```
SQL> alter user scott
```

```
2 account unlock;
```

User altered.

```
SQL> alter user scott
```

```
2 identified by tiger;
```

User altered.

```
SQL> show user;
```

USER is "SYS"

2. touch 명령어

" 파일의 용량이 0인 파일을 생성하는 명령어 "

예)

```
SQL> exit;
```

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

```
[orcl:~]$ touch a1.txt
[orcl:~]$ ls -l a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
```

설명)

-rw-r--r-- : 권한, 1 oracle : 파일 소유자, oinstall : 소유자가 속한 그룹, 0 : 파일 크기,
Jun 22 10:09 : 파일 생성 날짜, a1.txt : 파일명

아무것도 없는 것 확인)

```
[orcl:~]$ view a1.txt
```

빠져 나오는 명령어)

ESC키 2번 누르고
:q!

문제 3) 현재 디렉토리에 아래의 파일들을 크기가 0으로 생성하시오

보기)

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

답)

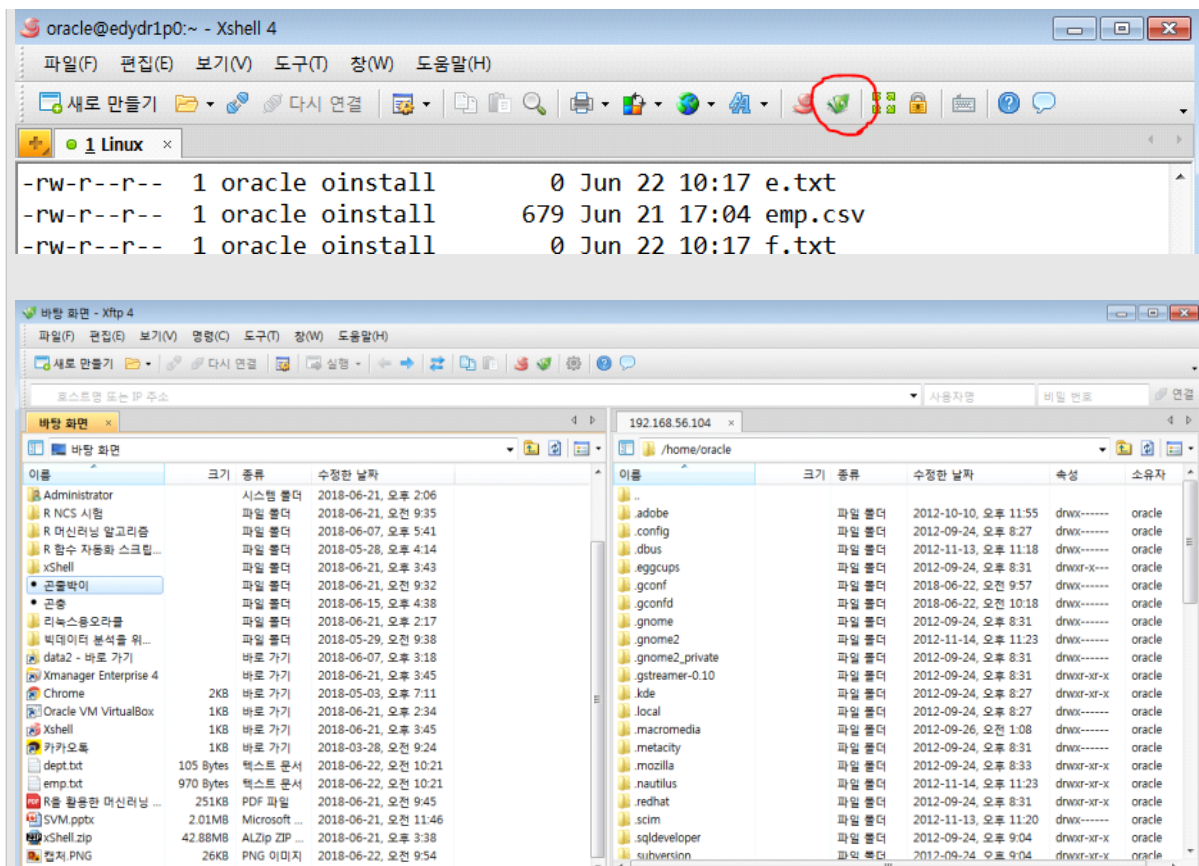
```
[orcl:~]$ touch a.txt b.txt c.txt d.txt e.txt f.txt
```

txt파일 조회)

```
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 d.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 f.txt
```

문제 4) 리눅스 서버에 /home/oracle 밑에 emp.txt와 dept.txt를 올리세요

답)



조회)

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

```
-rw-r--r-- 1 oracle oinstall 970 Jun 22 10:21 emp.txt
```

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

```
-rw-r--r-- 1 oracle oinstall 105 Jun 22 10:21 dept.txt
```

열람)

```
[orcl:~]$ cat emp.txt
```

3. mkdir 명령어

" 디렉토리를 생성하는 명령어 "

예)

```
[orcl:~]$ mkdir test2
```

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

```
17072993 test2
```

ls -h : 사람이 보기 좋게 출력됨

문제5) /home/oracle 밑에 아래의 하위 디렉토리를 아래와 같이 생성하시오

보기)

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

답)

```
[orcl:~]$ mkdir -p /home/oracle/test5/test6/test7/test8/test9
```

```
[orcl:test9]$ pwd
```

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

**문제 6) 지금 현재 디렉토리가 /home/oracle인 상태에서 한번에
/home/oracle/test5/test6/test7/test8/test9 로 이동하시오**

보기)

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

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

```
/home/oracle
```

답)

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

```
[orcl:test9]$ pwd
```

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

문제 7) 내가 바로 직전에 있었던 디렉토리로 가시오

답)

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

```
/home/oracle
```

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

```
/home/oracle
```

4. rm 명령어

" 파일이나 디렉토리를 삭제하는 명령어 "

주의사항

리눅스나 유닉스는 휴지통이 없기 때문에 삭제할 때 특히 주의해야 한다!
백업이 없으면 절대 복구가 불가능

예)

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

문제 8) 현재 디렉토리 밑에 있는 확장자가 txt인 파일들을 모두 삭제하시오

답)

rm -rf *.txt

무조건 조심해서 사용할 것

결과)

```
[orcl:~]$ rm -rf *.txt
[orcl:~]$ ls
Adobe    R-3.2.3      datamodeler.log  labs          test2
Desktop  R-3.2.3.tar.gz  emp.csv          oradiag_oracle test5
```

설명)

rm -rf *

-r 옵션 : 현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 다 삭제하겠다는 옵션

-f 옵션 : 삭제할 때 원래는 삭제할까요? 라고 물어보는데 물어보지 않고 그냥 다 삭제하라는 옵션

5. rmdir 명령어

" 디렉토리를 삭제하는 명령어 "

예)

```
[orcl:~]$ mkdir dd
[orcl:~]$ ls
Adobe    R-3.2.3      datamodeler.log  emp.csv  oradiag_oracle  test5
Desktop  R-3.2.3.tar.gz  dd              labs     test2
[orcl:~]$ rmdir dd
[orcl:~]$ ls
Adobe    R-3.2.3      datamodeler.log  labs          test2
Desktop  R-3.2.3.tar.gz  emp.csv          oradiag_oracle test5
```

문제 9) /home/oracle 밑에 test9 디렉토리를 만들고 test9 디렉토리 밑에 test10 디렉토리를 만드시오

답)

```
[orcl:~]$ mkdir -p ./test9/test10
```

```
Adobe      R-3.2.3      datamodeler.log  labs      test2  test9
Desktop    R-3.2.3.tar.gz  emp.csv          oradiag_oracle test5
```

설명)

./ : 현재 디렉토리

위의 점을 쓰지 않으면 현재 디렉토리에 생성됨

문제 10) /home/oracle 밑에 test9와 test10 디렉토리를 전부 삭제하시오

답)

```
[orcl:~]$ rm -r test9
```

6. alias 명령어

" 자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어 "

예)

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

```
[orcl:~]$ sql
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:16:35 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
```

문제 11) scott이라고 아래와 같이 명령어를 수행하면 오라클에 scott으로 접속되게 alias를 만드시오

답)

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

```
[orcl:~]$ scott
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:19:04 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
```

```
SQL> show user;
```

```
USER is "SCOTT"
```

alias 제거 및 조회

```
[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'  
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

문제 12) 오라클의 네트워크 설정에 대한 디렉토리가 있는 곳으로 이동하시오

보기)

```
pwd : print working directory
```

답)

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

```
[orcl:admin]$ pwd
```

```
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
```

```
[orcl:admin]$ ls
```

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

문제 13) 아래의 디렉토리로 쉽게 이동할 수 있는 alias를 생성하시오

보기)

```
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
```

```
$ oracle_network
```

답)

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

7. cat 명령어

" 파일의 내용을 화면에 출력하는 명령어 " \$ cat 파일명.txt

예)

```
[orcl:~]$ cat emp.txt
```

문제 14) cat으로 본 화면의 결과를 emp40.txt 라는 이름으로 저장하시오

답)

```
[orcl:~]$ cat emp.txt >> emp40.txt
```

문제 15) 현재 디렉토리 아래의 파일들을 touch 명령어로 생성하시오

답)

```
[orcl:~]$ touch a.txt b.txt c.txt e.txt
[orcl:~]$ ls -l *.txt
```

결과)

```
-rw-r--r-- 1 oracle oinstall  0 Jun 22 11:36 a.txt
-rw-r--r-- 1 oracle oinstall  0 Jun 22 11:36 b.txt
-rw-r--r-- 1 oracle oinstall  0 Jun 22 11:36 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:31 dept.txt
-rw-r--r-- 1 oracle oinstall  0 Jun 22 11:36 e.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:34 emp40.txt
```

문제 16) 지금 ls -l *.txt로 보고 있는 화면의 결과를 ls.txt라는 이름으로 저장하시오

답)

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

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

```
Adobe      R-3.2.3.tar.gz  c.txt      e.txt      emp40.txt  oradiag_oracle
Desktop    a.txt           datamodeler.log  emp.csv    labs       test2
R-3.2.3    b.txt           dept.txt     emp.txt    ls.txt     test5
```

```
[orcl:~]$ cat ls.txt
```

```
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 c.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:31 dept.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 e.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:34 emp40.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 ls.txt
```

8. redirection 명령어

" 화면의 출력되는 결과를 파일로 저장하는 명령어 "

>> : 없으면 파일을 생성하고, 있으면 기존 파일 뒤에 덧붙이겠다

> : 파일을 생성하겠다. 기존 파일이 있으면 그냥 덮어쓰겠다.

문제 17) emp.txt를 cat으로 보고 있는 결과를 emp50.txt 라는 이름으로 생성하고 다시 emp.txt의 결과를 emp50.txt에 덧붙여서 저장하시오

답)

```
[orcl:~]$ cat emp.txt > emp50.txt
```

```
[orcl:~]$ cat emp50.txt
```

```
[orcl:~]$ cat emp.txt >> emp50.txt
```

```
[orcl:~]$ cat emp50.txt
```

9. wc 명령어

" 파일 안에 단어의 개수 또는 라인 수를 출력하는 명령어 " \$ wc -l 파일명

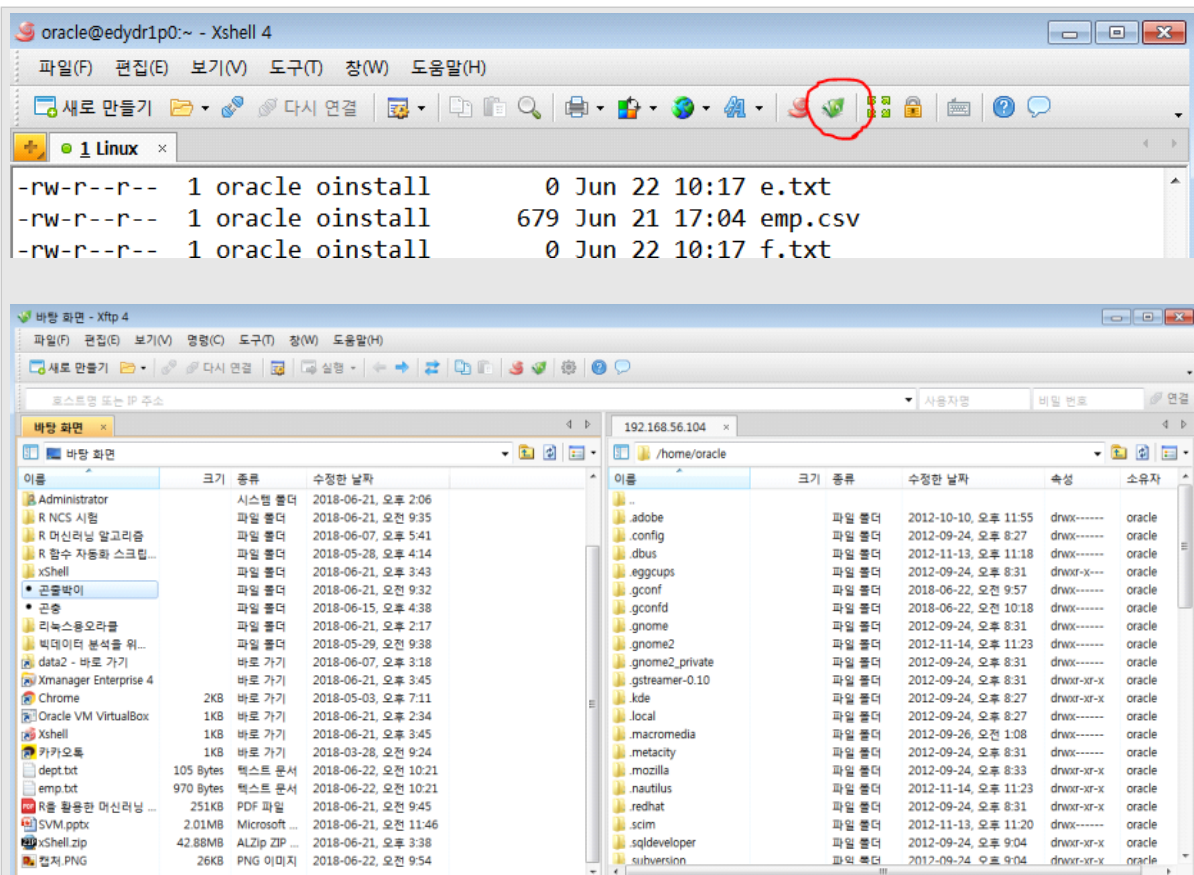
-l 옵션	라인 수 출력
-w 옵션	단어의 개수
-c 옵션	문자 수를 출력

예)

```
[orcl:~]$ wc -l emp.txt
14 emp.txt
[orcl:~]$ wc -w emp.txt
112 emp.txt
[orcl:~]$ wc -c emp.txt
970 emp.txt
```

문제 18) (점심시간 문제) 겨울왕국 대본의 단어 수가 어떻게 되는가?

답)



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

설명)

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

```
[orcl:~]$ man wc
```

매뉴얼 나가는 방법: q

문제 19) /home/oracle 밑에 확장자가 .txt로 끝나는 파일이 몇 개가 있는지 카운트하시오

답)

```
[orcl:~]$ ls -l *.txt | wc -l
10
```

설명)

| (파이프명령어) :

A(명령어) | B(명령어) : A명령어의 표준 출력을 B명령어의 표준 입력으로 보냄으로써 명령어의 실행결과를 다음 명령어로 전달하는 기능

문제 20) 아래의 경로로 이동해서 아래의 디렉토리 아래 있는 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]$ pwd
/u01/app/oracle/diag/rdbms/orcl/orcl/trace

[orcl:trace]$ ls -l *.trc | wc -l
185
```

10. more 명령어

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

예)

```
[orcl:~]$ more winter.txt
```

전진키 : space 키

후진키 : b

페이지 단위로 넘기기 : f

종료키 : q

11. head 명령어

" 문서의 처음 몇 줄을 화면에 출력하는 명령어 "

[orcl:~]\$ head -[출력 줄 수] [파일명]

예)

```
[orcl:~]$ head -20 winter.txt
```

문제 21) 겨울왕국 대본의 100줄까지의 내용을 winter_head.txt라는 이름으로 저장하시오

답)

```
[orcl:~]$ head -100 winter.txt >> winter_head.txt
```

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

결과)

Adobe	b.txt	emp.csv	ls.txt	winter_head.txt
Desktop	c.txt	emp.txt	oradiag_oracle	
R-3.2.3	datamodeler.log	emp40.txt	test2	
R-3.2.3.tar.gz	dept.txt	emp50.txt	test5	
a.txt	e.txt	labs	winter.txt	

12. tail 명령어

" 문서의 마지막 몇 줄을 화면에 출력하는 명령어 "

예)

```
[orcl:~]$ tail -10 winter.txt
```

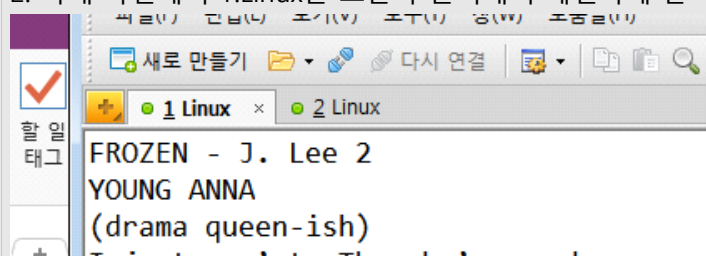
-f 옵션 : tail의 중요한 옵션으로 문서의 끝에 추가되는 새로운 내용을 실시간으로 확인할 수 있는 옵션

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

1. 디렉토리 이동

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

2. 아래 사진에서 1.Linux를 오른쪽 클릭해서 세션복제 한다



3. 원래 세션에서

```
[orcl:trace]$ ls -l alert*
```

```
[orcl:trace]$ tail -f alert_orcl.log
```

4. 새로 연 창에서

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

```
SQL> shutdown immediate
```

5. 원래 세션을 보면

실시간으로 계속 바뀌는 것을 볼 수 있다

6. 다시 새로 연 세션에서

```
SQL> startup
```

7. 원래 세션에서

Ctrl + C 로 종료 시킨다

예) 현재 리눅스 시스템의 i/o 성능을 확인하시오 (윈도우의 작업 관리자와 같은 것)

```
[orcl:~]$ sar 1 100
```

```
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/22/18
14:30:24      CPU      %user      %nice    %system    %iowait    %steal     %idle
le
14:30:25      all       0.50       0.00      13.86       0.00       0.00      85.64
14:30:26      all       0.00       0.00      12.06       0.00       0.00      87.94
14:30:27      all       0.50       0.00      15.50       0.00       0.00      84.00
14:30:28      all       0.50       0.00      10.40       0.00       0.00      89.10
```

새로 연 세션에서 scott에게 dba 권한을 준다

```
SQL> grant dba to scott;
```

```
Grant succeeded.
```

그리고 그 세션에서 악성 SQL을 열어본다

```
SQL> select count(*) from sh.sales, sh.sales;
```

그런 다음 원래 세션에 가보면 %user 가 오른 것을 볼 수 있다.

14:33:52	all	0.51	0.00	12.63	4.55	0.00	82.
32							
14:33:53	all	9.85	0.00	19.70	29.56	0.00	40.
89							
14:33:54	all	49.50	0.00	1.00	0.00	0.00	49.
50							
14:33:55	all	48.76	0.00	3.48	0.00	0.00	47.
76							
14:33:56	all	49.74	0.00	5.53	0.00	0.00	45.

새로운 세션, 원래 세션 모두 Ctrl + C 로 모든 작업을 중지시킨다.

문제 22) redirection을 이용해서 sar 1 100 명령어의 결과가 sar_0622.txt라는 이름으로 저장되게 하시오

답)

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

설명)

sar 1 100 : 1초 간 100번 조회하라는 뜻

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

```
Adobe      b.txt      emp.csv    ls.txt     winter.txt
Desktop    c.txt      emp.txt    oradiag_oracle winter_head.txt
R-3.2.3    datamodeler.log emp40.txt  sar_0622.txt
R-3.2.3.tar.gz dept.txt   emp50.txt test2
a.txt      e.txt      labs       test5
```

백그라운드로 돌리는 방법

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

```
[2] 10074
```

백그라운드로 돌렸던 거 중지시키는 법

Linux에서 kill 명령어로 안전하게 프로세스 종료 하는 방법

Kill은 이벤트가 발생 했음을 프로세스에 알려 옵션에서 지정한 동작을 하게 됩니다.

kill -KILL(9) <pid> : 하드웨어적 종료 (가장강력)

kill -TERM(15) <pid> : 소프트웨어적 종료 (소프트웨어에 따라 실행유무결정)

kill -HUP(1) <pid> : 데몬의 경우 종료 후, 다시시작 (프로세스 종료가 아님, 코드 및 데이터 refresh 역할)

kill -2 <pid> : 포그라운드에서 [ctrl] + [c] (작업취소) 를 누르는 것과 동일

kill -3 <pid> : 포그라운드에서 [ctrl] + [W]를 누르는 것과 동일

[ctrl] + [W] : 작업취소 [ctrl] + [c]와 동일하지만 더 강력함

13. grep 명령어

" 파일 안에 포함된 특정 단어나 구문을 검색하는 명령어 "

예)

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

설명)

emp.txt에서 scott이라는 단어가 포함된 라인을 보여달라는 뜻

-i 옵션 : 대소문자를 구분하지 않겠다

문제 23) 직업이 salesman인 직원들의 모든 행을 출력하시오

답)

```
[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
```

문제 24) 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
```

특정 컬럼을 뽑아낼 때 awk를 쓴다.

문제 25) 부서번호가 10번 인 직원들의 이름과 월급을 출력하시오

답)

```
[orcl:~]$ grep -iw '10' emp.txt
7782 CLARK MANAGER 7839 09-JUN-81 2450 0 10
```

7839 KING	PRESIDENT	0 17-NOV-81	5000	0	10
7934 MILLER	CLERK	7782 23-JAN-82	1300	0	10

```
[orcl:~]$ grep -iw '10' emp.txt | awk '{print $2, $6}'
CLARK 2450
KING 5000
MILLER 1300
```

설명)

-w 옵션 : 단어 별 검색

14. awk 명령어

" 특정 단어가 들어가 있는 라인에서 특정 컬럼명을 출력하고자 할 때 사용하는 명령어 "

예)

```
[orcl:~]$ awk '$3 == "SALESMAN" {print $2, $3}' emp.txt
ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
```

문제 26) 월급이 3000 이상인 직원들의 이름과 월급을 출력하시오

답)

```
[orcl:~]$ awk '$6 >= 3000 {print $2, $6}' emp.txt
SCOTT 3000
KING 5000
FORD 3000
```

리눅스의 연산자 3가지

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

문제 27) 직업이 **salesman**이 아닌 직원들의 이름과 직업을 출력하시오

답)

```
[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
```

대소문자 구별없이 나오게 하는 방법

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

문제 28) 직업이 **salesman**인 직원들의 이름과 월급을 출력하는데 월급이 1500 이상인 직원들만 출력하시오

답)

```
[orcl:~]$ awk '$3==toupper("salesmAN") && $6>=1500 {print $2, $6}' emp.txt
ALLEN 1600
TURNER 1500
```

문제 29) 이름이 **A**로 시작하는 직원들의 이름과 월급을 출력하시오

힌트)

grep과 정규식(시작: ^, 끝: \$)을 사용
awk와 grep과 |를 활용해서 출력

답)

```
[orcl:~]$ awk '{print $2, $6}' emp.txt | grep -i '^A'
ALLEN 1600
ADAMS 1100
```

문제 30) 81년도에 입사한 직원들의 이름과 입사일을 출력하시오

답)

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

결과)

```
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
```

문제 31) 81년도에 입사하지 않은 직원들의 이름과 입사일을 출력하시오

답)

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

결과)

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

설명)

-v 옵션 : 포함되지 않은 것을 검색할 때 사용

문제 32) 겨울왕국 대본에서 elsa가 포함되어져 있는 라인들의 개수가 전체 몇 개인지 출력하시오

답)

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

설명)

grep의 옵션

- c : 패턴이 일치하는 행의 수를 출력한다.
- i : 비교시 대소문자를 구별하지 않는다.
- v : 지정한 패턴과 일치하지 않는 행만 출력한다.
- n : 행의 번호를 함께 출력한다.
- l : 패턴이 포함된 파일의 이름을 출력한다.
- w : 패턴이 전체 단어와 일치하는 행만 출력한다.

문제 33) R에서 elsa가 포함되어져 있는 라인의 개수를 출력하시오

답)

cafe.daum.net/oracleoracle 카페/생각해야할문제 안에

R만 쳐도 R이 실행될 수 있도록 설정하는 방법

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

어떤 거라도 bin 디렉토리가 있는데 bin 디렉토리에 실행프로그램이 있다

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

위와 같이 하면 번거롭기 때문에
먼저 /home/oracle/R-3.2.3/ 를 복사해준다.

그리고 다시 cd로 되돌아간다

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

```
[orcl:~]$ vi .bash_profile
```

vi : 파일을 여는 명령어

.bash_profile : oracle 유저로 접속했을때 자동으로 실행되는 파일

```
# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export JAVA_HOME=/usr/java/jdk1.6.0_18
export PATH=$JAVA_HOME/bin:$PATH
export PS1='[\`echo $ORACLE_SID`:\W]$ '

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

위와 같이 쓴다.

위에서 export에 x에 커서를 두고 ESC 키를 누르고 알파벳 o를 누르면 아래에 쓸 수 있게 공간이 생긴다

거기 export PATH=/home/oracle/R-3.2.3/bin:\$PATH를 써준다

그리고 ESC키 눌러주고 :wq! 입력하고 엔터

그 다음 배쉬파일을 실행한다

```
[orcl:~]$ . .bash_profile
[orcl:~]$ R
```

15. sort 명령어

" 데이터를 특정 컬럼을 기준으로 정렬하는 명령어 "

sort

sort 명령은 사용자가 지정한 파일의 내용을 정렬하거나, 정렬된 파일의 내용을 병합(Merge)할 때 사용할 수 있다.
> sort[-옵션][-o 저장될 파일명]정렬할 파일명[-m 병합할 파일명....]

옵션	설명
-n	라인의 각 필드를 비교하는 대상을 숫자로 한정
-f	영어를 정렬할 때, 대소문자 구별안함
-r	출력 순서를 역순으로
-b	앞에 붙는 공백 무시
-t	필드 구분자 지정
-m	정렬된 파일을 병합
-u	정렬후, 중복행 제거
-o	저장할 파일명을 명시, 명시하지 않으면 화면에 출력

예)

```
[orcl:~]$ sort -k 6 emp.txt
```

결과)

7369 SMITH	CLERK	7902 17-DEC-80	800	0	20
7900 JAMES	CLERK	7698 03-DEC-81	950	0	30
7876 ADAMS	CLERK	7788 23-MAY-87	1100	0	20

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

문제 34) 위의 결과를 다시 출력하는데 이름과 월급만 나오게 하시오

답)

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

문제 35) 이름과 월급을 출력하는데 월급이 높은 것부터 출력되게 하시오

답)

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

결과)

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

문제 36) 직업이 salesman인 직원들의 이름과 월급을 출력하는데 월급이 높은 것부터 출력하시오

답)

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

결과)

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

문제 37) 직원 테이블의 월급의 토탈 값을 출력하시오

답)

```
[orcl:~]$ awk '{print $6}' emp.txt | awk '{sum += $1} END {print sum}'
29025
```

설명)

sum += \$1 은 sum = sum + \$1과 같다

문제 38) 직업이 SALESMAN인 직원들의 토탈 월급을 구하시오

답)

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

문제 39) 아래와 같이 결과가 출력되게 하시오

보기)

```
KING의 월급은 5000
ALLEN의 월급은 3000
.
.
```

답)

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

문제 40) 아래와 같이 결과를 출력하시오

보기)

```
SMITH의 직업은 CLERK입니다
ALLEN의 직업은 SALESMAN입니다
WARD의 직업은 SALESMAN입니다
JONES의 직업은 MANAGER입니다
.
.
```

답)

```
[orcl:~]$ awk '{print $2 "의 직업은 " $3 "입니다"}' emp.txt
```

문제 41) /home/oracle 에서 ls -lrt라고 조회한 결과를 분석하시오

결과)

```
[orcl:~]$ ls -lrt
total 29344
drwxr-xr-x 6 oracle oinstall 4096 Oct 13 2012 Adobe
drwxr-xr-x 4 oracle oinstall 4096 Oct 16 2012 labs
drwxr-xr-x 3 oracle oinstall 4096 Oct 16 2012 oradiag_oracle
-rw-r--r-- 1 oracle oinstall 2731 Oct 16 2012 datamodeler.log
drwx----- 2 oracle oinstall 4096 Nov 13 2012 Desktop
-rw-r--r-- 1 oracle oinstall 29816176 Dec 10 2015 R-3.2.3.tar.gz
drwxr-xr-x 15 oracle oinstall 4096 Jun 21 16:51 R-3.2.3
-rw-r--r-- 1 oracle oinstall 679 Jun 21 17:04 emp.csv
drwxr-xr-x 2 oracle oinstall 4096 Jun 22 10:25 test2
drwxr-xr-x 3 oracle oinstall 4096 Jun 22 10:28 test5
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:31 dept.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:31 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:34 emp40.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:36 a.txt
-rw-r--r-- 1 oracle oinstall 426 Jun 22 11:38 ls.txt
-rw-r--r-- 1 oracle oinstall 1940 Jun 22 11:42 emp50.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 22 11:47 winter.txt
-rw-r--r-- 1 oracle oinstall 3387 Jun 22 14:19 winter_head.txt
-rw-r--r-- 1 oracle oinstall 9134 Jun 22 14:41 sar_0622.txt
```

설명)

```
- l : 리스트  
- r : reverse  
- t : 시간  
파일 리스트를 보여주는데 오래된 것부터 보여달라  
  
-rw-r--r-- 1 oracle oinstall 1940 Jun 22 11:42 emp50.txt  
-rw-r--r-- : 권한  
1: 링크  
oracle : 소유자  
oinstall : 그룹  
1940 : 파일크기  
Jun 22 11:42 : 최종 변경시간  
emp50.txt : 파일명
```

문제 42) ls -lrt *.txt 했을 때 파일 크기에 해당하는 부분만 출력하시오

결과)

```
105  
970  
970  
0  
0  
0  
0  
426  
1940  
114548  
3387  
9134
```

답)

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

문제 43) 위 42번의 결과를 sum해서 출력하시오

답)

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


결과)

```
131480
```

문제 44) 아래와 같이 결과가 출력되게 하시오

보기)

현재 디렉토리에 있는 txt 파일들의 총 크기는 131480 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"}'
```

문제 45) 위의 스크립트를 쉘 스크립트로 저장해서 실행되게 하시오

답)

```
vi size.sh 치면 편집기가 뜬다.  
거기서 a를 누르고 위 44번의 답을 붙여넣고 ESC 키 여러 번 누른 다음  
:를 누르고 wq! 를 친다  
그리고 sh size.sh 치고 엔터!
```

결과)

```
[orcl:~]$ sh size.sh  
the size of the total file under the current directory is 131480 bytes.
```

16. uniq 명령어

" 중복된 라인을 제거하는 명령어 "

uniq

uniq 명령어는 중복된 내용의 행이 연속으로 있으면 하나만 남기고 삭제한다.
전체적으로 분산된 중복은 찾아내지 못한다. 따라서 정렬하여 순차적으로 만든 뒤에 적용한다. (sort와 같이 쓰기 좋음)
> **uniq [-옵션] [파일명]**

옵션	설명
-c	같은 라인이 몇번 나오는지 표시
-d	중복되어 나오는 라인 중 한 라인만 표시
-D	중복되는 모든 라인을 보여준다
-N	필터링은 무시할 라인을 정한다. 시작 라인부터 N번째 라인까지는 검사하지 않는다.
-i	중복 라인을 한 라인으로 생각하고 출력한다.
-u	중복 라인이 없는 것만 보여준다.
-w	N번째 문자까지만 비교대상으로 하여 uniq 명령을 수행
-s	N번째 문자까지는 비교대상에서 제외하고 uniq 명령을 수행
-f	N번째 필드를 비교대상에서 제외하고 uniq 명령을 수행

문제 46) emp.txt에서 직업만 출력하시오

답)

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

결과)

```
CLERK
SALESMAN
SALESMAN
.
.
```

문제 47) 위의 결과를 abcd 순으로 정렬해서 출력하시오

답)

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

문제 48) 위의 결과를 중복 제거해서 출력하시오

답)

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

문제 49) 부서번호를 출력하는데 중복 제거해서 출력하시오

답)

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

문제 50) emp.txt를 복사해서 emp7.txt를 생성하고 emp7.txt에 KING에 대한 데이터를 복사해서 맨 아래에 하나를 입력하고 저장하시오

답)

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

```
[orcl:~]$ cat emp7.txt
```

```
[orcl:~]$ vi emp7.txt
```

King에 커서를 두고 yy를 누른다 (복사) 그 다음 커서를 복사하고 싶은 곳에 두고 (제일 아래) p를 누른 다음 ESC키 여러 번 누르고 :wq! 치고 vi 편집기를 나온다

설명)

vi 편집기에서 dd : 한 줄 제거, yy : 복사, p : 붙여넣기, u : 실행취소

문제 51) emp7.txt에서 중복된 라인만 출력하시오

결과)

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

답)

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

문제 52) emp7.txt에서 중복된 라인을 제거하고 출력하시오

결과)

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

답)

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

17. echo 명령어

" 출력하고자 하는 글자를 출력할 때 사용하는 명령어 "

예)

```
[orcl:~]$ echo 'i am me'
```

설명)

```
echo -e " "
```

-e 옵션: 역슬래시(\)를 이스케이프 문자로서 쓸 수 있게 하는 옵션

예)

```
$ echo -e "\삭제하였습니다.\\"
"삭제하였습니다."
```

문제 52) 직업을 물어보게 하고 직업을 입력하면 해당 직업을 갖는 직원들의 이름과 직업이 출력되게 하시오

답)

```
$ vi job2.sh

a

#1/bin/bash

echo " "
echo -n "직업을 입력하세요 "
```

```
read job
echo "   "
grep -i $job emp.txt | awk '{print $2, $3}'
```

저장하고 나와서
\$ sh job2.sh

설명)

```
#1/bin/bash : 쉘 스크립트 중에 bash 스크립트를 쓰겠다
echo "   " : 공백 출력
echo -n "직업을 입력하세요 " : 메시지 출력 후 엔터가 수행된 것
read job : job 이라는 변수에 입력한 직업이 들어간다
```

문제 53) 부서번호를 물어보게 하고 부서번호를 입력하면 해당 부서번호에서 근무하는 직원들의 이름과 월급과 부서번호가 출력되게 하세요

보기)

```
$ sh deptno2.sh
```

부서번호를 입력하세요: 10

답)

```
#1/bin/bash

echo "   "
echo -n "부서번호를 입력하세요: "
read deptno
echo "   "
awk '{print $2, $6, $8}' emp.txt | grep -w $deptno
```

문제 54) 이름을 물어보게 하고 이름을 입력하면 아래와 같이 출력되게 하시오

보기)

```
vi find_sal.sh
```

이름을 입력하세요: scott
SCOTT의 월급은 3000 입니다.

답)

```
#1/bin/bash

echo " "
echo -n "이름을 입력하세요: "
read name
echo " "
grep -i $name emp.txt | awk '{print $2, "의 월급은 ", $6, "입니다"}'
```

문제 55) 직업을 물어보게 하고 직업을 입력하면 아래와 같이 출력되게 하시오

보기)

```
$ sh job_sumsal.sh

직업을 입력하세요: salesman
총 월급은 5600 입니다
```

답)

```
#1/bin/bash

echo " "
echo -n "직업을 입력하세요: "
read job
echo " "
grep -i $job emp.txt | awk '{sum += $6} END {print "총 월급은 ", sum, " 입니다"}'
```

문제 56) 입사일을 출력하는데 뒤에 년도만 잘라서 출력하시오

답)

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

문제 57) 년도를 물어보게 하고 년도를 입력하면 해당 년도에 입사한 직원들의 이름과 입사일이 출력되게 하시오

보기)

```
$ sh year.sh
```

입사년도를 입력하세요: 81

답)

```
#1/bin/bash

echo " "
echo -n "입사년도를 입력하세요: "
read hiredate
echo " "
grep -w $hiredate emp.txt | awk '{print $2, $5}'
```

문제 58) (점심시간) 년도를 물어보게 하고 년도를 입력하면 해당 년도의 토탈월급이 출력되게 하시오

보기)

```
$ sh year_sum.sh
```

년도를 입력하세요: 81

토탈 월급은 10290 입니다.

답)

```
#1/bin/bash

echo " "
echo -n "입사년도를 입력하세요: "
read hiredate
echo " "
grep -w $hiredate emp.txt | awk '{sum += $6} END {print "토탈월급은 ", sum, " 입니다."}'
```

18. diff 명령어

" 두 파일 간의 차이점을 찾아서 알려주는 명령어 "

예)

```
[orcl:~]$ diff emp.txt emp7.txt
```

```
14a15
```

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

문제 59) 아래와 같이 두 개의 파일 이름을 각각 물어보게 하고 결과로 파일의 내용의 차이가 출력되게 하시오

보기)

```
$ sh diff.sh

비교할 첫번째 파일명을 입력하세요: emp.txt

비교할 두번째 파일명을 입력하세요: emp7.txt

14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```

답)

```
#1/bin/bash

echo " "
echo -n "비교할 첫번째 파일명을 입력하세요: "
read file1
echo " "
echo -n "비교할 두번째 파일명을 입력하세요: "
read file2
echo "  "
diff $file1 $file2
```

19. find 명령어

" 검색하고 싶은 파일을 찾을 때 사용하는 명령어 "

예)

```
$ find 디렉토리 -name 파일명 -print

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

문제 60) /home/oracle 밑에 test100과 test200이라는 디렉토리를 생성하고 그 밑에 emp.txt를 각각 복사하시오

답)


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

[orcl:~]$ cp emp.txt ./test100/emp.txt
[orcl:~]$ cp emp.txt ./test200/emp.txt
```

문제 61) /home/oracle 밑에 있는 emp.txt를 검색하시오

답 1)

```
[orcl:~]$ find ~ -name emp.txt -print
/home/oracle/test100/emp.txt
/home/oracle/test200/emp.txt
/home/oracle/emp.txt
```

답 2)

```
[orcl:~]$ find . -maxdepth 1 -name 'emp.txt'
./emp.txt
[orcl:~]$ find . -maxdepth 2 -name 'emp.txt'
./test100/emp.txt
./test200/emp.txt
./emp.txt
```

문제 62) 파일을 검색하는 쉘 스크립트를 아래와 같이 생성하시오

보기)

```
$ sh find_file.sh

현재 디렉토리 밑에 검색할 파일명을 입력하세요: emp.txt

depth 값: 2
```

결과)

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

답)

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

```
echo " "
echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요: "
read file1
echo " "
echo -n "depth 값: "
read depth
echo " "
find . -maxdepth $depth -name $file1
```

20. tar 명령어

" 파일을 압축하거나 압축 해제하는 명령어 "

예)

1. 압축할 때
\$ tar cvf 압축파일명 압축파일대상
2. 압축 해제할 때
\$ tar xvf 압축파일명 압축해제할 위치

옵션)

-c	compress 여러 개의 파일을 하나로 만들어라
-v	view, 압축되는 과정을 보여달라
-f	file, 생성되는 파일명을 지정
-x	extract, 묶여있는 파일을 풀어줘라
-C	압축 풀릴 위치를 지정

문제 63) /home/oracle 밑에 test63 이라는 디렉토리를 만들고 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
```

문제 64) test63 디렉토리 밑에 있는 확장자 .txt 파일들을 all.tar라는 이름으로 압축하시오

답)

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

결과)

```
[orcl:~]$ ls -l
total 29408
drwxr-xr-x  6 oracle oinstall    4096 Oct 13  2012 Adobe
drwx-----  2 oracle oinstall    4096 Nov 13  2012 Desktop
drwxr-xr-x 15 oracle oinstall    4096 Jun 21 16:51 R-3.2.3
-rw-r--r--  1 oracle oinstall 29816176 Dec 10  2015 R-3.2.3.tar.gz
-rw-r--r--  1 oracle oinstall      0 Jun 22 11:36 a.txt
-rw-r--r--  1 oracle oinstall 10240 Jun 25 14:25 all.tar
```

문제 65) /home/oracle 밑에 test65라는 디렉토리를 생성하고 그 밑에 all.tar 압축파일의 압축을 해제하시오

답)

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

21. ln (link) 명령어

윈도우의 바로가기 기능과 유사함

" 내가 자주 열어봐야 하는 파일이 있다면 바로가기를 생성해 놓아 편하게 열어볼 수 있는 기능 "

예)

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

```
[orcl:~]$ ln -s /u01/app/oracle/product/11.2.0/dbhome_1/sqlplus/admin/glogin.sql glogin.sql
[orcl:~]$ ls
```

Adobe	c.txt	emp.txt	job2.sh	test100	winter_head.txt
Desktop	datamodeler.log	emp40.txt	job_sumsal.sh	test2	year.sh
R-3.2.3	dept.txt	emp50.txt	labs	test200	year_sum.sh
R-3.2.3.tar.gz	deptno2.sh	emp7.txt	ls.txt	test5	
a.txt	diff.sh	find_file.sh	oradiag_oracle	test63	
all.tar	e.txt	find_sal.sh	sar_0622.txt	test65	
b.txt	emp.csv	glogin.sql	size.sh	winter.txt	

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

```
--
-- Copyright (c) 1988, 2005, Oracle. All Rights Reserved.
--
-- NAME
--   glogin.sql
--
-- DESCRIPTION
--   SQL*Plus global login "site profile" file
--
--   Add any SQL*Plus commands here that are to be executed when a
--   user starts SQL*Plus, or uses the SQL*Plus CONNECT command.
--
-- USAGE
--   This script is automatically run
--
define_editor='gedit'
~
~
~
~
~
~
~
~
~
```

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

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Mon Jun 25 14:42:11 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
```

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

```
SQL> ed
```

22. .bash_profile 설정하는 방법

oracle 유저를 위한 리눅스 환경설정 파일

예)

```
[orcl:~]$ vi .bash_profile
```

```
oracle@edydr1p0:~ - Xshell 4
파일(F) 편집(E) 보기(V) 도구(T) 창(W) 도움말(H)
새로 만들기 폴더 다시 연결
1 Linux x
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin
export PATH
export JAVA_HOME=/usr/java/jdk1.6.0_18
export PATH=$JAVA_HOME/bin:$PATH
export PS1='[`echo $ORACLE_SID`:##W]$ '
export ORACLE_BASE=/u01/app/oracle
export ORACLE_SID=orcl
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=/home/oracle/R-3.2.3/bin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH
unset LANG
echo -e "여기는 운영서버입니다. 조심히 작업하세요~ "
~
~
```

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

Password:

여기는 운영서버입니다. 조심히 작업하세요~

문제 66) oracle 유저로 접속할 때마다 아래의 메시지가 출력되게 하시오

보기)

저한테 왜 그러시는 건데요 ——

답)

```
[orcl:~]$ vi .bash_profile
```

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

Password:

여기는 운영서버입니다. 조심히 작업하세요~

저한테 왜 그러시는 건데요 ——

설명)

su - oracle : 오라클로 접속하는 명령어

23. sed 명령어

" grep 명령어는 파일의 특정 내용을 검색하는 기능을 갖는다면
sed 명령어는 검색 뿐만 아니라 내용을 변경할 수도 있다 "

예)

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

emp.txt 를 출력할 때 KING을 yyy로 변경해서 출력한다
KING을 yyy로 스위치하겠다
실제로는 바뀌지 않고 출력되는 화면에서만 바뀐 것

문제 67) 위의 변경된 내용대로 출력한 내용을 emp9.txt로 저장하시오

답)

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

24. cp 명령어

" 파일을 복사하는 명령어 "

예)

```
$ cp 위치/원본 파일명 위치/복사할 파일명
```

백업할 때 쓰는 중요한 기능

문제 68) 리눅스의 oracle 계정을 위한 환경설정파일인 .bash_profile을 백업하시오

답)

```
[orcl:~]$ cp .bash_profile .bash_profile_backup  
[orcl:~]$ ls -a
```

문제 69) /home/oracle 밑에 있는 확장자 .txt 파일들 전부를 /home/oracle/backup 이라는 디렉토리에 복사하시오

답)

```
$ mkdir backup
```

```
$ cp /home/oracle/*.txt /home/oracle/backup/
```

25. mv 명령어

" 파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어 " move

예)

```
$ mv [기존 파일명] [새로운 파일명 또는 파일 위치]
[orcl:~]$ mv emp400.txt emp500.txt
[orcl:~]$ mv emp.txt ./backup/
```

문제 70) /home/oracle/backup 밑에 있는 emp.txt를 지금 현재 디렉토리인 /home/oracle 밑으로 이동시키시오

답)

```
[orcl:~]$ cd backup
[orcl:backup]$ mv emp.txt ~
[orcl:backup]$ cd ..
[orcl:~]$ ls
```

04 vi 편집기 명령어

2018년 6월 21일 목요일 오후 4:21

vi 편집기란?

리눅스 안에서 사용할 수 있는 문서 편집기
vi (Visual Editor)

vi 편집기의 명령모드 3가지

1. command 모드
vi의 기본 모드이며 vi를 실행하면 바로 보이는 기본화면
방향키로 왔다갔다 가능
2. edit 모드
a, i, o 등을 누르면서 내용을 입력 또는 삭제하는 명령모드
3. last line 모드
입력모드에서 저장 종료, 강제 종료 등의 명령어
.wq!

예)

```
[orcl:~]$ vi aaa.txt
```

편집기를 시작하는 명령어

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

vi 편집기를 열 때 정상적으로 열리지 않고 이상한 경고 메시지들이 나오면서 열릴 때 해결방법

```

E325: ATTENTION
Found a swap file by the name ".aaa.txt.swp"
    owned by: oracle   dated: Mon Jun 25 16:20:14 2018
    file name: ~oracle/aaa.txt
    modified: no
    user name: oracle   host name: edydr1p0.us.oracle.com
    process ID: 11246 (still running)
While opening file "aaa.txt"
    dated: Mon Jun 25 16:18:25 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.txt"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".aaa.txt.swp"
    to avoid this message.

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

```

위 처럼 나올때를 말한다.

위에 aaa.txt.swp이라는 것을 확인하고 저 파일을 지워준다. (임시로 만들어 놓은 파일)

```

[orcl:~]$ ls -a .*
.aaa.txt.swp

.adobe:
.  .. Acrobat Flash_Player Linguistics
[orcl:~]$ rm .aaa.txt.swp

[orcl:~]$ vi aaa.txt

```

vi 편집기 내에서 커서 이동

```

$ vi winter.txt

1. j : 아래로 이동
2. k : 위로 이동
3. h : 왼쪽으로 이동
4. l : 오른쪽으로 이동
5. G : 제일 아래로 이동
6. 1G : 제일 위로 이동
7. :set nu : 파일 내의 텍스트에 번호 표시
8. :set nonu : 번호 안 보이게 하는 방법

```

문제 71) 겨울왕국 대본의 1 ~ 200번째 내용을 winter200.txt 라는 이름으로 저장하시오

답)

```
[orcl:~]$ vi winter.txt
:set nu
:1,200 w winter500.txt
:wq!
```

결과)

```
[orcl:~]$ ls -l win*
/home/oracle/ 에 저장됨
```

vi 편집기의 입력 명령어

1. i : 커서의 왼쪽부터 입력하겠다
2. a : 커서의 오른쪽부터 입력하겠다
3. o : 커서의 다음라인부터 입력하겠다
4. O : 커서의 윗라인부터 입력하겠다

vi 편집기의 삭제 명령어

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

vi 편집기의 취소 명령어

u : 방금 작업한 거 취소하겠다

복사/붙여넣기 명령어

1. yy : 하나의 행을 복사
2. p : 붙여 넣기
3. yG : 현재 행부터 파일 끝까지 복사
4. :1,2 co 3 : 1~2행을 3행 다음으로 복사
5. :1,2 m 3 : 1~2행을 3행 다음으로 이동

문제 72) 겨울왕국 대본의 전체 라인 수를 먼저 확인하고 겨울왕국 대본 전체를 복사해서 아래의 붙여넣고 겨울왕국 대본의 라인 수가 2배가 되었는지 확인하시오

답)

```
$ vi winter.txt

:set nu
```

커서 첫글자에 두고 yG

커서 마지막에 두고 p

G 로 2배가 되었는지 확인

문자 변경하는 명령어

1. ./문자 : 문자 검색

예)

[orcl:~]\$ vi winter.txt

:/ ELSA

n : 다음

Shift + n : 다시 앞으로 검색

2. :s/기존문자/변경할 문자

" 지금 커서가 있는 현재행의 기존문자를 변경할 문자로 변경하겠다 "

3. :%s/기존문자/변경할 문자/g

" 문서 전체의 기존 문자를 변경할 문자로 변경하겠다 "

문제 73) emp.txt를 emp750.txt로 복사하고 emp750.txt를 열어서 SALESMAN을 전부 CLERK으로 변경하시오

답)

```
[orcl:~]$ cp emp.txt emp750.txt
```

```
[orcl:~]$ vi emp750.txt
```

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

```
:wq!
```

(Write and quit)

문제 74) emp.txt를 emp1.txt ~ emp20.txt로 복사하시오

답)

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

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

```
i=1  
while [ $i -le 20 ]  
do  
    cp emp.txt emp$i.txt  
    let i=i+1  
done
```

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

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

문제 75) emp1.txt ~ emp20.txt의 내용을 변경하는데 ANALYST를 jjj로 변경하시오

답)

```
$ vi emp*.txt  
:argdo %s/ANALYST/jjj/g | update
```

ZZ : vi편집기에서 저장하고 나오는 간단한 방법

**문제 76) winter.txt 대본을 100개로 복사하고 winter.txt ~ winter100.txt로 복사한다
winter.txt ~ winter100.txt의 내용 중에 by를 bybybyby로 한번에 변경하시오**

답)

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

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

```
i=1  
while [ $i -le 100 ]  
do  
    cp winter.txt winter$i.txt  
    let i=i+1  
done
```

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

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

```
[orcl:~]$ vi winter*.txt
:argdo %s/by/bybybyby/g | update
```

설명)

vi cp_winter.sh 에서
while 문 안에

숫자1 -eq 숫자2	두 숫자가 같으면	TRUE
숫자1 -ne 숫자2	두 숫자가 같지 않으면	TRUE
숫자1 -gt 숫자2	숫자1이 숫자2보다 크다면	TRUE
숫자1 -ge 숫자2	숫자1이 숫자2보다 크거나 같으면	TRUE
숫자1 -lt 숫자2	숫자1이 숫자2보다 작다면	TRUE
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면	TRUE
!숫자1	숫자1이 거짓이라면	TRUE

문제 77) winter1.txt ~ winter100.txt를 전부 삭제하시오

답)

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

```
#!/bin/bash

i=1
while [ $i -le 100 ]
do
    rm winter$i.txt
    let i=i+1
done
```

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

```
[orcl:~]$ ls -l winter*
```

05 권한 관리

2018년 6월 21일 목요일 오후 4:21

권한 관리를 잘 알아야 하는 이유

리눅스에서 파이썬 배치파일을 실행하거나 R 배치 파일을 실행하거나 하둡을 설치하고 운영을 할 때 여러가지 문제들이 발생하는데 그 중에 가장 많은 문제들이 권한에 관한 오류들이 많다.

권한관리에 대한 명령어 3가지

1. [chmod 명령어](#)
2. [chattr 명령어](#)
3. [chown 명령어](#)

권한의 종류

1. 읽기 권한
2. 쓰기 권한
3. 실행 권한

권한 관리표

번호	권한	대표문자	파일	디렉토리
1	읽기 권한	r	읽고, copy	디렉토리에서 ls 실행 가능
2	쓰기 권한	w	수정	디렉토리에서 파일을 생성 가능
3	실행 권한	x	실행	디렉토리에 cd로 접근 가능

ls -l로 특정 파일을 조회했을 때 나오는 권한 부분을 해서

```
$ ls -l emp.txt
```

```
-rw-r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

권한 부분을 좀더 상세하게 분석

-	디렉토리인지 파일인지를 나타냄
rw-	파일의 소유자의 권한
r--	그룹의 권한
r--	기타 유저의 권한

권한 대상

u	유저
g	그룹

chmod 명령어

문제 78) emp.txt의 소유자의 권한을 읽기만 가능하도록 수정하시오

보기)

```
[orcl:~]$ ls -l emp.txt
-rw-r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

답)

```
[orcl:~]$ chmod u-w emp.txt
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

설명)

u : 유저
g : 그룹
o : 기타 유저

r : 읽기
w : 쓰기
x : 실행

read only 파일도 '!' 를 붙이고 저장하면 저장된다

문제 79) emp.txt의 소유자의 권한에 읽기 권한도 없애시오

답)

```
[orcl:~]$ chmod u-r emp.txt
[orcl:~]$ ls -l emp.txt
---r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt

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

문제 80) emp.txt의 소유자의 권한에 읽기, 쓰기, 실행권한을 넣으시오

답)

```
[orcl:~]$ chmod u+rwx emp.txt
[orcl:~]$ ls -l emp.txt
-rwxr--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

문제 81) emp.txt에 대해서 소유자, 그룹, 기타 유저에 대해서 읽기, 쓰기, 실행 권한을 모두 주시오

답)

```
[orcl:~]$ chmod ugo+rwx emp.txt
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

답 2)

```
[orcl:~]$ chmod a+rwx emp.txt
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

문제 82) 오라클 유저는 emp.txt에 대해서 무조건 읽기만 할 수 있게 하고 쓰지는 못하게 하며 그룹과 다른 기타 유저에 대해서는 읽기, 쓰기, 실행 권한을 모두 주지 마시오

답)

```
[orcl:~]$ chmod u+r-wx,go-rwx emp.txt
[orcl:~]$ ls -l emp.txt
-r----- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

답 2)

```
[orcl:~]$ chmod a-rwx,u+r emp.txt
```

chattr 명령어

문제 83) oracle 유저가 emp.txt에 대해서 chmod 명령어 자체를 수행 못하게 하시오

답)

root 유저로 접속한다 (최상위 유저)

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



```
root
[root@edydr1p0 ~]# pwd
/root
[root@edydr1p0 ~]# cd /home/oracle/
[root@edydr1p0 oracle]# ls -l emp.txt
-r----- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
[root@edydr1p0 oracle]# chattr +i emp.txt
```

다시 oracle 유저로 접속해서 chmod 명령어를 emp.txt에 수행해본다

```
[root@edydr1p0 oracle]# su - oracle
여기는 운영서버입니다. 조심히 작업하세요~
저한테 왜 그러시는 건데요 --
[orcl:~]$ whoami
oracle
[orcl:~]$ chmod u+rx emp.txt
chmod: changing permissions of `emp.txt': Operation not permitted
```

다시 푸는 방법)

```
[orcl:~]$ su -
Password:
[root@edydr1p0 ~]# whoami
[root@edydr1p0 ~]# cd /home/oracle/
[root@edydr1p0 oracle]# chattr -i emp.txt
[root@edydr1p0 oracle]# su - oracle
```

여기는 운영서버입니다. 조심히 작업하세요~
저한테 왜 그러시는 건데요 --

```
[orcl:~]$ chmod u+rx emp.txt
[orcl:~]$ ls -l emp.txt
```

```
-rwx----- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

숫자로 권한을 설정하는 방법

```
0 ---> 퍼미션을 허용하지 않겠다
1 ---> 실행(x) 퍼미션을 허용하겠다
2 ---> 쓰기(w) 퍼미션을 허용하겠다
4 ---> 읽기(r) 퍼미션을 허용하겠다
```

예)

```
$ chmod u+rx,g+rw,o+r emp.txt
```

위의 명령어를 숫자로 권한 설정해보자

```
$ chmod 764 emp.txt
```

문제 84) emp.txt에 대해서 아래와 같이 권한 설정이 될 수 있도록 하시오

보기)

```
-r--r--r--
```

답)

```
[orcl:~]$ chmod 444 emp.txt
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

문제 85) 겨울왕국 대본의 권한을 -rwxr--r-- 로 변경하시오

답)

```
[orcl:~]$ chmod 744 winter.txt
[orcl:~]$ ls -l winter.txt
-rwxr--r-- 1 oracle oinstall 114820 Jun 26 10:16 winter.txt
```

점심시간 문제를 위한 명령어

```
겨울왕국 대본에 off라는 단어가 몇 번 나오는지 카운트 하시오
[orcl:~]$ grep -w -o -i 'off' winter.txt | wc -l
65
```

문제 86) 겨울왕국에는 elsa가 많이 나오는가 anna가 많이 나오는가

답)

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

문제 87) (점심시간 문제) R로 접속해서 elsa가 329개이고 anna가 678개인지 확인하시오

답)

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

chown 명령어

" 파일이나 디렉토리의 소유자를 변경하는 명령어 "

예)

```
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 962 Jun 26 10:08 emp.txt
[orcl:~]$ chmod 777 emp.txt
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 962 Jun 26 10:08 emp.txt
[orcl:~]$ su -
Password: oracle
[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 962 Jun 26 10:08 emp.txt
```

문제 88) 그럼 다시 emp.txt의 소유자를 oracle로 변경하고 그룹을 oinstall로 변경하시오

답)

```
[orcl:~]$ su -
Password:
[root@edydr1p0 ~]# cd /home/oracle/
[root@edydr1p0 oracle]# chown oracle:oinstall emp.txt
[root@edydr1p0 oracle]# exit
logout
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 962 Jun 26 10:08 emp.txt
```

문제 89) /home/oracle/ 밑에 bbb라는 디렉토리를 만들고 bbb라는 디렉토리 밑에 /home/oracle 밑에 있는 확장자 txt 파일들을 전부 copy하시오

답)

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

설명)

```
cp [복사할 파일] [복사할 위치]
```

```
[orcl:bbb]$ ls -ld /home/oracle/bbb
drwxr-xr-x 2 oracle oinstall 4096 Jun 26 14:09 /home/oracle/bbb
오라클에서 만들었기 때문에 디렉토리의 소유자가 oracle이다
```

문제 90) /home/oracle 밑에 bbb 디렉토리의 소유자를 root 유저로 변경하시오

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 14:09 /home/oracle/bbb
```

설명)

```
-R 옵션 : 그 디렉토리 안에 있는 모든 파일들의 소유자를 싹 변경하겠다
-R 옵션을 붙이지 않으면 디렉토리 소유자만 변경되고 파일들은 변경되지 않는다
```

문제 91) 다시 /home/oracle/bbb 디렉토리의 소유자를 oracle 유저, 그룹을 oinstall로 변경하시오

답)

```
[orcl:~]$ su -
Password:
[root@edydr1p0 ~]# cd /home/oracle/
[root@edydr1p0 oracle]# chown -R oracle:oinstall /home/oracle/bbb
[root@edydr1p0 oracle]# exit
```

logout

[orcl:~]\$ cd bbb

[orcl:bbb]\$ ls -ld .

drwxr-xr-x 2 oracle oinstall 4096 Jun 26 14:09 .

06 디스크 관리

2018년 6월 21일 목요일 오후 4:21

디스크 관리 명령어 3가지

1. [du 명령어](#)
2. [df 명령어](#)
3. [sar 명령어](#)

du 명령어

" 현재 파일/디렉토리의 디스크 사용량을 확인하는 명령어 "

```
[orcl:~]$ du -sh
```

```
682M
```

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

```
0      a.txt
4      aaa.txt
0      b.txt
0      c.txt
4      dept.txt
0      e.txt
4      emp.txt
4      emp40.txt
4      emp50.txt
4      emp500.txt
4      emp750.txt
4      ls.txt
12     sar_0622.txt
116    winter.txt
8      winter500.txt
4      winter_head.txt
```

문제 92) 위의 txt 파일의 총 크기를 출력하시오

답)

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

```
172
```

답 2)

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

df 명령어

" 현재 파일 시스템의 총 사용율을 확인하는 명령어 "

예)

```
[orcl:~]$ df -hP
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00  190G  45G  136G  25% /
/dev/sda1        99M   18M   76M   20% /boot
tmpfs            1014M  442M  572M   44% /dev/shm
```

sar 명령어

" 디스크의 i/o 성능을 모니터링 하는 명령어 "

예)

```
[orcl:~]$ sar 1 50
```

Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/26/18

	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:32:00							
14:32:01	all	0.00	0.00	1.49	0.00	0.00	98.51
14:32:02	all	0.50	0.00	1.00	0.00	0.00	98.50
14:32:03	all	0.00	0.00	1.99	0.50	0.00	97.51
14:32:04	all	0.00	0.00	3.50	0.00	0.00	96.50
14:32:05	all	0.00	0.00	1.00	0.00	0.00	99.00
14:32:06	all	0.50	0.00	0.50	0.00	0.00	99.00
14:32:07	all	0.00	0.00	1.00	0.00	0.00	99.00
14:32:08	all	0.00	0.00	1.99	1.00	0.00	97.01
14:32:09	all	0.00	0.00	0.50	0.00	0.00	99.50
14:32:10	all	0.50	0.00	1.99	0.00	0.00	97.51
14:32:11	all	0.00	0.00	1.51	0.00	0.00	98.49
14:32:12	all	0.00	0.00	1.00	0.00	0.00	99.00
14:32:13	all	0.00	0.00	3.00	0.00	0.00	97.00
14:32:14	all	0.50	0.00	0.50	0.00	0.00	99.00

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 얼마나 대기하는지
%idle	작업을 안하고 있는지, 얼마나 게으른 상태를 보여줌
%steal	찾아봐야함 πππ

문제 93) sar의 user i/o를 R로 시각화하시오

답)

1. R로 접속한다

```
[orcl:~]$ R
```

2. 아래의 R코드의 plot 그래프를 확인하시오

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

3. 세션 복제 후 리눅스 sar 명령어에서 아래와 같이 %user 컬럼 부분만 가져오게 awk 명령어를 작성하시오

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

4. R에서 리눅스 명령어를 수행하시오

원래 세션으로 돌아와서

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

를 적는다.

5. 위의 결과를 re라는 변수에 담고 5번째 요소만 가져오시오

```
> re <- system("sar 1 1 | awk '{print $3}'", intern = TRUE)
> re[5]
```

```
[1] "0.50"
```

6. 아래와 같이 NA를 10번 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 m 변수에 첫번째 요소로 담으시오

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

```
[1] "0.50" NA NA NA NA NA NA NA NA NA
```

8. 위의 m 데이터를 가지고 plot 그래프를 그려보시오

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

9. 실시간으로 그래프가 이어질 수 있도록 m에 데이터를 담고 plot 그래프를 그리시오

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



```
> m
```

```
[1] "0.50" "0.50" NA    NA    NA    NA    NA    NA    NA
```

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

10. 위의 R코드에 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)
```

문제 94) 위의 스크립트를 리눅스에서 별도로 저장해놓고 평상시에 쉽게 수행할 수 있도록 하시오

답)

```
[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)
```

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

돌리고 복제한 세션에 들어가서 SCOTT 계정으로 sqlplus 들어간 뒤

악성 SQL 하나 날려준다.

```
> select count(*) from sh.sales, sh,sales;
```

확 디스크의 i/o의 성능이 확 busy해 지는 것을 볼 수 있다.

문제 95) 위의 함수를 응용해서 emp.csv의 salary의 데이터가 실시간으로 plot 그래프로 그려지게 하시오

답)

```
x11()
sar <- function(n) {
  m <- rep(NA, n)
  emp <- read.csv("emp.csv")
  for (i in 1:n) {
    re <- emp[i, 6]
    m[i] <- re
    Sys.sleep(1)
    plot(m, type = 'o', col = 'red', xlab = 'time', ylab = 'ratio')
  }
  rm(m)
}

sar(14)
```

문제 96) 삼성 주식 등락을 1년치 데이터를 가지고 실시간 plot 그래프를 그려보시오

답)

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

2. R을 키고

```
x11()
s_stock <- function(n) {
  m <- rep(NA, n)
  stock <- read.csv("S_stock.csv")
  for (i in 1:n) {
    re <- stock$s_price[i]
    m[i] <- re
    Sys.sleep(0.01)
    plot(m, type = 'o', col = 'red', xlab = 'time', ylab = 'ratio')
```

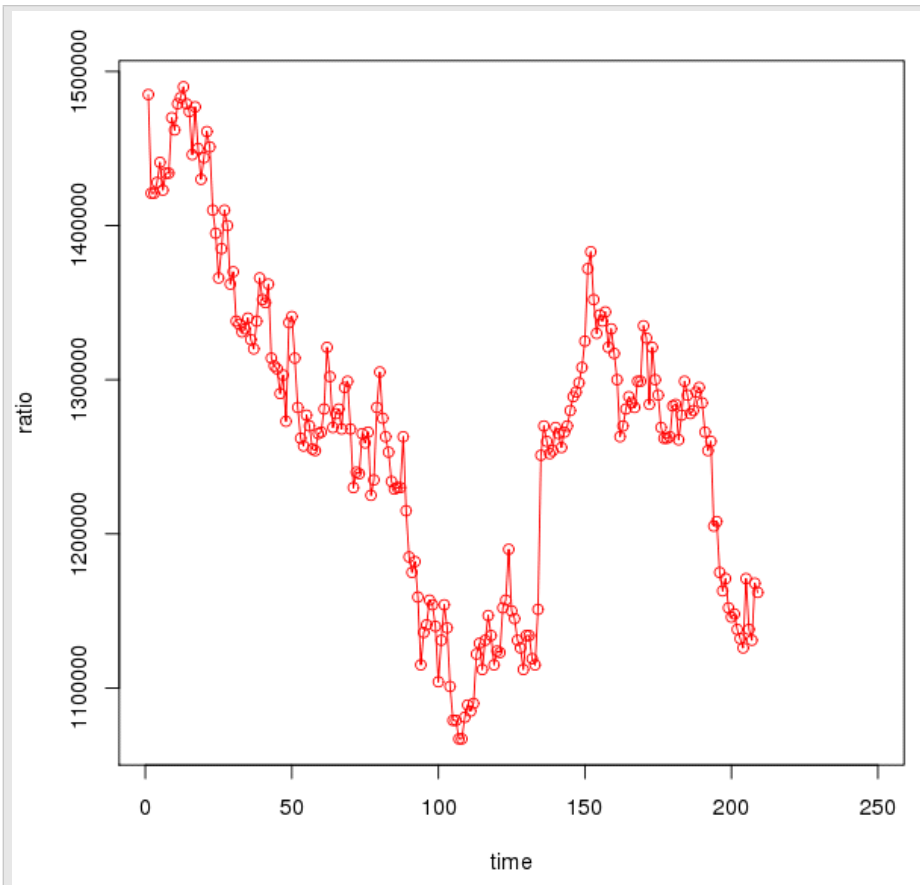
```

    }
    rm(m)
  }

```

```
s_stock(nrow(read.csv('S_stock.csv')))
```

결과)



문제 97) 현대 주식 등락을 1년치 데이터를 가지고 실시간 plot 그래프를 그려보시오

답)

```

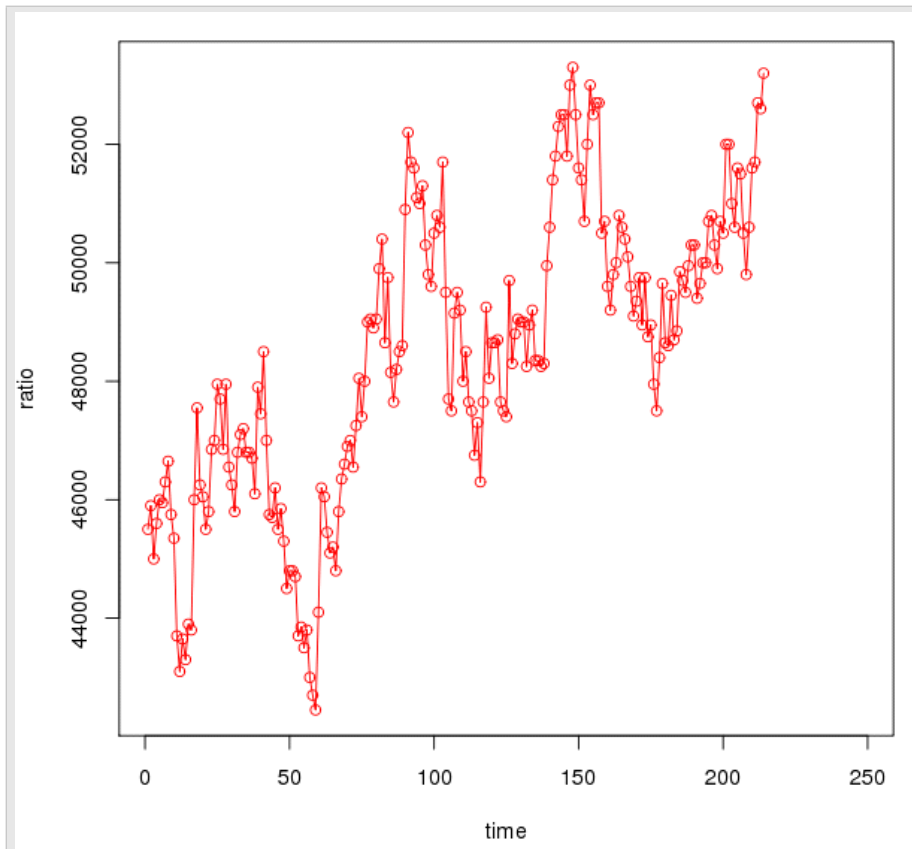
x11()
h_stock <- function(n) {
  m <- rep(NA, n)
  data <- read.csv("H_stock.csv")
  for (i in 1:n) {
    re <- data[i, 2]
    m[i] <- re
    Sys.sleep(0.1)
    plot(m, type = 'o', col = 'red', xlab = 'time', ylab = 'ratio')
  }
  rm(m)
}

```

```
}
```

```
h_stock(nrow(read.csv('H_stock.csv')))
```

결과)



문제 98) 지금 방금 작성한 스크립트를 이용해서 함수를 생성하는데 아래와 같이 함수를 실행시키면 csv 파일을 물어보게 하고 실시간 그래프가 그려지게 하시오

보기)

```
stock_fun()
```

입력파일을 로드하시오 : H_stock.csv

그래프

답)

```
stock_fun <- function() {  
  data <- read.csv(file.choose())  
  n <- nrow(data)  
  x11()  
  stock <- function(n) {  
    m <- rep(NA, n)
```

```

    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)
  }
  stock(n)
}

stock_fun()

```

문제 99) 리눅스에서 Rscript 명령어로 아래와 같이 수행되어서 실시간 그래프가 수행되게 하시오

보기)

```
$ sh s.sh
```

파일을 입력하세요: S_stock.csv

답)

1. s.R

```

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()

```

2. sh s.sh

```
#!/bin/bash
echo -n " 파일을 입력하세요 ~ "
read filecsv
Rscript /home/oracle/s.R $filecsv
```

3. 실행

```
$ sh s.sh
```

07 프로세서 관리

2018년 6월 21일 목요일 오후 4:21

프로세서 관리 명령어

1. [jobs 명령어](#)
2. [top 명령어](#)
3. [ps 명령어](#)
4. [kill 명령어의 옵션](#)

jobs 명령어

" 동작 중인 작업의 상태를 확인하는 명령어 "

예)

\$ jobs 옵션

상태 정보 4가지

1. running : 실행중
2. stopped : 일시 중단 중
3. done : 종료
4. terminated : 강제 종료

예)

```
[orcl:~]$ vi hhh.txt
```

vi 편집기에서

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

까지 쓰고 Ctrl키 + z 를 누르면

[1]	+	Stopped	vim hhh.txt
job 번호	현재 job	일시정지	동작 중인 job

일시 정지 했던 작업으로 돌아 가는 명령어

```
$ jobs
$ fg
```

문제 100) jobs 명령어를 수행하면 아래와 같이 결과가 출력되게 상황을 만드시오

보기)

```
[orcl:~]$ jobs
```

```
[1] Stopped          vim hhh.txt
[2]- Stopped          vim hhh2.txt
[3]+ Stopped          vim hhh3.txt
```

답)

```
[orcl:~]$ vi hhh.txt
```

vi 편집기에서 아무거나 쓰고 ctrl + z

```
[orcl:~]$ vi hhh2.txt
```

vi 편집기에서 아무거나 쓰고 ctrl + z

```
[orcl:~]$ vi hhh3.txt
```

vi 편집기에서 아무거나 쓰고 ctrl + z

문제 101) 위의 일시중단된 3가지 job 중에서 2번 job으로 fg(for ground) 하시오

답)

```
[orcl:~]$ jobs
```

```
[1] Stopped          vim hhh.txt
[2]- Stopped          vim hhh2.txt
[3]+ Stopped          vim hhh3.txt
```

```
[orcl:~]$ fg 2
```

top 명령어

" 지금 현재 작동 중인 프로세서들의 cpu 사용율과 메모리 사용율을 확인하는 명령어 "

```
[orcl:~]$ top
```



```
top - 16:47:18 up 17:26, 3 users, load average: 0.24, 0.26, 0.20
Tasks: 213 total, 2 running, 207 sleeping, 3 stopped, 1 zombie
Cpu(s): 0.0%us, 50.0%sy, 0.0%ni, 50.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2075456k total, 1944852k used, 130604k free, 165612k buffers
Swap: 4128760k total, 0k used, 4128760k free, 1454004k cached
Kill PID 21750 with signal [15]:
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
21750	oracle	15	0	2336	1076	792	R	121.2	0.1	0:00.08	top
1	root	15	0	2072	620	532	S	0.0	0.0	0:01.26	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.02	migration/0
3	root	34	19	0	0	0	R	0.0	0.0	0:00.01	ksoftirqd/0
4	root	RT	-5	0	0	0	S	0.0	0.0	0:00.06	watchdog/0
5	root	RT	-5	0	0	0	S	0.0	0.0	0:00.05	migration/1
6	root	34	19	0	0	0	S	0.0	0.0	0:00.05	ksoftirqd/1
7	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
8	root	10	-5	0	0	0	S	0.0	0.0	0:00.96	events/0
9	root	10	-5	0	0	0	S	0.0	0.0	0:17.91	events/1
10	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
11	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
15	root	10	-5	0	0	0	S	0.0	0.0	0:00.35	kblockd/0
16	root	10	-5	0	0	0	S	0.0	0.0	0:00.83	kblockd/1
17	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
55	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/0
56	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/1
59	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
61	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
130	root	19	0	0	0	0	S	0.0	0.0	0:00.00	pdf flush
131	root	15	0	0	0	0	S	0.0	0.0	0:02.32	pdf flush
132	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	kswapd0
133	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
134	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	aio/1
282	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	kpsmoused
320	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
321	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	ata/1
322	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	ata_aux
326	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	scsi eh 0

ps 명령어

" 현재 시스템에서 수행되고 있는 프로세서의 정보를 표시하는 명령어 "

예) \$ ps 옵션 프로세서 아이디

옵션	-e	현재 실행 중인 모든 프로세서
	-f	실제 유저명, 개시 시간 등을 표시
	-l	프로세서의 상태, 우선도 등과 같은 상세한 정보
	-p	프로세서 아이디

문제 102) 현재 리눅스 시스템에서 실행 중인 모든 프로세서를 출력하시오

답)

```
[orcl:~]$ ps -ef
[orcl:~]$ ps -ef | wc -l
```

결과)

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	?	00:00:23	[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]
root	55	11	0	Jun26	?	00:00:00	[cqueue/0]
root	56	11	0	Jun26	?	00:00:00	[cqueue/1]
root	59	11	0	Jun26	?	00:00:00	[khubd]
root	61	11	0	Jun26	?	00:00:00	[kseriod]
root	130	11	0	Jun26	?	00:00:00	[pdflush]
root	131	11	0	Jun26	?	00:00:00	[pdflush]

추가문제

세션을 복제하고 악성 SQL을 돌린 뒤 원래 세션에 돌아와서

```
[orcl:~]$ top
```

이라고 치면 제일 위에 %CPU가 가장 높은 걸 볼 수 있다. 그 PID를 기억 했다가
ctrl + c 로 나와서

```
[orcl:~]$ ps -ef | grep 23858
```

```
oracle  23858 23857 80 09:51 ?      00:01:19 oracleorcl (DESCRIPTION=(LOCAL=YES)
(ADDRESS=(PROTOCOL=beq)))
oracle  23884 23647  0 09:52 pts/1    00:00:00 grep 23858
```

악성 SQL 프로세스를 죽인다

```
[orcl:~]$ kill -9 23858
```

```
[orcl:~]$ top
```

kill 명령어의 옵션

\$ kill -0 프로세서번호

옵션	-9	프로세서를 죽인다
	-6	프로세서를 중단시킨다
	-2	프로세서에게 종료하도록 지시한다

문제 103) 지금 작업을 vim으로 수행하고 있는 프로세서들을 출력하시오

답)

복제한 세션에서

```
[orcl:~]$ vi aaaa2.txt
```

아무 작업이나 하다가

```
ctrl + z
```

```
[1]+  Stopped                  vim aaaa2.txt
```

원래 세션으로 돌아와서

```
[orcl:~]$ ps -ef | grep vim
```

```
oracle  23951 23813  0 09:56 pts/2    00:00:00 vim aaaa2.txt
```

```
oracle  23963 23647  0 09:58 pts/1    00:00:00 grep vim
```

```
[orcl:~]$ kill -9 23951
```

```
[orcl:~]$ ps -ef | grep vim
```

```
oracle  23974 23647  0 09:58 pts/1    00:00:00 grep vim
```

문제 104) kill 옵션 중 -2 옵션을 테스트하시오

터미널 창1

```
$ sar 1 10000
```

터미널 창2

```
$ ps -ef | grep sar
```

```
oracle  24010 23647  0 10:03 pts/1    00:00:00 sar 1 10000
```

```
oracle  24013 23813  0 10:03 pts/2    00:00:00 grep sar
```

```
$ kill -2 24010
```

```
$ ps -ef | grep sar
```

```
oracle  24022 23813  0 10:03 pts/2    00:00:00 grep sar
```

08 유저 생성 및 관리

2018년 6월 21일 목요일 오후 4:21

Cent OS 리눅스 설치 후

1. 호스트 이름

```
# hostname
```

2. 어떤 OS이고 버전이 어떻게 되는지 확인

```
# uname -an
```

09 셸 스크립트 작성법

2018년 6월 21일 목요일 오후 4:21

목차

09 셸 스크립트 작성법

- [셸\(Shell\)이란 무엇인가?](#)
- [셸 스크립트를 작성하고 실행하는 방법](#)
- [변수 사용법](#)
- [파라미터 변수](#)
- [IF문](#)
- [AND 와 OR 연산자](#)
- [LOOP문](#)
- [sed 명령어로 공백라인을 지우는 방법 \(데이터 정제 작업\)](#)
- [알고리즘 문제 \(factorial, power, 최대공약수\)](#)
- [대용량 데이터를 DB에서 직업 별로 각각의 txt 파일로 추출하기](#)
- [DB에서 데이터를 추출하는 방법 2가지](#)
- [While loop 문](#)
- [case 문](#)
- [swap file 생성되어 vi 편집기 실행 시 오류 해결 방법](#)
- [csv 파일 생성하는 스크립트 수정](#)

셸(Shell)이란 무엇인가?

운영체제에서 제공하는 명령을 실행하는 프로그램

셸(Shell) 스크립트란?

인터프리터 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

예) `#!/bin/bash` : 셸 중에 bash 셸을 쓰겠다

셸 스크립트 프로그래밍이란?

1. C 언어와 유사하게 프로그래밍이 가능
2. 변수, 반복문(loop문), 제어문(if문) 이 사용 가능
3. 별도로 컴파일하지 않고 텍스트 파일 형태로 바로 실행이 가능하다

4. vi나 gedit으로 작성 가능
5. 리눅스의 많은 부분이 쉘 스크립트로 작성되어 있다

쉘 스크립트를 작성하고 실행하는 방법

```
$ vi a.sh
echo "호스트 이름: " $HOSTNAME
$ sh a.sh
```

호스트 이름: edydr1p0.us.oracle.com

설명)

호스트 이름: 리눅스 컴퓨터의 이름

문제 105) a.sh에 소유자가 실행할 수 있도록 실행 권한을 부여하시오

보기)

```
$ ./a.sh
-bash: ./a.sh: Permission denied

$ ls -l a.sh
-rw-r--r-- 1 oracle oinstall 31 Jun 27 10:12 a.sh
```

답)

```
[orcl:~]$ chmod u+x a.sh
[orcl:~]$ ./a.sh
호스트 이름: edydr1p0.us.oracle.com
```

답 2)

```
$ chmod 744 a.sh
```

변수 사용법

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

예) \$ myvar="Hi ~~~~"

```
$ echo $myvar
```

변수의 숫자 계산하는 방법

1. 변수에 대입한 값은 모두 문자열로 취급이 된다
2. 변수에 들어있는 값을 숫자로 해서 사칙연산 등 연산을 하려면 `expr`을 사용해야 한다
3. 수식에 괄호 또는 곱하기(`*`)를 사용하려면 그 앞에 꼭 역슬래쉬(`\`)를 붙여야 한다

예)

```
$ num1=100
$ num2=200
$ echo $num1
$ echo $num2
$ echo $num1 + $num2 ( X )
$ expr $num1 + $num2 ( O )
```

결과)

```
[orcl:~]$ echo $num1+$num2
100+200
[orcl:~]$ echo $num1 + $num2
100 + 200
[orcl:~]$ expr $num1+$num2
100+200
[orcl:~]$ expr $num1 + $num2
300
```

➤ 공백에 민감하므로 주의할 것

문제 106) 위 예제에서 `num1`과 `num2`의 곱을 구하시오

답)

```
$ expr $num1 \* $num2
20000
```

문제 107) 아래의 계산식을 구현하시오

보기)

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

답)

```
$ expr W( $num2 + 200 W) W* $num1
```

40000

파라미터 변수

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

예)

```
vi b.sh
echo "실행파일 이름은 $0 입니다"
echo "첫번째 파라미터 값은 $1 입니다"
echo "두번째 파라미터 값은 $2 입니다"
echo "전체 파라미터 값은 $* 입니다"
sh b.sh 10 20
```

문제 108) 아래와 같이 두 정수를 파라미터 값으로 입력했을 때 나눈 나머지 값이 출력되게 하시오

보기)

```
$ sh f.sh 24 18
```

나눈 나머지 값은 6 입니다

답 1)

```
num1=$1
num2=$2
num3=`expr $num1 % $num2`
echo "나눈 나머지 값은 $num3 입니다"
```

답 2)

```
echo "$1을 $2로 나눈 나머지 값은 `expr $1 % $2` 입니다"
```

설명)

변수=`리눅스 명령어`

리눅스 명령어에 의해서 수행된 결과가 변수에 입력되어야 한다면 역따옴표를 사용해야 한다

문제 109) 이름이 scott인 사원의 부서번호가 출력되게 하는 셸을 작성하시오

답)

```
[orcl:~]$ vi a2.sh
#!/bin/bash

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

echo "$deptno 번에서 근무합니다."

[orcl:~]$ sh a2.sh
```

20 번에서 근무합니다.

문제 110) 위의 스크립트를 수정해서 이름을 물어보게 하고 이름을 입력하면 해당 사원의 부서번호가 출력되게 하시오

보기)

```
$ sh a2.sh

이름을 입력하세요: smith
10번에서 근무합니다.
```

답)

```
[orcl:~]$ vi a2.sh
#!/bin/bash

echo -n "이름을 입력하세요: "
read ename
deptno=`grep -i $ename emp.txt | awk '{print $8}'`

echo "$deptno 번에서 근무합니다."
```

문제 111) 위의 코드를 수정해서 아래와 같이 결과가 출력되게 하시오

보기)

```
이름을 입력하세요: allen
```

allen은 30번에서 근무합니다

답)

```
$ vi a2.sh
#!/bin/bash

echo -n "이름을 입력하세요: "
read ename
deptno=`grep -i $ename emp.txt | awk '{print $8}'`

echo "$ename은 $deptno 번에서 근무합니다."
$ sh a2.sh
```

결과)

```
[orcl:~]$ sh a2.sh
이름을 입력하세요: allen
allen은 30 번에서 근무합니다.
```

문제 112) dept.txt를 /home/oracle 에 올리고 부서번호를 물어보게 하고 부서번호를 입력하면 해당 부서번호의 부서위치가 출력되게 하시오

보기)

```
$ sh a3.sh
부서번호를 입력하세요: 20
DALLAS 입니다.
```

답)

```
$ vi a3.sh
#!/bin/bash

echo -n "부서번호 입력: "
read deptno
loc=`grep -w $deptno dept.txt | awk '{print $3}'`

echo "$loc 입니다"
```

문제 113) 이름을 물어보게 하고 이름을 입력하면 해당 사원의 부서위치가 출력되게 하시오

보기)

```
$ sh a99.sh
이름을 입력하세요: scott
scott은 DALLAS에서 근무합니다
```

답)

```
$ sh a99.sh

#!/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."
```

결과)

```
[orc1:~]$ sh a99.sh
이름을 입력하세요: scott
scott은 DALLAS 에서 근무합니다.
```

문제 114) 직업을 물어보게 하고 직업을 입력하면 해당 직업인 사원들의 월급이 출력되게 하시오

보기)

```
$ sh job_sal.sh

직업을 입력하세요: salesman

1600
1250
1250
1500
```

답)

```
$ vi job_sal.sh

#!/bin/bash

echo " "
echo -n "직업을 입력하세요: "
read job
```

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

```
echo "$salary"
```

```
$ sh job_sal.sh
```

문제 115) 직업을 물어보게 하고 직업을 입력하면 해당 직업의 토탈 월급이 출력되게 하시오

답)

```
$ vi job_sumsal.sh
```

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

```
echo " "
```

```
echo -n "직업을 입력하세요: "
```

```
read job
```

```
total=`grep -i $job emp.txt | awk '{print $6}' | awk '{sum+= $1} END {print sum}'`
```

```
echo "$total"
```

문제 116) 직업을 물어보게 하고 해당 직업의 평균 월급이 출력되게 하시오

보기)

```
$ sh job_avgsal.sh
```

직업 입력: salesman

salesman의 평균월급은 1400 입니다.

답)

```
$ vi job_avgsal.sh
```

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

```
echo " "
```

```
echo -n "직업 입력: "
```

```
read job
```

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

```
sum=`grep -i $job emp.txt | awk '{print $6}' | awk '{sum += $1} END {print sum}'`
```

```
avg=`expr $sum / $count`
```

```
echo "$job의 평균월급은 $avg 입니다."
```

문제 117) 컬럼번호를 물어보게 하고 컬럼번호를 입력하면 해당 컬럼의 데이터의 토탈 값과 평균 값이 아래와 같이 출력되게 하시오 (점심시간 문제)

보기)

```
$ sh total.sh
```

컬럼번호 입력: 6

토탈값: 10875

평균값: 6700

답)

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

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

```
echo " "
```

```
echo -n "컬럼번호 입력: "
```

```
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"
```

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

컬럼번호 입력: 6

토탈값 : 29025

평균값 : 2073

IF문

```
if [ 조건 ]; then
```

실행문

```
elif [ 조건 ]; then
```

실행문

```
else
```

실행문

```
fi
```

※ 주의사항

공백 처리를 잘 해줘야 한다

if문 조건문에 들어가는 비교 연산자

문자열 비교

"문자열1" = "문자열2"	두 문자열이 같으면	TRUE
"문자열1" != "문자열2"	두 문자열이 같지 않으면	TRUE

숫자열 비교

숫자1 -eq 숫자2	두 숫자가 같으면	TRUE
숫자1 -ne 숫자2	두 숫자가 같지 않으면	TRUE
숫자1 -gt 숫자2	숫자1이 숫자2보다 크다면	TRUE
숫자1 -ge 숫자2	숫자1이 숫자2보다 크거나 같으면	TRUE
숫자1 -lt 숫자2	숫자1이 숫자2보다 작다면	TRUE
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면	TRUE
!숫자1	숫자1이 거짓이라면	TRUE

예제)

```
$ vi if1.sh
```

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

```
if [ 100 -eq 200 ]; then  
    echo "100 과 200 은 같다 "
```

```
else  
    echo "100 과 200 은 같지 않다"
```

```
fi
```

```
$ sh if1.sh
```

문제 118) 위의 스크립트를 활용해서 아래처럼 숫자를 물어보게 하고 아래와 같이 결과가 출력되게 하시오

보기)

```
$ sh if2. sh
```

첫번째 숫자 입력: 200

두번째 숫자 입력: 250

250은 200보다 큼니다.

답)

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

```
echo " "
```

```
echo -n "첫번째 숫자 입력: "
```

```
read f_num
```

```
echo -n "두번째 숫자 입력: "
```

```
read s_num
```

```
if [ $f_num -gt $s_num ]; then
```

```
    echo "$f_num 은 $s_num 보다 크다."
```

```
elif [ $f_num -lt $s_num ]; then
```

```
    echo "$f_num 은 $s_num 보다 작다."
```

```
else
```

```
    echo "$f_num 와 $s_num 은 같다."
```

AND 와 OR 연산자

1. and 는 %% 또는 -a를 사용한다

2. or 는 || 또는 -o를 사용한다

예제)

```
if [ $sal -lt 200 ] && [ $job="SALESMAN" ]; then
```

또는

```
if [ $sal -lt 200 -a $job="SALESMAN" ]; then
```

문제 119) 사원 이름을 물어보게 하고 사원 이름을 입력했을때 해당 사원의 부서번호가 30번이고 월급이 2000보다 작다면 월급 인상 대상자 입니다. 라는 메시지가 출력되게 하고 그렇지 않으면 월급 인상 대상자가 아닙니다. 라는 메시지가 출력되게 하시오

결과)

```
이름 입력: scott
scott은 월급 인상 대상자가 아닙니다.
```

```
이름 입력: allen
allen은 월급 인상 대상자입니다.
```

답)

```
#!/bin/bash

echo " "
echo -n "이름 입력: "
read name

sal=`grep -i $name emp.txt | awk '{print $6}'`
deptno=`grep -i $name emp.txt | awk '{print $8}'`

if [ $deptno -eq 30 -a $sal -lt 2000 ]; then
    echo "$name은 월급 인상 대상자입니다."
else
    echo "$name은 월급 인상 대상자가 아닙니다."
fi
```

LOOP문

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

예제)


```
$ vi for.sh

#!/bin/bash

hap=0
for i in {1..10}
do
    hap=`expr $hap + $i`
done
    echo "1부터 10까지의 합은 "$hap

$ sh for.sh
```

문제 120) 위의 코드를 수정해서 구구단 2단을 출력하시오

결과)

```
[orcl:~]$ sh gugu2.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

답)

```
$ vi gugu2.sh

#!/bin/bash

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

$ sh gugu2.sh
```

답 2)

```
for i in {1..0}
do
```

```
let hap=2*$i
echo "2 x " $i " = " $hap
done
```

설명)

expr 대신에 let을 쓰면 역슬래시를 안써도 된다

문제 121) 구구단 2단부터 9단까지 출력하시오

결과)

```
[orcl:~]$ sh gugu.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
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
```

답)

```
#!/bin/bash

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

문제 122) 아래의 별을 출력하시오

보기)

```
$ sh star.sh
```

```
★
★★
★★★
★★★★
★★★★★
```

답)

```
#!/bin/bash

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

문제 123) 위의 코드를 수정해서 숫자를 물어보게 하고 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★"
done
```

```
echo $star  
done
```

설명)

eval 명령어는 문자열을 명령문으로 인식하고 실행하는 명령어

예)

```
$ str="ls -l emp.txt"  
$ echo $str  
$ eval $str
```

예 2)

```
$ eval echo {1..10}  
1 2 3 4 5 6 7 8 9 10
```

문제 124) 숫자를 입력하면 아래와 같이 ★ 가 출력되게 하시오

보기)

```
$ sh rev_star.sh  
숫자를 입력: 5
```

```
★★★★★  
★★★★★  
★★★★  
★★★  
★★  
★
```

답)

```
$ vi rev_star.sh
```

```
#!/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=""
done
```

```
$ sh rev_star.sh
```

디버깅

문제 125) 위의 스크립트를 디버깅하시오

답)

```
$ sh -x rev_star.sh
```

문제 126) emp.txt를 for loop 문을 이용해서 emp1.txt ~ emp100.txt로 복사하시오

답)

```
$ vi for_cp.sh

#!/bin/bash

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

$ sh for_cp.sh
```

문제 127) 짝공하고 자리를 바꿔서 emp1.txt ~ emp100.txt 중 하나의 파일을 랜덤으로 골라서 데이터를 변경하고 자리로 돌아오시오

답)

```
OK
```

문제 128) emp3.txt에 3900을 3000으로 바꾸시오

답)

```
$ vi emp3.txt
:%s/3900/3000/g
```

문제 129) emp.txt와 emp3.txt 파일 둘 간의 데이터의 차이가 있는지 확인하시오

답)

```
$ diff emp.txt emp3.txt

$ diff --brief emp.txt emp3.txt
Files emp.txt and emp3.txt differ
```

문제 130) 짝공이 어떠한 .txt 파일을 변경했는지 한번에 알아내시오 (emp1.txt ~ emp199.txt 중에서)

답)

```
$ vi a130.sh

#!/bin/bash

for i in {1..100}
do
    diff --brief emp.txt emp$i.txt
done

$ sh a130.sh
```

결과)

```
[orcl:~]$ sh a130.sh
Files emp.txt and emp3.txt differ
Files emp.txt and emp22.txt differ
```

정답)

emp22.txt를 바꿨습니다.

sed 명령어로 공백라인을 지우는 방법 (데이터 정제 작업)

1. emp.txt 를 열어서 중간에 공백 라인을 하나 넣으시오

```
$ vi emp55.txt
```

2. emp.txt 안에 공백 라인이 있는지 검색한다

```
$ sed -n /^$/p emp55.txt | wc -l
```

```
1
```

3. emp.txt 안에 공백 라인을 삭제한다

```
$ sed '/^$/d' emp55.txt >> emp_backup.txt
```

```
$ mv emp_backup.txt emp55.txt
```

```
$ sed -n /^$/p emp55.txt | wc -l
```

```
0
```

※ 엔터로 넣은 공백까지 다 지우는 방법

```
$ sed '/^ *$/d' emp55.txt >> emp_backup.txt
```

```
$ rm emp55.txt
```

```
$ mv emp_backup.txt emp55.txt
```

**문제 131) 짝공과 자리를 바꿔서 emp1.txt ~ emp100.txt 중 몇 개를 열어서 공백을 막 넣어주고
자리로 돌아오시오**

답)

```
OK
```

문제 132) 자리로 돌아와서 짝공이 어떠한 파일에 공백을 넣었는지 확인하시오

답)

```
$ sed -n /^$/p emp*.txt | wc -l
```

```
473
```

```
$ sh a130.sh
```

```
Files emp.txt and emp3.txt differ
```

```
Files emp.txt and emp13.txt differ
```

```
Files emp.txt and emp16.txt differ
```

```
Files emp.txt and emp22.txt differ
```

```
Files emp.txt and emp33.txt differ
```

```
Files emp.txt and emp46.txt differ
```

Files emp.txt and emp61.txt differ

Files emp.txt and emp65.txt differ

Files emp.txt and emp91.txt differ

```
$ vi a132.sh

#!/bin/bash

for i in {1..100}
do
    a=`sed -n '/^ */p' emp$i.txt |wc -l`
    if [ $a -ne 0 ]; then
        echo emp$i
    fi
done

$ sh a132.sh
```

문제 133) 짝공이 공백을 입력한 파일들의 공백을 다 지우시오

답)

```
$ vi a133.sh

#!/bin/bash

for i 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

$ sh a.133.sh
```

알고리즘 문제 (factorial, power, 최대공약수)

문제 134) (알고리즘 문제 : bc 카드 공채문제) 아래의 쉘을 생성하시오

보기)

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


5x4x3x2x1 = 120

답)

```
$ vi factorial.sh
```

```
#!/bin/bash

num=1
for i in `eval echo {1..$1}`
do
    num=`expr $num \* $i`
done
echo "$num"
```

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

문제 135) 아래의 power 함수를 쉘로 구현하시오

보기)

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

```
8
```

답)

```
#!/bin/bash

num=1
for (( i=1; i<=$2; i++ ))
do
    let num=num*$1
done

echo "   "
echo "$1의 $2 제곱: $num"
echo "   "
```

문제 136) 최대 공약수를 출력하시오 (중요)

보기)

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

답)

```
n=0
n1=$1
n2=$2
while [ $n2 -ne 0 ]
do
    n=`expr $n1 % $n2`
    n1=$n2
    n2=$n
done
echo $n1
```

문제 137) 직업을 중복 제거해서 출력하시오

답)

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

결과)

```
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제 138) 위의 결과를 변수에 담아서 for loop 문으로 변수의 내용을 출력하시오

답)

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

$ for i in $job
> do
> echo $i
> done
```

결과)

```
ANALYST
```

CLERK
MANAGER
PRESIDENT
SALESMAN

문제 139) 직업이 SALESMAN인 직원들의 토탈월급을 출력하시오

답)

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

결과)

```
5600
```

문제 140) 직업과 직업 별 토탈 월급을 출력하시오

보기)

```
$ sh a140.sh
```

답)

```
#!/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
```

문제 141) 부서번호, 부서번호 별 평균월급을 출력하시오

결과)

```
[orcl:~]$ sh a141.sh
10 3283
20 2079
30 2783
```

답)

```
#!/bin/bash

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

for i in $deptno
do
sum=`grep -i $i emp.txt | awk '{sum += $6} END {print sum}'`
count=`grep -w $i emp.txt | wc -l`
avg=`expr $sum / $count`
echo $i $avg
done
```

LG전자 고객이 의뢰한 회사의 사수에게 던진 질문

1. 라이나 생명에서 어떠한 분석을 했느냐
2. 데이터가 많았을 텐데 R로 어떻게 분석 했는가
 1. 하둡, 오라클, 쉘에서 데이터를 필터링
(40대만 따로 분리해서 별도의 text 파일을 생성)
 2. 40대의 데이터를 가지고 군집분석 (k-means)
 3. 40대의 데이터를 가지고 연관분석 (아프리오리)

위의 밑줄을 지금부터 해보자

대용량 데이터를 DB에서 직업 별로 각각의 txt 파일로 추출하기

문제 142) 직업이 SALESMAN인 직원들의 데이터만 추출해서 SALESMAN.txt로 생성하시오

답)

```
[orcl:~]$ grep -i 'salesman' emp.txt > salesman.txt
[orcl:~]$ cat salesman.txt
```

결과)

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

설명)

어떻게 R로 대용량의 파일을 돌리는가 물어보면
오라클이나 쉘이나 하둡으로 잘라서 돌려야 한다고 대답할 수 있다

문제 143) emp.txt를 직업 별로 데이터를 분리해서 아래와 같이 텍스트 파일들이 생성되게 하시오
(점심시간 문제)

보기)

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

위와 같이 한번에 생성되도록 하라

답)

```
$ vi partition_job.sh  
  
#!/bin/bash  
  
job=`awk '{print $3}' emp.txt | sort | uniq `
```



```
for i in $job  
do  
    grep -i $i emp.txt > $i.txt  
done
```



```
$ sh partition_job.sh
```

결과)

```
[orcl:~]$ cat MANAGER.txt
```

7566 JONES	MANAGER	7839 02-APR-81	2975	0	20
7698 BLAKE	MANAGER	7839 01-MAY-81	2850	0	30
7782 CLARK	MANAGER	7839 09-JUN-81	2450	0	10

DB에서 데이터를 추출하는 방법 2가지

1. dump file

```
$ alias scott='sqlplus scott/tiger'  
$ scott
```

SQL*Plus: Release 11.2.0.1.0 Production on Thu Jun 28 14:00:14 2018

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

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

emp를 dump파일로 떨어뜨린다

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

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

```
-rw-r--r-- 1 oracle oinstall 16384 Jun 28 14:01 emp.dmp
```

SCOTT계정으로 오라클 접속

```
[orcl:~]$ scott
```

위에서 덤프 파일로 떨어뜨려놨고 이제 실수로 SQL에서 drop 했다고 가정하고 아래 SQL문을 실행

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

Table dropped.

덤프파일로 살릴 수 있다

```
SQL> exit
```

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

```
[orcl:~]$ scott
```

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

2. csv 또는 text file

SQL 접속

```
$ scott
```

쉼표(,)로 구분

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

```
SQL> set pagesize 0
```

```
SQL> set linesize 120
```

spool off 할 때까지 다음의 작업을 emp222.csv에 저장한다

```
SQL> spool emp222.csv
```

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

```
SQL> spool off
```

```
SQL> exit()
```

```
$ cat emp222.csv
```

위와 같이 하면 불필요한 데이터까지 같이 딸려 들어온다. 따라서 설정해줘야 한다

exit로 SQL을 빠져 나갔다가 들어오면 set이 초기화되기 때문에 다시 설정해줘야 한다

```
SQL> set heading on      -- header(헤더) 포함 (디폴트값이 on)
SQL> set underline off   -- 밑줄 제거
SQL> set colsep ','      -- 쉼표(,) 로 구분
SQL> set pagesize 400    -- default가 14인데 지정 안하면 공백이 14줄 마다 나오기 때문에
                        -- 크게 설정해준다
SQL> set linesize 300    -- 라인도 record 크기만큼 설정해줘 아래로 구분되지 않도록 한다
SQL> set feedback off   -- 14 rows selected. 이것도 안나오게 하는 법
SQL> set echo off       -- select * from emp; 이것도 안나오게 하는 법
SQL> set sqlprompt ""    -- 왼쪽에 SQL> 지우는 법

spool emp225.csv
select * from emp;
spool off
exit()

[orcl:~]$ cat emp225.csv
```

While loop 문

예제)

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

문제 144) while loop 문으로 factorial을 구현하시오

보기)

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

```
120
```

답)

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

```

res=1
i=1
while [ $i -le $1 ]
do
    res=`expr $res W* $i`
    let i=$i+1
done
echo $res

```

문제 145) 최대 공약수를 while loop문으로 구현하시오

보기)

```

$ sh maxyaksu.sh 24 16
8

```

답)

```

n=0
n1=$1
n2=$2
while [ $n2 -ne 0 ]
do
    n=`expr $n1 % $n2`
    n1=$n2
    n2=$n
done
echo $n1

```

문제 146, 147) emp테이블의 데이터를 csv로 생성하고 R에서 로드하시오

답)

```

$ vi emp.sql

```

```

set heading on
set underline off
set colsep ','
set pagesize 20
set linesize 120

```



```

set feedback off
set echo off
set sqlprompt ""

spool emp.csv
select * from emp;
spool off
exit

```

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

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

```
$ R
```

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

```
dept
```

설명)

① \$ sed 's/emp/dept/' emp.sql > dept.sql

emp.sql 스크립트에서 emp를 dept로 바꿔서 dept.sql 이라는 이름으로 저장하겠다. (> : 덮어쓰기)

② \$ sqlplus scott/tiger @dept.sql

SQL 접속과 동시에 sqlsetting 스크립트를 돌리겠다

결과)

```

> 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

```

문제 148) 위의 스크립트를 이용해서 아래와 같이 테이블명을 물어보게 하고 테이블명을 입력하면 csv 파일이 생성되게 하시오

보기)

```
$ sh create_csv.sh
```

테이블명을 입력하세요: salgrade

```
$ cat salgrade.csv
```

답)

```
$ vi create_csv.sh
```

```
echo -n "테이블명을 입력하세요: "
```

```
read tablename
```

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

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

case 문

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

R의 menu 함수와 같다

예제)

```
$ vi m2.sh
```

```
echo "
```

```
    1. sar 그래프를 그릴거면 1번을
```

```
    2. csv 파일을 생성하려면 2번을
```

```
"
```

```
echo " "
```

```
echo -n "번호를 입력하세요: "
```

```
read choice
```

```
case $choice in
```

```
    1)
```

```
    /home/oracle/sar.sh ;;
```

```
    2)
```

```
    /home/oracle/create_csv.sh ;;
```

```
esac
```

권한을 주고

```
$ chmod 777 sar.R
```

```
$ chmod 777 sar.sh
```

```
$ chmod 777 create_csv.sh
```

sar그래프를 그리기 위해 sar.sh라는 쉘 스크립트를 만들어준다. 그 안에 실행 스크립트를 넣어준다

```
$ vi sar.sh
```

```
Rscript /home/oracle/sar.R
```

문제 149) find_file.sh 가 잘 실행되는지 확인하시오

답)

```
echo " "
echo -n "검색할 디렉토리를 입력하세요: "
read dir1
echo " "
echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요: "
read file1
echo " "
echo -n "depth 값: "
read depth
echo " "
find $dir1 -maxdepth $depth -name $file1

$ sh find_file.sh
```

문제 150) 리눅스 자동화 스크립트인 m2.sh의 세번째로 find_file.sh를 추가하시오

보기)

```
$ sh m2.sh
```

1. sar 그래프를 그릴거면 1번을
2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을

설명)

실행할때 \$ sh find_file.sh 라고 하면 실행이 되지만
그냥 /home/oracle/find_file.sh 라고 하면 실행이 안 된다.
따라서, 먼저 권한을 줘야한다

답)

```
$ chmod 766 find_file.sh
```

```
$ vi m2.sh
```

```
echo "  
    1. sar 그래프를 그릴거면 1번을  
    2. csv 파일을 생성하려면 2번을  
    3. file을 검색하려면 3번을  
    "  
echo " "  
echo -n "번호를 입력하세요: "  
read choice  
case $choice in  
    1)  
        /home/oracle/sar.sh ;;  
    2)  
        /home/oracle/create_csv.sh ;;  
    3)  
        /home/oracle/find_file.sh ;;  
esac
```

```
$ sh m2.sh
```

문제 151) 특정 파일 안에 특정 단어가 몇 건 있는지 검색하는 쉘을 만들기 위해서 겨울 왕국의 **elsa**라는 단어가 몇건이 있는지 조회하시오

답)

```
$ grep -iow 'elsa' winter.txt | wc -l
```

329

문제 152) 위의 스크립트로 아래와 같이 쉘 스크립트를 작성하시오

보기)

```
$ sh word_count.sh
```

단어를 검색할 대상의 파일명을 입력하세요: winter.txt

검색할 단어를 입력하세요: elsa

658 건 검색되었습니다

답)

```
#!/bin/bash

echo " "
echo "단어를 검색할 대상의 파일명을 입력하세요: "
read filename
echo " "
echo "검색할 단어를 입력하세요: "
read word
echo " "

result=`grep -iow $word $filename | wc -l`
echo "$result 건 검색되었습니다."
echo " "
```

문제 153) 자동화 스크립트 4번에 word.count.sh 를 추가하세요

보기)

```
$ sh m2.sh
```

1. sar 그래프를 그릴거면 1번을
2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을
4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을

답)

```
$ vi m2.sh
$ chmod 766 word_count.sh

echo "
    1. sar 그래프를 그릴거면 1번을
    2. csv 파일을 생성하려면 2번을
    3. file을 검색하려면 3번을
    4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을
"
echo " "
echo -n "번호를 입력하세요: "
read choice
case $choice in
```

```

1)
/home/oracle/sar.sh ;;
2)
/home/oracle/create_csv.sh ;;
3)
/home/oracle/find_file.sh ;;
4)
/home/oracle/word_count.sh ;;
esac

```

sar 그래프 ctrl z 했을때 그래프가 안꺼지는데 찾아서 죽이는 방법

```
$ ps -ef | grep sar.R
```

```

oracle 10479 10478 0 11:04 pts/1 00:00:00 /home/oracle/R-3.2.3/bin/exec/R --slave
e --file=/home/oracle/sar.R
oracle 10588 9155 0 11:07 pts/1 00:00:00 grep sar.R

```

```
$ kill -9 10479
```

swap file 생성되어 vi 편집기 실행 시 오류 해결 방법

vi aaa.sh 에서 아무거나 막 편집하다가 Ctrl + z 를 누르고 나온 뒤에
다시 vi aaa.sh 로 들어가면 아래와 같이 나온다

```

E325: ATTENTION
Found a swap file by the name ".aaa.sh.swp"
    owned by: oracle   dated: Fri Jun 29 11:10:50 2018
    file name: ~oracle/aaa.sh
    modified: YES
    user name: oracle  host name: edydr1p0.us.oracle.com
    process ID: 10616 (still running)
While opening file "aaa.sh"

(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:

```

이거 안 뜨게 하려면 위의 동그라미를 없애줘야 한다

```
$ ls -al *.swp
$ rm *.swp
$ ls -al *.swp
```

문제 154) 아래의 명령어를 `rm_swp.sh` 스크립트에 저장하는데 파일이 없으면 "삭제할 파일이 없습니다." 라는 메시지가 출력되게 하시오

답)

```
$ vi rm_swp.sh

#!/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

$ sh rm_swp.sh
```

문제 155) 위의 `rm_swp.sh` 를 자동화 스크립트 5번에 추가하시오

보기)

```
$ sh m2.sh

1. sar 그래프를 그릴거면 1번을
2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을
4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을
5. swp 파일을 모두 삭제하려면 5번을
```

답)

```
echo "
1. sar 그래프를 그릴거면 1번을
2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을
```

```

4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을
5. swp 파일을 모두 삭제하려면 5번을
"
echo " "
echo -n "번호를 입력하세요: "
read choice
case $choice in
    1)
        /home/oracle/sar.sh ;;
    2)
        /home/oracle/create_csv.sh ;;
    3)
        /home/oracle/find_file.sh ;;
    4)
        /home/oracle/word_count.sh ;;
    5)
        sh rm_swp.sh ;;
esac

```

문제 156) 두개의 파일의 차이를 확인하는 diff명령어를 이용한 스크립트 diff.sh 를 확인하시오

답)

```
$ vi diff.sh
```

문제 157) diff.sh 를 자동화 스크립트 6번에 추가하시오

보기)

```
$ sh m2.sh
```

1. sar 그래프를 그릴거면 1번을
2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을
4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을
5. swp 파일을 모두 삭제하려면 5번을
6. 두 개의 파일의 차이를 확인하려면 6번을

답)

```
echo "
1. sar 그래프를 그릴거면 1번을
```



```

2. csv 파일을 생성하려면 2번을
3. file을 검색하려면 3번을
4. 원하는 파일 안에 특정 단어의 개수를 검색하려면 4번을
5. swp 파일을 모두 삭제하려면 5번을
6. 두 개의 파일의 차이를 확인하려면 6번을
"
echo " "
echo -n "번호를 입력하세요: "
read choice
case $choice in
    1)
        /home/oracle/sar.sh ;;
    2)
        /home/oracle/create_csv.sh ;;
    3)
        /home/oracle/find_file.sh ;;
    4)
        /home/oracle/word_count.sh ;;
    5)
        sh rm_swp.sh ;;
    6)
        sh diff.sh ;;
esac

```

문제 158) 오라클의 hr 계정의 락을 해제하고 hr 계정이 가지고 있는 테이블 리스트를 조회하시오

답)

```
$ 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;
```

User altered.

hr 계정으로 접속

```
SQL> connect hr/hr
Connected.
```

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

답)

```
$ vi emp2.sql
```

```
set heading on
set underline off
set colsep ','
set pagesize 100
set linesize 120
set feedback off
set echo off
set sqlprompt ""

spool emp.csv
select * from emp;
spool off
exit
```

```
$ vi create_csv2.sh
```

```
echo -n "테이블명을 입력하세요: "
read tablename
```

```
sed 's/emp/"$tablename"/' emp2.sql > $tablename.sql
```

```
sqlplus hr/hr @$tablename.sql
```

```
$ sh create_csv2.sh
```

한 다음 위에 테이블명 하나하나 입력해서 만든다

csv 파일 생성하는 스크립트 수정

```
$ vi csv7.sh

#!/bin/bash
echo " "
echo -n "테이블명을 입력하세요: "
read table
echo " "
sqlplus scott/tiger << EOF
set colsep ','
set pagesize 20
set linesize 120
set sqlprompt ""
set feedback off
set echo off
set underline off

spool $table.csv

select * from $table;

spool off

EOF

$ sh csv7.sh
emp

$ cat emp.csv
```

EOF (End Of File)

결과)

```
[orcl:~]$ cat emp.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
```

문제 160) emp.csv 에서 위에 1번과 2번 라인을 삭제하고 emp2.csv 로 저장하시오

답)

```
$ sed '1,2d' emp.csv > emp2.csv
```

결과)

```
[orcl:~]$ cat emp2.csv
```

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
```

문제 161) emp2.csv의 맨 밑에 spool off를 삭제하고 emp3.csv로 저장하시오

답)

```
$ sed '/spool/d' emp2.csv > emp3.csv
```

결과)

```
[orcl:~]$ cat emp3.csv
```

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

문제 162) 혹시 공백라인이 있을 수 있으므로 공백라인을 emp3.csv에서 지우고 emp4.csv로 저장하시오

답)

```
$ sed '/^$/d' emp3.csv > emp4.csv
```

결과)

```
$ cat emp4.csv
```

 다음 공백 줄이 지워지고 바로 테이블이 출력된다

문제 163) emp4.csv 가 R 에서 잘 로드되는지 확인하시오

답)

```
$ R
```

```
emp <- read.csv("emp4.csv", strip.white=T)
```

결과)

```
> 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
```

문제 164) 위에서 사용한 아래의 7개의 스크립트를 쉘로 저장해서 R로 깔끔하게 로드되는 csv 파일을 생성하시오

보기)

\$ sh csv7.sh 를 다시 실행해서 emp.csv 를 다시 정제하기 전으로 돌려놓는다.

\$ sed '1,2d' emp.csv > emp2.csv

\$ sed '/spool/d' emp2.csv > emp3.csv

\$ sed '/^\$/d' emp3.csv > emp4.csv

\$ rm emp.csv

\$ rm emp2.csv

\$ rm emp3.csv

\$ mv emp4.csv emp.csv

위 7개를 이용하여 새로운 clean.sh 라는 쉘 스크립트를 만든다

답)

\$ vi clean.sh

#!/bin/bash

echo " "

echo -n "정제할 테이블이 있는 위치를 입력하세요: "

read tableloc

echo -n "정제할 테이블명을 입력하세요: "

read tablename

echo " "

sed '1,2d' \$tableloc\$tablename.csv > \$tableloc{\$tablename}2.csv

sed '/spool/d' \$tableloc{\$tablename}2.csv > \$tableloc{\$tablename}3.csv

sed '/^\$/d' \$tableloc{\$tablename}3.csv > \$tableloc{\$tablename}4.csv

rm \$tablename.csv

```
rm ${tablename}2.csv
rm ${tablename}3.csv
mv ${tablename}4.csv ${tablename}.csv
```

문제 165) csv7.sh와 clean.sh를 하나로 합쳐서 스크립트를 생성하시오

답)

```
#!/bin/bash
echo " "
echo -n "테이블명을 입력하세요: "
read table
echo " "
sqlplus scott/tiger << EOF
set colsep ','
set lines 4000
set pagesize 4000
set linesize 120
set sqlprompt ""
set feedback off
set echo off
set underline off

spool $table.csv
select * from $table;
spool off

EOF

sh /home/oracle/clean.sh
```

문제 166) 자동화 스크립트 2번을 위의 스크립트로 수정하세요

답)

```
위의 스크립트를 전에 만들었던 create_csv.sh 안에 넣고
$ sh m2.sh
```

10 리눅스 환경에서 R과 오라클 연동시키기

2018년 6월 21일 목요일 오후 4:21

목차

- [리눅스 서버에서 R 설치](#)
- [리눅스에서 RJDBC 를 이용하여 오라클과 연동하는 방법](#)

리눅스 서버에서 R 설치

```
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>
```

리눅스에서 RJDBC 를 이용하여 오라클과 연동하는 방법

1. 사전확인사항 : 리스너의 IP 주소와 RJDBC의 IP 주소는 같아야 한다.

```
$ lsnrctl status
```

```
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.56.104)(PORT=1521)))  
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
```

원래 HOST= 가 위처럼 숫자로 나와야 하는데 리눅스에서 해보면

Listening Endpoints Summary...

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))  
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p0.us.oracle.com)(PORT=1521)))
```

위와 같이 나온다

2. 아래와 같이 오라클에 접속이 되는지 확인한다

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

edydr1p0.us.oracle.com	서버의 주소	여삼 빌딩
1521	포트	10층
orcl.us.oracle.com	서비스 이름	아이티월

3. 오라클과 R 연동을 위해 필요한 패키지를 설치한다

```
$ R  
> install.packages("rJava")  
우리나라 선택  
22  
50  
  
> install.packages("DBI")  
> install.packages("RJDBC")
```

4. 자바홈을 설정 / rJava의 memory limit default 2기가까지 확장 / rJava 라이브러리 불러오기

```
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"))  
[1] "1.6.0_18"
```

6. RJDBC 라이브러리

```
library(DBI)  
library(RJDBC)
```

7. 커넥션 드라이버 생성

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

8. 연결: scott 커넥션 열기

```
scott <- dbConnect(jdbcDriver, "jdbc:oracle:thin:  
@//edydr1p0.us.oracle.com:1521/orcl.us.oracle.com", "scott", "tiger")
```

9. 쿼리 실행 및 프린트

```
emp <- dbGetQuery(scott, "SELECT * from emp")  
print(emp)
```

10. working directory에 emp900.csv 라고 저장 후 확인

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

세션 복제 후 확인해본다
[orcl:~]\$ cat emp900.csv

11. 커넥션 종료

```
dbDisconnect(jdbcConnection)  
dbDisconnect(scott)
```

문제 169) dept 테이블을 오라클에서 가져와서 dept900.csv 를 생성하시오

답)

```
> dept <- dbGetQuery(scott, "SELECT * from dept")  
> write.csv(dept, '/home/oracle/dept900.csv')  
> dept900 <- read.csv("dept900.csv")  
> dept900
```

문제 170) scott이 가지고 있는 테이블 리스트를 가져와 변수에 저장하시오

답)

```
> scott_table <- dbGetQuery(scott, "SELECT table_name from user_tables")  
> scott_table
```

TABLE_NAME

1 DEPT

- 2 BONUS
 - 3 SALGRADE
 - 4 EMP
-