

1. VirtualBox, Ubuntu, xshell 설치 및 연결

2018년 6월 30일 토요일 오후 11:09

1. VirtualBox 설치

Virtual box 홈페이지에서 설치파일을 받는다.

+ 홈페이지에서 확장팩도 함께 설치한다.

1. VirtualBox 설치파일 다운로드함

VirtualBox사이트에서 설치 파일 다운로드함.(다운로드 사이트 : <https://www.virtualbox.org/wiki/Downloads>)



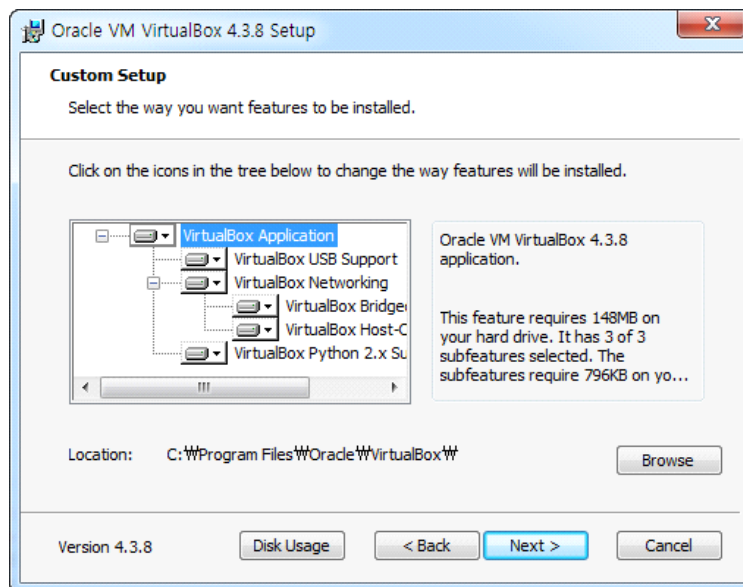
2. 설치 파일 실행함.

설치 파일 실행후 Next버튼 클릭함.



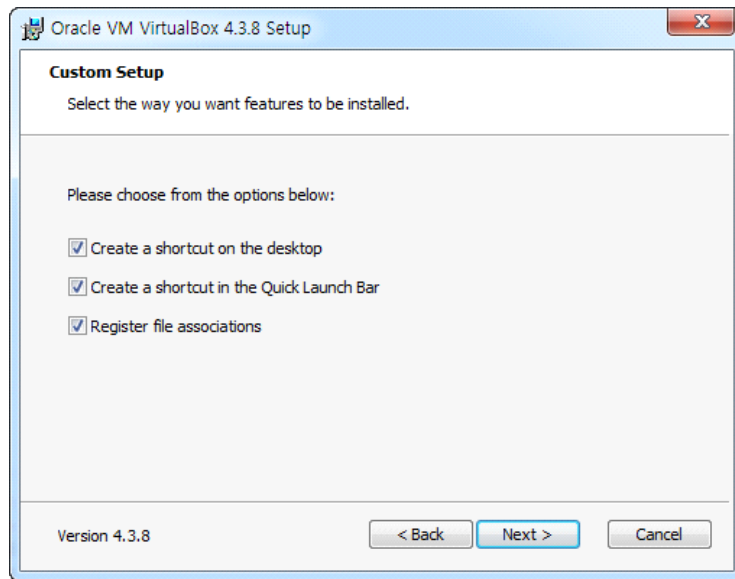
3. 설치 파일 확인 및 설치경로 지정함.

특별히 변경해야할 사항이 없다면, Next버튼 클릭함.



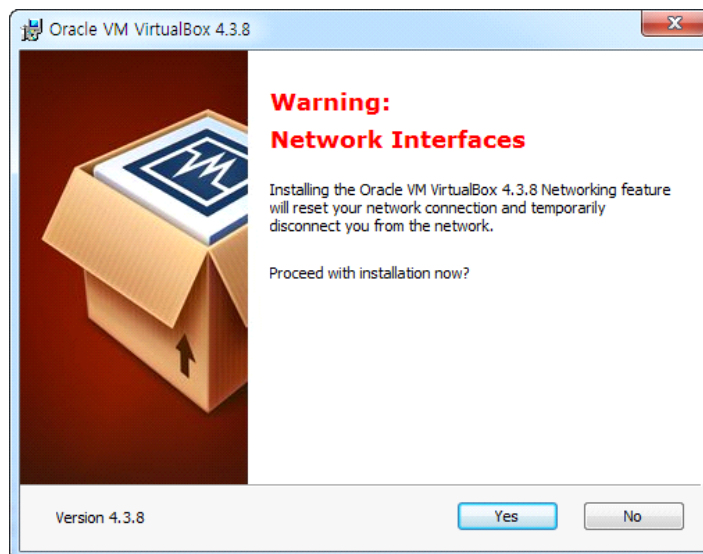
4. 실행 아이콘 생성 위치 지정함.

실행아이콘 생성 위치를 지정함. 기본적으로 다 체크 되어 있으나, 입맛에 따라 체크를 해지 후 Next버튼을 클릭함.



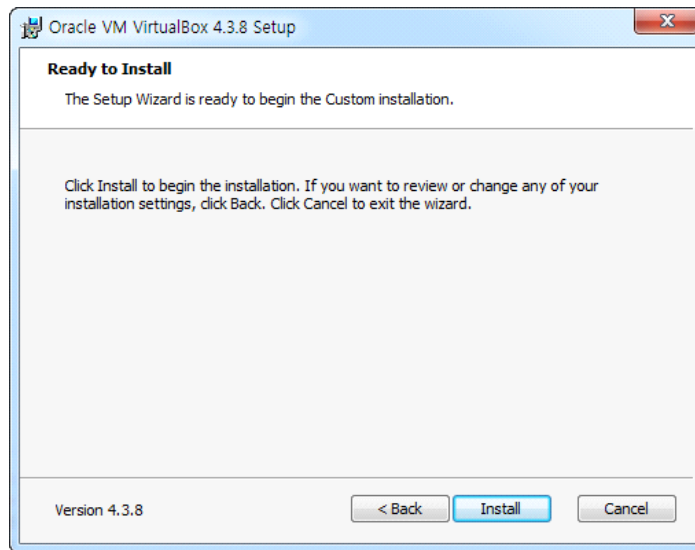
5. VirtualBox의 가상 Network Device를 설치 함.

VirtualBox의 가상 Network Device를 설치를 진행하며, Install하는 동안 일시적으로 모든 Network가 끊길 수 있다고함. Yes버튼 클릭함.



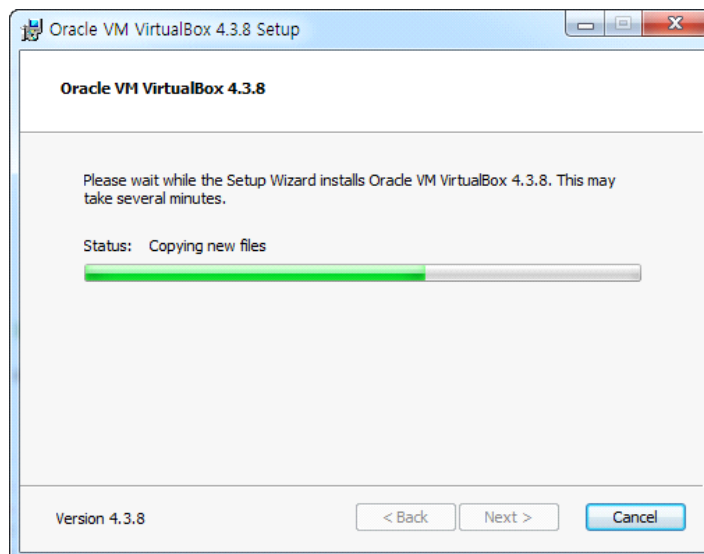
6. Configuration 정보가 설정된 후 Install 준비가 완료됨.

설치 구성 요소를 다 설정하였기 때문에 이제 Install버튼을 클릭함.



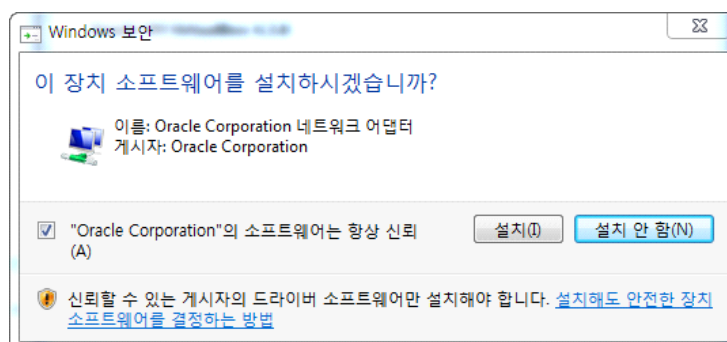
7. 설치 진행화면.

설치 진행 프로그래스바가 완료될 때까지 기다림.

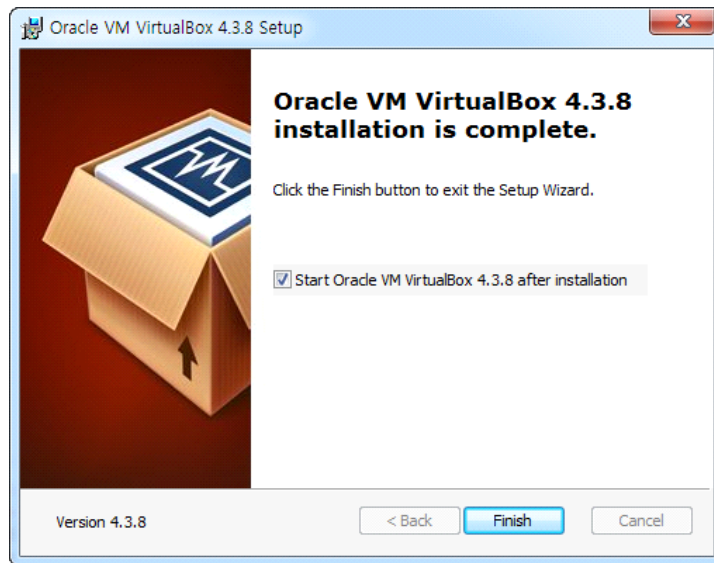


8. 컨트롤 설치함.(항상 신뢰 체크함)

VirtualBox설치를 위한 기타 컨트롤들을 같이 설치하며, 항상 신뢰 체크박스에 체크해둬.



9. 설치완료



10. 실행 화면



참고 : <http://blog.happydong.kr/251>

2. 우분투 설치

■ Virtual box에서 구동할 우분투(ubuntu)를 설치한다.

<https://www.ubuntu.com/download>

Ubuntu downloads

Ubuntu Desktop >

Download Ubuntu desktop and replace your current operating system whether it's Windows or Mac OS, or, run Ubuntu alongside it.

Do you want to upgrade? Follow our simple guide

Ubuntu Server >

Whether you want to configure a simple file server or build a fifty thousand-node cloud, you can rely on Ubuntu Server and its five years of guaranteed free upgrades.

Ubuntu Cloud >

Ubuntu is the reference OS for OpenStack. Try Canonical's OpenStack on a single machine or start building a production cloud on a cluster — just add servers.

Ubuntu flavours >

Ubuntu flavours offer a unique way to experience Ubuntu

Ubuntu for IoT >

Are you a developer who wants to try snappy Ubuntu Core?

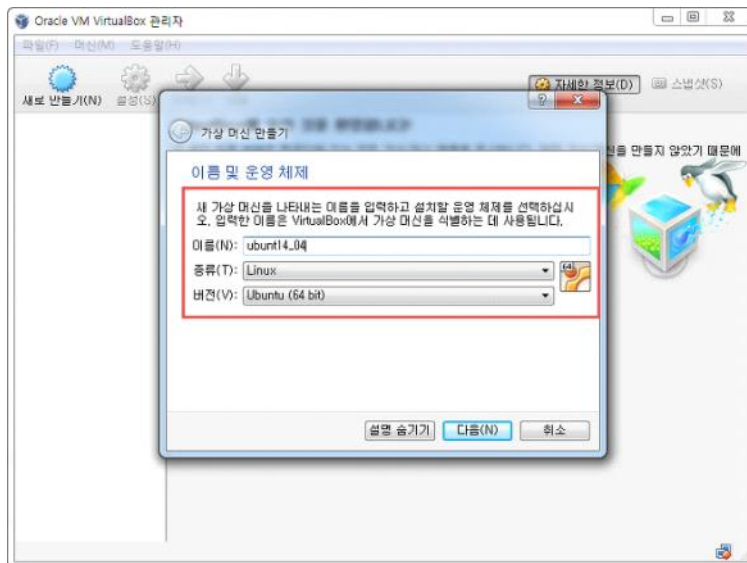
■ VirtualBox에 세팅

우분투 설치파일을 다운로드 다 받았다면, 이제 VirtualBox에 설치를 위해 세팅을 해보자.



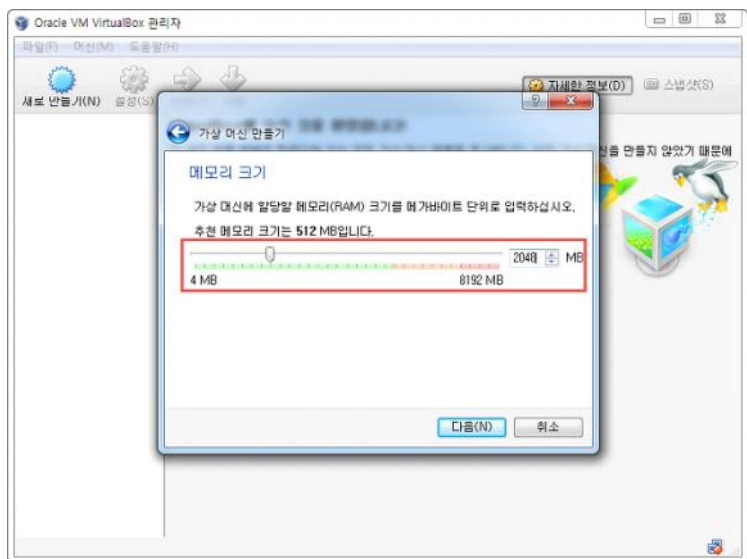
[그림 5 - VirtualBox 실행화면]

설치된 VirtualBox를 실행 했다면 "새로 만들기" 버튼을 클릭해서 가상 머신을 생성하도록 하자.



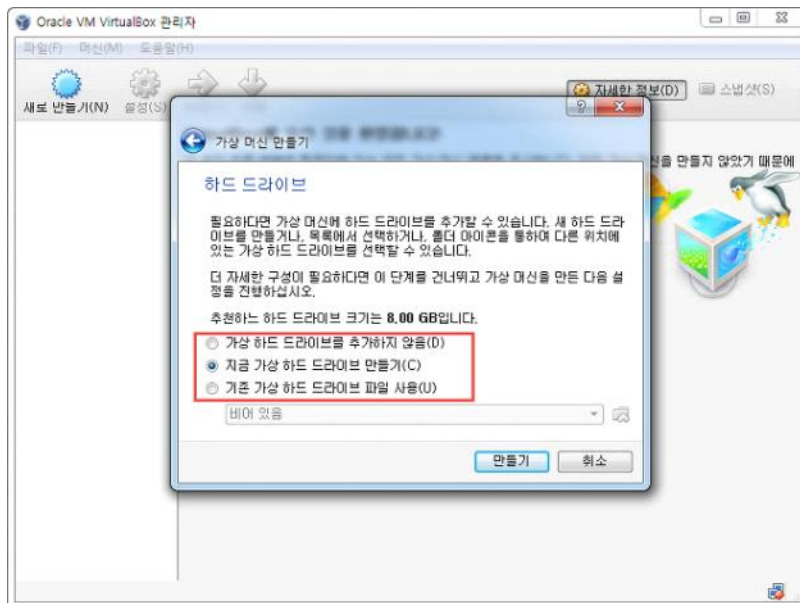
[그림 6 - 가상 머신 만들기 화면]

이름은 적당히 본인이 알 수 있는 이름으로 짓도록 한다. 그리고 종류를 선택한다. 우분투는 Linux 계열이기 때문에 Linux로 선택한다. 그리고 버전도 설치하고자 하는 내용에 맞게 잘 확인하고 선택한다. (이름에 ubuntu라고 입력만 해도 자동으로 맞춰 지기도 할 것이다. 만약에 안된다면 수동으로 종류와 버전을 선택하도록 한다.)



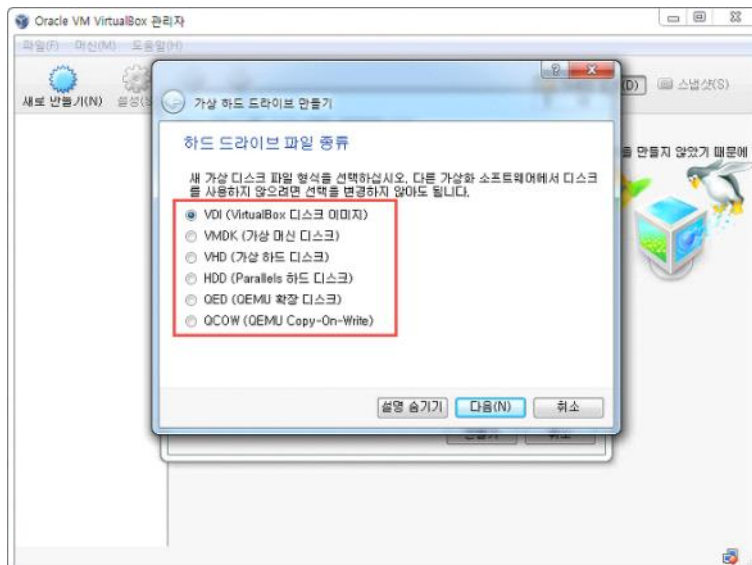
[그림 7 - 가상 머신 메모리 할당 화면]

가상 머신의 메모리를 할당해 줘야한다. 본인의 PC에 메모리가 여유 있다면 적당히 1024mb ~ 2048mb정도를 주면 되지 않을까 싶다. 메모리 설정은 본인 PC에 맞게 정당히 주도록 하자. 최소 512mb는 주자!! 내가 쓰는 데스크탑에는 메모리가 여유가 있으므로 2048mb 설정했다.



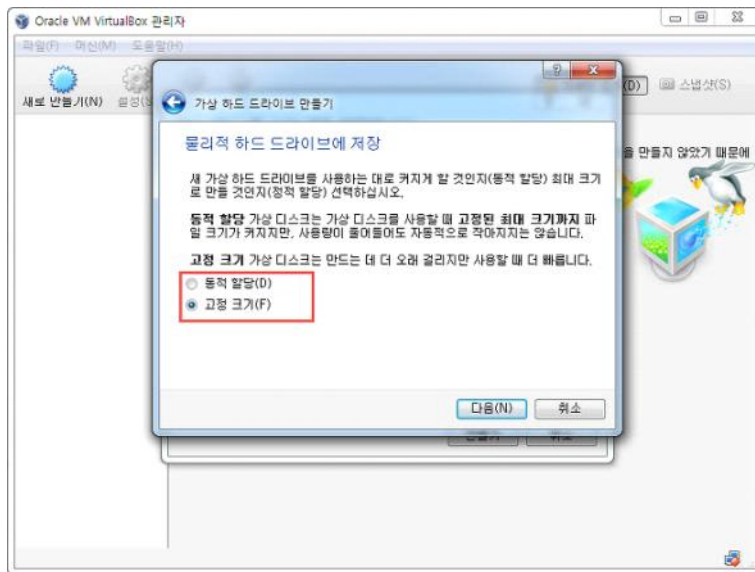
[그림 8 - 가상 머신 하드 드라이브 할당 화면]

설치할 하드 드라이브가 잡혀 있어서 OS를 올리수 있기때문에 "지금 가상 하드 드라이브 만들기"를 선택해서 하드 드라이브를 만들도록 하자.



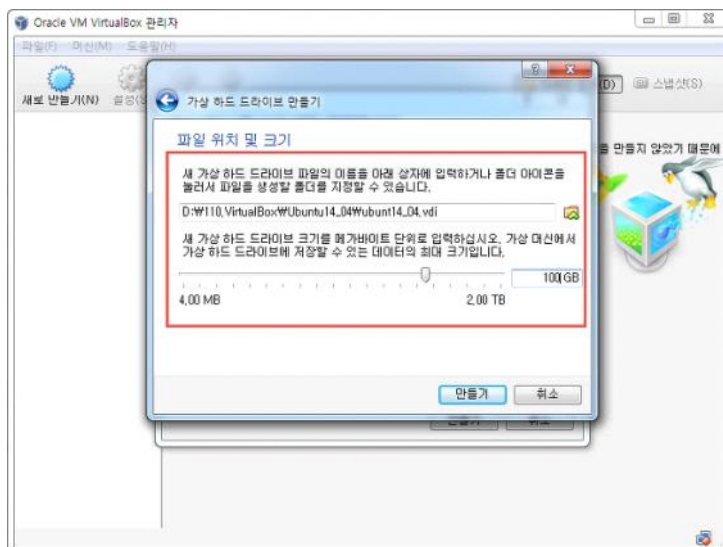
[그림 9 - 하드 드라이브 종류 선택 화면]

만들고자 하는 하드 드라이브 종류를 선택하도록 하는데, 해당 하드 드라이브는 VirtualBox에서만 사용할 것이므로 "VDI(VirtualBox) 디스크 이미지"를 선택 하도록 한다.



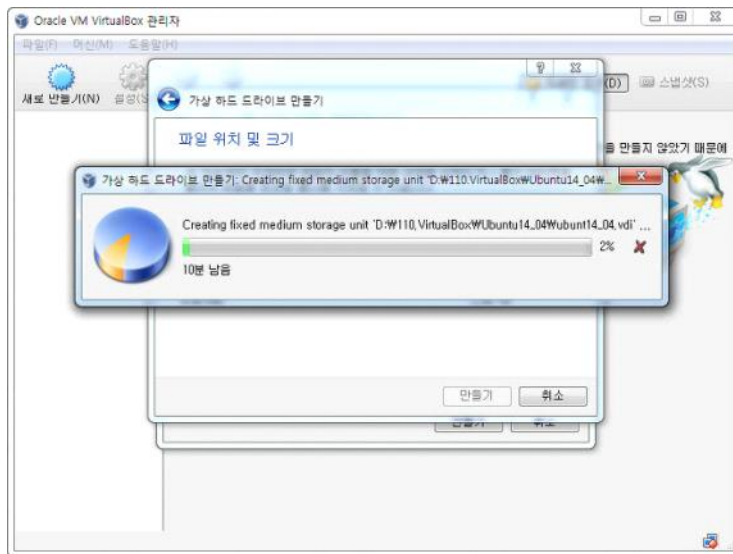
[그림 10 - 물리적 하드 드라이브 저장 형태선택 화면]

물리적 하드 드라이브 저장 형태에 대해 선택하도록 한다. 동적 할당은 기존 크기에서 용량이 늘어나면 자동으로 증가하는 형태이며, 고정 크기는 최초에 잡은 용량을 유지하는 형태이다. 나는 동적 할당 보다, 고정 크기를 지향한다. 그래서 고정 크기를 선택 후 다음으로 넘어 갔다.



[그림 11 - 파일 위치 및 크기 선택 화면]

물리적 하드 드라이브의 파일 위치를 지정한다. 나는 D드라이브에서 1테라를 사용 중이므로, D드라이브에 적당한 폴더를 만들어 위치 시켰다. 그리고 용량에 대한 여유가 아주 많아 100기가 정도 할당 했다. 위 설정은 본인이 설치하고자하는 PC의 환경에 맞게 고려하여 설정해 주면된다.



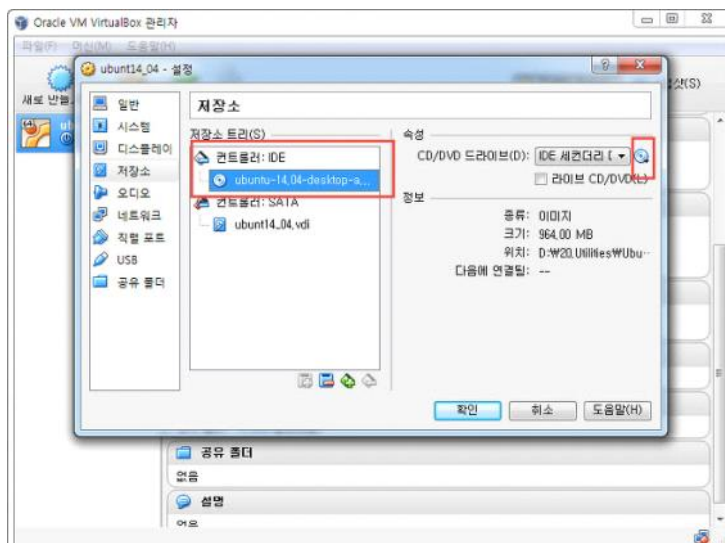
[그림 12 - 하드 드라이브 만드는 중 화면]

설정일 끝나면 하드 드라이브를 잡도록 하는데, 나는 100기가를 잡아서 시간이 좀 걸렸다. 적당히 잡았다면 금방 끝날 것이지만, 저 같이 많이 잡아 졌다면 커피 한잔하고 오면 딱 되었을 것 이다.



[그림 13 - VirtualBox 설정 완료 화면]

모든 설정이 끝났다면 [그림 13]과 같은 화면으로 나타날 것이다. 그림 설치 파일(iso파일)을 연결하기 위해 "저장소"라는 메뉴를 클릭하도록 한다.



[그림 14 - 저장소 CD/DVD 드라이브 선택화면]

[그림 14]와 같은 화면이 나타났다면 "컨트롤러 : IDE" 자식 트리를 선택한 후 CD/DVD 드라이브에서 씨디아이콘을 클릭해 우분투 설치 ISO파일 선택하도록 한다.

그리고 "확인"버튼을 클릭해 설정을 완료 하도록 한다.



[그림 15 - CD/DVD 드라이브 우분투 설치파일 연결 화면]

[그림 15]와 같이 설치 ISO파일을 연결했다면, "시작"버튼을 클릭해서 가상 머신을 작동 하도록 한다.

참고 : <http://blog.happydong.kr/252>

3.Xshell 설치 및 연결

1.우분투에서 ssh 설치

```
oracle@oracle-VirtualBox:~$ sudo apt-get install ssh
[sudo] oracle의 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
ncurses-term openssh-server openssh-sftp-server ssh-import-id
제안하는 패키지:
molly-guard monkeysphere rssh ssh-askpass
다음 새 패키지를 설치할 것입니다:
ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
6개 업그레이드, 5개 새로 설치, 0개 제거 및 1개 업그레이드 안 함.
542 k바이트 아카이브를 받아야 합니다.
이 작업 후 5,422 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] y
받기:1 http://kr.archive.ubuntu.com/ubuntu bionic/main amd64 openssh-sftp-serve
amd64 1:7.6p1-4 [45.5 kB]
받기:2 http://kr.archive.ubuntu.com/ubuntu bionic/main amd64 openssh-server amd
4 1:7.6p1-4 [332 kB]
받기:3 http://kr.archive.ubuntu.com/ubuntu bionic/main amd64 ssh all 1:7.6p1-4
5,188 B]
받기:4 http://kr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 ncurses-te
```

2. 포트 포워딩

```
oracle@oracle-VirtualBox:~$ netstat -ntl
Command 'netstat' not found, but can be installed with:
sudo apt install net-tools
```

명령어를 찾을 수 없으면 설치를 해주어야 한다.

```

oracle@oracle-VirtualBox:~$ sudo apt install net-tools
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다.
상태 정보를 읽는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
 net-tools
0개 업그레이드, 1개 새로 설치, 0개 제거 및 1개 업그레이드 안 함.
194 k바이트 아카이브를 받아야 합니다.
이 작업 후 803 k바이트의 디스크 공간을 더 사용하게 됩니다.
받기:1 http://kr.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.
61116.90da8a0-1ubuntu1 [194 kB]
내려받기 194 k바이트, 소요시간 0초 (1,074 k바이트/초)
Selecting previously unselected package net-tools.
(데이터베이스 읽는중 ...현재 165388개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack ../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...

```

우분투에서 22번 포트가 열려 있는지 확인한다.

```
$ netstat -ntl
```

```

oracle@oracle-VirtualBox:~$ netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN
tcp6       0      0 :::22                  :::*                    LISTEN
tcp6       0      0 :::1:631               :::*                    LISTEN

```

내 아이피를 확인한다.

```
cmd < ipconfig
```

```

이더넷 어댑터 VirtualBox Host-Only Network:

연결별 DNS 접미사. . . . :
링크-로컬 IPv6 주소 . . . : fe80::94b0:7bf5:3fbb:a73d%41
IPv4 주소 . . . . . : 192.168.56.1
서브넷 마스크 . . . . . : 255.255.255.0
기본 게이트웨이 . . . . . :

```

우분투의 ip를 확인한다.

```
$ ifconfig
```

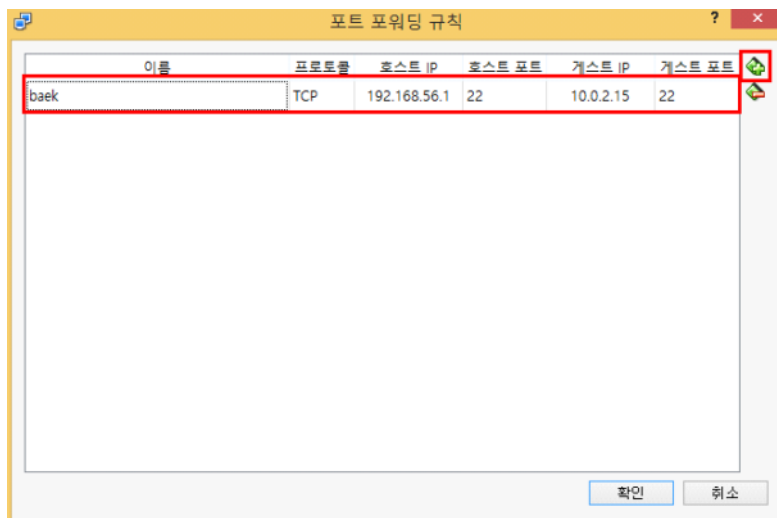
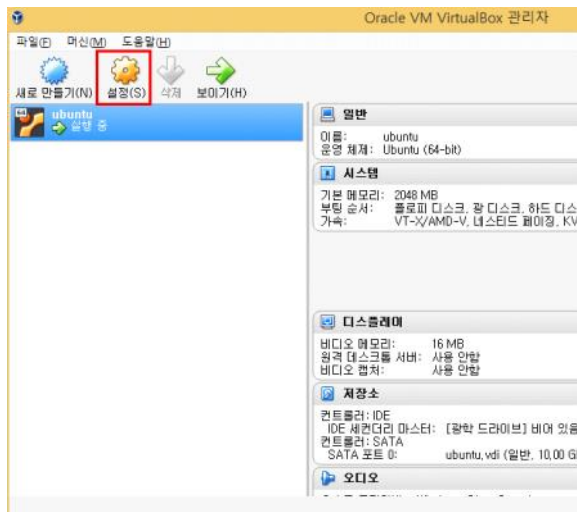
```

oracle@oracle-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::b1d0:ba71:21dd:1bf8 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1f:cc:f5 txqueuelen 1000 (Ethernet)
    RX packets 116153 bytes 167773579 (167.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6320 bytes 506795 (506.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 327 bytes 24491 (24.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 327 bytes 24491 (24.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

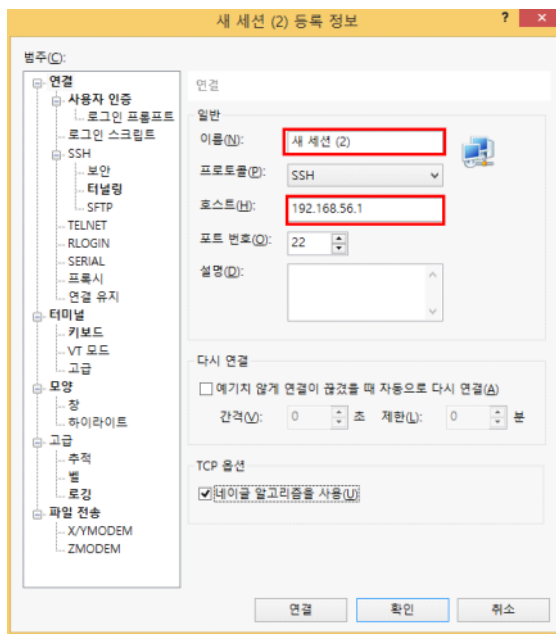
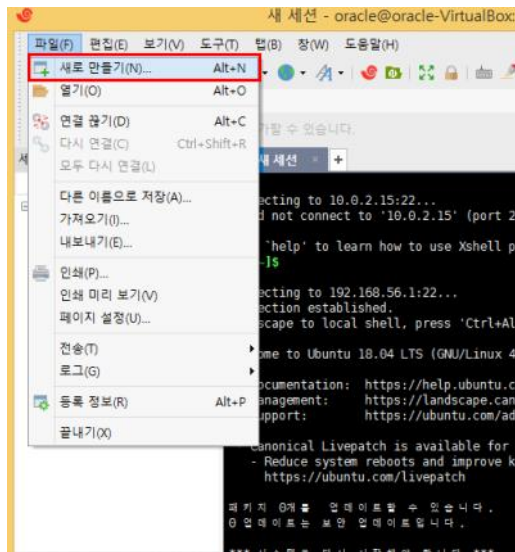
```

3. Virtual box 설정

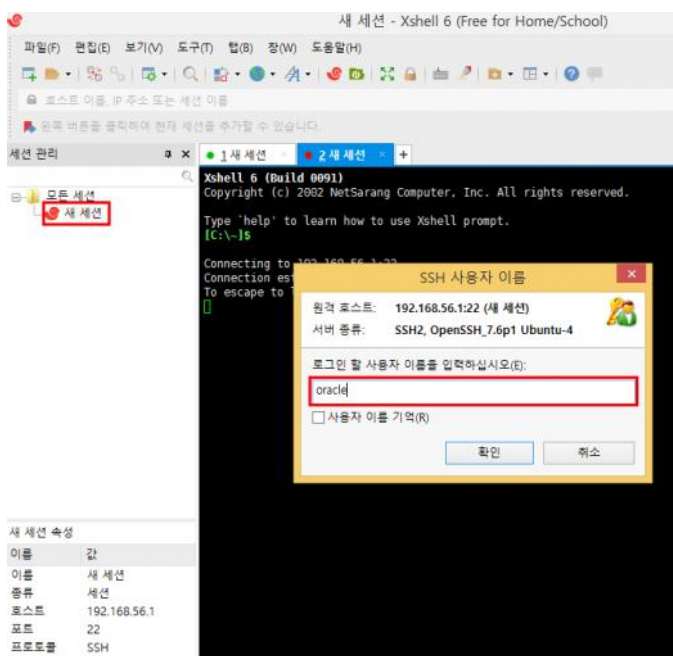


호스트 아이피엔 내 pc의 ip를, 호스트 포트는 22, 게스트 ip 는 우분투의 ip, 게스트 포트는 22로 설정한다.

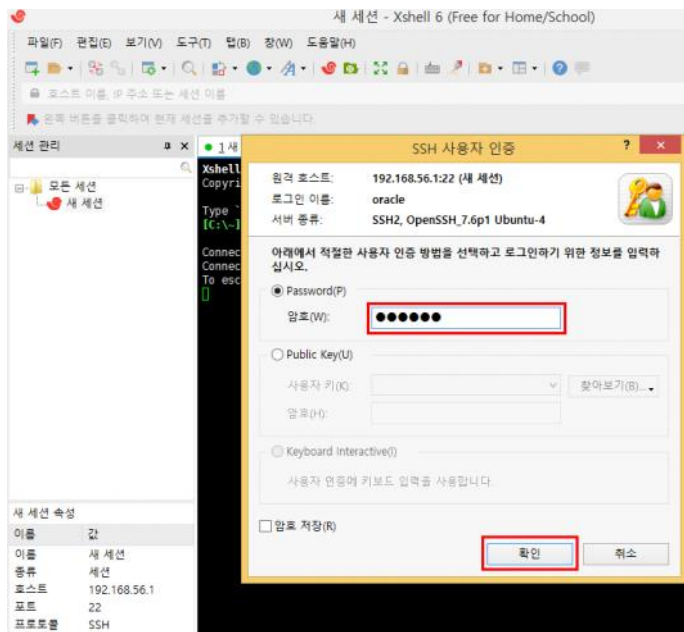
4. Xshell 접속



이름을 지정하고, 호스트를 현재 pc의 ip주소로 설정



#우분투를 설치할 때 설정한 사용자 이름 입력



#해당 사용자의 비밀번호 입력

```
Xshell 6 (Build 0091)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 192.168.56.1:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

패키지 0개를 업데이트할 수 있습니다.
0 업데이트는 보안 업데이트입니다.

*** 시스템을 다시 시작해야 합니다 ***
Last login: Sat Jun 30 23:07:48 2018 from 10.0.2.2
oracle@oracle-VirtualBox:~$ whoami
oracle
oracle@oracle-VirtualBox:~$
```

#oracle 유저로 접속이 되었는지 확인해본다.

참고 : <http://blog.naver.com/PostView.nhn?blogId=dreamspotter&logNo=220875488068>

2. 리눅스

2018년 6월 21일 목요일 오후 1:59

목차

1. 리눅스가 무엇인가?
2. 리눅스 기본 명령어
3. vi 편집기 명령어
4. 권한 관리
5. 디스크 관리
6. 프로세서 관리
7. 유저 생성 및 관리
8. 셸스크립트 작성법
9. 리눅스 환경에서 R 사용하기

■ vm웨어에 리눅스 설치하기

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

오라클 리눅스 로그인 화면

유저 : oracle

비밀번호 : oracle

* 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 분석가도 리눅스를 다루는게 필수이다. (하둡을 다룰 수 있어야 하니까)

1. 리눅스가 무엇인가?

유닉스가 너무 고가여서 리눅스 오픈소스를 리누즈 토발즈라는 핀란드 학생이 1991년 11월에 개발한 os이다. 리누즈 토발즈가 개발한 것은 리눅스의 커널(자동차 엔진)을 개발하고 소스를 무료로 공개하고 전세계의 많은 개발자들이 이 오픈소스를 가져다가 더 좋게 개선해서 다시 인터넷에 올리고 하는 작업을 반복하다 보니 리눅스 os가 유닉스보다 더 가볍고 안정적이게 되었다.

GNU 프로젝트 -----> 누구든지 배포된 오픈 소스를 가져다 개발할 수 있다. 돈을 벌 목적으로 상용화 할수도 있는데 한가지 지켜야할 약속은 이 소스를 가져다 더 좋게 수정했으면 수정한 코드를 인터넷에 공개 해야 한다.

레드햇 (상용버전, 유료) -----> cent os (무료)

Ex)

- i. Oracle linux
- ii. Ubuntu

3. 리눅스 기본 명령어

2018년 6월 25일 월요일 오후 7:26

1. 리눅스 기본 명령어

1. cd 명령어

사용방법	의미
cd (== cd ~)	자기 자신의 홈 디렉토리로 이동
cd /	루트 디렉토리(/)로 이동
cd ..	한 단계 상위 디렉토리로 이동
cd -	바로전에 위치했던 디렉토리로 이동
cd /경로 ..	해당 경로로 이동

■ 경로의 2가지 종류

1. 절대 경로 : cd 내가 가고자하는 정확한 위치
\$ cd /home/oracle/labs
2. 상대 경로
\$ cd .. : 나의 현재 위치를 상대로 상위 디렉토리로 이동하겠다.
\$ cd ./labs : 현재 디렉토리의 labs 디렉토리로 이동하겠다.

문제 1. /home/oracle 경로로 이동 하시오.

\$ cd ..

문제 2. 현재 디렉토리에서 /(루트) 디렉토리로 바로 이동 하시오.

\$ cd /

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

Cd - : 전에 있는 경로로 이동
Cd .. : 하위 경로로 이동
Cd [절대 경로] : 절대 경로로 이동

2. touch 명령어

" 용량이 0인 파일을 생성하거나 파일의 타임 스탬프를 변경하는 용도로 사용한다. "

Ex 1) 용량이 0인 빈 파일 생성하는 용도

```
$ touch a1.txt
```

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

```
[orcl:~]$ touch a1.txt
[orcl:~]$ ls
Adobe      R-3.2.3          a1.txt          labs
Desktop    R-3.2.3.tar.gz   datamodeler.log oradiag_oracle
[orcl:~]$ ls -l a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
```

권한 소유자 소유자가 속한 그룹 파일용량 생성날짜

Ex 2) 파일의 타임스탬프를 변경하는 용도

touch 파일명

(파일이 존재하는 경우 해당 파일의 타임 스탬프 값이 현재의 시간으로 업데이트 된다.)

```
[orcl:~]$ stat emp.txt
  File: `emp.txt'
  Size: 970          Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 16876427   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 500/  oracle)   Gid: ( 504/ oinstall)
Access: 2018-06-25 15:32:08.000000000 +0900
Modify: 2018-06-22 11:34:10.000000000 +0900
Change: 2018-06-25 15:38:48.000000000 +0900
[orcl:~]$ touch emp.txt
[orcl:~]$ stat emp.txt
  File: `emp.txt'
  Size: 970          Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 16876427   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 500/  oracle)   Gid: ( 504/ oinstall)
Access: 2018-06-25 19:49:17.000000000 +0900
Modify: 2018-06-25 19:49:17.000000000 +0900
Change: 2018-06-25 19:49:17.000000000 +0900
```

■ 리눅스 시스템의 파일 타임 스탬프 종류

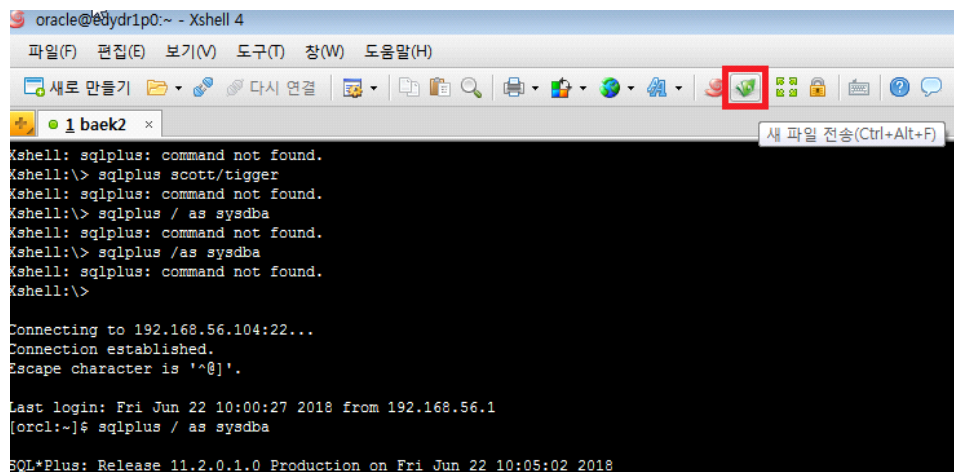
종류	설명
atime	최종 접근시각(access time)
mtime	최종 수정시각(modify time)
ctime	최종 상태변경시각(change time)

문제 3. 현재 디렉토리에 아래의 파일들을 크기를 0으로 생성 하시오.

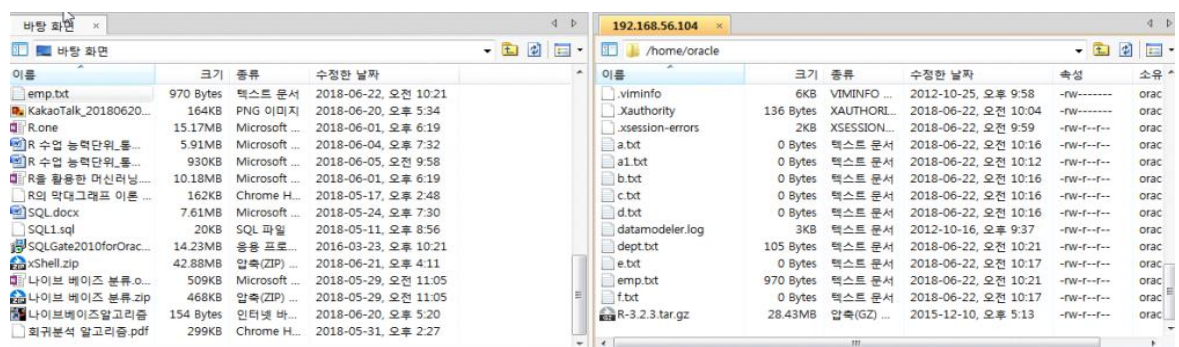
```
[orcl:~]$ touch a.txt
[orcl:~]$ touch b.txt
[orcl:~]$ touch c.txt
[orcl:~]$ touch d.txt
[orcl:~]$ touch e.txt
[orcl:~]$ touch f.txt
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 d.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 f.txt
```

문제 4. 리눅스 서버에 /home/oracle 경로에 emp.txt 를 전송 하시오.

- a. Xshell에서 새 파일 전송 버튼을 누르고



- b. 윈도우에서 리눅스 경로로 파일을 드래그 해주면 된다.



3. li 명령어

" 파일이나 디렉토리의 목록과 정보를 출력할 때 사용하는 명령어 "

옵션	설명
-a	숨겨진 파일을 포함해서 모든 내용을 출력한다.
-l	자세한 내용을 표시한다.
-r	내림차순 정렬 (기본은 오름차순 정렬)
-f	디스크에 저장된 순서대로 출력
-c	최근에 변경된 파일부터 정렬해서 출력
-R	하위 디렉토리 출력

4. mkdir 명령어

" 디렉토리를 생성하는 명령어 "

옵션	Long 옵션	설명
-m	--mode=MODE	디렉토리를 생성할 때 권한을 설정, 디폴트는 755
-p	--parents	상위 경로도 함께 생성

-v	--verbose	디렉토리를 생성 후, 생성된 디렉토리에 대해 메시지 출력
----	-----------	---------------------------------

Ex 1) 디렉토리 생성할 때 권한 설정

\$ mkdir -m 755 디렉토리명

Ex 2) 상위 경로와 함께 생성

\$ mkdir -p /경로/디렉토리명

Ex 3) 디렉토리 생성 후 각각 성공, 실패 여부를 출력한다.

\$ mkdir -v 디렉토리명1 디렉토리명2 ...

문제 5.	/home/oracle 경로에 /home/oracle/test5/test6/test7/test8/test9 디렉토리를 생성 하시오.
--------------	---

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

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

-p 옵션을 사용하면 한번에 상위-하위 디렉토리 생성 가능

문제 6.	현재 디렉토리에서 한번에 /home/oracle 로 이동 하시오.
--------------	--------------------------------------

```
[orcl:test6]$ cd /home/oracle
```

```
[orcl:test6]$ cd /home/oracle
[orcl:~]$ pwd
/home/oracle
```

문제 7.	
--------------	--

5. rm 명령어

" 파일이나 디렉토리를 삭제하는 명령어 "

* 리눅스나 유닉스는 휴지통이 없기 때문에 삭제할 때 특히 주의해야한다. (백업이 없으면 복구 불가능)

옵션	Long 옵션	설명
-f	--force	강제로 파일이나 디렉토리를 삭제하고, 삭제할 대상이 없을 경우 메시지를 출력하지 않음
-i	--interactive	매번 삭제할 때마다 사용자에게 질문함
-I		셋 이상의 파일을 삭제하거나 하위의 파일이나 디렉토리가 있을 경우 질문함
	--interactive[=WHEN]	상호대화형 모드로 값(WHEN)을 지정함 WHEN 대신 once(-i 옵션과 같음)와 always(-i 옵션과 같으며 디폴트 값)가 올 수 있음
	--no-preserve-root	'/'를 특별하게 취급하지 않음
	--preserve-root	'/'를 삭제하지 않음(디폴트 값)
-r, -R	--recursive	하위 디렉토리를 포함하여 모든 내용을 삭제
-d	--dir	빈 디렉토리들만 제거
-v	--verbose	지워지는 파일의 정보를 출력
	--help	rm 명령어 사용법을 출력

	--version	rm 명령어의 버전 정보를 출력
--	-----------	-------------------

■ rmdir 명령어

" 디렉토리를 삭제하는 명령어 (옵션이 좋은 rm 명령어를 사용하는 것을 추천) "

옵션	설명
-p	상위 디렉토리 삭제

Ex. \$ rmdir 디렉토리명
 \$ mkdir ddd
 \$ ls
 \$ rmdir ddd

문제 8.	현재 디렉토리 밑에 있는 확장자가 txt인 파일들을 모두 삭제 하시오.
-------	---

```
[orcl:~]$ rm -rf *.txt
```

-r : 현재 디렉토리 밑에 있는 모든 파일과 디렉토리를 삭제한다.
 -f : 삭제할 때 확인메시지를 띄우지않고 강제로 삭제한다.

문제 9.	/home/oracle 경로에 test9 디렉토리를 만들고 rm 안에 test10 디렉토리를 만드시오.
-------	---

```
[orcl:~]$ mkdir -p test9/test10
[orcl:~]$ ls
Adobe      R-3.2.3      datamodeler.log  oradiag_oracle  test9
Desktop    R-3.2.3.tar.gz  labs             test5
[orcl:~]$ cd test9
[orcl:test9]$ ls
test10
```

문제 10.	/home/oracle 경로에 test9 와 그 안에 있는 test10 디렉토리를 삭제 하시오.
--------	---

```
[orcl:~]$ rm -rf test9
[orcl:~]$ ls
Adobe      R-3.2.3      datamodeler.log  oradiag_oracle
Desktop    R-3.2.3.tar.gz  labs             test5
```

6. alias 명령어

" 자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어 "

Ex. 오라클 db에 접속

기존 명령어 : \$ sqlplus / as sysdba

새로운 명령어 : \$ alias sql = 'sqlplus / as sysdba
 \$ sql

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

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:15:47 2018

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

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

* alias 만 입력할 경우 현재 존재하는 별칭을 보여준다.

```
[orcl:~]$ alias
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

문제 11.	Scott 이라고 아래와 같이 명령어를 수행하면 오라클에 scott으로 접속되게 alias를 만드시오.
--------	---

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

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:18:16 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"
```

* 별칭 지우는 방법 : unalias sql

문제 12.	오라클의 네트워크 설정에 대한 디렉토리가 있는 곳으로 이동 하시오.
--------	---------------------------------------

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

문제 13.	오라클의 네트워크 설정에 대한 디렉토리가 있는 곳 바로 이동하는 별칭을 만드시오.
--------	---

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



```
[orcl:~]$ ora
[orcl:admin]$ ls
samples shrept.lst tnsnames.ora
[orcl:admin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
```

7. cat 명령어

" 파일의 내용을 화면에 출력하는 명령어 "

옵션	설명
-b	각 행에 번호를 붙여서 출력
-n	빈 행에도 번호를 붙여서 출력
-s	연속되는 2개 이상의 빈 행을 한 행으로 출력

Ex. \$ Cat 파일명.txt

\$ cat emp.txt

문제 14. Cat 으로 본 화면의 결과를 emp40.txt 라는 이름으로 저장 하시오.

\$ cat emp.txt >> emp40.txt

```
[orcl:~]$ cat emp.txt >> emp40.txt
[orcl:~]$ ls
Adobe      R-3.2.3      datamodeler.log  emp.txt  labs  test5
Desktop    R-3.2.3.tar.gz  dept.txt  emp40.txt  oradiag_oracle
```

>> : 리다이렉션 redirection // R에서 %>% 랑 비슷

문제 15. 현재 디렉토리에 아래의 파일들을 touch 명령어로 생성 하시오.

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

\$ li -l *.txt

문제 16. 지금 ls -l *.txt로 보고 있는 화면의 결과를 ls.txt라는 이름으로 저장 하시오.

```
[orcl:~]$ ls -l *.txt >> ls.txt
[orcl:~]$ ls
Adobe      R-3.2.3.tar.gz  c.txt      dept.txt  emp40.txt  oradiag_oracle
Desktop    a.txt           d.txt      e.txt     labs       test5
R-3.2.3    b.txt           datamodeler.log  emp.txt  ls.txt
```

8. redirection 명령어

" 화면에 출력되는 결과를 파일로 저장하는 명령어 "

>> : 없으면 파일을 생성하고 있으면 기존 파일 뒤에 덧붙이겠다.

> : 파일을 생성하겠다. 기존 파일이 있으면 그냥 덮어 쓰겠다.

문제 17. emp.txt의 내용을 emp50.txt 라는 이름으로 생성하고 emp.txt의 결과를 emp50.txt에 덧붙여서 저장 하시오.

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

9. wc 명령어

" 파일 안의 단어의 개수 또는 라인수를 출력 하는 명령어 "

옵션	설명
-l	라인수를 출력
-w	단어의 개수를 출력
-c	문자 수를 출력

Ex. \$ wc -l 파일명
\$ wc -l emp.txt

문제 18. 겨울 왕국 대본의 단어수가 어떻게 되는가?

```
[orcl:~]$ wc -c winter.txt
114548 winter.txt

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

# 라인수 , 단어수, 문자수
```

문제 19. /home/oracle 경로의 확장자가 .txt 로 끝나는 파일이 몇 개가 있는지 카운트 하시오.

```
[orcl:~]$ ls -l *.txt | wc -l
11
```

* |: **파이프 명령어** (앞의 명령어의 표준 출력을 뒤에 명령어의 표준 입력으로 보냄으로써 명령어의 실행 결과를 다음 명령어로 전달하는 기능

문제 20. 아래의 경로로 이동해서 아래의 디렉토리 아래있는 확장자 trc 파일이 총 몇 개가 있는지 확인 하시오.
/u01/app/oracle/diag/rdbms/orcl/orcl/trace

```
[orcl:~]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
[orcl:trace]$ ls -l *.trc | wc -l
185
```

10. more 명령어

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

■ 단축키

- h : more 명령어상태에서 사용할 수 있는 키 도움말 확인
- Space Bar : 한 화면씩 뒤로 이동하기 (f와 동일)
- Enter : 현재행에서 한 행씩 뒤로 이동하기
- q : more 명령어 종료하기
- f : 한 페이지씩 뒤로 이동하기(Space Bar 와 동일)
- b : 한 페이지씩 앞으로 이동하기
- = : 현재 위치의 행번호 표시하기
- /문자열 : 지정한 문자열을 검색하기
- n : /문자열로 검색한 문자열을 차례대로 계속해서 찾기
- !셸명령어 : more 명령어상태에서 셸명령어를 실행하기
- v : more 명령어로 열려있는 파일의 현재위치에서 vi를 실행하기

Ex. \$ more winter.txt

전진키 : space 키

후진키 : b

페이지 단위로 이동 : f

```
[orcl:~]$ more winter.txt
"Final Shooting Draft
by
Jennifer Lee
9/23/13
Property of the Walt Disney Animation Studios
OPEN ON: ICE.
We were underwater looking up at it. A saw cuts through,
heading right for us.
EXT. SNOW-CAPPED MOUNTAINS ? DUSK
ICE HARVESTERS, dressed in traditional Sami clothing, score
frozen lake. They SING.
The Frozen Heart (Ice Worker's Song)
ICE HARVESTERS
BORN OF COLD AND WINTER AIR
AND MOUNTAIN RAIN COMBINING,
THIS ICY FORCE BOTH FOUL AND FAIR
```

11. head / tail 명령어

Head : " 문서의 처음 몇 줄을 화면에 출력하는 명령어 "

옵션	설명
-n 숫자	앞에서 부터 숫자행까지 출력 (기본은 10행까지 출력)
-c 데이터크기	앞에서 부터 데이터 크기(byte)만큼 출력

Ex. \$ head 출력줄수 파일명

\$ head -20 winter.txt

Tail : "문서의 마지막 몇 줄을 화면에 출력 하는 명령어 "

옵션	설명
-n 숫자	마지막 행 부터 숫자행까지 출력 (기본은 10행까지 출력)
-c 데이터크기	마지막 행 부터 데이터 크기(byte)만큼 출력
-f	로그파일을 실시간 모니터링(파일을 실시간으로 읽을 수 있다), 종료는 Ctrl+c

Ex. \$ tail 출력줄 수 파일명

\$ tail -10 winter.txt

* tail 명령어의 중요한 옵션 : -f 옵션

문서의 끝에 추가되는 새로운 내용을 실시간으로 확인할 수 있는 옵션

Ex. 오라클 db가 올라가거나 내려갈때의 그 정보를 적는 파일을 tail -f 로 여시오.

\$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace

```
[orcl:trace]$ tail -f alert_orcl.log
db_recovery_file_dest_size of 3852 MB is 4.26% used. This is a
user-specified limit on the amount of space that will be used by this
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Fri Jun 22 10:00:54 2018
Starting ORACLE instance (normal)
Fri Jun 22 10:05:03 2018
Starting background process SMC0
Fri Jun 22 10:05:03 2018
SMC0 started with pid=24, OS id=5755
```

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 14:26:51 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> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
[orcl:trace]$ tail -f alert_orcl.log
ARCH: Archival disabled due to shutdown: 1089
Shutting down archive processes
Archiving is disabled
Archive process shutdown avoided: 0 active
Fri Jun 22 14:27:04 2018
Stopping background process VKTM:
Fri Jun 22 14:27:04 2018
NOTE: Shutting down MARK background process
Fri Jun 22 14:27:06 2018
```

예제. 현재 리눅스 시스템의 i/o 성능을 확인 하시오.

\$ sar 1 100 ---> 1 ~100 동안 확인

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

악성 sql을 날리면 실시간으로 성능이 좋지 않은 것을 확인 할 수 있다.

```
[orcl:~]$ sar 1 100
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/22/18
14:33:16          CPU      %user      %nice      %system      %iowait      %steal      %idle
```

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

14:33:16      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:33:17      all       49.75        0.00        0.50        0.00        0.00       49.75
14:33:18      all       49.25        0.00        1.49        0.00        0.00       49.25
14:33:19      all       50.25        0.00        0.00        0.00        0.00       49.75
14:33:20      all       49.75        0.00        1.00        0.00        0.00       49.25
14:33:21      all       49.75        0.00        0.50        0.00        0.00       49.75
14:33:22      all       49.50        0.00        1.50        0.00        0.00       49.00
```

Ctrl + c 를 눌러서 악성 sql을 종료한 후의 모습

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

14:35:14      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:35:15      all        0.50        0.00        1.49        0.00        0.00      98.01
14:35:16      all        0.00        0.00        1.50        0.00        0.00      98.50
14:35:17      all        0.00        0.00        1.50        0.00        0.00      98.50
14:35:18      all        0.50        0.00        2.50        0.00        0.00      97.00
14:35:19      all        0.00        0.00        1.50        0.00        0.00      98.50
```

문제 21. 겨울 왕국 대본의 100줄까지의 내용을 winter_head.txt 라는 이름으로 저장 하시오.

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

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

Sar 1 100 >> sar_0622.txt

현업에서는 시스템 부하 상태를 기록하기 위해 값을 많이 주고 항상 커논다 --> 백그라운드로 실행하는게 좋다 (끝수도 있으니까)

Sar 1 100 >> sar_0622.txt & # & : 백그라운드로 명령 실행

12. grep 명령어

" 파일안에 포함된 특정 단어나 구문을 검색하는 명령어 "

옵션	설명
-c	count 옵션은 파일 이름 다음에 패턴이 들어 있는 라인의 수를 표시
-h	hide 옵션은 파일이름을 출력하지 않음
-i	ignore 옵션은 대소문자 구분하지 않고 찾음
-n	line number 옵션은 출력하는 각 라인에 라인번호를 붙임
-v	inverse 옵션은 패턴이 없는 라인만 출력
-w	word 옵션은 패턴 자체가 완전한 단어로 독립되었을 경우만 글자로 인식함

Ex. \$ grep [옵션] '찾고싶은단어' 파일명

\$ grep -i 'scott' emp.txt # emp.txt에서 'scott'이란 단어가 포함된 라인을 보여달라.

-i : 대소문자를 구분하지 않겠다.

문제 23.	직업이 salesman인 직원들의 모든 행을 출력 하시오.
---------------	----------------------------------

```
[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 이름, 월급을 출력 하시오.
---------------	------------------------------------

```
[orcl:~]$ grep -i 'salesman' emp.txt | awk '{print $2,$6}'
ALLEN 1600
WARD 1250
MARTIN 1250
TURNER 1500
```

13. awk 명령어

" 특정 단어가 들어가 있는 라인에서 특정 컬럼명을 출력할 때 사용하는 명령어 "

패 턴	내 용
BEGIN	입력화일을 읽어들이기 전에 옆에 제시되는 문자를 실행시키도록 한다.
END	awk가 모든 입력을 처리한 후, 옆에 제시되는 문장을 실행시키도록 한다.
expression (식)	식을 평가한 후 이 식이 참, 즉 non-zero이거나 non-null인 경우 문장을 실행한다.
/re/	정규식과 일치하는 문자열을 포함하고 있는 라인에서 문장을 실행한다.
compound-pattern	복합패턴이라는 것으로 &&(and), (or) , !(not) 그리고 괄호에 의해 연결시킨 것이다. expression의 경우와 마찬가지로 복합 패턴도 참인 경우의 문장을 실행시킨다.
pattern1, pattern2	이러한 패턴을 범위 패턴이라한다. 현재 처리되고 있는 라인이 pattern1과 일치되고, 다음에 따라오는 라인

중 임의의 라인이 pattern2와 일치할 때, 범위 패턴은 두 라인 사이의 각 라인과 일치한다.

Ex. \$ awk '패턴(action)' 대상파일명

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

\$ awk '\$3 == "salesman" {print \$2, \$3}' emp.txt

[orcl:~]\$ awk '\$8 == 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

문제 25. 부서번호가 10번인 직원들의 이름, 월급을 출력 하시오.

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

-w : 정확히 매치 하는것만 보여줌 (10이라고 검색했을때 1000 은 뜨지 않고 10일때만 뜸)

문제 26. 월급이 3000 이상인 직원들의 이름, 월급을 출력 하시오.

```
[orcl:~]$ awk '$6>=3000 {print $2, $3}' emp.txt
SCOTT ANALYST
KING PRESIDENT
FORD ANALYST
```

* 리눅스의 연산자 3가지

- i. 산술연산자 : +, -, *, /
- ii. 비교연산자 : >, <, >=, <=, ==, !=
- iii. 논리연산자 : &&, ||, !

문제 27. 직업이 salesman이 아닌 직원들의 이름, 직업을 출력 하시오.

```
[orcl:~]$ awk '$3 != toupper("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
```

문제 28. 직업이 salesman인 직원들의 이름, 월급을 출력 하는데 월급이 1500 이상인 직원들만 출력 하시오.

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

문제 29. 이름 A로 시작하는 직원들의 이름과 월급을 출력 하시오.
힌트 : grep과 정규식 (시작 : ^ , 끝 : \$)을 사용하고 awk, grep과 |를 활용해서 출력

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

문제 30. 81년도에 입사한 직원들의 이름, 입사일을 출력 하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep '81$'
ALLEN 20-FEB-81
WARD 22-FEB-81
JONES 02-APR-81
MARTIN 28-SEP-81
BLAKE 01-MAY-81
CLARK 09-JUN-81
KING 17-NOV-81
TURNER 08-SEP-81
JAMES 03-DEC-81
FORD 03-DEC-81
```

문제 31. 81년도에 입사하지 않은 직원들의 이름, 입사일을 출력 하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep -v 81
SMITH 17-DEC-80
SCOTT 19-APR-87
ADAMS 23-MAY-87
MILLER 23-JAN-82
```

문제 32. 겨울왕국 대본에서 elsa가 포함되어있는 라인들의 개수가 몇 개인지 출력 하시오.

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

* 리눅스에서 R을 쉽게 실행 하는 방법

```
[orcl:~]$ ls
Adobe          b.txt          e.txt          ls.txt          winter_head.txt
Desktop        c.txt          emp.txt        oradiag_oracle
R-3.2.3        d.txt          emp40.txt      sar_0622.txt
R-3.2.3.tar.gz datamodeler.log emp50.txt      test5
a.txt          dept.txt       labs           winter.txt
[orcl:~]$ vi .bash_profile
```

/home/oracle 경로에서 vi .bash_profile을 열어 명령어를 추가해준다.

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



```
# .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=$ORACLE_HOME/bin:$PATH
export PATH=/home/oracle/R-3.2.3/bin:$PATH
unset LANG

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

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i686-pc-linux-gnu (32-bit)
```

14. sort 명령어

" data를 특정 컬럼을 기준으로 정렬하는 명령어 "

sort [-옵션] [-o 저장될 파일명] 정렬할 파일명 [병합할 파일명....]

옵션	설명
-n	라인의 각 필드를 비교하는 대상을 숫자로 한정
-f	영어를 정렬할 때, 대소문자 구별 안함
-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
-r	출력 순서를 역순으로

■ 사용 예

우선 sort 명령을 확인해보기 위해서 다음과 같이 sort.txt 파일을 생성했다.

```
[naddru@ ~]$ cat sort.txt
banana 3
camera 1
apple 4
date 2
```

그 후, 기본적인 sort 명령을 사용해 보면 다음과 같은결과가 나타난다.

```
sort [파일명]
```

```
[naddru@ ~]$ sort sort.txt
```

■ 사용 예

우선 sort 명령을 확인해보기 위해서 다음과 같이 sort.txt 파일을 생성했다.

```
[naddru@ ~]$ cat sort.txt
banana 3
camera 1
apple 4
date 2
```

그 후, 기본적인 sort 명령을 사용해 보면 다음과 같은결과가 나타난다.

sort [파일명]

```
[naddru@ ~]$ sort sort.txt
apple 4
banana 3
camera 1
date 2
```

만약 **두번째 필드를 기준으로 정렬**하고 싶다면,

sort +1 [파일명]

```
[naddru@ ~]$ sort +1 sort.txt
camera 1
date 2
banana 3
apple 4
```

마지막으로 **파일에 저장**하고 싶다면 -o 옵션을 사용한다.

sort [파일명] -o [저장할 파일명]

예제 : \$ sort 옵션 파일명

\$ sort -k 6 emp.txt

-k : 오름차순 , -rk : 내림차순

Emp.txt 의 모든 데이터를 출력하는데 6번째 컬럼 (월급)이 적은 것부터 출력 하시오.

문제 34.	위의 결과를 다시 출력하는데 이름, 월급만 나오게 하시오.
--------	----------------------------------

```
[orcl:~]$ sort -k 6 emp.txt | awk '{print $2,$6}'
SMITH 800
JAMES 950
ADAMS 1100
WARD 1250
MARTIN 1250
MILLER 1300
TURNER 1500
ALLEN 1600
CLARK 2450
BLAKE 2850
JONES 2975
SCOTT 3000
FORD 3000
KING 5000
```

문제 35.	이름, 월급을 출력하는데 월급이 높은 것부터 출력 하시오.
--------	----------------------------------

```
[orcl:~]$ sort -rk 6 emp.txt | awk '{print $2,$6}'
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 이름, 월급을 출력하는데 월급이 높은 것부터 출력 하시오.

```
[orcl:~]$ sort -k 6 emp.txt | awk '$3==toupper("salesman") {print $2,$6}'
WARD 1250
MARTIN 1250
TURNER 1500
ALLEN 1600
```

문제 37. 직원 테이블의 월급의 토탈 값을 출력 하시오.

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

문제 38. 직업이 SALESMAN인 직원들의 토탈 월급을 출력 하시오.

```
[orcl:~]$ awk '$3==toupper("salesman") {print $6}' emp.txt | awk '{sum+=$1} END {print sum}'
5600
```

문제 39. 아래와 같이 결과가 출력되게 하시오.

King 의 월급은 5000입니다. ...

```
[orcl:~]$ awk '{print $2 "의 월급은 " $6 "입니다."}' emp.txt
SMITH의 월급은 800입니다.
ALLEN의 월급은 1600입니다.
WARD의 월급은 1250입니다.
JONES의 월급은 2975입니다.
MARTIN의 월급은 1250입니다.
BLAKE의 월급은 2850입니다.
CLARK의 월급은 2450입니다.
SCOTT의 월급은 3000입니다.
KING의 월급은 5000입니다.
TURNER의 월급은 1500입니다.
ADAMS의 월급은 1100입니다.
JAMES의 월급은 950입니다.
FORD의 월급은 3000입니다.
MILLER의 월급은 1300입니다.
```

문제 40. 아래와 같이 결과가 출력되게 하시오.

```
[orcl:~]$ awk '{print $2 "의 직업은 " $3 "입니다."}' emp.txt
SMITH의 직업은 CLERK입니다.
ALLEN의 직업은 SALESMAN입니다.
WARD의 직업은 SALESMAN입니다.
JONES의 직업은 MANAGER입니다.
MARTIN의 직업은 SALESMAN입니다.
BLAKE의 직업은 MANAGER입니다.
CLARK의 직업은 MANAGER입니다.
SCOTT의 직업은 ANALYST입니다.
KING의 직업은 PRESIDENT입니다.
TURNER의 직업은 SALESMAN입니다.
ADAMS의 직업은 CLERK입니다.
JAMES의 직업은 CLERK입니다.
FORD의 직업은 ANALYST입니다.
MILLER의 직업은 CLERK입니다.
```

문제 41. /home/oracle 에서 ls -lrt 라고 조회한 결과를 분석 하시오.

```
$pwd
```

```
$ls -lrt
```

-l : 리스트

-r : 리버스

-t : 시간

----> 파일 리스트를 옛날 것 부터 보여달라 !

```
[orcl:~]$ ls -lrt
total 29336
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
-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 16:55 R-3.2.3
drwx----- 2 oracle oinstall 4096 Jun 22 09:52 Desktop
drwxr-xr-x 3 oracle oinstall 4096 Jun 22 10:28 test5
-rw-r--r-- 1 oracle oinstall 970 Jun 22 11:34 emp.txt
-rw-r--r-- 1 oracle oinstall 105 Jun 22 11:34 dept.txt
-rw-r--r-- 1 oracle oinstall 1940 Jun 22 11:34 emp40.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 d.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 11:37 a.txt
-rw-r--r-- 1 oracle oinstall 433 Jun 22 11:38 ls.txt
-rw-r--r-- 1 oracle oinstall 3880 Jun 22 11:43 emp50.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 22 11:47 winter.txt
-rw-r--r-- 1 oracle oinstall 3387 Jun 22 14:18 winter_head.txt
-rw-r--r-- 1 oracle oinstall 9790 Jun 22 14:45 sar_0622.txt
```

문제 42. ls -lrt *.txt 했을 때 파일 크기에 해당하는 부분만 출력 하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{print $5}'
970
105
1940
0
0
0
0
0
0
433
3880
114548
3387
9790
```

문제 43. 위에서 출력된 숫자들을 모두 sum 해서 출력 하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{sum+= $5 } END {print sum}'
135053
```

문제 44. 아래와 같이 결과를 출력되게 하시오.

현재 디렉토리에 있는 txt 파일들의 총 크기는 135053 바이트 입니다.

```
[orcl:~]$ ls -lrt *.txt | awk '{sum+= $5 } END {print "현재 디렉토리에 있는 txt 파일들의 총 크기는 "sum "바이트 입니다"}'
현재 디렉토리에 있는 txt 파일들의 총 크기는 135053바이트 입니다
```

문제 45. 위의 스크립트를 쉘 스크립트로 저장해서 실행되게 하시오.

* 자주 쓰는 명령어를 쉘 스크립트로 저장해 뒀다가 사용한다.

```
ls -lrt *.txt | awk '{sum+= $5 } END {print "현재 디렉토리에 있는 txt 파일들의 총 크기는 "sum "바이트 입니다"}'
```

```
[orcl:~]$ vi size.sh
[orcl:~]$ sh size.sh
현재 디렉토리에 있는 txt 파일들의 총 크기는 135053바이트 입니다
```

15. uniq 명령어

" 중복된 라인을 제거하는 명령어 "

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 라인까진 검사하지 않는다
-i	중복 라인을 한라인으로 생각하고 출력한다.
-u	중복 라인이 없는 것만 보여준다.
-w	N 번째 문자까지만 비교대상으로 하여 uniq 명령을 수행
-s	N 번째 문자까지는 비교 대상에서 제외하고 uniq 명령을 수행
-f	N 번째 필드를 비교대상에서 제외하고 uniq 명령을 수행

Ex. \$ uniq [옵션] 파일명

문제 46. emp.txt에서 직업만 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK
```

문제 47. 위의 결과를 abcd 순으로 정렬해서 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort
ANALYST
ANALYST
CLERK
CLERK
CLERK
CLERK
CLERK
MANAGER
MANAGER
MANAGER
MANAGER
PRESIDENT
SALESMAN
SALESMAN
SALESMAN
SALESMAN
```

* 내림 차순

```
[orcl:~]$ awk '{print $3}' emp.txt | sort -rf
SALESMAN
SALESMAN
SALESMAN
SALESMAN
PRESIDENT
MANAGER
MANAGER
MANAGER
CLERK
CLERK
CLERK
CLERK
ANALYST
ANALYST
```

문제 48. 위의 결과를 중복을 제거해서 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort | uniq
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제 49. 부서번호를 출력하는데 중복 제거해서 출력 하시오.

```
[orcl:~]$ awk '{print $8}' emp.txt | sort | uniq
10
20
30
```

문제 50. Emp.txt 를 복사해서 emp7.txt를 생성하고 emp7.txt에 king에 대한 데이터를 복사해서 맨 아래에 하나를 입력하고 저장 하시오.

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

복사할 로우에 커서를 위치한 후 YY를 눌러서 복사하고 맨 아래로우로 커서를 옮긴 후 P를 눌러서 붙여넣는다.

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

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

문제 51. emp7.txt 에서 중복된 라인만 출력 하시오.

```
[orcl:~]$ sort emp7.txt | uniq -d
7839 KING PRESIDENT 0 17-NOV-81 5000 0 10
```

문제 52. emp7.txt 에서 중복된 라인을 제거하고 출력 하시오.

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

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

중복된 King이 하나도 출력되지 않는다.

16. echo 명령어

" 출력하고자 하는 글자를 출력할 때 사용하는 명령어 (print 함수와 비슷한 역할 --> 주로 스크립트에 사용)"

옵션	설명
-n	마지막에 따라오는 개행 문자를 출력하지 않음 (줄 바꾸지 않음)
-e	문자열 중에 백슬래쉬와 조합되는 특수 문자를 인식하게 된다.

이스케이프 문자	설명
\a	경고음 소리 출력
\b	백스페이스
\c	마지막 개행 문자를 출력 안 함
\f	폼 피드 형식(프린터에서 용지 바꿈)으로 출력
\n	개행 문자 출력
\r	개리지 리턴
\t	수평 탭
\v	수직 탭
\w	역슬래시
\wnnn	ASCII Code 가 nnn(8진수)인 문자

Ex. \$ echo '나는 나다'

문제 53. 직업을 입력 받아 해당 직업을 갖는 직원들의 이름, 직업이 출력되게 하시오.

#!/bin/bash 리눅스 스크립트 중에 shell 스크립트를 사용 하겠다.

```
#!/bin/bash
echo "
echo -n "직업을 입력하세요 ~~~"
read job
echo "
grep -i $job emp.txt | awk '{print $2, $3}'
```

echo -n 옵션을 안주면 줄 바꿈

```
직업을 입력하세요 ~~~
salesman
```



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

직업을 입력하세요 ~~~SALESMAN

ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
[orcl:~]$ vi job2.sh
[orcl:~]$ sh job2.sh

직업을 입력하세요 ~~~salesman

ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
```

문제 54. 부서번호를 입력 받아 해당 부서번호에 해당되는 직원들의 이름, 월급, 부서번호가 출력되게 하시오.

```
[orcl:~]$ vi deptno2.sh
[orcl:~]$ sh deptno2.sh

부서번호를 입력 하시오 : 20

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

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

문제 55. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 월급이 아래와 같이 출력되게 하시오.
scott의 월급은 3000 입니다.

```
echo "
echo -n "이름을 입력 하시오 : "
read ename
echo "
awk ' {print $2 "의 월급은 " $6 "입니다."}' emp.txt | grep -iw $ename
```

```
[orcl:~]$ sh ename2.sh

이름을 입력 하시오 : scott

SCOTT의 월급은 3000입니다.
```

문제 56. 직업을 물어보게 하고 직업을 입력하면 해당 직업의 토탈 월급이 아래와 같이 출력되게 하시오.

```
echo "
echo -n "직업 입력하세요 ~ "
read job
echo "
grep -iw $job emp.txt | awk '{sumsal += $6} END {print "토탈월급은" sumsal "입니다." }'
```

```
[orcl:~]$ sh job_sumsal.sh
직업 입력하세요 ~ salesman
토탈월급은5600입니다.
```

문제 57. 입사일을 출력하는데 뒤에 년도만 잘라서 출력 하시오.

* awk {} 괄호 안에 substr 함수 사용가능 !

```
[orcl:~]$ awk '{print substr($5,8,10)}' emp.txt
80
81
81
81
81
81
81
87
81
81
87
81
81
82
```

문제 58. 년도를 물어보게 하고 년도를 입력하면 해당 년도에 입사한 직원들의 이름, 입사일이 출력되게 하시오.

```
echo "
echo -n "년도를 입력하세요 : "
read year

awk 'substr($5,8,10)=="$year" {print $2, $5}' emp.txt

[orcl:~]$ vi year.sh
[orcl:~]$ sh year.sh

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

문제 59. 년도를 물어보게 하고 년도를 입력하면 해당 년도의 토탈 월급이 출력되게 하시오.

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

년도를 입력하세요 : 81
토탈월급은 22825 입니다.

echo "
echo -n "년도를 입력하세요 : "
read year

awk 'substr($5,8,10)=="$year" {sum += $6} END {print "토탈월급은 " sum " 입니다."}' emp.txt
```

17. diff 명령어

" 두 파일간의 차이점을 찾아서 알려주는 명령어 "

옵션	설명
-b	연속된 공백 무시
-i	대소문자를 구분하지 않음
-t	출력 라인에 TAB 문자를 넣음
-w	두 행의 비교 시 공백을 무시
--brief	파일 비교 후 결과 표시 (같은지, 다른지)
-d	세세한 차이까지 검색
-H	큰 파일을 빠르게 처리할 때 사용
-q	두 파일의 차이점만 출력
-s	두 파일이 같은지 확인
-r	두 디렉토리의 차이점 출력
-N	검색 중 new file도 적용 가능

Ex. \$ diff emp.txd emp7.txt

문제 60. 아래와 같이 두개의 파일 이름을 각각 물어보게 하고 결과로 파일의 내용의 차이가 출력되게 하시오.

```
echo "      "
echo -n "비교할 첫 번째 파일을 입력하세요."
read f1
echo -n "비교할 두 번째 파일을 입력하세요."
read f2

diff $f1 $f2

[orcl:~]$ vi diff.sh
[orcl:~]$ sh diff.sh

비교할 첫 번째 파일을 입력하세요.emp.txt
비교할 두 번째 파일을 입력하세요.emp7.txt
14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```

18. find 명령어

" 검색하고 하는 파일을 찾을 때 사용하는 명령어 "

옵션	설명
-empty	비어 있는 파일, 단 일반파일 또는 일반 디렉토리에 한함
-uid n	지정된 n값의 UID를 갖는 파일
-gid n	지정된 n값의 GID를 갖는 파일
-group 그룹명	지정한 그룹명에 소속된 파일
-name	지정된 형식의 패턴을 가지는 파일

-newer 파일	지정한 파일보다 더 최근에 변경된 적이 있는 파일
-used n	파일이 변경된 이후 최근 n일 이후에 액세스된 적이 있는 파일
-user uname	지정한 uname의 소유로 된 모든 파일(UID로 지정가능)
-amin n	최근 n분 이전에 액세스된 적이 있는 파일
-anewer 파일	지정한 파일보다 더 최근에 액세스된 적이 있는 파일
-atime n	최근 n*24 시간 전에 액세스된 적이 있는 파일
-mmin n	최근 n분 이전에 변경된 적이 있는 파일
-mtime n	최근 n*24 시간 이전에 변경된 적이 있는 파일
-executable	실행 가능한 파일
-readable	읽기 가능한 파일
-writable	쓰기 가능한 파일
-fstype type	지정한 파일시스템타입(type)에 존재하는 파일
-nogroup	소속그룹이 존재하지 않는 파일
-nouser	소속 사용자(ID)가 존재하지 않는 파일
-path 패턴	파일명이 패턴에 매칭되는 파일
-perm 모드	모드에서 지정된 퍼미션과 일치하는 퍼미션을 가진 파일만 해당됨.
-perm -mode	모드에서 지정된 퍼미션을 가진 모든 파일.
-regex 패턴	패턴에서 지정한 정규표현식에 매칭되는 파일
-size n[cwbkMG]	지정된 n 단위의 디스크공간을 사용하는 파일. 디폴트는 b
-type 타입	지정한 파일타입에 해당하는 파일. b,c,d,p,f,l,s 등이 있음

Ex. \$ find 디렉토리 -name 파일명 -print

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

설명 : /home/oracle 경로에 emp.txt라는 파일이 있는지 검색 하시오.

문제 61. /home/oracle 경로에 test100과 test200 이라는 디렉토리를 생성하고 그 밑에 emp.txt를 각각 복사하시오.

```
[orcl:~]$ mkdir test100 test200
[orcl:~]$ ls
Adobe          a.txt  datamodeler.log  e.txt      emp7.txt      labs        size.sh  winter.txt
Desktop        b.txt  dept.txt         emp.txt    ename2.sh     ls.txt      test100  winter_head.txt
R-3.2.3        c.txt  deptno2.sh       emp40.txt  job2.sh       oradiag_oracle test200  year.sh
R-3.2.3.tar.gz d.txt  diff.sh          emp50.txt  job_sumsal.sh sar_0622.txt test5    year_sum.sh
[orcl:~]$ cp emp.txt ./test100/emp.txt
[orcl:~]$ cp emp.txt ./test200/emp.txt
```

문제 62. /home/oracle 밑에 있는 emp.txt를 검색 하시오.

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

----> 현재 디렉토리 안에 있는게 모두 검색됨

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

-----> maxdepth 옵션을 줘서 몇 번째 하위 디렉토리까지 검색할지 정할 수 있다.

문제 63. 파일을 검색하는 셸 스크립트를 아래와 같이 생성 하시오.

```
echo " "
echo -n "현재 경로에서 찾을 파일명을 입력 하시오. : "
read f_name
echo -n "depth 를 입력하세요 : "
read dpth

find . -maxdepth $dpth -name $f_name

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

현재 경로에서 찾을 파일명을 입력 하시오. : emp.txt
depth 를 입력하세요 : 2
./test100/emp.txt
./test200/emp.txt
./emp.txt
[orcl:~]$
```

19. tar 명령어

" 파일을 압축하고 압출 해제하는 명령어 "

옵션	설명
-c	compress, 여러 개의 파일을 하나로 만들어라 (tar로 묶음)
-p	파일 권한을 저장
-v	view, 묶거나 파일을 풀 때 과정을 화면으로 출력
-f	file, 생성되는 파일명을 지정
-C	압축이 풀릴 위치를 지정
-x	tar 압축을 풀
-z	gzip으로 압축하거나 해제함

1. 압출할 때 : \$ tar cvf 압출파일명 압출파일대상
2. 압출 풀 때 : \$ tar xvf 압축파일명 압축 해제할 대상

문제 64. /home/oracle 밑에 test64 이라는 디렉토리를 만들고 그 밑에 emp.txt를 복사해서 아래의 파일들을 생성 하시오.
a.txt, b.txt, c.txt, d.txt, e.txt

```
[orcl:~]$ mkdir test64
[orcl:~]$ cp emp.txt ./test64/a.txt
[orcl:~]$ cp emp.txt ./test64/b.txt
[orcl:~]$ cp emp.txt ./test64/c.txt
[orcl:~]$ cp emp.txt ./test64/d.txt
[orcl:~]$ cp emp.txt ./test64/e.txt
```

문제 65. Test64 디렉토리 밑에 있는 확장자 txt 파일들을 all.tar라는 이름으로 압축 하시오.

```
[orcl:~]$ tar -cvf txttar.tar ./test64/*.txt
./test64/a.txt
./test64/b.txt
./test64/c.txt
./test64/d.txt
./test64/e.txt
```

문제 66. /home/oracle 밑에 test66이라는 디렉토리를 생성하고 그 밑에 all.tar 압축파일의 압축을 해제하시오.

```
[orcl:~]$ tar -xvf all.tar -C ./test66
./test64/a.txt
./test64/b.txt
./test64/c.txt
./test64/d.txt
./test64/e.txt
```

```
[orcl:~]$ ls
Adobe          b.txt          diff.sh        ename2.sh      oradiag_oracle test64          year_sum.sh
Desktop        c.txt          e.txt          find_file.sh   sar_0622.txt   test66
R-3.2.3        d.txt          emp.txt        job2.sh        size.sh        txttar.tar
R-3.2.3.tar.gz datamodeler.log emp40.txt      job_sumsal.sh test100        winter.txt
a.txt          dept.txt       emp50.txt     labs           test200        winter_head.txt
all.tar        deptno2.sh     emp7.txt      ls.txt         test5          year.sh
```

```
[orcl:~]$ cd test66
[orcl:test66]$ ls
test64
[orcl:test66]$ cd test64
[orcl:test64]$ ls
a.txt b.txt c.txt d.txt e.txt
```

20. ln(link) 명령어

" 윈도우 바로가기 기능과 유사함 "

" 내가 자주 열어봐야하는 파일이 있다면 바로가기를 생성해 놓으면 편하게 열어볼 수 있다. "

■ 심볼릭링크 (Symbolic Link)

단순히 원본파일을 가리키도록 링크만 시켜둔 것으로 MS의 윈도우시스템에서 흔히 사용하는 '바로가기' 같은 것이며, 원본파일을 가리키고 만 있으므로 원본파일의 크기와는 무관한다. 그리고 심볼릭링크에서는 원본파일이 삭제되어 존재하지 않을 경우에 링크파일은 깜박거리면서 링크파일의 원본파일이 없다는 것을 알려준다.

■ 하드링크 (Hard Link)

원본파일과 다른 이름으로 존재하는 동일한 파일이며 원본파일과 동일한 내용의 다른 파일이라고 할 수 있다. 그리고 하드링크에서는 원본 파일과 링크파일 두개가 서로 다른 파일이기 때문에 둘 중 하나를 삭제하더라도 나머지 하나는 그대로 남아 있다. 또한 하드링크에서는 원본 파일의 내용이 변경될 경우에는 링크파일의 내용 또한 자동으로 변경된다.

옵션	설명
--backup[=CONTROL]	대상파일이 이미 존재할 경우에 백업파일을 만든 후에 링크파일 생성
-b	링크파일 생성시에 대상 파일이 이미 존재하면 백업파일을 만든 후에 링크 파일을 생성
-d	디렉토리에 대한 하드링크 파일생성을 가능하게 함. 단 root 권한으로 수행하더라도 시스템의 권한 제한으로 인하여 실패할 가능성이 높음
-f	대상파일이 존재할 경우에 대상파일을 지우고 링크파일을 생성
-i	대상파일이 존재할 경우에 대상파일을 지울것인가를 확인요청 (--interactive 와 동일)
-s	심볼릭 링크파일을 생성
-S	백업파일 생성시에 원하는 접미사(suffix)를 지정할 수 있음
-t	링크파일을 생성할 디렉토리를 지정

Ex. \$ ln -s "링크를 걸 파일위치와 파일명" "바로가기 이름"

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

```
[orcl:test64]$ ln -s /u01/app/oracle/product/11.2.0/dbhome_1/sqlplus/admin/glogin.sql glogin.sql
[orcl:test64]$ glogin.sql
-bash: glogin.sql: command not found
[orcl:test64]$ view glogin.sql
```

.bash_profile 설정하는 방법

.bash_profile 이란 ? Oracle 유저를 위한 리눅스 환경설정 파일

~/bashrc (bash run control)

bash를 사용할 때, bash가 참고할 사항을 정의해 놓는 파일이다.

환경변수(JAVA_HOME 등)나 Alias(ls -al --> ll 등)등을 미리 정의해 놓고 사용할 수 있도록 한다.

~/bash_profile

.bash_profile은 bash를 login shell로 사용할 때 참조된다. 즉 **bash로 로그인하면 bash_profile에 정의되어 있는 걸 사용할 수 있다.** .bash_profile에서 보통 .bashrc를 부르기 때문에 .bashrc에 있는건 자동으로 참조 된다.

리눅스 GUI에서 터미널을 열어 bash를 사용할 땐, .bash_profile이 참조 안 될 수 있으니 주의해야 한다.

/etc/bashrc

~/bashrc가 개인용 환경 설정이라면 /etc/bashrc는 전체용 환경 설정이다.

/etc/profile

bashrc와 마찬가지로 시스템 전체 환경 설정 파일이다. (로그인시 수행되는...)

* 위에서 커맨드 셸이란?

도스에서 command.com 셸이 사용자의 명령어(DIR 같은..)를 받아 특정 행동을 수행해주는 것 과 같이 사용자의 특정 명령어를 받아 OS한테 요로요러 한걸 해주세요라고 대신 말 해주는 놈이라 보면 된다.

* 참고로 shell에서 export 명령어는 환경변수를 지정하는 명령어이다.

Ex. vi .bash_profile

맨 아래쪽에 아래의 내용을 적고 저장하고 나온다.

Echo -e "여기는 운영서버 입니다. 조심히 작업 하세요 ~"

\$ su - oracle

```
# .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=$ORACLE_HOME/bin:$PATH
export PATH=/home/oracle/R-3.2.3/bin:$PATH
unset LANG

echo -e "여기는 운영 서버 입니다. 조심히 작업하세요 ~"

[orcl:~]$ su - oracle
Password:
여기는 운영 서버 입니다. 조심히 작업하세요 ~
```

문제 67.	Oracle 유저로 접속할 때마다 아래의 메시지가 출력되게 하시오. 이랏샤이 마세이~~
---------------	---

```
echo -e "이랏샤이 마세이 ~~"

[orcl:~]$ su - oracle
Password:
이랏샤이 마세이 ~~
```

21. sed 명령어

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

Ex. \$ sed 's/KING/yyy/' emp.txt

----> emp.txt 를 출력할 때 KING을 YYY로 변경해서 출력한다.

■ 명령어

aW : 현재 행에 하나 이상의 새로운 행을 추가한다.
cW : 현재 행의 내용을 새로운 내용으로 바꾼다.
iW : 현재 행의 위에 삽입한다.
d : 행을 삭제한다.
l : 출력되지 않는 특수문자를 출력한다.
p : 행을 출력한다.
n : 다음 입력 행을 첫 번째 명령어가 아닌 다음 명령어에서 처리하게 한다.
q : sed를 종료한다.
r : 파일로부터 행을 읽어온다.
! : 선택된 행을 제외한 나머지 전체 행에 명령어를 적용한다.
s : 문자열을 치환한다.

■ 옵션

-e : 다중 편집시 사용.
-f : sed 스크립트 파일 지정에 사용.
-n : 출력을 생략한다.

■ 정규식

- []: 한문자 매치
- ^: 라인 맨앞
- \$: 라인 끝
- &: 검색문자열 치환
- \w<: 단어의 시작
- >: 단어의 끝
- x{m,n}: m,n 구간 반복

■ 연산자

- [범위]/p: 지정 범위 출력
- [범위]/d: 지정 범위 삭제
- [범위]/문자1/문자2: 지정된 범위의 처음 나타난 문자를 문자2로
- s/문자1/문자2/: 문자1을 2로
- g: 모든 라인에 적용

■ 예제

*특정라인 출력

sed -n '3,7p' <파일> # <파일>에서 3~7라인만 출력

*특정라인 제외 출력

sed '3,7d' <파일> # 3~7라인 제외하고 출력

*특정라인(비연속) 출력

sed -n -e '3,5p' -e '7,9p' <파일> # 3~5, 3~7라인만 출력

*치환하여 출력

sed 's/1/2/g' <파일> # 1을 2로 바꿔서 출력

sed 's/1/2/gi' <파일> # 1을 2로 바꿔서 출력(대소문자 무시)

*특정라인에서 치환

sed '2,5 s/4/7/g' <파일> # <파일>의 2~5줄내에서 4를 7로 치환 출력

*정규식(예제1)

sed '/^#W|^\$W|^*#/d' <파일> # "#"로 시작하거나 or 빈 줄인 라인 제거

*정규식(예제2)

sed 's/[Zz]ip/rar/g' <파일> # zip 또는 Zip을 rar로 바꾸기

*패턴 출력

sed -n '/^May 22/ p' <파일> # "May 22" 로 시작하는 라인 출력

*공백 라인 넣기

sed G <파일> # 라인마다 공백라인 1줄 추가

sed 'G;G' <파일> # 라인마다 공백라인 2줄 추가

*에물레이팅 dos2unix

sed -i 's/\r//' <파일> # hidden new line 제거

*In-place 편집 및 백업 (-i 옵션)

sed -i'.org' 's/this/that/gi' <파일> # 원본 수정 후 .org 확장자 파일 생성하여 백업

*쌍단어 스위칭

sed 's/^\(.*\),.*\$/\2\1,g' <파일> # 특정 구분자(comma)를 기준으로 앞뒤 바꿈

*특정 조건내 변환

sed '/services/ s/start/stop/g' <파일> # services를 갖는 줄에서 start를 stop으로 global하게 변환

*두개 이상 동시 변환

sed -i 's/that/this/gi;s/line/verse/gi' <파일> # that을 this로 / line을 verse로 동시에 바꿈

*다른 명령어와 동시 사용

ip route show | sed -n '/src/p' | sed -e 's/ */ /g' | cut -d' ' -f9

"|"를 이용하여 최종적으로 ip 추출("src"를 갖는 줄에서 연속된 공백을 하나로 치환하고 9번째 컬럼 값인 ip를 출력)

```
[orcl:~]$ sed 's/KING/yyy' emp.txt
sed: -e expression #1, char 10: unterminated `s' command
[orcl:~]$ sed 's/KING/yyy/' emp.txt
7369 SMITH      CLERK          7902 17-DEC-80      800          0          20
7499 ALLEN      SALESMAN       7698 20-FEB-81     1600         300         30
7521 WARD       SALESMAN       7698 22-FEB-81     1250         500         30
7566 JONES      MANAGER        7839 02-APR-81     2975          0          20
7654 MARTIN     SALESMAN       7698 28-SEP-81     1250        1400         30
7698 BLAKE      MANAGER        7839 01-MAY-81     2850          0          30
7782 CLARK      MANAGER        7839 09-JUN-81     2450          0          10
7788 SCOTT      ANALYST        7566 19-APR-87     3000          0          20
7839 yyy        PRESIDENT      0 17-NOV-81      5000         0          10
7844 TURNER     SALESMAN       7698 08-SEP-81     1500          0          30
7876 ADAMS      CLERK          7788 23-MAY-87     1100          0          20
7900 JAMES      CLERK          7698 03-DEC-81      950          0          30
7902 FORD       ANALYST        7566 03-DEC-81     3000          0          20
7934 MILLER     CLERK          7782 23-JAN-82     1300          0          10
```

출력화면에서만 KING이 yyy로 치환된 것이지 원본 파일은 그대로

문제 68. 위의 변경된 내용대로 출력한 내용을 emp9.txt로 저장 하시오.

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

22. cp 명령어

" 파일을 복사하는 명령어 "

Ex. \$ cp 위치/원본 파일명 위치/복사할 파일명

\$ cp emp.txt emp400.txt

\$ cat emp400.txt

옵션	Long 옵션	설명
-a	--archive	-dR --preserve=all 옵션과 같음
-b	--backup	복사 대상 파일이 이미 존재할 경우를 대비하여 백업 파일을 만들
-d	--no-dereference	원본 파일이 소프트링크 파일이면 소프트링크 원본을 삭제
-f	--force	복사 대상이 이미 존재하면 복사 대상 파일을 강제로 지우고 복사

-i	--interactive	복사 대상이 이미 존재하면 사용자에게 덮어씌울건지 여부를 확인
-l	--link	심볼릭 링크 형식으로 복사
-P	--parents	원본 파일명에 디렉토리 경로도 같이 입력했다면 그 경로 그대로 복사
-p	--preserve	원본 파일의 소유주, 그룹, 권한, 시간 정보들을 그대로 보존하여 복사
-s	--symbolic-link	디렉터리가 아닌 일반 파일을 심볼릭 링크 형식으로 복사
-u	--update	변경일이 같거나 더 최신의 복사 대상 파일이 있으면 복사 안 함
-v	--verbose	복사 상태를 자세하게 출력
-x	--one-file-system	원본 파일과 대상 파일의 파일시스템이 서로 다를 경우, 복사 안 함
-R, -r	--recursive	디렉토리를 복사할 경우 하위 디렉토리와 파일을 모두 복사

문제 69. 리눅스의 oracle 계정을 위한 환경설정 파일인 .bash_profile을 백업 하시오.

```
[orcl:~]$ cp .bash_profile .copy_bash_profile
[orcl:~]$ cp .bash_profile .copy_bash_profile
[orcl:~]$ ls
Adobe      b.txt      diff.sh    emp9.txt   ls.txt     test5      year.sh
Desktop    c.txt      e.txt      ename2.sh  oradiag_oracle test64      year_sum.sh
R-3.2.3    d.txt      emp.txt    find_file.sh sar_0622.txt test66
R-3.2.3.tar.gz datamodeler.log emp40.txt  job2.sh    size.sh    txttar.tar
a.txt      dept.txt   emp50.txt  job_sumsal.sh test100    winter.txt
all.tar    deptno2.sh emp7.txt   labs        test200    winter_head.txt
```

문제 70. /home/oracle 밑에 있는 확장자 .txt 파일들 전부를 /home/oracle/backup 이라는 폴더를 만들고 거기에 복사 하시오.

```
[orcl:~]$ mkdir backup
[orcl:~]$ cp /home/oracle/*.txt /home/oracle/backup/
[orcl:~]$ cd backup/
[orcl:backup]$ ls
a.txt  c.txt  dept.txt  emp.txt  emp50.txt  emp9.txt  sar_0622.txt  winter_head.txt
b.txt  d.txt  e.txt     emp40.txt  emp7.txt  ls.txt    winter.txt
```

23. mv 명령어

" 파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어 "

옵션	Long 옵션	설명
	--backup[= CONTROL]	대상 파일이 존재하면 백업 파일을 만들
-b		--backup과 비슷하게 백업 파일을 생성
-f	--force	사용자에게 묻지 않고 파일을 강제로 덮어쓰
-i	--interactive	존재하는 파일을 덮어 쓸 경우 확인
-n	--no-clobber	존재하는 파일을 덮어쓰지 않음
-S	--suffix= SUFFIX	지정한 접미사로 백업을 생성
-t	--target-directory= DIRECTORY	전체 원본 파일을 대상 디렉토리로 이동
-T	--no-target-directory	파일을 대상 디렉토리로 취급하지 않고 원본 파일로 취급
-u	--update	파일이 업데이트된 경우에만 이동
-v	--verbose	진행 상황에 대한 상세한 정보를 출력

Ex. \$ mv 기존파일명 새로운파일명
\$ mv emp400.txt emp500.txt
\$ mv emp.txt ./backup/

문제 71.	다시 /home/oracle/backup 밑에 있는 emp.txt를 지금 현재 디렉토리인 /home/oracle 밑으로 이동 시키시오.
--------	---

```
[orcl:~]$ mv ./backup/emp.txt .  
[orcl:~]$ ls  
Adobe          b.txt          deptno2.sh     emp7.txt       labs           test200        winter_head.txt  
Desktop        backup         diff.sh        emp9.txt       ls.txt         test5          year.sh  
R-3.2.3        c.txt         e.txt         ename2.sh      oradiag_oracle test64         year_sum.sh  
R-3.2.3.tar.gz d.txt        emp.txt       find_file.sh   sar_0622.txt  test66  
a.txt          datamodeler.log emp40.txt      job2.sh        size.sh       txttar.tar  
all.tar        dept.txt      emp50.txt     job_sumsal.sh test100        winter.txt
```

4. vi 편집기

2018년 6월 25일 월요일 오전 10:02

■ vi 편집기 명령어

- Vi 편집기란 ? : 리눅스 안에서 사용할 수 있는 문서 편집기

Vi (visual editor)의 뜻에서 유래됨

■ vi 편집기의 명령모드 3가지

- Command 모드 : vi의 기본 모드이며 vi를 실행하면 바로 보이는 기본 화면 (방향키로 왔다갔다 할 수 있는 화면) (esc)
- Edit 모드 : a, i, o 등을 누르면서 내용을 입력 또는 삭제하는 명령모드 (a, i, o)
- Last line 모드 : 입력 모드에서 저장 종료, 강제 종료 등의 명령어를 입력하는 모드 (:))

Ex. \$ vi aaa.txt

```
when i was young, our family was poor.  
My mother said she doesn't like chineses noodles.  
Yah-ee yah-ee ya~
```

이상태로 ctrl + z를 눌러서 저장하지 않고 나오면 다음에 실행 시

```
E325: ATTENTION  
Found a swap file by the name ".aaa.txt.swp"  
    owned by: oracle   dated: Mon Jun 25 16:11:09 2018  
    file name: ~oracle/aaa.txt  
    modified: YES  
    user name: oracle   host name: edydrip0.us.oracle.com  
    process ID: 17689 (still running)  
While opening file "aaa.txt"  
  
(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:
```

이런 화면이 뜬다.

```
[orcl:~]$ ls -al .aaa.txt.swp  
-rw----- 1 oracle oinstall 12288 Jun 25 16:11 .aaa.txt.swp
```

임시 저장 파일이 저장하기 때문 !

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

임시 저장 파일을 지우면 위의 화면이 나오지 않는다.

■ 편집기를 시작하는 명령어

- a. \$ vi 파일명 : 읽기, 쓰기가 가능한 상태로 파일이 열린다.
- b. \$view 파일명 : 읽기만 가능한 상태로 파일이 열린다.
- c. \$vi -r 파일명 : 손상된 파일을 복구하는 명령어

■ vi 편집기 내에서 커서 이동

- a. j : 아래로 이동
- b. k : 위로 이동
- c. h : 왼쪽 이동
- d. l : 오른쪽 이동
- e. G : 가장 아래로 이동 , 123G --> 123번째 줄로 이동
- f. 1G/gg : 맨 위로 이동
- g. :set nu : 파일내의 텍스트에 번호 표시
- h. :set nonu : 번호 안보이게 한다.

■ vi 편집기의 입력 명령어

- a. i : 커서의 왼쪽부터 입력하겠다.
- b. a : 커서의 오른쪽부터 입력하겠다.
- c. o : 커서의 다음라인부터 입력하겠다.
- d. O : 커서의 위 라인부터 입력하겠다.

■ vi 편집기의 삭제 명령어

- a. x : 철자 하나 삭제
- b. dd : 한 행 삭제
- c. dw : 커서에 있는 단어를 삭제
- d. :5,10 d : 5~10행 삭제
- e. D : 커서 오른쪽 행 삭제

■ 취소 명령어

u : 방금 작업 한거 취소 하겠다.

■ 복사//붙여넣기 명령어

- a. yy : 하나의 행을 복사
- b. P : 붙여 넣기
- c. yG : 현재 행 부터 파일 끝까지 복사
- d. :1,2 co 3 : 1~2행을 3행 다음으로 복사하겠다.
- e. :1,2 m 3 : 1~2행을 3행 다음으로 이동하겠다.

■ 문자 변경하는 명령어

- a. ./문자 : 문자 검색 (n을 누르면 다음 문자, shift+n을 누르면 이전 문자)
b. :s/기존문자/변경할 문자 : 지금 커서가 있는 현재행의 기존문자를 변경할 문자로 변경하겠다.

```

YOUNG ANNA ← 커서위치
Elsa. Psst. Elsa
Elsa doesn't stir. Anna sits on Elsa and bounces.
YOUNG ANNA (CONT'D)
Wake up. Wake up. Wake up.
YOUNG ELSA
(grumbling)
Anna, go back to sleep.
Anna rolls onto her back and spreads all her weight on Elsa.
FROZEN - J. Lee 2
YOUNG ANNA
(drama queen-ish)
I just can't. The sky's awake, so
I'm awake, so we have to play.
YOUNG ELSA
...Go play by yourself.
Elsa shoves Anna off the bed.
Anna lands butt to floor, sighs, defeated. But then she gets
an idea. She hops back on the bed and lifts one of Elsa's
eyelids.
YOUNG ANNA
W10: Warning: Changing a readonly file
:s/ANNA/BAEK

```

- c. :%s/기존문자/변경할 문자/g : 문서 전체의 기존문자를 변경할 문자로 변경하겠다.

■ 숫자열 비교

숫자1 -eq 숫자2	두 숫자가 같으면 true
숫자1 -ne 숫자2	두 숫자가 같지 않으면 true
숫자1 -gt 숫자2	숫자 1이 숫자 2보다 크면 true
숫자1 -ge 숫자2	숫자 1이 숫자 2보다 크거나 같으면 true
숫자1 -lt 숫자2	숫자 1이 숫자 2보다 작으면 true
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면 true
!숫자1	숫자 1이 거짓이라면 true

문제 72. 겨울왕국 대본의 1~200 번째까지의 내용을 winter200.txt라는 이름으로 저장 하시오.

```

STRONGER THAN ONE, STRONGER THAN TEN
STRONGER THAN A HUNDRED MEN!
Massive fjord horses drag heavy ice plows.
FROZEN - J. Lee 1
ICE HARVESTERS (CONT'D)
BORN OF COLD AND WINTER AIR
AND MOUNTAIN RAIN COMBINING
:1,200 w winter200.txt

```

```

[orcl:~]$ ls
Adobe      b.txt      deptno2.sh emp7.txt    labs        test200    winter200.txt
Desktop    backup     diff.sh     emp9.txt    ls.txt      test5      winter500.txt
R-3.2.3    c.txt      e.txt       ename2.sh   oradiag_oracle test64     winter_head.txt
R-3.2.3.tar.gz d.txt     emp.txt     find_file.sh sar_0622.txt test66     year.sh
a.txt      datamodeler.log emp40.txt   job2.sh     size.sh     txttar.tar year_sum.sh
all.tar     dept.txt   emp50.txt   job_sumsal.sh test100      winter.txt
[orcl:~]$

```

문제 73. 겨울 왕국 대본의 전체 라인수를 확인하고, 대본 전체를 복사해서 아래에 붙여 넣고 겨울왕국의 라인수가 2배가 되었는지 확인 하시오.

GG 명령으로 마지막 행 확인

```
repaired with a ice.
All is right in Arendelle.
FINAL FADE OUT.
THE END
FROZEN - J. Lee 112"
```

█

4240,1

Bot

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

```
:1,4240 co 4240 █
```

```
glide and pivot.
We pull away slowly, into the sky. We arrive at a bird' s-eye
view to see that where the castle had crumbled has been
repaired with a ice.
All is right in Arendelle.
FINAL FADE OUT.
THE END
FROZEN - J. Lee 112"
```

█

4240 more lines

8480,1

문제 74. emp.txt를 emp750.txt로 복사하고 emp750.txt를 열어서 SALESMAN을 전부다 CLERK로 변경 하시오.

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

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

```
Adobe      b.txt      deptno2.sh  emp7.txt    job_sumsal.sh  test100    winter.txt
Desktop     backup     diff.sh     emp750.txt  labs           test200    winter200.txt
R-3.2.3     c.txt      e.txt       emp9.txt    ls.txt         test5      winter500.txt
R-3.2.3.tar.gz d.txt     emp.txt     ename2.sh   oradiag_oracle test64     winter_head.txt
a.txt       datamodeler.log emp40.txt   find_file.sh sar_0622.txt  test66     year.sh
all.tar     dept.txt   emp50.txt   job2.sh     size.sh        txttar.tar year_sum.sh
```

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

```
:%s/SALESMAN/CLERK/g █
```

```
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      CLERK      7698 20-FEB-81      1600     300     30
7521 WARD        CLERK      7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER    7839 02-APR-81      2975      0      20
7654 MARTIN     CLERK      7698 28-SEP-81      1250    1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81      2850      0      30
7782 CLARK       MANAGER    7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST    7566 19-APR-87      3000      0      20
7839 KING       PRESIDENT      0 17-NOV-81      5000      0      10
7844 TURNER     CLERK      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
```

문제 74.+ Emp.txt를 emp1.txt ~ emp20.txt로 복사 하시오.


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

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

while(공백)[(공백) ~~~ (공백)] <--- 공백 주의하자.

```
[orcl:~]$ vi cp_emp.sh
[orcl:~]$ sh cp_emp.sh
[orcl:~]$ ls -l emp*.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 25 19:49 emp.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp1.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp10.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp11.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp12.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp13.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp14.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp15.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp16.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp17.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp18.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp19.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp2.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp20.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp3.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp4.txt
-rw-r--r-- 1 oracle oinstall 1940 Jun 22 11:34 emp40.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp5.txt
-rw-r--r-- 1 oracle oinstall 3880 Jun 22 11:43 emp50.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp6.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp7.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 09:59 emp750.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp8.txt
-rw-r--r-- 1 oracle oinstall 970 Jun 26 10:04 emp9.txt
```

문제 75. Emp1.txt ~ emp20.txt의 내용을 변경하는데 ANALYST를 jjj로 변경하시오.

```
[orcl:~]$ vi emp*.txt
24 files to edit
```

```

"emp.txt" 14L, 962C written
"emp1.txt" 14L, 962C written
"emp10.txt" 14L, 962C written
"emp11.txt" 14L, 962C written
"emp12.txt" 14L, 962C written
"emp13.txt" 14L, 962C written
"emp14.txt" 14L, 962C written
"emp15.txt" 14L, 962C written
"emp16.txt" 14L, 962C written
"emp17.txt" 14L, 962C written
"emp18.txt" 14L, 962C written
"emp19.txt" 14L, 962C written
"emp2.txt" 14L, 962C written
"emp20.txt" 14L, 962C written
"emp3.txt" 14L, 962C written
"emp4.txt" 14L, 962C written
"emp40.txt" 28 lines, 1940 characters
"emp40.txt" 28L, 1924C written
"emp5.txt" 14L, 962C written
"emp50.txt" 56 lines, 3880 characters
"emp50.txt" 56L, 3848C written
"emp6.txt" 14L, 962C written
"emp7.txt" 14L, 962C written
"emp750.txt" 14L, 962C written
"emp8.txt" 14L, 962C written
"emp9.txt" 14L, 962C written
:argdo %s/ANALYST/jjj/g | update

```

저장하고 vi 빠져나오는 방법 : ZZ

문제 76.	Winter.txt 대본을 winter1.txt ~ winter100.txt 로 100개 복사하고 복사한 내용 중에 by를 bybyby로 변경 하시오,.
--------	---

```

#!/bin/bash

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

[orcl:~]$ vi winter_cp.sh
[orcl:~]$ sh winter_cp.sh
[orcl:~]$ vi winter_cp.sh
[orcl:~]$ ls -l winter*.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 22 11:47 winter.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter1.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter10.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter100.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter11.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter12.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter13.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter14.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter15.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter16.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter17.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter18.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter19.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter2.txt
-rw-r--r-- 1 oracle oinstall 114548 Jun 26 10:19 winter20.txt
-rw-r--r-- 1 oracle oinstall 6524 Jun 25 16:31 winter200.txt
...

[orcl:~]$ vi winter*.txt
104 files to edit

:argdo %s/by/bybyby/g | update

```

문제 77. winter1.txt ~ winter100.txt를 전부 삭제 하시오.

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

#!/bin/bash

i=1

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

5. 권한 관리 (chmod, chown, chattr, lsattr)

2018년 6월 26일 화요일 오전 10:33

리눅스에서 파이썬 배치파일을 실행하거나 R 배치 파일을 실행하거나 하둡을 설치하고 운영을 할 때 여러가지 문제들이 발생하는데 그 중에 가장 많은 문제들이 권한에 관련한 오류들이 많다. 그래서 권한 관리를 잘 알아둬야 한다.

■ 권한 관리에 대한 명령어 3가지

1. chmod : 파일이나 디렉토리의 권한을 변경하는 명령어 (허가권)
ex. chmod g+w test.txt # 그룹에 쓰기 권한을 추가
2. chown : 파일이나 디렉토리의 소유자를 변경하는 명령어 (소유권)
ex. chown 소유자명:소유그룹명 /home/oracle/test.txt
3. chattr : 파일이나 디렉토리의 속성을 설정 또는 제거하는 명령어
4. lsattr : 파일이나 디렉토리의 속성을 확인할 수 있는 명령어 ex) lsattr 파일명

• ls -l 로 특정 파일을 조회했을 때 나오는 권한 부분을 해석

```
[orcl:~]$ ls -l emp.txt  
-rw-r--r-- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

drwxr-xr-x 2 root root 4096 Apr 22 16:59 conory

파일Type 퍼미션정보 링크수 소유자 소유그룹 용량 생성날짜 파일이름

파일 Type : "d" -> 디렉토리, "l" -> 링크파일, "-" -> 일반파일 등등..

퍼미션정보 : 해당 파일에 어떠한 퍼미션이 부여되어있는 지 표시!

링크수 : 해당 파일이 링크된 수! 링크는 윈도우의 "바로가기"와 같습니다. "in [대상파일] [링크파일]" 명령으로 링크파일을 만듭니다.

▼ chmod

사용법 : chmod 권한설정값 파일or디렉토리

참고사항

- | | |
|--------------|-----------|
| 1. u : 유저 | 1. r : 읽기 |
| 2. g : 그룹 | 2. w : 쓰기 |
| 3. o : 기타 유저 | 3. x : 실행 |

• 권한 관리표

번호	권한	대표문자	파일일 경우	디렉토리일 경우
1	읽기권한	r	읽고,copy	디렉토리에서 ls 실행가능
2	쓰기권한	w	수정	디렉토리에서 파일을 생성 가능
3	실행권한	x	실행	디렉토리에 cd로 접근 가능

▼ chown

사용법 : chown 소유자명 : 소유그룹명 파일or디렉토리

예제:

```
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 970 Jun 26 10:11 emp.txt

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

[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 root root 970 Jun 26 10:11 emp.txt
```

▼ chattr

사용법 : chattr [옵션] [+,-,= 속성] 파일명

옵션

-R : 현재 디렉토리 이하의 모든 디렉토리와 파일의 속성을 변환
-V : 변환된 속성의 자세한 정보를 출력한다.
-v : 버전 정보를 출력한다.

설정모드

+ : 속성을 추가한다
- : 속성을 제거한다
= : 원래 파일이 가지고 있던 그 속성만을 유지하게 한다.

속성

a : 파일을 추가 모드로만 열 수 있다.
c : 커널에 의해 디스크 상에 압축 상태로 저장한다.
d : dump 명령 수행 시 백업하지 않는다.
i : 파일의 수정을 방지한다. 오직 슈퍼 유저만이 다시 이 속성을 변경 할 수 있다.
s : 파일이 지워질 때 일단 블록들이 모두 0이 된 다음 디스크에 기록한다.
S : 파일이 수정될 때 그 변화가 디스크 상에 동기화한다.
u : 파일이 지워지면 내용을 저장한다.

문제 78. emp.txt의 소유자의 권한을 읽기만 가능하도록 수정 하시오.

```
[orcl:~]$ ls -l emp.txt
-rw-r--r-- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

참고사항

- | | |
|--------------|-----------|
| 1. u : 유저 | 1. r : 읽기 |
| 2. g : 그룹 | 2. w : 쓰기 |
| 3. o : 기타 유저 | 3. x : 실행 |

-rw- r-- r--

\$ chmod u-w emp.txt

\$ ls -l emp.txt

\$ vi emp.txt

설명 : read only 파일도 !(느낌표)를 붙이고 저장하면 강제 종료되어서 저장된다.

문제 79. emp.txt의 소유자의 권한에 읽기권한도 없애시오.

```
$ chmod u-r emp.txt
```

```
[orcl:~]$ chmod u-r emp.txt
[orcl:~]$ ls -l emp.txt
--w-r--r-- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 80. Emp.txt의 소유자의 권한에 읽기,쓰기,실행권한을 넣으시오.

```
[orcl:~]$ chmod u+rwX emp.txt
[orcl:~]$ ls -l emp.txt
-rwxr-xr-x 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 81. Emp.txt의 소유자의 권한에 읽기,쓰기,실행권한을 모두 넣으시오.

```
[orcl:~]$ chmod 777 emp.txt
[orcl:~]$ ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 82. Oracle 유저는 emp.txt에 대해서 무조건 읽기만 할 수 있게 하고 쓰지는 못하게 하며 그룹과 다른 기타 유저에 대해서는 읽기,쓰기, 실행 다 못하게 하시오.

```
[orcl:~]$ chmod 400 emp.txt
[orcl:~]$ ls -l emp.txt
-r----- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

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

```
[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 10:09 emp.txt
[root@edydr1p0 oracle]# chattr +i emp.txt
[root@edydr1p0 oracle]# ls -l emp.txt
-r----- 1 oracle oinstall 962 Jun 26 10:09 emp.txt

[root@edydr1p0 oracle]# su - oracle
이랏샤이 마세이 ~~
[orcl:~]$ ls -l emp.txt
-r----- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
[orcl:~]$ chmod u+rwX emp.txt
chmod: changing permissions of `emp.txt': Operation not permitted
```

다시 oracle 유저가 수정할 수 있도록 하는

```
[root@edydr1p0 ~]# cd /home/oracle
[root@edydr1p0 oracle]# chatter -i emp.txt
[root@edydr1p0 oracle]# su - oracle
이랏샤이 마세이 ~~
[orcl:~]$ chmod u+rwX emp.txt
[orcl:~]$ ls -l emp.txt
-rwx----- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 84.	emp.txt에 대해서 아래와 같이 권한 설정이 될 수 있도록 하시오. \$ ls -l emp.txt -r--r--r--
---------------	---

```
[orcl:~]$ chmod 444 emp.txt
[orcl:~]$ ls -l emp.txt
-r--r--r-- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 85.	겨울왕국 대본의 권한을 rwx r-- r-- 로 변경 하시오.
---------------	------------------------------------

```
[orcl:~]$ chmod 744 winter.txt
[orcl:~]$ ls -l winter.txt
-rwxr--r-- 1 oracle oinstall 114730 Jun 26 10:28 winter.txt
```

- 점심시간 문제를 위한 명령어

```
[orcl:~]$ grep -w -o -i 'off' winter.txt | wc -l
65
```

문제 86.	겨울왕국에 elsa란 단어가 많이 나오는지 anna란 단어가 많이 나오는지 확인하시오.
---------------	--

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

문제 87.	R에서 겨울왕국에 elsa란 단어가 많이 나오는지 anna란 단어가 많이 나오는지 확인하시오.
---------------	--

```
> length(grep('elsa',winter,ignore.case=T))
[1] 318
> length(grep('anna',winter,ignore.case=T))
[1] 677
```

문제 88.	그럼 다시 emp.txt의 소유자를 oracle로 변경하고 그룹을 oinstall로 변경 하시오.
---------------	--

```
[orcl:~]$ su -
Password:
[root@edydr1p0 ~]# cd /home/oracle
[root@edydr1p0 oracle]# chown oracle:oinstall emp.txt
[root@edydr1p0 oracle]# ls -l emp.txt
-rwxrwxrwx 1 oracle oinstall 970 Jun 26 10:11 emp.txt
```

문제 89.	/home/oracle 밑에 bbb라는 디렉토리를 만들고 bbb라는 디렉토리 밑에 /home/oracle 밑에 있는 확장자 txt 파일들 을 전부 copy 하시오.
---------------	--

```
[root@edydr1p0 ~]# cd /home/oracle
[root@edydr1p0 oracle]# mkdir bbb
[root@edydr1p0 oracle]# ls
Adobe          cp_emp.sh      emp10.txt      emp2.txt      emp8.txt      size.sh        winter_cp.sh
Desktop        d.txt          emp11.txt      emp20.txt     emp9.txt      test100        winter_head.txt
R-3.2.3        datamodeler.log emp12.txt      emp3.txt      ename2.sh     test200        year.sh
R-3.2.3.tar.gz del_winter.sh  emp13.txt      emp4.txt      find_file.sh  test5          year_sum.sh
a.txt          dept.txt       emp14.txt      emp40.txt     job2.sh       test64
all.tar         deptno2.sh     emp15.txt      emp5.txt      job_sumsal.sh test66
b.txt          diff.sh        emp16.txt      emp50.txt     labs          txttar.tar
backup         e.txt          emp17.txt      emp6.txt      ls.txt        winter.txt
bbb            emp.txt        emp18.txt      emp7.txt      oradiag_oracle winter200.txt
c.txt          emp1.txt       emp19.txt      emp750.txt    sar_0622.txt  winter500.txt

[root@edydr1p0 oracle]# cp *.txt /home/oracle/bbb
[root@edydr1p0 oracle]# cd bbb
[root@edydr1p0 bbb]# ls
a.txt  dept.txt  emp10.txt  emp14.txt  emp18.txt  emp3.txt  emp50.txt  emp8.txt  winter.txt
b.txt  e.txt     emp11.txt  emp15.txt  emp19.txt  emp4.txt  emp6.txt  emp9.txt  winter200.txt
c.txt  emp.txt   emp12.txt  emp16.txt  emp2.txt   emp40.txt emp7.txt   ls.txt    winter500.txt
d.txt  emp1.txt  emp13.txt  emp17.txt  emp20.txt  emp5.txt  emp750.txt sar_0622.txt winter head.txt
```

문제 90. /home/oracle 밑에 bbb 디렉토리의 소유자를 oracle로 바꾸시오.

```
[root@edydr1p0 oracle]# ls -ld bbb
drwxr-xr-x 2 root root 4096 Jun 26 14:09 bbb

[root@edydr1p0 oracle]# chown -R oracle:oracle /home/oracle/bbb
[root@edydr1p0 oracle]# ls -ld bbb
drwxr-xr-x 2 oracle oracle 4096 Jun 26 14:09 bbb

[root@edydr1p0 bbb]# ls -l *.txt
-rw-r--r-- 1 oracle oracle 0 Jun 26 14:09 a.txt
-rw-r--r-- 1 oracle oracle 0 Jun 26 14:09 b.txt
-rw-r--r-- 1 oracle oracle 0 Jun 26 14:09 c.txt
-rw-r--r-- 1 oracle oracle 0 Jun 26 14:09 d.txt
-rw-r--r-- 1 oracle oracle 105 Jun 26 14:09 dept.txt
-rw-r--r-- 1 oracle oracle 0 Jun 26 14:09 e.txt
-r--r--r-- 1 oracle oracle 962 Jun 26 14:09 emp.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp1.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp10.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp11.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp12.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp13.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp14.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp15.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp16.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp17.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp18.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp19.txt
-rw-r--r-- 1 oracle oracle 962 Jun 26 14:09 emp2.txt
```

-R 옵션을 써줬기 때문에 디렉토리, 디렉토리 안의 파일까지 모두 소유자가 바꼈다.

문제 91. 다시 /home/oracle 밑에 bbb 디렉토리의 소유자를 root로 변경하고 그룹도 oinstall로 바꾸시오.

```
[root@edydr1p0 oracle]# chown -R root:oinstall /home/oracle/bbb
[root@edydr1p0 oracle]# ls -ld bbb
drwxr-xr-x 2 root oinstall 4096 Jun 26 14:09 bbb
[root@edydr1p0 oracle]# whoami
root
```


6. 디스크 관리(du, df, sar)

2018년 6월 26일 화요일 오후 2:20

■ 디스크 관리 목록

■ du 명령어

" 현재 파일의 용량을 확인하는 명령어 "

ex. \$ du -sh

```
[orcl:~]$ du -sh
638M  .
[orcl:~]$ du -s
653308  .
```

```
[orcl:~]$ du -s *.txt
0      a.txt
0      b.txt
0      c.txt
0      d.txt
4      dept.txt
0      e.txt
4      emp.txt
4      emp1.txt
4      emp10.txt
4      emp11.txt
4      emp12.txt
4      emp13.txt
4      emp14.txt
4      emp15.txt
4      emp16.txt
4      emp17.txt
4      emp18.txt
4      emp19.txt
4      emp2.txt
4      emp20.txt
4      emp3.txt
4      emp4.txt
4      emp40.txt
4      emp5.txt
4      emp50.txt
4      emp6.txt
4      emp7.txt
4      emp750.txt
4      emp8.txt
4      emp9.txt
4      ls.txt
12     sar_0622.txt
120    winter.txt
8      winter200.txt
8      winter500.txt
4      winter_head.txt
```

■ df 명령어

" 현재 파일 시스템의 총 사용율을 확인하는 명령어 "

```
[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

" disk의 i/o 성능을 모니터링 하는 명령어 "

ex. \$ sar 1 50

```
[orcl:~]$ sar 1 1
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/26/18

14:39:16      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:39:17      all       0.50       0.00       4.00       2.00       0.00      93.50
Average:      all       0.50       0.00       4.00       2.00       0.00      93.50
```

* 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 있는 idel(게으른) 한 상태
%steal	?

문제 92. 위의 txt 파일의 총 크기를 출력 하시오.

```
[orcl:~]$ du -c *.txt
0      a.txt
0      b.txt
0      c.txt
0      d.txt
:
4      ls.txt
12     sar_0622.txt
120    winter.txt
8      winter200.txt
8      winter500.txt
4      winter_head.txt
256    total
```

또는

```
[orcl:~]$ du -s *.txt | awk '{sum = sum + $1} END {print sum}'
256
```

문제 93. sar의 user i/o를 R로 시각화 하시오.

1. R로 접속한다.
2. 아래의 R 코드의 plot 그래프를 확인 하시오.

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

3. 리눅스 아래의 명령어를 수행하시오.

```
[orcl:~]$ sar 1 1 | awk '{print $3}'
(edydr1p0.us.oracle.com)

%user
0.00
0.00
```

4. r에서 리눅스 명령어를 수행 하시오.

```
> system("sar 1 1 | awk '{print $3}' ",intern=TRUE)
[1] "(edydr1p0.us.oracle.com)" ""
[3] "$user" "0.50"
[5] "0.50"
```

5. 위의 결과를 RE라는 변수에 담고 5번째 요소만 가져오시오.

```
> re<-system("sar 1 1 | awk '{print $3}' ",intern=TRUE)
> re
[1] "(edydr1p0.us.oracle.com)" ""
[3] "$user" "0.00"
[5] "0.00"
> re[5]
[1] "0.00"
```

6. 아래와 같이 NA를 반복하게 하고 m이라는 변수에 담는다.

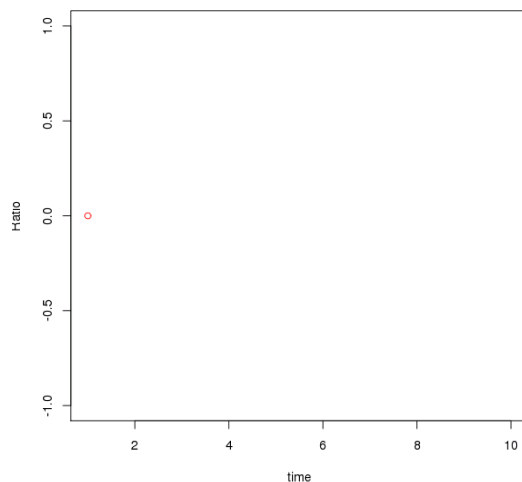
```
> rep(NA,10)
[1] NA NA NA NA NA NA NA NA NA NA
> m<-rep(NA,10)
> m
[1] NA NA NA NA NA NA NA NA NA NA
```

7. re[5]의 값을 m변수에 첫번째 요소로 담으시오.

```
> m[1]<-re[5]
> m
[1] "0.00" NA NA NA NA NA NA NA NA NA
```

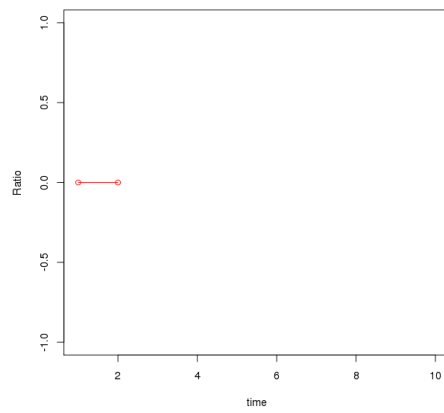
8. 위의 m 데이터를 가지고 plot 그래프를 그리시오.

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



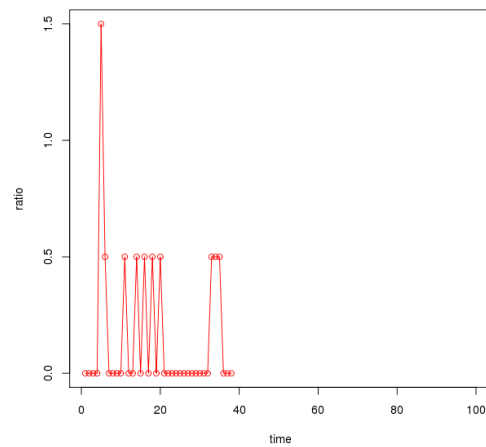
9. 실시간으로 그래프가 이어질 수 있도록 m에 데이터를 담고 plot 그래프를 그리시오.

```
> m[2]<-re[5]
> m
[1] "0.00" "0.00" NA NA NA NA NA NA NA
> plot(m,type='o',col='red',xlab='time',ylab='Ratio')
```



10. 위의 R코드에 for loop문을 사용해서 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 저장해서 평상시에 쉽게 수행할 수 있도록 하시오.

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

```
[orcl:~]$ Rscript 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)

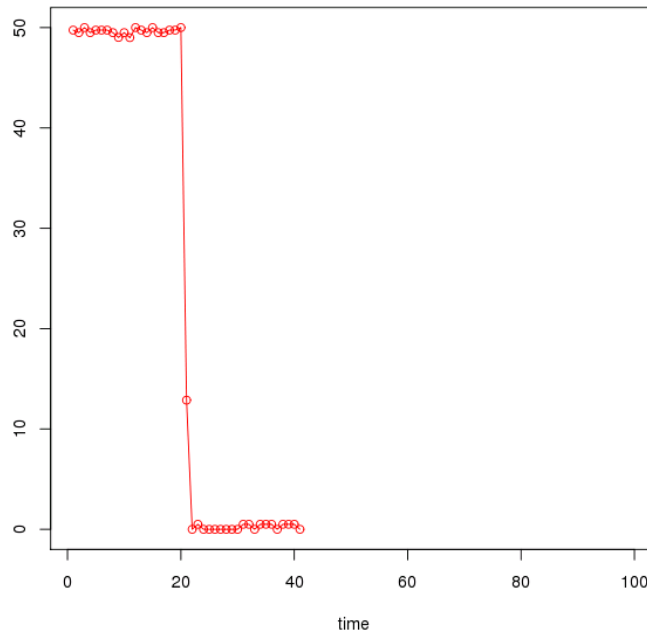
```

악성 쿼리를 날려보면 ..

```

SQL> select count(*)
2  from sh.sales, sh.sales
3  ;

```



실시간으로 부하상태를 볼 수 있다.

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

```

sar <- function(n){
    m<-rep(NA,n)
    emp<-read.csv("emp.csv")
    for (i in 1:n) {
        re <- emp[i,8]
        re <- re[5]
        m[i] <- re
        plot(m, type = 'o', col = 'red' , xlab='time', ylab='ratio')
    }
    rm(m)
}

```

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

1. /home/oracle 경로에 s_stock.csv 파일을 올린다.

2. s_stock 함수를 생성한다.

```
x11()

s_stock <- function(n) {

  m <- rep(NA, n)
  stock <- read.csv("/home/oracle/S_stock.csv")

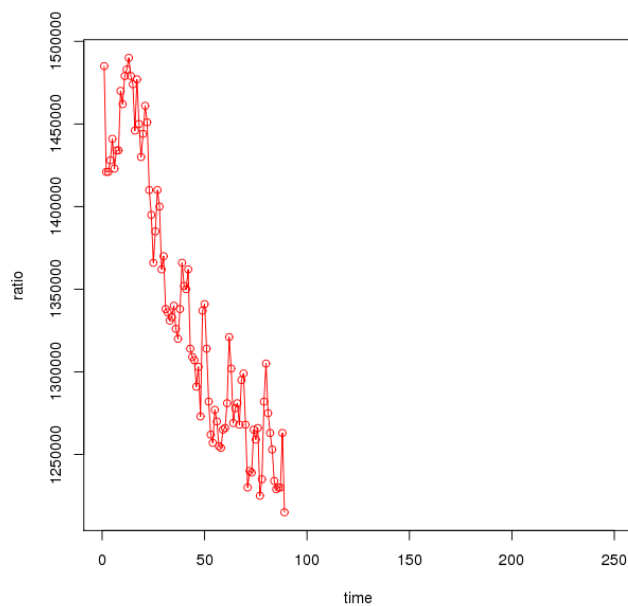
  for ( i in 1:n) {

    re <- stock$s_price[i]
    m[i] <- re
    Sys.sleep(0.1)
    plot(m, type='o', col='red', xlab='time', ylab='ratio')
  }

  rm(m)
}

s_stock(nrow(read.csv('/home/oracle/S_stock.csv')))
```

3. 저장 후 Rscript s_stock.R 로 실행



문제 97.	현재 주식 등락을 1년치 데이터를 가지고 실시간 plot 그래프를 그려보시오.
--------	---

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

```
x11()

h_stock <- function(n) {

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

```
data <- read.csv("/home/oracle/H_stock.csv")

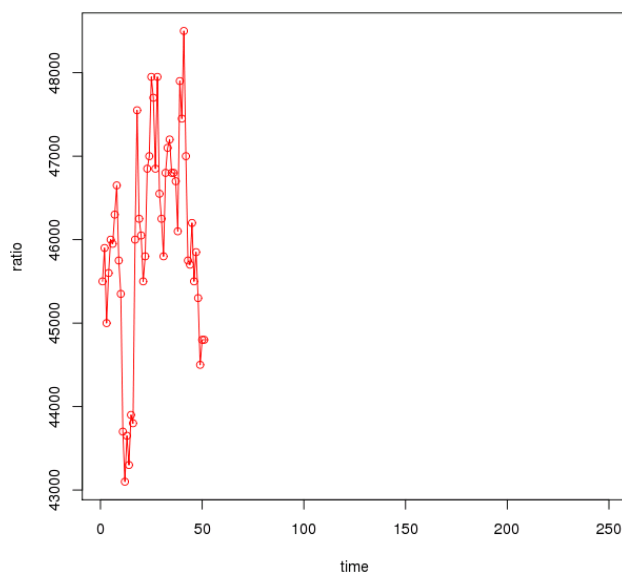
for ( i in 1:n) {

    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('/home/oracle/H_stock.csv')))

[orcl:~]\$ Rscript h_stock.R



문제 98.	R에서 위의 스크립트를 이용해서 함수를 생성하는데 아래와 같이 함수를 실행하면 csv 파일을 물어보게 하고 실시간 그래프가 그려지게 하시오.
---------------	--

```
stock_fun<-function(){
  file<-readline(prompt='파일을 입력하세요')
  data<-read.csv(file)
  n<-nrow(data)
  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)
}
```

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

1. s.sh

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

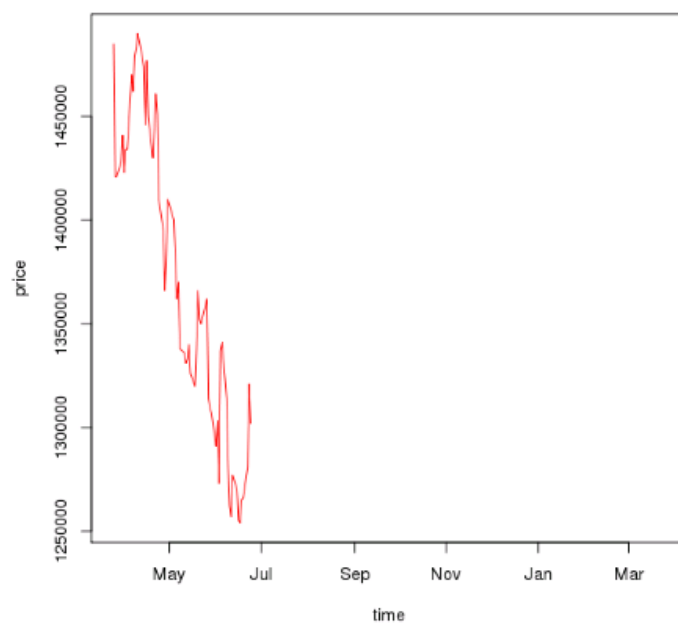
2. s.R

```
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()
```

3. 실행하는 방법

```
[orcl:~]$ sh s.sh
파일을 입력하세요 ~ S_stock.csv
```



7. 프로세서 관리

2018년 6월 26일 화요일 오후 4:32

■ 프로세서 관리

명령어	설명
ps	
job	동작중인 작업의 상태를 확인하는 명령어
top	

■ jobs 명령어

" 동작중인 작업의 상태를 확인하는 명령어 "

-상태정보 4가지

1. running : 실행 중
2. stopped : 일시 중단 중
3. Done : 종료
4. terminated : 강제종료

\$ vi hhh.txt

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

ctrl + z <-- 하던 작업 종료

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

```
[7]+  Stopped                  vim hhh.txt
     ↑ job번호   ↑ 현재상태      ↑ 동작중인 job
```

\$ jobs

\$ fg 하면

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

작업중이던 위치로 이동한다.

■ top 명령어

" 현재 작성중인 프로세서들의 cpu 사용율과 메모리 사용율을 확인하는 명령어 "

```
[orcl:~]$ top
top - 16:47:06 up 1 day, 1:50, 4 users, load average: 0.26, 0.30, 0.26
Tasks: 236 total, 1 running, 223 sleeping, 11 stopped, 1 zombie
Cpu(s): 0.2%us, 1.2%sy, 0.0%ni, 98.3%id, 0.2%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 2075456k total, 2004960k used, 70496k free, 157568k buffers
Swap: 4128760k total, 0k used, 4128760k free, 1370532k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4961	oracle	18	0	400m	23m	11m	S	0.7	1.2	6:54.76	oraagent.bin
9095	oracle	15	0	2336	1120	800	R	0.7	0.1	0:00.03	top
4358	oracle	15	0	335m	29m	18m	S	0.3	1.4	4:45.98	ohasd.bin
4933	root	15	0	33968	14m	6196	S	0.3	0.7	1:17.65	Xorg
5009	oracle	18	0	97880	11m	5664	S	0.3	0.6	1:20.87	diskmon.bin
5104	oracle	15	0	7776	1332	888	S	0.3	0.1	1:29.15	VBoxClient
5185	oracle	15	0	37792	7140	5244	S	0.3	0.3	0:48.99	scim-panel-gtk
5378	oracle	-2	0	409m	11m	10m	S	0.3	0.6	5:18.75	oracle
9828	oracle	-2	0	612m	13m	11m	S	0.3	0.6	4:23.69	oracle
1	root	15	0	2072	616	532	S	0.0	0.0	0:01.29	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.10	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.02	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	0:00.12	migration/1
6	root	38	19	0	0	0	S	0.0	0.0	0:00.22	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	0:11.95	events/0
9	root	10	-5	0	0	0	S	0.0	0.0	0:19.08	events/1
10	root	10	-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	0:00.50	kblockd/0
16	root	10	-5	0	0	0	S	0.0	0.0	0:01.46	kblockd/1
17	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
57	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/0
58	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/1
61	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
63	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	pdflush
131	root	15	0	0	0	0	S	0.0	0.0	0:03.43	pdflush
132	root	10	-5	0	0	0	S	0.0	0.0	0:00.03	kswapd0
133	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
134	root	14	-5	0	0	0	S	0.0	0.0	0:00.00	aio/1
283	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	knfsd

문제 100. jobs 명령어를 수행하면 아래와 같이 결과가 출력되게 하시오.

```
$ jobs
```

```
[1] stopped hhh.txt
[2] stopped hhh2.txt
[3] stopped hhh3.txt
```

```
[orcl:~]$ jobs
[1] Stopped R
[2] Stopped R
[3] Stopped Rscript h_stock.R
[4] Stopped Rscript s_stock.R
[5] Stopped vim stock.R
[6] Stopped R
[7] Stopped vim hhh.txt
[8] Stopped vim hhh.txt
[9]- Stopped vim hhh2.txt
[10]+ Stopped vim hhh3.txt
```

문제 101. 위의 일시중단된 세가지 job 중에서 2번 job으로 fg하려면 어떻게 해야 하는가?

```
$ fg [job번호]
```

```

[orcl:~]$ jobs
[1]  Stopped          R
[2]  Stopped          R
[3]  Stopped          Rscript h_stock.R
[4]  Stopped          Rscript s_stock.R
[5]  Stopped          vim stock.R
[6]  Stopped          R
[7]  Stopped          vim hhh.txt
[8]  Stopped          vim hhh.txt
[9]- Stopped          vim hhh2.txt
[10]+ Stopped         vim hhh3.txt
[orcl:~]$ fg 6
R

[6]+ Stopped          R
[orcl:~]$ fg 7

[orcl:~]$ jobs
[1]  Stopped          R
[2]  Stopped          R
[3]  Stopped          Rscript h_stock.R
[4]  Stopped          Rscript s_stock.R
[5]- Stopped          vim stock.R
[6]+ Stopped          R
[8]  Stopped          vim hhh.txt
[9]  Stopped          vim hhh2.txt
[10] Stopped          vim hhh3.txt

```

■ ps 명령어

" 현재 시스템에서 수행되고 있는 프로세서의 정보를 표시하는 명령어 "

예제 : \$ ps 옵션 프로세서아이디

옵션

- e : 현재 실행중인 모든 프로세서
- f : 실제 유저명, 개시시간등을 표시
- l : 프로세서의 상태, 우선도 등과 같은 상세한 정보
- p : 프로세서 아이디

문제 102. 현재 리눅스 시스템에서 실행중인 모든 프로세서를 출력 하시오.

```

0 S oracle    9868      1 0 75   0 - 158646 stext Jun26 ?      00:00:04 ora_mark_orcl
0 S oracle    9926      1 0 78   0 - 156854 stext Jun26 ?      00:00:00 ora_qmnc_orcl
0 S oracle    9940      1 0 75   0 - 157896 stext Jun26 ?      00:00:11 ora_cjq0_orcl
0 S oracle    9972      1 0 75   0 - 157122 stext Jun26 ?      00:00:00 ora_q000_orcl
0 S oracle    9974      1 0 78   0 - 156855 stext Jun26 ?      00:00:00 ora_q001_orcl
0 S oracle   10016      1 0 78   0 - 156726 stext Jun26 ?      00:00:05 ora_smco_orcl
4 S root     10760    2351 0 76   0 - 2481 stext 09:48 ?      00:00:00 sshd: oracle [priv]
5 S oracle   10762  10760 0 75   0 - 2481 stext 09:48 ?      00:00:00 sshd: oracle@pts/1
0 S oracle   10763  10762 0 75   0 - 1169 wait  09:48 pts/1    00:00:00 -bash
0 S oracle   10821      1 0 75   0 - 156817 stext 09:49 ?      00:00:00 ora_o000_orcl
0 S oracle   10823      1 0 80   0 - 104869 pipe_w 09:49 ?      00:00:00 oracle+ASM_o000_orcl (DESCRIPTION
0 S oracle   10825      1 0 75   0 - 156725 stext 09:49 ?      00:00:00 ora_o001_orcl
0 R oracle   10826  10763 0 77   0 - 546 -      09:49 pts/1    00:00:00 ps -elf
0 S oracle   16431      1 0 75   0 - 18076 stext Jun26 ?      00:00:21 gnome-terminal
0 S oracle   16434  16431 0 79   0 - 620 stext Jun26 ?      00:00:00 gnome-pty-helper
0 S oracle   16435  16431 0 76   0 - 1167 -      Jun26 pts/3    00:00:00 -bash
[orcl:~]$ ps -elf

```

세션을 복사해서 악성 sql을 날려본다.

```

SQL> select count(*)
      2 from sh.sales a, sh.sales b;

```

Top 명령어로 어떤 프로세스가 부하를 주는지 확인한다.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7088	oracle	25	0	626m	51m	38m	R	99.8	2.6	130:42.42	oracle
10869	oracle	25	0	626m	38m	25m	R	99.4	1.9	0:59.73	oracle
4961	oracle	18	0	400m	23m	11m	S	0.3	1.2	7:18.15	oraagent.bin
10964	oracle	15	0	2336	1076	792	R	0.3	0.1	0:00.01	top
1	root	15	0	2072	616	532	S	0.0	0.0	0:01.34	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.10	migration/0

Ps 명령어로 정보를 확인한다.

Ps -ef | grep 7088 (악성 프로세스의 pid)

Kill -9 pid 명령어를 이용해서 프로세스를 제거 한다.

```
[orcl:~]$ kill -9 7088
```

문제 103. 지금 작업을 vim으로 수행하고 있는 프로세서들을 출력 하고 삭제 하시오.

```
[orcl:~]$ ps -ef | grep vim
oracle  11007 10763  0 09:55 pts/1    00:00:00 vim aaa.txt
oracle  11067 11024  0 09:57 pts/4    00:00:00 grep vim
```

```
[orcl:~]$ kill -9 11007
[orcl:~]$ ps -ef | grep vim
oracle  11110 11024  0 09:58 pts/4    00:00:00 grep vim
```

■ kill 명령어의 옵션

\$ kill -9 프로세서번호

옵션

- 9 : 프로세서를 죽인다.
- 6 : 프로세서를 중단시킨다.
- 2 : 프로세서에게 종료하도록 지시한다.

문제 104. Kill 옵션 중 -2 옵션을 테스트 하시오.

- 터미널 창1

```
$ sar 1 10000
```

```
[orcl:~]$ sar 1 10000
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com)      06/27/18
```

	CPU	%user	%nice	%system	%iowait	%steal	%idle
10:02:34							
10:02:35	all	0.00	0.00	1.00	0.00	0.00	99.00
10:02:36	all	1.51	0.00	2.51	0.00	0.00	95.98
10:02:37	all	0.00	0.00	1.00	0.00	0.00	99.00
10:02:38	all	0.00	0.00	0.00	0.00	0.00	100.00
10:02:39	all	0.00	0.00	1.49	0.00	0.00	98.51

- 터미널 창2

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

```
$ kill -2 프로세스 번호
```

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

```
oracle  11191 11154  0 10:02 pts/5    00:00:00 sar 1 10000
```

```
oracle  11198 11024  0 10:02 pts/4    00:00:00 grep sar
```

```
[orcl:~]$ kill -2 11191
```

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

```
oracle  11206 11024  0 10:03 pts/4    00:00:00 grep sar
```

8. 쉘 스크립트 작성법

2018년 6월 27일 수요일 오전 10:05

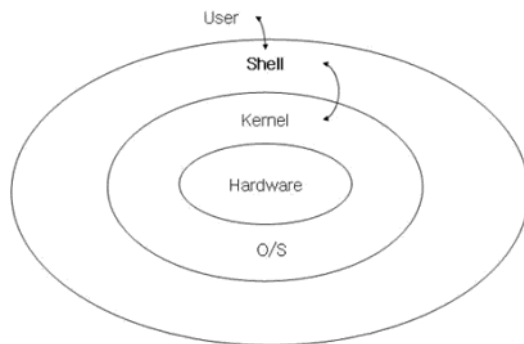
리눅스 쉘 스크립트 작성법을 배우면서 알고리즘 문제를 해결하는 체력을 키운다.

목표 : 카카오 블라인드 코딩 시험 대비 (이메일 주소, 전화번호 등록)

온라인 시험 2번 --> 오프라인 시험 1번

■ 쉘(shell)이란 무엇인가?

" 운영체제에서 제공하는 명령을 실행하는 프로그램 "



쉘의 종류 ?

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

Ex. `#!/bin/bash` ---> 쉘 중에 bash 쉘을 쓰겠다.

-Shell

우리가 도스상에서 내리는 대부분의 명령은 `command.com`이라는 파일이 처리한다. 전문가들은 이 `command.com`파일을 운영체제의 쉘(Shell)프로그램이라고 말한다. 쉘 프로그램은 사용자가 내린 명령을 운영체제가 수행할 수 있는 명령어를 입력받고 이를 컴퓨터의 운영체제에 전달하는 일을 한다. 다시 말해 사용자가 내린 명령을 운영체제가 수행할 수 있도록 통역해주는 일을 한다. 그래서 명령어해석기라고도 불린다.

-kernel

운영체제의 핵심은 커널 부분이다.

도스로 말하면 `io.sys`와 같은 파일이 이에 해당한다. 실질적으로 사용자가 내린 명령을 수행하면서 컴퓨터의 하드웨어를 조작하는 파일이다. 이 처럼 실질적으로 **사용자의 명령을 수행하면서 파일의 입출력을 담당하고, 하드웨어를 관리하는 프로그램을 커널이라고 한다.** 그리고 운영체제의 핵심은 커널이라고 부른다.

쉘은 사용자가 내린 명령을 커널이 이해할 수 있도록 통역해준다

그렇다면 왜 쉘과 커널로 구분을 하는가? 운영체제의 핵심이 커널이지만 사용자가 바로 커널을 제어하기는 매우 어렵다.

기계적인 부분이 많기 때문이다. 그래서 사용자가 내린 명령을 운영체제의 핵심인 커널 부분이 이해할 수 있도록 번역해주는 프로그램이 필요 했고 이 번역장치에 해당하는게 셸이다. 그래서 **우리가 운영체제라고 부르는 프로그램은 크게 커널과 셸로 구성 된다.**

■ 셸 스크립트란 ?

" 인터프리터 역할을 하는 것으로 시스템에서 지원하는 명령어들의 집합을 묶어서 프로그램화 한것을 말한다. "

셸 스크립트 프로그래밍이란?

1. C언어와 유사하게 프로그래밍이 가능하다.
2. 변수, 반복문, 제어문이 사용가능하다.
3. 별도로 컴파일 하지 않고 텍스트 파일 형태로 바로 실행이 가능하다.
4. vi나 gedit으로 작성이 가능
5. 리눅스의 많은 부분이 셸 스크립트로 작성되어 있다.

셸 스크립트를 작성하고 실행하는 방법

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

```
$ sh a.sh
```

```
[orcl:~]$ vi a.sh
[orcl:~]$ sh a.sh
호스트이름 : edydr1p0.us.oracle.com
```

\$./a.sh 을 사용 했을 때 ----> 실행 권한이 없다.

```
[orcl:~]$ ./a.sh
-bash: ./a.sh: Permission denied
```

```
[orcl:~]$ ls -l a.sh
-rw-r--r-- 1 oracle oinstall 35 Jun 27 10:13 a.sh
```

문제 105.	a.sh에 소유자가 실행할 수 있는 권한을 부여하시오.
----------------	--------------------------------

```
[orcl:~]$ chmod 744 a.sh
[orcl:~]$ ./a.sh
호스트이름 : edydr1p0.us.oracle.com
```

■ 변수 사용법

1. 모든 변수는 문자열(string)로 취급된다.
2. 변수이름은 대소문자를 구분한다.
3. **변수에 값을 대입할 때는 '=' 좌우에 공백이 없어야 한다!!!**
4. 변수의 값(문자열)을 출력하려면 \$를 붙이고 echo 명령어로 출력하면 된다.

```
ex. $ my_var='Hi ~~~'
    $ echo $my_var
```

■ 변수 산술연산 하는 방법

1. 변수에 대입한 값은 모두 문자열로 취급되기 때문에 변수에 들어있는 값을 숫자로 사칙연산 하려면 `expr`을 사용해야 한다.
2. 수식에 괄호 또는 곱하기를 사용하려면 그 앞에 꼭 역슬래쉬(`\`)를 붙여줘야한다.

```
ex. $ num1=100
    $ num2=200

[orcl:~]$ echo $num1 + $num2
100 + 200
[orcl:~]$ expr $num1 + $num2    (산술연산 사이에 한칸씩 띄어줘야한다)
300
```

■ 파라미터 변수

1. 파라미터 변수는 `$0`, `$1`, `$2 ..` 의 형태를 가진다.
2. 전체 파라미터는 `$*`로 표시된다.

```
ex.
    $ vi b.sh
        echo "실행파일 이름은 $0 입니다. "
        echo "첫 번째 파라미터 값은 $1 입니다. "
        echo "두 번째 파라미터 값은 $2 입니다. "
        echo "전체 파라미터 값은 $* 입니다. "
```

```
$ sh b.sh 10 20
```

```
[orcl:~]$ vi b.sh
[orcl:~]$ sh b.sh
실행파일 이름 : b.sh
첫 번째 파라미터 :
두 번째 파라미터 :
전체 파라미터 값 :
[orcl:~]$ sh b.sh 10 20
실행파일 이름 : b.sh
첫 번째 파라미터 : 10
두 번째 파라미터 : 20
전체 파라미터 값 : 10 20
```

문제 106.	num1과 num2의 곱을 구하시오.
----------------	----------------------

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

문제 107.	아래의 계산식을 구현 하시오.
----------------	------------------

	<code>(\$num2 + 200) * \$num1</code>
--	--

`[orcl:~]$ expr W($num2 + 200 W) W* $num1` ---> 괄호 뒤에 띄어줘야 한다.
40000

문제 108.	아래와 같이 두 정수를 파라미터 값으로 입력했을 때 나눈 나머지 값이 출력되게 하시오. <code>\$ sh f.sh 24 18</code> 나눈 나머지 값은 6 입니다.
----------------	---

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

```
num1=$1
num2=$2
num3=`expr $num1 % $num2`    <--- 계산된 결과를 변수에 넣기 위해서는 역따옴표(`)를 써야한다.
```

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

`echo "나눈 나머지 값은 $num3 입니다."`

설명 : 변수=`리눅스 명령어`

리눅스 명령어에 의해서 수행된 결과가 변수에 입력 되어야 한다면 역따옴표를 사용해야 한다.

문제 109.	이름이 scott인 사원의 부서번호가 출력되게 하는 셸을 작성 하시오.
----------------	---

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

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

`$ sh a2.sh`

`[orcl:~]$ vi a2.sh`
`[orcl:~]$ sh a2.sh`
20 번에서 근무합니다.

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

```
#!/bin/bash
echo -n "이름을 입력하세요."
read ename
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
echo "부서번호는 $deptno 입니다."
```

`[orcl:~]$ sh a3.sh`

이름을 입력하세요.scott
 부서번호는 20 입니다.
 [orcl:~]\$ sh a3.sh
 이름을 입력하세요.SCOTT
 부서번호는 20 입니다.

문제 111.	위의 코드를 수정해서 결과가 아래와 같이 출력되게 하시오. 이름을 입력 하세요 ~ allen allen은 30번 부서에서 근무합니다.
----------------	--

```
#!/bin/bash
echo -n "이름을 입력하세요."
read ename
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
echo "$ename은 $deptno번 부서에서 근무합니다."
```

```
[orcl:~]$ vi a4.sh
[orcl:~]$ sh a4.sh
이름을 입력하세요.scott
scott은 20번 부서에서 근무합니다.
```

문제 112.	dept.txt 를 /home/oracle에 올리고 부서번호를 물어보게 하고 부서번호를 입력하면 해당 부서번호의 부서위치가 출력되게 하시오. \$ sh a3.sh 부서번호를 입력 하시오 ~
----------------	---

```
#!/bin/bash

echo -n "입력하세요 부서번호를 "
read deptno

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

echo "위치는 $loc입니다.." <---- 작은 따옴표로 하면 $loc 이라고 출력됨
```

```
[orcl:~]$ vi a5.sh
[orcl:~]$ sh a5.sh
입력하세요 부서번호를 20
위치는 $loc입니다..
[orcl:~]$ vi a5.sh
[orcl:~]$ sh a5.sh
입력하세요 부서번호를 20
위치는 DALLAS입니다..
```

문제 113.	이름을 물어보게하고 이름을 입력하면 해당 사원의 부서위치가 출력되게 하시오. \$ sh a99.sh
----------------	--

이름을 입력하세요 ~~~ scott scott은 dallas에서 근무합니다.

```
#!/bin/bash
echo -n "입력하세요 이름을 "
read ename
deptno=`grep -i $ename emp.txt | awk '{print $8}'`
loc=`grep -i $deptno dept.txt | awk '{print $3}'`
echo "$ename은 $loc에서 근무합니다."
```

```
[orcl:~]$ sh a6.sh
입력하세요 이름을 scott
scott은 DALLAS에서 근무합니다.
```

문제 114.	직업을 물어보게하고 직업을 입력하면 해당 직업인 사원들의 월급이 출력되게 하시오.
----------------	---

```
$ sh job_sal.sh
직업을 입력 하시오 ~ salesman
```

```
#!/bin/bash
echo -n "직업을 입력 하세요 : "
read job
sal=`grep -i $job emp.txt | awk '{print $6}'`
echo "$sal"
```

```
직업을 입력 하세요 : salesman
1600
1250
1250
1500
```

문제 115.	직업을 물어보게 하고 직업을 입력하면 해당 직업의 토탈월급이 출력되게 하시오.
----------------	---

```
#!/bin/bash
echo -n "직업을 입력 하세요 : "
read job
sal=`grep -i $job emp.txt | awk '{sum += $6} END {print sum}'`
echo "$sal"
```

```
[orcl:~]$ vi a8.sh
[orcl:~]$ sh a8.sh
직업을 입력 하세요 : salesman
5600
```

문제 116.	직업을 물어보게 하고 해당 직업의 평균 월급이 출력되게 하시오.
----------------	-------------------------------------

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

```
echo -n "직업을 입력 하세요 : "
read job

cnt=`grep -i $job emp.txt | wc -l`
sal=`grep -i $job emp.txt | awk '{sum += $6} END {print sum}'`
mean=`expr $sal / $cnt`
echo "$mean"

[orcl:~]$ vi a9.sh
[orcl:~]$ sh a9.sh
직업을 입력 하세요 : salesman
1400
```

문제 117.	컬럼번호를 물어보게 하고 컬럼번호를 입력하면 해당 컬럼의 데이터의 토탈값과 평균값이 아래와 같이 출력되게 하시오. (점심시간 문제)
----------------	---

```
#!/bin/bash

echo -n "컬럼번호 입력 : "
read colnum
numb=`awk '{print $colnum}' emp.txt | wc -l`
awk '{x+="$colnum"}' END {print "토탈값 : " x " 평균값 : " x / '$numb'}' emp.txt

[orcl:~]$ vi a10.sh
[orcl:~]$ sh a10.sh
컬럼번호 입력 :6
토탈값 : 29025 평균값 :2073.21
```

■ 리눅스 셸에서 if문 사용방법

```
if [조건]; then
    실행문
elif [조건]; then
    실행문
else
    실행문
fi

* if문 사용시 주의 사항 : 공백 처리를 잘해줘야함.
```

■ if 문 조건문에 들어가는 비교 연산자

- 문자열 비교
 1. "문자열" = "문자열" : 두 문자열이 같으면 true
 2. "문자열1" != "문자열2" : 두 문자열이 같이 않으면 true
- 숫자 비교

숫자1 -eq 숫자2	두 숫자가 같으면 true
숫자1 -ne 숫자2	두 숫자가 같지 않으면 true

숫자1 -gt 숫자2	숫자 1이 숫자 2보다 크면 true
숫자1 -ge 숫자2	숫자 1이 숫자 2보다 크거나 같으면 true
숫자1 -lt 숫자2	숫자 1이 숫자 2보다 작으면 true
숫자1 -le 숫자2	숫자1이 숫자2보다 작거나 같으면 true
! 숫자1	숫자 1이 거짓이라면 true

예제 : \$ vi if1.sh

```
#!/bin/bash

if [ 100 -eq 200 ]; then
    echo "100과 200이 같다 "
else
    echo "100과 200은 같지 않다."
fi

[orcl:~]$ vi if1.sh
[orcl:~]$ sh if1.sh
100과 200은 같지 않다.
```

문제 118.	위의 스크립트를 활용해서 아래 처럼 숫자를 물어보게하고 아래와 같이 결과가 출력되게 하시오. 첫 번째 숫자를 입력하세요 ~ 10 두 번째 숫자를 입력하세요 ~ 20 20은 10보다 큽니다.
----------------	--

```
#!/bin/bash

echo -n "첫 번째 숫자를 입력하세요 ~ : "
read num1

echo -n "두 번째 숫자를 입력하세요 ~ : "
read num2

if [ $num1 -eq $num2 ]; then
    echo "$num1 과 $num2 는 같습니다."

elif [ $num1 -gt $num2 ]; then
    echo "$num1이 $num2 보다 큽니다."
else
    echo "$num2가 $num1 보다 큽니다."
fi

[orcl:~]$ sh if2.sh
첫 번째 숫자를 입력하세요 ~ : 3
두 번째 숫자를 입력하세요 ~ : 56
56가 3 보다 큽니다.
```

■ 리눅스에서 and와 or 연산자

1. and 는 && 또는 -a를 사용한다.
2. or는 || 또는 -o를 사용한다.

ex. if [\$sal -lt 2000] && [\$job='SALESMAN']; then
또는
if [\$sal -lt 2000 -a \$job='SALESMAN']; then

문제 119.	사원 이름을 물어보게하고 사원 이름을 입력했을 때 해당 사원의 부서번호가 30번 이고 월급이 2000 보다 작다면 월급 인상 대상자 입니다. 라는 메시지가 출력되게 하고 그렇지 않다면 월급 인상 대상자가 아닙니다. 라는 메시지가 출력되게 하시오.
----------------	---

```
#!/bin/bash

echo -n "이름을 입력 하세요"
read name

sal=`grep -i $name emp.txt | awk '{print $6}`
deptno=`grep -i $name emp.txt | awk '{print $8}`
if [ $sal -lt 2000 -a $deptno -eq 30 ]; then
    echo "월급인상 대상자 입니다."
else
    echo "월급인상 대상자가 아닙니다."
fi
```

```
[orcl:~]$ vi a119.sh
[orcl:~]$ sh a119.sh
이름을 입력 하세요allen
월급인상 대상자 입니다.
```

문제 120.	위의 코드를 수정해서 구구단 2단을 출력 하시오.
----------------	-----------------------------

```
#!/bin/bash

hap=0
for i in {1..9}
do
    echo " 2 x $i = `expr 2 \*$i` "
done

[orcl:~]$ sh a124.sh
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
```

$2 \times 6 = 12$
 $2 \times 7 = 14$
 $2 \times 8 = 16$
 $2 \times 9 = 18$

문제 121. 구구단 2단부터 9단까지 출력 하시오.

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

```

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

```

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

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

```

2 x 1 = 2 3 x 1 = 3 4 x 1 = 4 5 x 1 = 5 6 x 1 = 6 7 x 1 = 7 8 x 1 = 8 9 x 1 = 9
2 x 2 = 4 3 x 2 = 6 4 x 2 = 8 5 x 2 = 10 6 x 2 = 12 7 x 2 = 14 8 x 2 = 16 9 x 2 = 18
2 x 3 = 6 3 x 3 = 9 4 x 3 = 12 5 x 3 = 15 6 x 3 = 18 7 x 3 = 21 8 x 3 = 24 9 x 3 = 27
2 x 4 = 8 3 x 4 = 12 4 x 4 = 16 5 x 4 = 20 6 x 4 = 24 7 x 4 = 28 8 x 4 = 32 9 x 4 = 36
2 x 5 = 10 3 x 5 = 15 4 x 5 = 20 5 x 5 = 25 6 x 5 = 30 7 x 5 = 35 8 x 5 = 40 9 x 5 = 45
2 x 6 = 12 3 x 6 = 18 4 x 6 = 24 5 x 6 = 30 6 x 6 = 36 7 x 6 = 42 8 x 6 = 48 9 x 6 = 54
2 x 7 = 14 3 x 7 = 21 4 x 7 = 28 5 x 7 = 35 6 x 7 = 42 7 x 7 = 49 8 x 7 = 56 9 x 7 = 63
2 x 8 = 16 3 x 8 = 24 4 x 8 = 32 5 x 8 = 40 6 x 8 = 48 7 x 8 = 56 8 x 8 = 64 9 x 8 = 72
2 x 9 = 18 3 x 9 = 27 4 x 9 = 36 5 x 9 = 45 6 x 9 = 54 7 x 9 = 63 8 x 9 = 72 9 x 9 = 81

```

문제 122. 아래의 별표를 출력 하시오.

```

★
★★
★★★
★★★★
★★★★★

```

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

```

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

```

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

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

```

★
★★
★★★
★★★★
★★★★★

```

문제 123.	<p>위의 코드를 수정해서 입력받은 숫자만큼 행이 생성되게 하시오.</p> <pre>★ ★★ ★★★ ★★★★</pre>
---------	--

```
#!/bin/bash

echo -n "숫자입력 : "
read num

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

[orcl:~]$ vi star.sh
[orcl:~]$ sh star.sh
숫자입력 : 6
★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★
```

■ eval 명령어

" 문자열을 명령문으로 인식하고 실행한 결과를 반환 "

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

[orcl:~]$ str="ls -l emp.txt"
[orcl:~]$ echo $str
ls -l emp.txt
[orcl:~]$ eval $str
-r--r--r-- 1 oracle oinstall 962 Jun 26 10:09 emp.txt
```

문제 124.	<p>숫자를 입력하면 아래와 같이 ★가 출력되게 하시오.</p> <pre>\$ sh star.sh 숫자를 입력 하세요 ~ 5</pre>
---------	---

```
#!/bin/bash

echo -n "숫자입력 : "
```



```
[orcl:~]$ vi star.sh
[orcl:~]$ sh star.sh
숫자입력 : 6
★★★★★★
★★★★★★
★★★★★
★★★★
★★★
★★
★
```

```
[orcl:~]$ sh -x wtar.sh
+ echo -n '숫자를 입력하세요 '
숫자를 입력하세요 + read num
5
+ num1=5
++ eval echo '{1..5}'
+++ echo 1 2 3 4 5
+ for i in ``eval echo {1..$num}``
+ star=
++ eval echo '{1..5}'
+++ echo 1 2 3 4 5
+ for j in ``eval echo {1..$num1}``
+ star=${star}\241\332
+ for j in ``eval echo {1..$num1}``
+ star=${star}\241\332\241\332
+ for j in ``eval echo {1..$num1}``
+ star=${star}\241\332\241\332\241\332
+ for j in ``eval echo {1..$num1}``
+ star=${star}\241\332\241\332\241\332\241\332
+ for j in ``eval echo {1..$num1}``
+ star=${star}\241\332\241\332\241\332\241\332\241\332
+ echo ${star}\241\332\241\332\241\332\241\332\241\332\241\332
★★★★★
```

```
#!/bin/bash

for i in {1..100}
do
```

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

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

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

```
Adobe      bbb      emp21.txt emp44.txt emp67.txt emp89.txt  s.R
Desktop    c.txt    emp22.txt emp45.txt emp68.txt emp9.txt   s.sh
H_stock.csv cp_emp.sh emp23.txt emp46.txt emp69.txt emp90.txt  s_stock.R
R-3.2.3     d.txt    emp24.txt emp47.txt emp7.txt   emp91.txt  sar.R
R-3.2.3.tar.gz datamodeler.log emp25.txt emp48.txt emp70.txt emp92.txt  sar_0622.txt
S_stock.csv del_winter.sh emp26.txt emp49.txt emp71.txt emp93.txt  size.sh
a.sh        dept.txt emp27.txt emp5.txt  emp72.txt emp94.txt  sss.sh
a.txt       deptno2.sh emp28.txt emp50.txt emp73.txt emp95.txt  star.sh
a10.sh      diff.sh   emp29.txt emp51.txt emp74.txt emp96.txt  stock.R
a119.sh     e.txt     emp3.txt  emp52.txt emp75.txt emp97.txt  test100
a123.sh     emp.txt   emp30.txt emp53.txt emp750.txt emp98.txt  test200
a124.sh     emp1.txt  emp31.txt emp54.txt emp76.txt emp99.txt  test5
a2.sh       emp10.txt emp32.txt emp55.txt emp77.txt ename2.sh  test64
a3.sh       emp100.txt emp33.txt emp56.txt emp78.txt f.sh       test66
a4.sh       emp11.txt  emp34.txt emp57.txt emp79.txt find_file.sh txttar.tar
a5.sh       emp12.txt  emp35.txt emp58.txt emp8.txt  gugu.sh    winter.txt
a6.sh       emp13.txt  emp36.txt emp59.txt emp80.txt h_stock.R  winter200.txt
a7.sh       emp14.txt  emp37.txt emp6.txt  emp81.txt hhh.txt    winter500.txt
a8.sh       emp15.txt  emp38.txt emp60.txt emp82.txt if1.sh     winter_cp.sh
a9.sh       emp16.txt  emp39.txt emp61.txt emp83.txt if2.sh     winter_head.txt
all.tar     emp17.txt  emp4.txt  emp62.txt emp84.txt job2.sh    year.sh
b.sh        emp18.txt  emp40.txt emp63.txt emp85.txt job_sumsal.sh year_sum.sh
b.txt       emp19.txt  emp41.txt emp64.txt emp86.txt labs
backup      emp2.txt   emp42.txt emp65.txt emp87.txt ls.txt
baek.sh     emp20.txt  emp43.txt emp66.txt emp88.txt oradiag_oracle
```

문제 127.	짜궁하고 자리를 바꿔서 emp1.txt~emp100.txt 중 하나의 파일을 랜덤으로 골라서 데이터를 변경 하고 자리로 돌아 오시오~
---------	--

문제 128.	emp3.txt를 열어서 3000을 3900으로 변경 하시오.
---------	------------------------------------

```
$ vi emp3.txt
```

```
:%s/3000/3900/g
```

문제 129.	emp.txt와 emp3.txt 파일 둘간의 데이터 차이가 있는지 확인 하시오.
---------	--

```
$ diff emp.txt emp3.txt
```

```
$ diff --brief emp.txt emp3.txt
```

```
[orcl:~]$ diff emp.txt emp3.txt
```

```

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

```

문제 130. 짝공이 어떠한 .txt를 변경했는지 한번에 알아내시오.

```

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

```

```

[orcl:~]$ vi find_emp.sh
[orcl:~]$ sh find_emp.sh
Files emp.txt and emp3.txt differ
Files emp.txt and emp35.txt differ

```

■ sed 명령어로 공백라인을 지우는 방법 (데이터 정제 작업)

1. emp.txt를 열어서 중간에 공백라인을 하나 넣으시오.
2. emp.txt 안에 공백라인이 있는지 검색한다.

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

[orcl:~]\$ vi emp5.txt <--- 커서를 위치하고 o를 누르면 빈 줄이 생김.

```

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      jjj        7566 19-APR-87      3000     0      20
7839 KING        PRESIDENT  0 17-NOV-81      5000     0      10
7844 TURNER     SALESMAN   7698 08-SEP-81      1500     0      30
7876 ADAMS      CLERK      7788 23-MAY-87      1100     0      20
7900 JAMES      CLERK      7698 03-DEC-81      950      0      30
7902 FORD       jjj        7566 03-DEC-81      3000     0      20
7934 MILLER     CLERK      7782 23-JAN-82      1300     0      10

```

```

[orcl:~]$ sed -n /^$/p emp5.txt | wc -l
1

```

3. emp.txt 안의 공백라인을 삭제한다.

```
[orcl:~]$ sed '/^$/d' emp5.txt
```

```
[orcl:~]$ sed '/^$/d' emp5.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN   7698 20-FEB-81     1600     300     30
7521 WARD        SALESMAN   7698 22-FEB-81     1250     500     30
7566 JONES      MANAGER    7839 02-APR-81     2975      0      20
7654 MARTIN     SALESMAN   7698 28-SEP-81     1250    1400     30
7698 BLAKE      MANAGER    7839 01-MAY-81     2850      0      30
7782 CLARK      MANAGER    7839 09-JUN-81     2450      0      10
7788 SCOTT      jjj        7566 19-APR-87     3000      0      20
7839 KING      PRESIDENT   0 17-NOV-81     5000      0      10
7844 TURNER     SALESMAN   7698 08-SEP-81     1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87     1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950      0      30
7902 FORD       jjj        7566 03-DEC-81     3000      0      20
7934 MILLER     CLERK      7782 23-JAN-82     1300      0      10
```

문제 131. 짝공과 자리를 바꿔서 emp1.txt ~ emp100.txt 중 몇 개를 열어서 공백을 막 넣어주고 자리로 돌아오시오 ~~~0.

문제 132. 자리로 돌아와서 짝공이 어떠한 파일에 공백을 넣었는지 확인 하시오.

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

```
[orcl:~]$ vi find_0100.sh
[orcl:~]$ sh find_0100.sh
emp5 : 1
emp24 : 18
emp77 : 8
emp89 : 20
emp100 : 8
```

문제 133. 짝공이 공백을 입력한 파일들의 공백을 모두 지우시오.

```
#!/bin/bash

for i in {1..100}
do
    num=`sed -n /^$/p emp$i.txt | wc -l`
    if [ $num -ne 0 ]; then
        sed '/^ *$/d' emp$i.txt >> emp$i_backup.txt
        rm emp$i.txt
        mv emp$i_backup.txt emp$i.txt
        echo "emp$i의 공백이 지워졌습니다."
    fi
done

[orcl:~]$ sh space1.sh
emp12의 공백이 지워졌습니다.
emp55의 공백이 지워졌습니다.
```

문제 134. (알고리즘 문제 :bc 카드 공채 문제) 아래의 셸을 생성 하시오.

```
var=1
for ((i=$1;i>=1;i--))
do
    var=`expr $var W* $i`
    if [ $i -ne 1 ]; then
        echo -n "$i x "
    else
        echo "$i = $var"
    fi
done
```

[orcl:~]\$ vi factorial.sh

[orcl:~]\$ sh factorial.sh 5

5 x 4 x 3 x 2 x 1 = 120

[orcl:~]\$ sh factorial.sh 4

4 x 3 x 2 x 1 = 24

문제 135. (알고리즘 문제) 아래의 power 함수를 셸로 구현 하시오.

[orcl:~]\$ vi power.sh

[orcl:~]\$ sh power.sh 2 3

8

```
num=$1
```

```
t=1
```

```
for ((i=1;i<=$2;i++))
```

```
do
```

```
    t=`expr $t W* $num`
```

```
done
```

```
echo $t
```

문제 136. (알고리즘 문제) 최대 공약수를 출력 하시오.

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

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

문제 137. 직업과 직업별 토달월급을 출력하시오.

```
[orcl:~]$ cat emp.txt | awk '{print $3}' | sort | uniq
CLERK
MANAGER
PRESIDENT
SALESMAN
jjj
```

문제 138. 위의 결과를 변수에 담아서 for loop 문으로 변수의 내용을 출력 하시오.

```
[orcl:~]$ job=`awk '{print $3}' emp.txt | sort | uniq`
[orcl:~]$ for i in $job
> do
> echo $i
> done
CLERK
MANAGER
PRESIDENT
SALESMAN
jjj
```

문제 139. 직업이 SALESMAN인 사원들의 토달월급을 출력 하시오.

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

문제 140. 직업과 직업별 토달월급을 출력 하시오.

```
[orcl:~]$ job=`awk '{print $3}' emp.txt | sort | uniq`
[orcl:~]$ for i in $job
> do
> grep -i $i emp.txt | awk '{sum += $6} END {print $3 " : " sum}'
> done
CLERK : 4150
MANAGER : 8275
PRESIDENT : 5000
SALESMAN : 5600
jjj : 6000
[orcl:~]$
```

문제 141. 부서번호, 부서번호별 평균 월급을 출력 하시오.

```
deptno=`awk '{print $8}' emp.txt | sort | uniq`
for i in $deptno
do
    cnt=`grep -i $i emp.txt | awk '{c+=1} END {print c}'`
    grep -i $i emp.txt | awk '{sum += $6} END {print $8 " : " sum/"$cnt}'
done

[orcl:~]$ vi deptmean.sh
[orcl:~]$ sh deptmean.sh
10 : 2462.5
20 : 2079.17
10 : 1855.56
```

문제 142. 직업이 salesman인 사원들의 데이터만 추출해서 salesman.txt로 생성 하시오.

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

문제 143. emp.txt를 직업별로 데이터를 분리해서 아래와 같이 텍스트 파일이 생성되게 하시오.

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

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

■ 데이터를 db에서 추출하는 방법 2가지

1. dump file

ex.

emp.dmp 파일로 db의 emp 테이블을 백업한다.

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

Export: Release 11.2.0.1.0 - Production on Thu Jun 28 14:00:58 2018

Copyright (c) 1982, 2009, Oracle and/or its affiliates. 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
Export done in US7ASCII character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)

About to export specified tables via Conventional Path ...

. . exporting table EMP 14 rows exported

EXP-00091: Exporting questionable statistics.

EXP-00091: Exporting questionable statistics.

Export terminated successfully with warnings.

emp.dmp 가 생성됐는지 확인

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

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

sqlplus 에서 emp 테이블을 drop 한다.

[orcl:~]\$ sqlplus scott/tiger

SQL*Plus: Release 11.2.0.1.0 Production on Thu Jun 28 14:03:13 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> drop table emp;

Table dropped.

다시 emp.dmp 파일을 db에 import 해준다.

[orcl:~]\$ imp scott/tiger file=emp.dmp tables=emp

Import: Release 11.2.0.1.0 - Production on Thu Jun 28 14:04:29 2018

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

Export file created by EXPORT:V11.02.00 via conventional path
import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing SCOTT's objects into SCOTT
. importing SCOTT's objects into SCOTT
. . importing table "EMP" 14 rows imported
About to enable constraints...
Import terminated successfully without warnings.

emp 테이블이 생겼는지 확인한다.

SQL> select count(*) from emp;

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

2. csv 또는 text file

SQL> set colsep ','

SQL> set pagesize 0

SQL> set linesize 120

SQL> set sqlprompt ""

SQL> spool emp222.csv

SQL>

SQL> select * from emp;

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

14 rows selected.

SQL> spool off

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:~]\$

[orcl:~]\$ cat emp222.csv

SQL>

SQL> select * from emp;

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

14 rows selected.

SQL> spool off

[orcl:~]\$

문제 144.	while loop 문으로 factorial을 구현하시오.
----------------	----------------------------------

```
n=1
i=1
while [ $i -ne $1 ]
do
    i=`expr $i + 1`
    n=`expr $n \* $i`
done
echo $n
```

\$ sh factorial.sh 5

문제 145.	while loop 문으로 factorial을 구현하시오.
----------------	----------------------------------

```
if [ $1 > $2 ]; then
    n1=$1
    n2=$2
elif [ $2 > $1 ]; then
    n1=$2
    n2=$1
fi
n=0
```

```

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

[orcl:~]$ vi a.sh
[orcl:~]$ sh a.sh 24 16
8

```

문제 146.	emp 테이블의 데이터를 csv로 생성하고 R에서 로드 하시오.
----------------	-------------------------------------

```

# sql 파일을 만들고

$ vi aaa.sql

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

spool emp227.csv

select * from emp;

spool off

exit

# 실행과 동시에 aaa.sql 문이 실행된다.
$ sqlplus scott/tiger @aaa.sql

$ R

emp<-read.csv('emp227.csv',strip.white=T)

```

문제 147.	sed 명령어로 aaa.sql의 emp를 dept로 변경 하시오.
----------------	--------------------------------------

```

$ sed 's/emp/dept/' aaa.sql > aaa.sql
$ sqlplus scott/tiger @aaa.sql

```

문제 148.	위의 스크립트를 이용해서 아래와 같이 테이블명을 물어보게 하고 테이블명을 입력하면 csv 파일이 생성되게 하시오. \$ create_csv.sh
----------------	---

```
테이블명을 입력 하세요 ~salgrade
$ cat salgrade.csv
```

```
echo -n "csv로 변경할 테이블명을 입력 하세요 : "
read tb
```

```
sed 's/emp/'$tb'/' aaa.sql > $tb.sql
sqlplus scott/tiger @$tb.sql
```

```
[orcl:~]$ vi create_csv.sh
[orcl:~]$ sh create_csv.sh
csv로 변경할 테이블명을 입력 하세요 : salgrade
```

SQL*Plus: Release 11.2.0.1.0 Production on Thu Jun 28 16:50:53 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

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

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:~]$ cat salgrade.csv
"GRADE","LOSAL","HISAL"
1,700,1200
2,1201,1400
3,1401,2000
4,2001,3000
5,3001,9999
```

9. 반복문

2018년 6월 28일 목요일 오전 10:19

■ for문

사용법 :

1. **for** 변수 **in** 숫자1 숫자2 숫자3
2. **for** 변수 **in** {숫자1..숫자2} <--숫자의 순서 상관없이 작은 숫자부터 큰숫자 까지
(\$1 과 같은 입력값을 받을때 문자로 받아들여서 불편... {'eval echo 1..\$1'}와 같이 바꿔줘야한다..)
3. **for** ((i=1;i<\$2;i++)) <--- 2번째로 입력받은 값보다 적은 i값을 가질 동안 ..

모두 do done있어야함.

```
for 변수 in 값1,값2,값3
do
    [let 변수명=계산식] <-- expr 대신 let을 사용할수 있다 let은 역슬래시 안써도 됨
    반복할 문장
done
```

ex.

```
#!/bin/bash
hap=0
for l in 1 2 3 4 5 6 7 8 9 10 <----- {1..10} 으로 변경 가능
do
    hap=`expr $hap + $i`
done
echo "1부터 10까지의 합은 "$hap

[orcl:~]$ vi a123.sh
[orcl:~]$ sh a123.sh
1부터 10까지의 합은 0+1+2+3+4+5+6+7+8+9+10
```

■ while문

사용법 :

```
while [ 값1 조건식 값2 ]
do
    수행문
    수행문
done
```

ex.

```
n=0
```

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

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

10. case문을 이용한 자동화 스크립트

2018년 6월 29일 금요일 오전 10:00

1. Case 문

" if 문과 유사한 문법인데 특정 쉘 스크립트를 실행할 수 있도록 도와주는 문법 "
R의 menu 함수와 같다.

Ex. \$ vi m2.sh

```
echo "  
    1. sar 그래프  
    2. csv 파일 생성  
    "  
echo " "  
echo -n "번호를 입력하세요"  
read choice  
case $choice in  
    1)  
        /home/oracle/sar.sh;;  
    2)  
        /home/oracle/create.sh ;; # 여기에 있는 파일들의 권한이 모두 777 이여야함  
esac
```

[orcl:~]\$ vi sar.sh

Rscript /home/oracle/sar.R

[orcl:~]\$ vi sar.R

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

문제 149.	이전에 만들었던 find_file.sh가 잘 실행되는지 확인 하여라.
----------------	--

[orcl:~]\$ vi find_file.sh

```
echo " "  
echo "현재 디렉토리는 다음과 같습니다"  
pwd  
echo -n "현재 디렉토리 밑에 검색할 파일명을 입력하세요."
```

```

read file
echo -n "depth는 어떻게 지정하시겠습니까?"
read depth
find . -maxdepth $depth -name $file -print

```

```

[orcl:~]$ sh find_file.sh
현재 디렉토리는 다음과 같습니다
/home/oracle
현재 디렉토리 밑에 검색할 파일명을 입력하세요.emp.txt
depth는 어떻게 지정하시겠습니까?3
./test100/emp.txt
./test200/emp.txt
./bbb/emp.txt

```

문제 150.	리눅스 자동화 스크립트인 m2.sh의 세번째로 find_file.sh를 추가하여라.
----------------	--

```

[orcl:~]$ vi m2.sh

echo "
    1. sar 그래프
    2. csv 파일 생성
    3. file 찾기
"

echo " "
echo -n "번호를 입력하세요"
read choice
case $choice in
    1)
        /home/oracle/sar.sh;;
    2)
        /home/oracle/create.sh ;;
    3)
        /home/oracle/find_file.sh ;;
esac

```

```

[orcl:~]$ sh m2.sh

1. sar 그래프
2. csv 파일생성
3. file 찾기

번호를 입력하세요3

현재 디렉토리는 다음과 같습니다
/home/oracle
현재 디렉토리 밑에 검색할 파일명을 입력하세요.emp.txt
depth는 어떻게 지정하시겠습니까?4
./test100/emp.txt
./test200/emp.txt
./bbb/emp.txt

```


문제 151.	특정 파일 안에 특정 단어가 몇 건이 있는지 검색하는 쉘을 만들기 위해서 겨울왕국의 elsa라는 단어가 몇건이 나오는지 검색하여라.
----------------	---

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

-i	ignore 옵션은 대소문자 구분하지 않고 찾을
-w	word 옵션은 패턴 자체가 완전한 단어로 독립되었을 경우만 글자로 인식함

문제 152.	위의 스크립트로 아래와 같이 쉘 스크립트를 작성하여라
----------------	-------------------------------

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

```
[orcl:~]$ vi find_word.sh
[orcl:~]$ sh find_word.sh
검색 대상 파일명을 입력하세요.winter.txt
검색할 단어를 입력하세요.elsa
329
```

문제 153.	자동화 스크립트 4번에 find_word.sh를 추가 하여라
----------------	-----------------------------------

```
[orcl:~]$ vi m2.sh
echo "
1. sar 그래프
2. csv 파일 생성
3. file 찾기
4. 파일의 특정 단어 수 찾기
"
echo " "
echo -n "번호를 입력하세요"
read choice
case $choice in
1)
    /home/oracle/sar.sh;;
2)
    /home/oracle/create.sh ;;
3)
    /home/oracle/find_file.sh ;;
4)
    /home/oracle/find_word.sh
esac
```

```
[orcl:~]$ chmod 777 find_word.sh
[orcl:~]$ sh m2.sh
```

1. sar 그래프
2. csv 파일생성
3. file 찾기
4. 파일의 특정 단어수 찾기

번호를 입력하세요4

검색 대상 파일명을 입력하세요.winter.txt

검색할 단어를 입력하세요.elsa

329

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

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

A ! 근데 그냥 Ctrl + C하면 꺼짐.

Ctrl + C : kill

Ctrl + Z : stop

```
E325: ATTENTION
Found a swap file by the name ".aaa.sh.swp"
    owned by: oracle    dated: Fri Jun 29 11:12:01 2018
    file name: ~oracle/aaa.sh
    modified: YES
    user name: oracle   host name: edydr1p0.us.oracle.com
    process ID: 21879 (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:
```

작성 도중 ctrl + z 를 누르면 다음 실행부터 저런 메시지가 뜨는데

Swp 파일을 지우면 해결 된다.

```
rm *.swp
```

문제 154.	아래의 명령어를 rmswp.sh 스크립트에 저장하고 수행 하시오. 파일이 없으면 파일이 없습니다 라는 메시지가 출력 되게 하시오.
----------------	--

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

```

[orcl:~]\$ sh rmswp.sh

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

문제 155.	위의 rmswp.sh 를 자동화 스크립트 5번에 추가하시오.
----------------	-----------------------------------

[orcl:~]\$ vi m2.sh

```

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

echo " "
echo -n "번호를 입력하세요"
read choice
case $choice in
    1)
        /home/oracle/sar.sh;;
    2)
        /home/oracle/create.sh ;;
    3)
        /home/oracle/find_file.sh ;;
    4)
        /home/oracle/find_word.sh ;;
    5)
        /home/oracle/rmswp.sh ;;
Esac

```

[orcl:~]\$ chmod 777 rmswp.sh

[orcl:~]\$ sh m2.sh

1. sar 그래프
2. csv 파일생성
3. file 찾기
4. 파일의 특정 단어수 찾기
5. swp파일 모두 삭제

번호를 입력하세요5

ls: *.swp: No such file or directory

파일이 없습니다

문제 156. 두개의 파일의 차이를 확인하는 diff 명령어가 잘 수행되는지 확인 하시오.

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

비교할 첫 번째 파일을 입력하세요.emp.txt

비교할 두 번째 파일을 입력하세요.emp35.txt

```
< 7788 SCOTT    jjj    7566 19-APR-87    3000 0    20
13,14c12
< 7902 FORD    jjj    7566 03-DEC-81    3000 0    20
< 7934 MILLER  CLERK    7782 23-JAN-82    1300 0    10
---
> 89 BAAAM    DRUNK    33
```

문제 157. diff.sh를 m2.sh 6번에 추가하여라.

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

```
echo " "
echo -n "비교할 첫번째 파일명을 입력하세요"
read first
echo -n "비교할 두번째 파일명을 입력하세요"
read second
diff $first $second
```

```
echo "
1. sar 그래프
2. csv 파일생성
3. file 찾기
4. 파일의 특정 단어수 찾기
5. swp파일 모두 삭제
6. 파일 비교하기
"
```

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

```
case $choice in
1)
    /home/oracle/sar.sh;;
2)
    /home/oracle/create.sh;;
3)
    /home/oracle/find_file.sh;;
4)
    /home/oracle/find_word.sh;;
5)
    /home/oracle/rmswp.sh;;
6)
```

```
/home/oracle/diff.sh;;  
esac
```

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

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 29 11:37:54 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 hr
```

```
2 account unlock;      # hr 계정의 lock 을 해제  
User altered.
```

```
SQL> alter user hr
```

```
2 identified by hr;     # hr 계정의 패스워드를 hr로 변경하는 명령어  
User altered.
```

```
SQL> connect hr/hr      # hr 계정으로 접속
```

```
Connected.
```

```
SQL> select * from tab;    # hr 계정이 가지고 있는 테이블 리스트를 조회
```

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 테이블들을 전부 csv 파일로 생성 하시오.

```
read -p "oracle ID : " user  
read -p "oracle PW : " pw  
echo -n "csv로 변경할 테이블명을 입력 하세요 : "  
read tb
```

```

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

      spool $tb.csv
      select * from $tb;
      spool off
      exit

```

```

sqlplus $user/$pw << EOF
$query
EOF

```

```

[orcl:~]$ sh create_csv2.sh
oracle ID : hr
oracle PW : hr
csv로 변경할 테이블명을 입력 하세요 : jobs

```

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 29 14:02:34 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> SQL> SQL> SQL>

JOB_ID	JOB_TITLE		MIN_SALARY	MAX_SALARY
AD_PRES	President		20080	40000
AD_VP	Administration Vice President		15000	30000
AD_ASST	Administration Assistant		3000	6000
FI_MGR	Finance Manager		8200	16000
FI_ACCOUNT	Accountant		4200	9000
AC_MGR	Accounting Manager		8200	16000
AC_ACCOUNT	Public Accountant		4200	9000
SA_MAN	Sales Manager		10000	20080
SA_REP	Sales Representative		6000	12008
PU_MAN	Purchasing Manager			8000, 15000
PU_CLERK	Purchasing Clerk		2500	5500
ST_MAN	Stock Manager		5500	8500
ST_CLERK	Stock Clerk		2008	5000
SH_CLERK	Shipping Clerk		2500	5500
IT_PROG	Programmer		4000	10000
MK_MAN	Marketing Manager			9000, 15000
MK_REP	Marketing Representative		4000	9000
HR_REP	Human Resources Representative		4000	9000

JOB_ID	JOB_TITLE		MIN_SALARY	MAX_SALARY
--------	-----------	--	------------	------------

PR_REP ,Public Relations Representative , 4500, 10500
 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:~]\$ ls -l jobs.csv
 -rw-r--r-- 1 oracle oinstall 2623 Jun 29 14:02 jobs.csv

문제 160. Emp.csv 에서 위의 1번과 2번 라인을 삭제하고 emp2.csv로 저장 하시오.

```
[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
```

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

```
[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 저장 하시오.

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

```
[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 저장 하시오.

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

```
[orcl:~]$ sed '/^$/d' 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

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

문제 163. Emp.csv가 R에서 잘 로드되는지 확인 하시오.

```
[orcl:~]$ R
> emp<-read.csv("/home/oracle/emp4.csv", strip.white=T)
> emp
  EMPNO ENAME   JOB MGR HIREDATE SAL COMM DEPTNO
1  7369 SMITH   CLERK 7902 17-DEC-80 800  NA   20
2  7499 ALLEN  SALESMAN 7698 20-FEB-81 1600 300   30
3  7521 WARD   SALESMAN 7698 22-FEB-81 1250 500   30
4  7566 JONES   MANAGER 7839 02-APR-81 2975  NA   20
5  7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400   30
6  7698 BLAKE   MANAGER 7839 01-MAY-81 2850  NA   30
```



```

7 7782 CLARK  MANAGER 7839 09-JUN-81 2450  NA   10
8 7788 SCOTT  ANALYST 7566 19-APR-87 3000  NA   20
9 7839 KING  PRESIDENT   NA 17-NOV-81 5000  NA   10
10 7844 TURNER SALESMAN 7698 08-SEP-81 1500   0   30
11 7876 ADAMS   CLERK 7788 23-MAY-87 1100  NA   20
12 7900 JAMES   CLERK 7698 03-DEC-81  950  NA   30
13 7902 FORD    ANALYST 7566 03-DEC-81 3000  NA   20
14 7934 MILLER   CLERK 7782 23-JAN-82 1300  NA   10

```

문제 164. 위에서 사용한 아래의 5개의 스크립트를 쉘로 저장해서 R로 깔끔하게 로드되는 csv파일을 생성 하시오.

/home/oracle/**create_all_csv.sh**

```

read -p "oracle ID : " user
read -p "oracle PW : " pw

query=" set colsep ','
      set pagesize 200
          set linesize 120
          set sqlprompt "
          set feedback off
          set echo off
          set underline off

      spool tmp1.txt
      select tname from tab where TABTYPE = 'TABLE';
      spool off
      exit"

sqlplus $user/$pw << EOF
$query
EOF

sed '1,3d' tmp1.txt > tmp2.txt
sed '/spool/d' tmp2.txt | awk '{print $1}' > tmp3.txt

rm tmp1.txt
rm tmp2.txt

num=`cat tmp3.txt | wc -l`
for i in `eval echo {1..$num}`
do
    name=`cat tmp3.txt | head -n $i | tail -n 1`
    query2=" set colsep ','
          set pagesize 200
              set linesize 120
              set sqlprompt "
              set feedback off
              set echo off
              set underline off

          spool $name.csv

```

```

        select * from $name;
        spool off
        exit"

sqlplus $user/$pw << EOF
$query2
EOF

sed '1,2d' $name.csv > tmp1.csv
sed '/spool/d' tmp1.csv > tmp2.csv
sed '/^$/d' tmp2.csv > tmp3.csv
rm tmp1.csv
rm tmp2.csv
mv tmp3.csv $name.csv

done

```

문제 165. 위에서 사용한 아래의 5개의 스크립트를 쉘로 저장해서 R로 깔끔하게 로드되는 csv파일을 생성 하시오.

/home/oracle/**create_all_csv.sh**

```

read -p "oracle ID : " user
read -p "oracle PW : " pw

query=" set colsep ';'
        set pagesize 200
                set linesize 120
                set sqlprompt ""
                set feedback off
                set echo off
                set underline off

        spool tmp1.txt
        select tname from tab where TABTYPE = 'TABLE';
        spool off
        exit"

sqlplus $user/$pw << EOF
$query
EOF

sed '1,3d' tmp1.txt > tmp2.txt
sed '/spool/d' tmp2.txt | awk '{print $1}' > tmp3.txt

rm tmp1.txt
rm tmp2.txt

num=`cat tmp3.txt | wc -l`
for i in `eval echo {1..$num}`
do
        name=`cat tmp3.txt | head -n $i | tail -n 1`

```

```

query2=" set colsep ','
set pagesize 200
        set linesize 120
        set sqlprompt "
        set feedback off
        set echo off
        set underline off

        spool $name.csv
        select * from $name;
        spool off
        exit"

sqlplus $user/$pw << EOF
$query2
EOF

sed '1,2d' $name.csv > tmp1.csv
sed '/spool/d' tmp1.csv > tmp2.csv
sed '/^$/d' tmp2.csv > tmp3.csv
rm tmp1.csv
rm tmp2.csv
mv tmp3.csv $name.csv

done

```

문제 166. 위에서 사용한 아래의 5개의 스크립트를 쉘로 저장해서 R로 깔끔하게 로드되는 csv파일을 생성 하시오.

```

echo "
1. sar 000000
2. csv 00000000
3. file 0000
4. 0000000000000000000000000000 0
5. swp 0000000000000000
6. 00000000000000000000
7. sql create table
8. sql create table all
"

echo " "
echo -n "0000000000000000000000000000"
read choice

case $choice in
1)
    /home/oracle/sar.sh;;
2)
    /home/oracle/create.sh;;
3)
    /home/oracle/find_file.sh;;

```

```
4)      /home/oracle/find_word.sh;;
5)      /home/oracle/rmswp.sh;;
6)      /home/oracle/diff.sh;;
7)      /home/oracle/create_csv2.sh;;
8)      /home/oracle/create_all_csv.sh;;
esac
```

11. 리눅스 환경에서 R과 오라클 연동시키기

2018년 6월 29일 금요일 오후 3:11

■ 리눅스에서 RJDBC 를 이용하여 R과 오라클 연동하는 방법

1. 리눅스에서 오라클 리스너 상태를 확인한다.

```
# 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)))

```
$ lsnrctl status
```

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

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 29-JUN-2018 15:11:25

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

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))

STATUS of the LISTENER

Alias	LISTENER
Version	TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date	22-JUN-2018 09:58:53
Uptime	7 days 5 hr. 12 min. 32 sec
Trace Level	off
Security	ON: Local OS Authentication
SNMP	OFF
Listener Parameter File	/u01/app/oracle/product/11.2.0/grid/network/admin/listener.ora
Listener Log File	/u01/app/oracle/diag/tnslsnr/edydr1p0/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p0.us.oracle.com)(PORT=1521)))

Services Summary...

Service "+ASM" has 1 instance(s).

Instance "+ASM", status READY, has 1 handler(s) for this service...

Service "orcl.us.oracle.com" has 1 instance(s).

Instance "orcl", status READY, has 1 handler(s) for this service...

Service "orclXDB.us.oracle.com" has 1 instance(s).

Instance "orcl", status READY, has 1 handler(s) for this service...

The command completed successfully

2. 아래와 같이 오라클에 접속이 되는지 확인 하시오.

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

서버의 주소 포트주소 서비스 이름

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

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 29 15:23:09 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
```

3. 오라클과 R연동을 위해 필요한 패키지를 설치한다 .

```
# 라이브러리 설치

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

4. 자바 홈을 설치한다.

```
# 자바 홈 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

```
# 자바 버전확인
```

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

```
# RJDBC 라이브러리 불러오기
```

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

```
# 커넥션 드라이버 생성
```

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

```
# 연결 : scott 커넥션 열기, 쿼리 실행 및 프린트
```

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

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

커넥션 종료

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

문제 168. Emp 변수의 내용을 emp900.csv로 생성하시오.

```
> write.csv(emp, 'emp900.csv')
> write.csv(emp, '/home/oracle/emp900.csv')

[orcl:~]$ ls -l emp900.csv
-rw-r--r-- 1 oracle oinstall 976 Jun 29 16:15 emp900.csv
```

문제 169. Dept 테이블을 오라클에서 가져와서 dept900.csv로 생성하시오.

```
> dept <- dbGetQuery(scott,"SELECT * from dept")
> print(dept)
  DEPTNO  DNAME      LOC
1    10 ACCOUNTING NEW YORK
2    20  RESEARCH  DALLAS
3    30    SALES  CHICAGO
4    40 OPERATIONS  BOSTON
> write.csv(dept,'dept900.csv')

[orcl:~]$ ls -l dept900.csv
-rw-r--r-- 1 oracle oinstall 138 Jun 29 16:20 dept900.csv
```

문제 170. Scott이 가지고있는 테이블 리스트를 가져와 변수에 저장 하시오.

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

문제 171.	배포된 리눅스 설치 파일로 리눅스를 설치하고 설치 성공 마지막 화면과 자신의 호스트 이름을 캡처해서 붙여넣으세요
----------------	--

```
[root@BKH ~]# hostname
BKH
[root@BKH ~]# uname -an
Linux BKH 3.10.0-327.el7.x86_64 #1 SMP Thu Nov 19 22:10:57 UTC 2015 x86_64
x86_64 x86_64 GNU/Linux
[root@BKH ~]#
```


cd 명령어

2018년 6월 28일 목요일 오후 6:09

1. cd 명령어

사용방법	의미
cd (== cd ~)	자기 자신의 홈 디렉토리로 이동
cd /	루트 디렉토리(/)로 이동
cd ..	한 단계 상위 디렉토리로 이동
cd -	바로전에 위치했던 디렉토리로 이동
cd /경로 ..	해당 경로로 이동

■ 경로의 2가지 종류

1. 절대 경로 : cd 내가 가고자하는 정확한 위치
\$ cd /home/oracle/labs
2. 상대 경로
\$ cd .. : 나의 현재 위치를 상대로 상위 디렉토리로 이동하겠다.
\$ cd ./labs : 현재 디렉토리의 labs 디렉토리로 이동하겠다.

문제 1.	/home/oracle 경로로 이동 하시오.
-------	--------------------------

\$ cd ..

문제 2.	현재 디렉토리에서 /(루트) 디렉토리로 바로 이동 하시오.
-------	----------------------------------

\$ cd /

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

Cd - : 전에 있는 경로로 이동

Cd .. : 하위 경로로 이동

Cd [절대 경로] : 절대 경로로 이동

touch 명령어

2018년 6월 28일 목요일 오후 6:09

1. touch 명령어

" 용량이 0인 파일을 생성하거나 파일의 타임 스탬프를 변경하는 용도로 사용한다. "

Ex 1) 용량이 0인 빈 파일 생성하는 용도

```
$ touch a1.txt
```

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

```
[orcl:~]$ touch a1.txt
[orcl:~]$ ls
Adobe      R-3.2.3          a1.txt          labs
Desktop    R-3.2.3.tar.gz   datamodeler.log oradiag_oracle
[orcl:~]$ ls -l a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
```

권한 소유자 소유자가 속한 그룹 파일용량 생성날짜

Ex 2) 파일의 타임스탬프를 변경하는 용도

touch 파일명

(파일이 존재하는 경우 해당 파일의 타임 스탬프 값이 현재의 시간으로 업데이트 된다.)

```
[orcl:~]$ stat emp.txt
  File: `emp.txt'
  Size: 970          Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 16876427   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 500/  oracle)   Gid: ( 504/oinstall)
Access: 2018-06-25 15:32:08.000000000 +0900
Modify: 2018-06-22 11:34:10.000000000 +0900
Change: 2018-06-25 15:38:48.000000000 +0900
[orcl:~]$ touch emp.txt
[orcl:~]$ stat emp.txt
  File: `emp.txt'
  Size: 970          Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 16876427   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 500/  oracle)   Gid: ( 504/oinstall)
Access: 2018-06-25 19:49:17.000000000 +0900
Modify: 2018-06-25 19:49:17.000000000 +0900
Change: 2018-06-25 19:49:17.000000000 +0900
```

■ 리눅스 시스템의 파일 타임 스탬프 종류

종류	설명
atime	최종 접근시각(access time)
mtime	최종 수정시각(modify time)
ctime	최종 상태변경시각(change time)

문제 3.	현재 디렉토리에 아래의 파일들을 크기를 0으로 생성 하시오.
-------	-----------------------------------

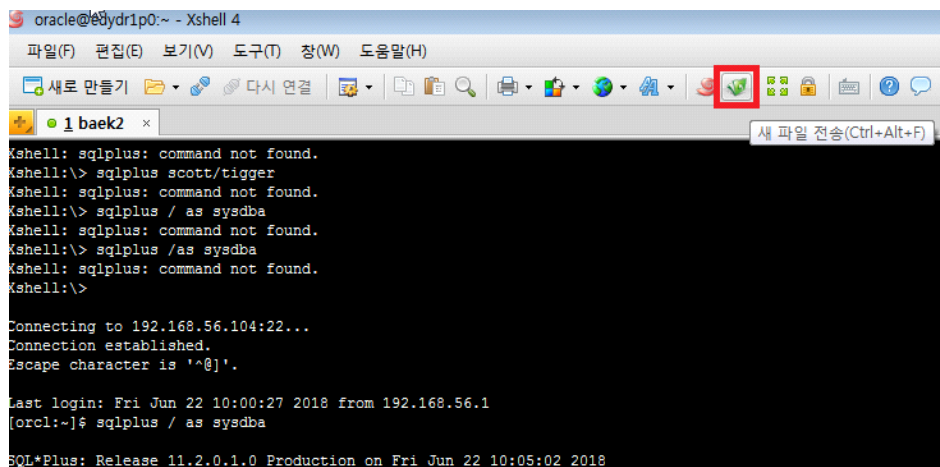
```

[orcl:~]$ touch a.txt
[orcl:~]$ touch b.txt
[orcl:~]$ touch c.txt
[orcl:~]$ touch d.txt
[orcl:~]$ touch e.txt
[orcl:~]$ touch f.txt
[orcl:~]$ ls -l *.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 a.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:12 a1.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 b.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 c.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:16 d.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 e.txt
-rw-r--r-- 1 oracle oinstall 0 Jun 22 10:17 f.txt

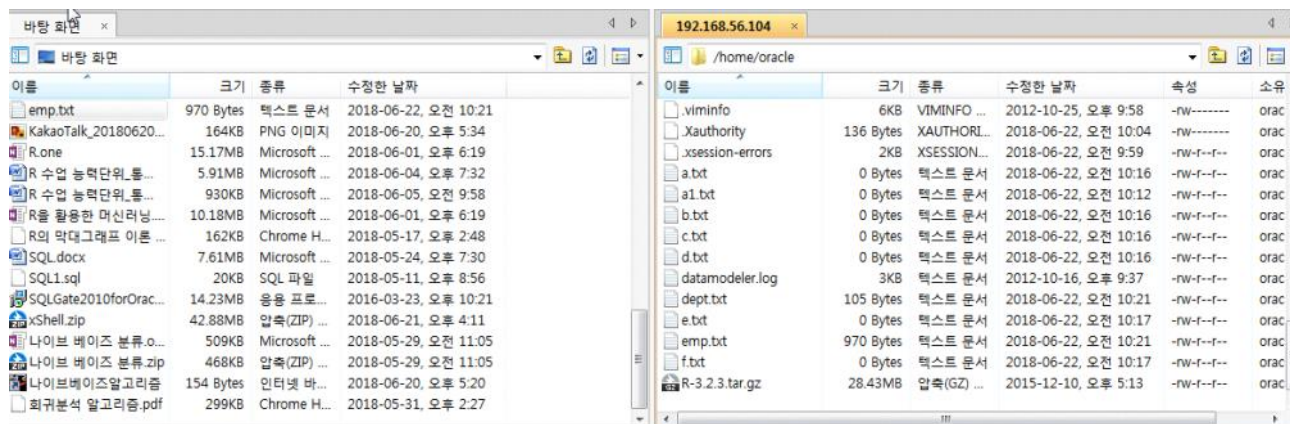
```

문제 4. 리눅스 서버에 /home/oracle 경로에 emp.txt 를 전송 하시오.

- a. Xshell에서 새 파일 전송 버튼을 누르고



- b. 윈도우에서 리눅스 경로로 파일을 드래그 해주면 된다.



li 명령어

2018년 6월 28일 목요일 오후 6:09

3. li 명령어

" 파일이나 디렉토리의 목록과 정보를 출력할 때 사용하는 명령어 "

옵션	설명
-a	숨겨진 파일을 포함해서 모든 내용을 출력한다.
-l	자세한 내용을 표시한다.
-r	내림차순 정렬 (기본은 오름차순 정렬)
-f	디스크에 저장된 순서대로 출력
-c	최근에 변경된 파일부터 정렬해서 출력
-R	하위 디렉토리 출력

mkdir 명령어

2018년 6월 28일 목요일 오후 6:09

mkdir 명령어

" 디렉토리를 생성하는 명령어 "

옵션	Long 옵션	설명
-m	--mode=MODE	디렉토리를 생성할 때 권한을 설정, 디폴트는 755
-p	--parents	상위 경로도 함께 생성
-v	--verbose	디렉토리를 생성 후, 생성된 디렉토리에 대해 메시지 출력

Ex 1) 디렉토리 생성할 때 권한 설정

```
$ mkdir -m 755 디렉토리명
```

Ex 2) 상위 경로와 함께 생성

```
$ mkdir -p /경로/디렉토리명
```

Ex 3) 디렉토리 생성 후 각각 성공, 실패 여부를 출력한다.

```
$ mkdir -v 디렉토리명1 디렉토리명2 ...
```

문제 5.	/home/oracle 경로에 /home/oracle/test5/test6/test7/test8/test9 디렉토리를 생성 하시오.
--------------	---

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

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

-p 옵션을 사용하면 한번에 상위-하위 디렉토리 생성 가능

문제 6.	현재 디렉토리에서 한번에 /home/oracle 로 이동 하시오.
--------------	--------------------------------------

```
[orcl:test6]$ cd /home/oracle
```

```
[orcl:test6]$ cd /home/oracle
[orcl:~]$ pwd
/home/oracle
```

문제 7.	
--------------	--

rm 명령어

2018년 6월 28일 목요일 오후 6:09

rm 명령어

" 파일이나 디렉토리를 삭제하는 명령어 "

* 리눅스나 유닉스는 휴지통이 없기 때문에 삭제할 때 특히 주의해야한다. (백업이 없으면 복구 불가능)

옵션	Long 옵션	설명
-f	--force	강제로 파일이나 디렉토리를 삭제하고, 삭제할 대상이 없을 경우 메시지를 출력하지 않음
-i	--interactive	매번 삭제할 때마다 사용자에게 질문함
-I		셋 이상의 파일을 삭제하거나 하위의 파일이나 디렉토리가 있을 경우 질문함
	-- interactive[=WHEN]	상호대화형 모드로 값(WHEN)을 지정함 WHEN 대신 once(-I 옵션과 같음)와 always(-i 옵션과 같으며 디폴트 값)가 올 수 있음
	--no-preserve-root	'/'를 특별하게 취급하지 않음
	--preserve-root	'/'를 삭제하지 않음(디폴트 값)
-r, -R	--recursive	하위 디렉토리를 포함하여 모든 내용을 삭제
-d	--dir	빈 디렉토리들만 제거
-v	--verbose	지워지는 파일의 정보를 출력
	--help	rm 명령어 사용법을 출력
	--version	rm 명령어의 버전 정보를 출력
-rf		사용자에게 확인을 하지 않고 즉시 하위 디렉토리까지 삭제

■ rmdir 명령어

" 디렉토리를 삭제하는 명령어 (옵션이 좋은 rm 명령어를 사용하는 것을 추천) "

옵션	설명
-p	상위 디렉토리 삭제

Ex. \$ rmdir 디렉토리명

\$ mkdir ddd

\$ ls

\$ rmdir ddd

alias 명령어

2018년 6월 28일 목요일 오후 6:09

1. alias 명령어

" 자주 수행하는 명령어들을 쉽게 사용할 수 있도록 설정하는 명령어 "

Ex. 오라클 db에 접속

기존 명령어 : `$ sqlplus / as sysdba`

새로운 명령어 : `$ alias sql = 'sqlplus / as sysdba'`
`$ sql`

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:15:47 2018
```

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

```
Connected to:
```

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

* alias 만 입력할 경우 현재 존재하는 별칭을 보여준다.

```
[orcl:~]$ alias
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

문제 11.	Scott 이라고 아래와 같이 명령어를 수행하면 오라클에 scott으로 접속되게 alias를 만드시오.
--------	---

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 11:18:16 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"
```

* 별칭 지우는 방법 : unalias sql

문제 12.	오라클의 네트워크 설정에 대한 디렉토리가 있는 곳으로 이동 하시오.
--------	---------------------------------------

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

문제 13.	오라클의 네트워크 설정에 대한 디렉토리가 있는 곳 바로 이동하는 별칭을 만드시오.
--------	---

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

[orcl:~]$ ora
[orcl:admin]$ ls
samples  shrept.lst  tnsnames.ora
[orcl:admin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
```


cat 명령어

2018년 6월 28일 목요일 오후 6:09

1. cat 명령어

" 파일의 내용을 화면에 출력하는 명령어 "

옵션	설명
-b	각 행에 번호를 붙여서 출력
-n	빈 행에도 번호를 붙여서 출력
-s	연속되는 2개 이상의 빈 행을 한 행으로 출력

Ex. \$ Cat 파일명.txt

\$ cat emp.txt

문제 14. Cat 으로 본 화면의 결과를 emp40.txt 라는 이름으로 저장 하시오.

```
$ cat emp.txt >> emp40.txt
```

```
[orcl:~]$ cat emp.txt >> emp40.txt
[orcl:~]$ ls
Adobe      R-3.2.3      datamodeler.log  emp.txt      labs          test5
Desktop    R-3.2.3.tar.gz  dept.txt         emp40.txt    oradiag_oracle
```

>> : 리다이렉션 redirection // R에서 %>% 랑 비슷

문제 15. 현재 디렉토리에 아래의 파일들을 touch 명령어로 생성 하시오.

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

```
$ li -l *.txt
```

문제 16. 지금 ls -l *.txt로 보고 있는 화면의 결과를 ls.txt라는 이름으로 저장 하시오.

```
[orcl:~]$ ls -l *.txt >> ls.txt
[orcl:~]$ ls
Adobe      R-3.2.3.tar.gz  c.txt      dept.txt  emp40.txt  oradiag_oracle
Desktop    a.txt           d.txt      e.txt     labs       test5
R-3.2.3    b.txt           datamodeler.log  emp.txt    ls.txt
```

redirection 명령어

2018년 6월 28일 목요일 오후 6:09

1. redirection 명령어

" 화면에 출력되는 결과를 파일로 저장하는 명령어 "

>> : 없으면 파일을 생성하고 있으면 기존 파일 뒤에 덧붙이겠다.

> : 파일을 생성하겠다. 기존 파일이 있으면 그냥 덮어 쓰겠다.

문제 17.	emp.txt의 내용을 emp50.txt 라는 이름으로 생성하고 emp.txt의 결과를 emp50.txt에 덧붙여서 저장 하시오.
---------------	--

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

wc 명령어

2018년 6월 28일 목요일 오후 6:09

1. wc 명령어

" 파일 안의 단어의 개수 또는 라인수를 출력 하는 명령어 "

옵션	설명
-l	라인수를 출력
-w	단어의 개수를 출력
-c	문자 수를 출력

Ex. \$ wc -l 파일명
\$ wc -l emp.txt

문제 18. 겨울 왕국 대본의 단어수가 어떻게 되는가?

```
[orcl:~]$ wc -c winter.txt
114548 winter.txt

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

라인수 , 단어수, 문자수

문제 19. /home/oracle 경로의 확장자가 .txt 로 끝나는 파일이 몇 개가 있는지 카운트 하시오.

```
[orcl:~]$ ls -l *.txt | wc -l
11
```

* |: **파이프 명령어** (앞의 명령어의 표준 출력을 뒤에 명령어의 표준 입력으로 보냄으로써 명령어의 실행 결과를 다음 명령어로 전달하는 기능

문제 20. 아래의 경로로 이동해서 아래의 디렉토리 아래있는 확장자 trc 파일이 총 몇 개가 있는지 확인 하시오.
/u01/app/oracle/diag/rdbms/orcl/orcl/trace

```
[orcl:~]$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
[orcl:trace]$ ls -l *.trc | wc -l
185
```

more 명령어

2018년 6월 28일 목요일 오후 6:09

1. more 명령어

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

■ 단축키

- h : more 명령어상태에서 사용할 수 있는 키 도움말 확인
- Space Bar : 한 화면씩 뒤로 이동하기 (f와 동일)
- Enter : 현재행에서 한 행씩 뒤로 이동하기
- q : more 명령어 종료하기
- f : 한 페이지씩 뒤로 이동하기(Space Bar 와 동일)
- b : 한 페이지씩 앞으로 이동하기
- = : 현재 위치의 행번호 표시하기
- /문자열 : 지정한 문자열을 검색하기
- n : /문자열로 검색한 문자열을 차례대로 계속해서 찾기
- !셸명령어 : more 명령어상태에서 셸명령어를 실행하기
- v : more 명령어로 열려있는 파일의 현재위치에서 vi를 실행하기

Ex. \$ more winter.txt

전진키 : space 키

후진키 : b

페이지 단위로 이동 : f

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

head / tail 명령어

2018년 6월 28일 목요일 오후 6:09

1. head / tail 명령어

Head : " 문서의 처음 몇 줄을 화면에 출력하는 명령어 "

옵션	설명
-n 숫자	앞에서 부터 숫자행까지 출력 (기본은 10행까지 출력)
-c 데이터 크기	앞에서 부터 데이터 크기(byte)만큼 출력

Ex. \$ head 출력줄수 파일명
\$ head -20 winter.txt

Tail : "문서의 마지막 몇 줄을 화면에 출력 하는 명령어 "

옵션	설명
-n 숫자	마지막 행 부터 숫자행까지 출력 (기본은 10행까지 출력)
-c 데이터 크기	마지막 행 부터 데이터 크기(byte)만큼 출력
-f	로그파일을 실시간 모니터링(파일을 실시간으로 읽을 수 있다), 종료는 Ctrl+c

Ex. \$ tail 출력줄 수 파일명
\$ tail -10 winter.txt

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

Ex. 오라클 db가 올라가거나 내려갈때의 그 정보를 적는 파일을 tail -f 로 여시오.

```
$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
```

```
[orcl:trace]$ tail -f alert_orcl.log
db_recovery_file_dest_size of 3852 MB is 4.26% used. This is a
user-specified limit on the amount of space that will be used by this
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Fri Jun 22 10:00:54 2018
Starting ORACLE instance (normal)
Fri Jun 22 10:05:03 2018
Starting background process SMC0
Fri Jun 22 10:05:03 2018
SMC0 started with pid=24, OS id=5755
```

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

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jun 22 14:26:51 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> shutdown immediate
```

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

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

```
ARCH: Archival disabled due to shutdown: 1089
```

```
Shutting down archive processes
```

```
Archiving is disabled
```

```
Archive process shutdown avoided: 0 active
```

```
Fri Jun 22 14:27:04 2018
```

```
Stopping background process VKTM:
```

```
Fri Jun 22 14:27:04 2018
```

```
NOTE: Shutting down MARK background process
```

```
Fri Jun 22 14:27:06 2018
```

예제. 현재 리눅스 시스템의 i/o 성능을 확인 하시오.

```
$ sar 1 100 ---> 1 ~100 동안 확인
```

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

악성 sql을 날리면 실시간으로 성능이 좋지 않은 것을 확인 할 수 있다.

```
[orcl:~]$ sar 1 100  
Linux 2.6.18-164.el5 (edydr1p0.us.oracle.com) 06/22/18  
  
14:33:16      CPU      %user      %nice      %system      %iowait      %steal      %idle  
e  
14:33:17      all      49.75      0.00      0.50      0.00      0.00      49.7  
5  
14:33:18      all      49.25      0.00      1.49      0.00      0.00      49.2  
5  
14:33:19      all      50.25      0.00      0.00      0.00      0.00      49.7  
5  
14:33:20      all      49.75      0.00      1.00      0.00      0.00      49.2  
5  
14:33:21      all      49.75      0.00      0.50      0.00      0.00      49.7  
5  
14:33:22      all      49.50      0.00      1.50      0.00      0.00      49.0  
0
```

Ctrl + c 를 눌러서 악성 sql을 종료한 후의 모습

```
[orcl:~]$ sar 1 100
Linux 2.6.18-164.el5 (edydrp0.us.oracle.com) 06/22/18

14:35:14      CPU      %user      %nice      %system      %iowait      %steal      %idle
14:35:15      all        0.50        0.00         1.49         0.00         0.00       98.01
14:35:16      all        0.00        0.00         1.50         0.00         0.00       98.50
14:35:17      all        0.00        0.00         1.50         0.00         0.00       98.50
14:35:18      all        0.50        0.00         2.50         0.00         0.00       97.00
14:35:19      all        0.00        0.00         1.50         0.00         0.00       98.50
```

문제 21. 겨울 왕국 대본의 100줄까지의 내용을 winter_head.txt 라는 이름으로 저장 하시오.

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

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

```
Sar 1 100 >> sar_0622.txt
# 현업에서는 시스템 부하 상태를 기록하기 위해 값을 많이 주고 항상 켜놓다 --> 백그라운드로
실행하는게 좋다 (끝수도 있으니까)
```

```
Sar 1 100 >> sar_0622.txt & # & : 백그라운드로 명령 실행
```

grep 명령어

2018년 6월 28일 목요일 오후 6:09

1. grep 명령어

" 파일안에 포함된 특정 단어나 구문을 검색하는 명령어 "

옵션	설명
-c	count 옵션은 파일 이름 다음에 패턴이 들어 있는 라인의 수를 표시
-h	hide 옵션은 파일이름을 출력하지 않음
-i	ignore 옵션은 대소문자 구분하지 않고 찾음
-n	line number 옵션은 출력하는 각 라인에 라인번호를 붙임
-v	inverse 옵션은 패턴이 없는 라인만 출력
-w	word 옵션은 패턴 자체가 완전한 단어로 독립되었을 경우만 글자로 인식함

Ex. \$ grep [옵션] '찾고싶은단어' 파일명

\$ grep -i 'scott' emp.txt # emp.txt에서 'scott'이란 단어가 포함된 라인을 보여달라.

-i : 대소문자를 구분하지 않겠다.

문제 23. 직업이 salesman인 직원들의 모든 행을 출력 하시오.

```
[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 이름, 월급을 출력 하시오.

```
[orcl:~]$ grep -i 'salesman' emp.txt | awk '{print $2,$6}'
ALLEN 1600
WARD 1250
MARTIN 1250
TURNER 1500
```


awk 명령어

2018년 6월 28일 목요일 오후 6:14

1. awk 명령어

" 특정 단어가 들어가 있는 라인에서 특정 컬럼명을 출력할 때 사용하는 명령어 "

패 턴	내 용
BEGIN	입력화일을 읽어들이기 전에 옆에 제시되는 문자를 실행시키도록 한다.
END	awk가 모든 입력을 처리한 후, 옆에 제시되는 문장을 실행시키도록 한다.
expression (식)	식을 평가한 후 이 식이 참, 즉 non-zero이거나 non-null인 경우 문장을 실행한다.
/re/	정규식과 일치하는 문자열을 포함하고 있는 라인에서 문장을 실행한다.
compound- pattern	복합패턴이라는 것으로 &&(and), (or) , !(not) 그리고 괄호에 의해 연결시킨 것이다. expression의 경우와 마찬가지로 복합 패턴도 참인 경우의 문장을 실행시킨다.
pattern1, pattern2	이러한 패턴을 범위 패턴이라한다. 현재 처리되고 있는 라인이 pattern1과 일치되고, 다음에 따라오는 라인 중 임의의 라인이 pattern2와 일치할 때, 범위 패턴은 두 라인 사이의 각 라인과 일치한다.

Ex. \$ awk '패턴(action)' 대상파일명

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

\$ awk '\$3 == "salesman" {print \$2, \$3}' emp.txt

```
[orcl:~]$ awk '$8 == 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
```

문제 25. 부서번호가 10번인 직원들의 이름, 월급을 출력 하시오.

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

-w : 정확히 매치 하는것만 보여줌 (10이라고 검색했을때 1000 은 뜨지 않고 10일때만 뜸)

문제 26. 월급이 3000 이상인 직원들의 이름, 월급을 출력 하시오.

```
[orcl:~]$ awk '$6>=3000 {print $2, $3}' emp.txt
SCOTT ANALYST
KING PRESIDENT
FORD ANALYST
```

* 리눅스의 연산자 3가지

- i. 산술연산자 : +,0,*,/
- ii. 비교연산자 : >,<,>=,<=,==,!=
- iii. 논리연산자 : &&, || , !

문제 27. 직업이 salesman이 아닌 직원들의 이름, 직업을 출력 하시오.

```
[orcl:~]$ awk '$3 != toupper("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
```

문제 28. 직업이 salesman인 직원들의 이름, 월급을 출력 하는데 월급이 1500 이상인 직원들만 출력 하시오.

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

문제 29. 이름 A로 시작하는 직원들의 이름과 월급을 출력 하시오.
힌트 : grep과 정규식 (시작 : ^ , 끝 : \$)을 사용하고 awk, grep과 |를 활용해서 출력

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

문제 30. 81년도에 입사한 직원들의 이름, 입사일을 출력 하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep '81$'
ALLEN 20-FEB-81
WARD 22-FEB-81
JONES 02-APR-81
MARTIN 28-SEP-81
BLAKE 01-MAY-81
CLARK 09-JUN-81
KING 17-NOV-81
TURNER 08-SEP-81
JAMES 03-DEC-81
FORD 03-DEC-81
```

문제 31. 81년도에 입사하지 않은 직원들의 이름, 입사일을 출력 하시오.

```
[orcl:~]$ awk '{print $2,$5}' emp.txt | grep -v 81
SMITH 17-DEC-80
SCOTT 19-APR-87
ADAMS 23-MAY-87
MILLER 23-JAN-82
```

문제 32. 겨울왕국 대본에서 elsa가 포함되어있는 라인들의 개수가 몇 개인지 출력 하시오.

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

* 리눅스에서 R을 쉽게 실행 하는 방법

```
[orcl:~]$ ls
Adobe          b.txt          e.txt          ls.txt          winter_head.txt
Desktop        c.txt          emp.txt        oradiag_oracle
R-3.2.3        d.txt          emp40.txt      sar_0622.txt
R-3.2.3.tar.gz datamodeler.log emp50.txt      test5
a.txt          dept.txt       labs           winter.txt
[orcl:~]$ vi .bash_profile
```

/home/oracle 경로에서 vi .bash_profile을 열어 명령어를 추가해준다.

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

# .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=$ORACLE_HOME/bin:$PATH
export PATH=/home/oracle/R-3.2.3/bin:$PATH
unset LANG

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

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i686-pc-linux-gnu (32-bit)
```

sort 명령어

2018년 6월 28일 목요일 오후 6:14

1. sort 명령어

" data를 특정 컬럼을 기준으로 정렬하는 명령어 "

sort [-옵션] [-o 저장될 파일명] 정렬할 파일명 [병합할 파일명....]

옵션	설명
-n	라인의 각 필드를 비교하는 대상을 숫자로 한정
-f	영어를 정렬할 때, 대소문자 구별 안함
-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
-r	출력 순서를 역순으로

사용 예

우선 sort 명령을 확인해보기 위해서 다음과 같이 sort.txt 파일을 생성했다.

```
[naddru@ ~]$ cat sort.txt
banana 3
camera 1
apple 4
date 2
```

그 후, 기본적인 sort 명령을 사용해 보면 다음과 같은결과가 나타난다.

sort [파일명]

```
[naddru@ ~]$ sort sort.txt
apple 4
banana 3
camera 1
date 2
```

만약 **두번째 필드를 기준으로 정렬**하고 싶다면,

sort +1 [파일명]

```
[naddru@ ~]$ sort +1 sort.txt
camera 1
date 2
banana 3
apple 4
```

마지막으로 **파일에 저장**하고 싶다면 -o 옵션을 사용한다.

sort [파일명] -o [저장할 파일명]

예제 : \$ sort 옵션 파일명

\$ sort -k 6 emp.txt

-k : 오름차순 , -rk : 내림차순

Emp.txt 의 모든 데이터를 출력하는데 6번째 컬럼 (월급)이 적은 것부터 출력 하시오.

문제 34. 위의 결과를 다시 출력하는데 이름, 월급만 나오게 하시오.

```
[orcl:~]$ sort -k 6 emp.txt | awk '{print $2,$6}'
SMITH 800
JAMES 950
ADAMS 1100
WARD 1250
MARTIN 1250
MILLER 1300
TURNER 1500
ALLEN 1600
CLARK 2450
BLAKE 2850
JONES 2975
SCOTT 3000
FORD 3000
KING 5000
```

문제 35. 이름, 월급을 출력하는데 월급이 높은 것부터 출력 하시오.

```
[orcl:~]$ sort -rk 6 emp.txt | awk '{print $2,$6}'
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 이름, 월급을 출력하는데 월급이 높은 것부터 출력 하시오.

```
[orcl:~]$ sort -k 6 emp.txt | awk '$3==toupper("salesman") {print $2,$6}'
WARD 1250
MARTIN 1250
TURNER 1500
ALLEN 1600
```

문제 37. 직원 테이블의 월급의 토탈 값을 출력 하시오.

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

문제 38. 직업이 SALESMAN인 직원들의 토탈 월급을 출력 하시오.

```
[orcl:~]$ awk '$3==toupper("salesman") {print $6}' emp.txt | awk '{sum+=$1} END {print sum}'
5600
```

문제 39. 아래와 같이 결과가 출력되게 하시오.

King 의 월급은 5000입니다. ...

```
[orcl:~]$ awk '{print $2 "의 월급은 " $6 "입니다."}' emp.txt
SMITH의 월급은 800입니다.
ALLEN의 월급은 1600입니다.
WARD의 월급은 1250입니다.
JONES의 월급은 2975입니다.
MARTIN의 월급은 1250입니다.
BLAKE의 월급은 2850입니다.
CLARK의 월급은 2450입니다.
SCOTT의 월급은 3000입니다.
KING의 월급은 5000입니다.
TURNER의 월급은 1500입니다.
ADAMS의 월급은 1100입니다.
JAMES의 월급은 950입니다.
FORD의 월급은 3000입니다.
MILLER의 월급은 1300입니다.
```

문제 40. 아래와 같이 결과가 출력되게 하시오.

King 의 직업은 manager입니다. ...

```
[orcl:~]$ awk '{print $2 "의 직업은 " $3 "입니다."}' emp.txt
SMITH의 직업은 CLERK입니다.
ALLEN의 직업은 SALESMAN입니다.
WARD의 직업은 SALESMAN입니다.
JONES의 직업은 MANAGER입니다.
MARTIN의 직업은 SALESMAN입니다.
BLAKE의 직업은 MANAGER입니다.
CLARK의 직업은 MANAGER입니다.
SCOTT의 직업은 ANALYST입니다.
KING의 직업은 PRESIDENT입니다.
TURNER의 직업은 SALESMAN입니다.
ADAMS의 직업은 CLERK입니다.
JAMES의 직업은 CLERK입니다.
FORD의 직업은 ANALYST입니다.
MILLER의 직업은 CLERK입니다.
```

문제 41. /home/oracle 에서 ls -lrt 라고 조회한 결과를 분석 하시오.

\$pwd

\$ls -lrt

-l : 리스트

-r : 리버스

-t : 시간

----> 파일 리스트를 옛날 것 부터 보여달라 !

```
[orcl:~]$ ls -lrt
total 29336
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
-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 16:55 R-3.2.3
drwx----- 2 oracle oinstall      4096 Jun 22 09:52 Desktop
drwxr-xr-x  3 oracle oinstall      4096 Jun 22 10:28 test5
-rw-r--r--  1 oracle oinstall       970 Jun 22 11:34 emp.txt
-rw-r--r--  1 oracle oinstall       105 Jun 22 11:34 dept.txt
-rw-r--r--  1 oracle oinstall      1940 Jun 22 11:34 emp40.txt
-rw-r--r--  1 oracle oinstall         0 Jun 22 11:37 e.txt
-rw-r--r--  1 oracle oinstall         0 Jun 22 11:37 d.txt
-rw-r--r--  1 oracle oinstall         0 Jun 22 11:37 c.txt
-rw-r--r--  1 oracle oinstall         0 Jun 22 11:37 b.txt
-rw-r--r--  1 oracle oinstall         0 Jun 22 11:37 a.txt
-rw-r--r--  1 oracle oinstall       433 Jun 22 11:38 ls.txt
-rw-r--r--  1 oracle oinstall      3880 Jun 22 11:43 emp50.txt
-rw-r--r--  1 oracle oinstall     114548 Jun 22 11:47 winter.txt
-rw-r--r--  1 oracle oinstall      3387 Jun 22 14:18 winter_head.txt
-rw-r--r--  1 oracle oinstall      9790 Jun 22 14:45 sar_0622.txt
```

문제 42. ls -lrt *.txt 했을 때 파일 크기에 해당하는 부분만 출력 하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{print $5}'
970
105
1940
0
0
0
0
0
0
433
3880
114548
3387
9790
```

문제 43. 위에서 출력된 숫자들을 모두 sum 해서 출력 하시오.

```
[orcl:~]$ ls -lrt *.txt | awk '{sum+= $5 } END {print sum}'
135053
```

문제 44. 아래와 같이 결과를 출력되게 하시오.

현재 디렉토리에 있는 txt 파일들의 총 크기는 135053 바이트 입니다.

```
[orcl:~]$ ls -lrt *.txt | awk '{sum+= $5 } END {print "현재 디렉토리에 있는 txt 파일들의 총 크기는 "sum "바이트 입니다"}'
현재 디렉토리에 있는 txt 파일들의 총 크기는 135053바이트 입니다
```

문제 45. 위의 스크립트를 쉘 스크립트로 저장해서 실행되게 하시오.

* 자주 쓰는 명령어를 쉘 스크립트로 저장해 뒀다가 사용한다.

```
ls -lrt *.txt | awk '{sum+= $5 } END {print "현재 디렉토리에 있는 txt 파일들의 총 크기는 "sum "바이트 입니다"}'
```

```
[orcl:~]$ vi size.sh
[orcl:~]$ sh size.sh
현재 디렉토리에 있는 txt 파일들의 총 크기는 135053바이트 입니다
```


uniq 명령어

2018년 6월 28일 목요일 오후 6:14

1. uniq 명령어

" 중복된 라인을 제거하는 명령어 "

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 라인까진 검사하지 않는다
-i	중복 라인을 한라인으로 생각하고 출력한다.
-u	중복 라인이 없는 것만 보여준다.
-w	N 번째 문자까지만 비교대상으로 하여 unique 명령을 수행
-s	N 번째 문자까지는 비교 대상에서 제외하고 unique 명령을 수행
-f	N 번째 필드를 비교대상에서 제외하고 unique 명령을 수행

Ex. \$ uniq [옵션] 파일명

문제 46. emp.txt에서 직업만 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK
```

문제 47. 위의 결과를 abcd 순으로 정렬해서 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort
ANALYST
ANALYST
CLERK
CLERK
CLERK
CLERK
MANAGER
MANAGER
MANAGER
PRESIDENT
SALESMAN
SALESMAN
SALESMAN
SALESMAN
```

* 내림 차순

```
[orcl:~]$ awk '{print $3}' emp.txt | sort -rf
SALESMAN
SALESMAN
SALESMAN
SALESMAN
PRESIDENT
MANAGER
MANAGER
MANAGER
CLERK
CLERK
CLERK
CLERK
ANALYST
ANALYST
```

문제 48. 위의 결과를 중복을 제거해서 출력 하시오.

```
[orcl:~]$ awk '{print $3}' emp.txt | sort | uniq
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

문제 49. 부서번호를 출력하는데 중복 제거해서 출력 하시오.

```
[orcl:~]$ awk '{print $8}' emp.txt | sort | uniq
10
20
30
```

문제 50. Emp.txt 를 복사해서 emp7.txt를 생성하고 emp7.txt에 king에 대한 데이터를 복사해서 맨 아래에 하나를 입력하고 저장 하시오.

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

복사할 로우에 커서를 위치한 후 YY를 눌러서 복사하고 맨 아래로우로 커서를 옮긴 후 P를 눌러서 붙여넣는다.

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

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

문제 51. emp7.txt 에서 중복된 라인만 출력 하시오.

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

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

문제 52. emp7.txt 에서 중복된 라인을 제거하고 출력 하시오.

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

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

중복된 King이 하나도 출력되지 않는다.

echo 명령어

2018년 6월 28일 목요일 오후 6:14

1. echo 명령어

" 출력하고자 하는 글자를 출력할 때 사용하는 명령어 (print 함수와 비슷한 역할 --> 주로 스크립트에 사용)"

옵션	설명
-n	마지막에 따라오는 개행 문자를 출력하지 않음 (줄 바꾸지 않음)
-e	문자열 중에 백슬래쉬와 조합되는 특수 문자를 인식하게 된다.

이스케이프 문자	설명
\a	경고음 소리 출력
\b	백스페이스
\c	마지막 개행 문자를 출력 안 함
\f	폼 피드 형식(프린터에서 용지 바꿈)으로 출력
\n	개행 문자 출력
\r	개리지 리턴
\t	수평 탭
\v	수직 탭
\w	역슬래시
\nnn	ASCII Code 가 nnn(8진수)인 문자

Ex. \$ echo '나는 나다'

문제 53.	직업을 입력 받아 해당 직업을 갖는 직원들의 이름, 직업이 출력되게 하시오.
--------	--

#!/bin/bash 리눅스 스크립트 중에 shell 스크립트를 사용 하겠다.

```
#!/bin/bash
echo "
echo -n "직업을 입력하세요 ~~~"
read job
echo "
grep -i $job emp.txt | awk '{print $2, $3}'
```

echo -n 옵션을 안주면 줄 바뀜

```
직업을 입력하세요 ~~~
salesman
```

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

직업을 입력하세요 ~~~SALESMAN

ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
[orcl:~]$ vi job2.sh
[orcl:~]$ sh job2.sh

직업을 입력하세요 ~~~salesman

ALLEN SALESMAN
WARD SALESMAN
MARTIN SALESMAN
TURNER SALESMAN
```

문제 54. 부서번호를 입력 받아 해당 부서번호에 해당되는 직원들의 이름, 월급, 부서번호가 출력되게 하시오.

```
[orcl:~]$ vi deptno2.sh
[orcl:~]$ sh deptno2.sh

부서번호를 입력 하시오 : 20

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

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

문제 55. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 월급이 아래와 같이 출력되게 하시오.
scott의 월급은 3000 입니다.

```
echo " "
echo -n "이름을 입력 하시오 : "
read ename
echo " "
awk ' {print $2 "의 월급은 " $6 "입니다."}' emp.txt | grep -iw $ename
```

```
[orcl:~]$ sh ename2.sh

이름을 입력 하시오 : scott

SCOTT의 월급은 3000입니다.
```

문제 56. 직업을 물어보게 하고 직업을 입력하면 해당 직업의 토탈 월급이 아래와 같이 출력되게 하시오.

```
echo "      "
echo -n "직업 입력하세요 ~ "
read job
echo "      "

grep -iw $job emp.txt | awk '{sumsal += $6} END {print "토달월급은" sumsal "입니다." }'
```

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

직업 입력하세요 ~ salesman

토달월급은5600입니다.
```

문제 57. 인사일을 출력하는데 뒤에 년도만 잘라서 출력 하시오.

* awk {} 괄호 안에 substr 함수 사용가능 !

```
[orcl:~]$ awk '{print substr($5,8,10)}' emp.txt
80
81
81
81
81
81
81
87
81
81
87
81
81
82
```

문제 58. 년도를 물어보게 하고 년도를 입력하면 해당 년도에 입사한 직원들의 이름, 입사일이 출력되게 하시오.

```
echo "      "
echo -n "년도를 입력하세요 : "
read year

awk 'substr($5,8,10)=="$year" {print $2, $5}' emp.txt
```

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

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

문제 59. 년도를 물어보게 하고 년도를 입력하면 해당 년도의 토달 월급이 출력되게 하시오.

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

```
년도를 입력하세요 : 81
토탈월급은 22825 입니다.
```

```
echo "      "
echo -n "년도를 입력하세요 : "
read year
```

```
awk 'substr($5,8,10)=="$year" {sum += $6} END {print "토탈월급은 " sum " 입니다."}' emp.txt
```

diff 명령어

2018년 6월 28일 목요일 오후 6:14

1. diff 명령어

" 두 파일간의 차이점을 찾아서 알려주는 명령어 "

옵션	설명
-b	연속된 공백 무시
-i	대소문자를 구분하지 않음
-t	출력 라인에 TAB 문자를 넣음
-w	두 행의 비교 시 공백을 무시
--brief	파일 비교 후 결과 표시 (같은지, 다른지)
-d	세세한 차이까지 검색
-H	큰 파일을 빠르게 처리할 때 사용
-q	두 파일의 차이점만 출력
-s	두 파일이 같은지 확인
-r	두 디렉토리의 차이점 출력
-N	검색 중 new file도 적용 가능

Ex. \$ diff emp.txt emp7.txt

문제 60. 아래와 같이 두개의 파일 이름을 각각 물어보게 하고 결과로 파일의 내용의 차이가 출력되게 하시오.

```
echo "      "
echo -n "비교할 첫 번째 파일을 입력하세요."
read f1
echo -n "비교할 두 번째 파일을 입력하세요."
read f2

diff $f1 $f2

[orcl:~]$ vi diff.sh
[orcl:~]$ sh diff.sh

비교할 첫 번째 파일을 입력하세요.emp.txt
비교할 두 번째 파일을 입력하세요.emp7.txt
14a15
>      7839 KING      PRESIDENT      0 17-NOV-81      5000      0      10
```


find 명령어

2018년 6월 28일 목요일 오후 6:16

1. find 명령어

" 검색하고 하는 파일을 찾을 때 사용하는 명령어 "

옵션	설명
-empty	비어 있는 파일, 단 일반파일 또는 일반 디렉토리에 한함
-uid n	지정된 n값의 UID를 갖는 파일
-gid n	지정된 n값의 GID를 갖는 파일
-group 그룹명	지정한 그룹명에 소속된 파일
-name	지정된 형식의 패턴을 가지는 파일
-newer 파일	지정한 파일보다 더 최근에 변경된 적이 있는 파일
-used n	파일이 변경된 이후 최근 n일 이후에 액세스된 적이 있는 파일
-user uname	지정한 uname의 소유로 된 모든 파일(UID로 지정가능)
-amin n	최근 n분 이전에 액세스된 적이 있는 파일
-anewer 파일	지정한 파일보다 더 최근에 액세스된 적이 있는 파일
-atime n	최근 n*24 시간 전에 액세스된 적이 있는 파일
-mmin n	최근 n분 이전에 변경된 적이 있는 파일
-mtime n	최근 n*24 시간 이전에 변경된 적이 있는 파일
- executa ble	실행 가능한 파일
- readable	읽기 가능한 파일
-writable	쓰기 가능한 파일
-fstype type	지정한 파일시스템타입(type)에 존재하는 파일
- nogroup	소속그룹이 존재하지 않는 파일
-nouser	소속 사용자(ID)가 존재하지 않는 파일
-path 패 턴	파일명이 패턴에 매칭되는 파일
-perm 모드	모드에서 지정된 퍼미션과 일치하는 퍼미션을 가진 파일만 해당됨.
-perm - mode	모드에서 지정된 퍼미션을 가진 모든 파일.
-regex 패턴	패턴에서 지정한 정규표현식에 매칭되는 파일

-size n[cwbk MG]	지정된 n 단위의 디스크공간을 사용하는 파일. 디폴트는 b
-type 타 입	지정한 파일타입에 해당하는 파일. b,c,d,p,f,l,s 등이 있음

Ex. \$ find 디렉토리 -name 파일명 -print

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

설명 : /home/oracle 경로에 emp.txt라는 파일이 있는지 검색 하시오.

문제 61. /home/oracle 경로에 test100과 test200 이라는 디렉토리를 생성하고 그 밑에 emp.txt를 각각 복사하시오.

```
[orcl:~]$ mkdir test100 test200
[orcl:~]$ ls
Adobe          a.txt  datamodeler.log  e.txt      emp7.txt      labs          size.sh  winter.txt
Desktop        b.txt  dept.txt         emp.txt     ename2.sh     ls.txt        test100  winter_head.txt
R-3.2.3        c.txt  deptno2.sh       emp40.txt  job2.sh       oradiag_oracle test200  year.sh
R-3.2.3.tar.gz d.txt  diff.sh          emp50.txt  job_sumsal.sh sar_0622.txt  test5   year_sum.sh
[orcl:~]$ cp emp.txt ./test100/emp.txt
[orcl:~]$ cp emp.txt ./test200/emp.txt
```

문제 62. /home/oracle 밑에 있는 emp.txt를 검색 하시오.

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

----> 현재 디렉토리 안에 있는게 모두 검색됨

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

-----> maxdepth 옵션을 줘서 몇 번째 하위 디렉토리까지 검색할지 정할 수 있다.

문제 63. 파일을 검색하는 쉘 스크립트를 아래와 같이 생성 하시오.

```
echo " "
echo -n "현재 경로에서 찾을 파일명을 입력 하시오. : "
read f_name
echo -n "depth 를 입력하세요 : "
read dpth

find . -maxdepth $dpth -name $f_name

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

현재 경로에서 찾을 파일명을 입력 하시오. : emp.txt
depth 를 입력하세요 : 2
./test100/emp.txt
./test200/emp.txt
./emp.txt
[orcl:~]$
```

tar 명령어

2018년 6월 28일 목요일 오후 6:16

1. tar 명령어

" 파일을 압축하고 압축 해제하는 명령어 "

옵션	설명
-c	compress, 여러 개의 파일을 하나로 만들어라 (tar로 묶음)
-p	파일 권한을 저장
-v	view, 묶거나 파일을 풀 때 과정을 화면으로 출력
-f	file, 생성되는 파일명을 지정
-C	압축이 풀릴 위치를 지정
-x	tar 압축을 풀
-z	gzip으로 압축하거나 해제함

- Ex. 1. 압축할 때 : \$ tar cvf 압축파일명 압축파일대상
2. 압축 풀 때 : \$ tar xvf 압축파일명 압축 해제할 대상

문제 64.	/home/oracle 밑에 test64 이라는 디렉토리를 만들고 그 밑에 emp.txt를 복사해서 아래의 파일들을 생성 하시오. a.txt, b.txt, c.txt, d.txt, e.txt
--------	---

```
[orcl:~]$ mkdir test64
[orcl:~]$ cp emp.txt ./test64/a.txt
[orcl:~]$ cp emp.txt ./test64/b.txt
[orcl:~]$ cp emp.txt ./test64/c.txt
[orcl:~]$ cp emp.txt ./test64/d.txt
[orcl:~]$ cp emp.txt ./test64/e.txt
```

문제 65.	Test64 디렉토리 밑에 있는 확장자 txt 파일들을 all.tar라는 이름으로 압축 하시오.
--------	---

```
[orcl:~]$ tar -cvf txttar.tar ./test64/*.txt
./test64/a.txt
./test64/b.txt
./test64/c.txt
./test64/d.txt
./test64/e.txt
```

문제 66.	/home/oracle 밑에 test66이라는 디렉토리를 생성하고 그 밑에 all.tar 압축파일의 압축을 해제하시오.
--------	--

```
[orcl:~]$ tar -xvf all.tar -C ./test66
./test64/a.txt
./test64/b.txt
./test64/c.txt
./test64/d.txt
./test64/e.txt
```

```

[orcl:~]$ ls
Adobe          b.txt          diff.sh        ename2.sh      oradiag_oracle test64          year_sum.sh
Desktop        c.txt          e.txt          find_file.sh   sar_0622.txt   test66
R-3.2.3        d.txt          emp.txt        job2.sh        size.sh        txttar.tar
R-3.2.3.tar.gz datamodeler.log emp40.txt      job_sumsal.sh test100         winter.txt
a.txt          dept.txt       emp50.txt     labs           test200        winter_head.txt
all.tar        deptno2.sh     emp7.txt      ls.txt         test5          year.sh

[orcl:~]$ cd test66
[orcl:test66]$ ls
test64
[orcl:test66]$ cd test64
[orcl:test64]$ ls
a.txt  b.txt  c.txt  d.txt  e.txt

```

In 명령어

2018년 6월 28일 목요일 오후 6:16

1. ln(link) 명령어

" 윈도우 바로가기 기능과 유사함 "

" 내가 자주 열어봐야하는 파일이 있다면 바로가기를 생성해 놓으면 편하게 열어볼 수 있다. "

■ 심볼릭링크 (Symbolic Link)

단순히 원본파일을 가리키도록 링크만 시켜둔 것으로 MS의 윈도우시스템에서 흔히 사용하는 '바로가기' 같은 것이며, 원본파일을 가리키고만 있으므로 원본파일의 크기와는 무관한다. 그리고 심볼릭링크에서는 원본파일이 삭제되어 존재하지 않을 경우에 링크파일은 깜박거리면서 링크파일의 원본파일이 없다는 것을 알려준다.

■ 하드링크 (Hard Link)

원본파일과 다른 이름으로 존재하는 동일한 파일이며 원본파일과 동일한 내용의 다른 파일이라고 할 수 있다. 그리고 하드링크에서는 원본파일과 링크파일 두개가 서로 다른 파일이기 때문에 둘 중 하나를 삭제 하더라도 나머지 하나는 그대로 남아 있다. 또한 하드링크에서는 원본파일의 내용이 변경될 경우에는 링크파일의 내용 또한 자동으로 변경된다.

옵션	설명
-- backup[= CONTRO L]	대상파일이 이미 존재할 경우에 백업파일을 만든 후에 링크파일 생성
-b	링크파일 생성시에 대상 파일이 이미 존재하면 백업파일을 만든 후에 링크 파일을 생성
-d	디렉토리에 대한 하드링크 파일생성을 가능하게 함. 단 root 권한으로 수행하더라도 시스템의 권한 제한으로 인하여 실패할 가능성이 높음
-f	대상파일이 존재할 경우에 대상파일을 지우고 링크파일을 생성
-i	대상파일이 존재할 경우에 대상파일을 지울것인가를 확인요청 (--interactive 와 동일)
-s	심볼릭 링크파일을 생성
-S	백업파일 생성시에 원하는 접미사(suffix)를 지정할 수 있음
-t	링크파일을 생성할 디렉토리를 지정

Ex. \$ ln -s "링크를 걸 파일위치와 파일명" "바로가기 이름"

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

```
[orcl:test64]$ ln -s /u01/app/oracle/product/11.2.0/dbhome_1/sqlplus/admin/glogin.sql glogin.sql
[orcl:test64]$ glogin.sql
-bash: glogin.sql: command not found
[orcl:test64]$ view glogin.sql
```

.bash_profile 설정하는 방법

.bash_profile 이란 ? Oracle 유저를 위한 리눅스 환경설정 파일

~/bashrc (bash run control)

bash를 사용할 때, bash가 참고할 사항을 정의해 놓는 파일이다.

환경변수(JAVA_HOME 등)나 Alias(ls -al --> ll 등)등을 미리 정의해 놓고 사용할 수 있도록 한다.

~/bash_profile

.bash_profile은 bash를 login shell로 사용할 때 참조된다. 즉 **bash로 로그인하면 bash_profile에 정의되어 있는 걸 사용할 수** 있다. .bash_profile에서 보통 .bashrc를 부르기 때문에 .bashrc에 있는건 자동으로 참조 된다.

리눅스 GUI에서 터미널을 열어 bash를 사용할 땐, .bash_profile이 참조 안 될 수 있으니 주의해야 한다.

/etc/bashrc

~/bashrc가 개인용 환경 설정이라면 /etc/bashrc는 전체용 환경 설정이다.

/etc/profile

bashrc와 마찬가지로 시스템 전체 환경 설정 파일이다. (로그인시 수행되는...)

* 위에서 커맨드 셸이란?

도스에서 command.com 셸이 사용자의 명령어(DIR 같은..)를 받아 특정 행동을 수행해주는 것 과 같이 사용자의 특정 명령어를 받아 OS한테 요로요리 한걸 해주세요라고 대신 말 해주는 놈이라 보면 된다.

* 참고로 shell에서 export 명령어는 환경변수를 지정하는 명령어이다.

Ex. vi .bash_profile

맨 아래쪽에 아래의 내용을 적고 저장하고 나온다.

Echo -e "여기는 운영서버 입니다. 조심히 작업 하세요 ~"

\$ su - oracle

```
# .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=$ORACLE_HOME/bin:$PATH
export PATH=/home/oracle/R-3.2.3/bin:$PATH
unset LANG

Echo -e "여기는 운영 서버 입니다. 조심히 작업하세요 ~"

[orcl:~]$ su - oracle
Password:
여기는 운영 서버 입니다. 조심히 작업하세요 ~
```

문제 67.	Oracle 유저로 접속할 때마다 아래의 메시지가 출력되게 하시오. 이랏샤이 마세이~~
--------	---

```
echo -e "이랏샤이 마세이 ~~"
```

```
[orcl:~]$ su - oracle  
Password:  
이랏샤이 마세이 ~~
```

sed 명령어

2018년 6월 28일 목요일 오후 6:16

1. sed 명령어

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

Ex. \$ sed 's/KING/yyy/' emp.txt

----> emp.txt 를 출력할 때 KING을 YYY로 변경해서 출력한다.

■ 명령어

aW : 현재 행에 하나 이상의 새로운 행을 추가한다.
cW : 현재 행의 내용을 새로운 내용으로 바꾼다.
iW : 현재 행의 위에 삽입한다.
d : 행을 삭제한다.
l : 출력되지 않는 특수문자를 출력한다.
p : 행을 출력한다.
n : 다음 입력 행을 첫 번째 명령어가 아닌 다음 명령어에서 처리하게 한다.
q : sed를 종료한다.
r : 파일로부터 행을 읽어온다.
! : 선택된 행을 제외한 나머지 전체 행에 명령어를 적용한다.
s : 문자열을 치환한다.

■ 옵션

-e : 다중 편집시 사용.
-f : sed 스크립트 파일 지정에 사용.
-n : 출력을 생략한다.

■ 정규식

- []: 한문자 매치
- ^: 라인 맨앞
- \$: 라인 끝
- &: 검색문자열 치환
- \w<: 단어의 시작
- >: 단어의 끝
- x{m,n}: m,n 구간 반복

■ 연산자

- [범위]/p: 지정 범위 출력
- [범위]/d: 지정 범위 삭제
- [범위]/문자1/문자2: 지정된 범위의 처음 나타난 문자를 문자2로
- s/문자1/문자2/: 문자1을 2로
- g: 모든 라인에 적용

■ 예제

*특정라인 출력

sed -n '3,7p' <파일> # <파일>에서 3~7라인만 출력

*특정라인 제외 출력

sed '3,7d' <파일> # 3~7라인 제외하고 출력

*특정라인(비연속) 출력

sed -n -e '3,5p' -e '7,9p' <파일> # 3~5, 3~7라인만 출력

*치환하여 출력

sed 's/1/2/g' <파일> # 1을 2로 바꿔서 출력

sed 's/1/2/gi' <파일> # 1을 2로 바꿔서 출력(대소문자 무시)

*특정라인에서 치환

sed '2,5 s/4/7/g' <파일> # <파일>의 2~5줄내에서 4를 7로 치환 출력

*정규식(예제1)

sed '/^#W|^\$W|^#/' <파일> # "#"로 시작하거나 or 빈 줄인 라인 제거

*정규식(예제2)

sed 's/[Zz]ip/rar/g' <파일> # zip 또는 Zip을 rar로 바꾸기

*패턴 출력

sed -n '/^May 22/ p' <파일> # "May 22" 로 시작하는 라인 출력

*공백 라인 넣기

sed G <파일> # 라인마다 공백라인 1줄 추가

sed 'G;G' <파일> # 라인마다 공백라인 2줄 추가

*에뮬레이팅 dos2unix

sed -i 's/Wr//' <파일> # hidden new line 제거

*In-place 편집 및 백업 (-i 옵션)

sed -i'.org' 's/this/that/gi' <파일> # 원본 수정 후 .org 확장자 파일 생성하여 백업

*쌍단어 스위칭

sed 's/^.*,.*\$/W2W, W1/g' <파일> # 특정 구분자(comma)를 기준으로 앞뒤 바꿈

*특정 조건내 변환

sed '/services/ s/start/stop/g' <파일> # services를 갖는 줄에서 start를 stop으로 global 하게 변환

*두개 이상 동시 변환

sed -i 's/that/this/gi;s/line/verse/gi' <파일> # that을 this로 / line을 verse로 동시에 바꿈

*다른 명령어와 동시 사용

ip route show | sed -n '/src/p' | sed -e 's/ */ /g' | cut -d' ' -f9

"|"를 이용하여 최종적으로 ip 추출("src"를 갖는 줄에서 연속된 공백을 하나로 치환하고 9 번째 컬럼 값인 ip를 출력)

```
[orcl:~]$ sed 's/KING/yyy' emp.txt
sed: -e expression #1, char 10: unterminated `s' command
[orcl:~]$ sed 's/KING/yyy/' emp.txt
7369 SMITH      CLERK      7902 17-DEC-80      800      0      20
7499 ALLEN      SALESMAN    7698 20-FEB-81      1600     300     30
7521 WARD      SALESMAN    7698 22-FEB-81      1250     500     30
7566 JONES      MANAGER     7839 02-APR-81      2975      0      20
7654 MARTIN    SALESMAN    7698 28-SEP-81      1250     1400     30
7698 BLAKE      MANAGER     7839 01-MAY-81      2850      0      30
7782 CLARK      MANAGER     7839 09-JUN-81      2450      0      10
7788 SCOTT      ANALYST     7566 19-APR-87      3000      0      20
7839 yyy      PRESIDENT   0 17-NOV-81      5000      0      10
7844 TURNER    SALESMAN    7698 08-SEP-81      1500      0      30
7876 ADAMS      CLERK      7788 23-MAY-87      1100      0      20
7900 JAMES      CLERK      7698 03-DEC-81      950       0      30
7902 FORD      ANALYST     7566 03-DEC-81      3000      0      20
7934 MILLER    CLERK      7782 23-JAN-82      1300      0      10
```

출력화면에서만 KING이 yyy로 치환된 것이지 원본 파일은 그대로

문제 68. 위의 변경된 내용대로 출력한 내용을 emp9.txt로 저장 하시오.

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

cp 명령어

2018년 6월 28일 목요일 오후 6:17

1. cp 명령어

" 파일을 복사하는 명령어 "

Ex. \$ cp 위치/원본 파일명 위치/복사할 파일명

\$ cp emp.txt emp400.txt

\$ cat emp400.txt

옵션	Long 옵션	설명
-a	--archive	-dR --preserve=all 옵션과 같음
-b	--backup	복사 대상 파일이 이미 존재할 경우를 대비하여 백업 파일을 만듦
-d	--no-dereference	원본 파일이 소프트링크 파일이면 소프트링크 원본을 삭제
-f	--force	복사 대상이 이미 존재하면 복사 대상 파일을 강제로 지우고 복사
-i	--interactive	복사 대상이 이미 존재하면 사용자에게 덮어씌울건지 여부를 확인
-l	--link	심볼릭 링크 형식으로 복사
-P	--parents	원본 파일명에 디렉토리 경로도 같이 입력했다면 그 경로 그대로 복사
-p	--preserve	원본 파일의 소유주, 그룹, 권한, 시간 정보들을 그대로 보존하여 복사
-s	--symbolic-link	디렉터리가 아닌 일반 파일을 심볼릭 링크 형식으로 복사
-u	--update	변경일이 같거나 더 최신의 복사 대상 파일이 있으면 복사 안 함
-v	--verbose	복사 상태를 자세하게 출력
-x	--one-file-system	원본 파일과 대상 파일의 파일시스템이 서로 다를 경우, 복사 안 함
-R, -r	--recursive	디렉토리를 복사할 경우 하위 디렉토리도 파일을 모두 복사

문제 69. 리눅스의 oracle 계정을 위한 환경설정 파일인 .bash_profile을 백업 하시오.

```
[orcl:~]$ cp .bash_profile .copy_bash_profile
[orcl:~]$ cp .bash_profile .copy_bash_profile
[orcl:~]$ ls
Adobe          b.txt          diff.sh        emp9.txt       ls.txt         test5          year.sh
Desktop        c.txt          e.txt          ename2.sh      oradiag_oracle test64         year_sum.sh
R-3.2.3        d.txt          emp.txt        find_file.sh   sar_0622.txt  test66
R-3.2.3.tar.gz datamodeler.log emp40.txt      job2.sh        size.sh        txttar.tar
a.txt          dept.txt       emp50.txt     job_sumsal.sh test100        winter.txt
all.tar        deptno2.sh     emp7.txt      labs           test200       winter_head.txt
```

문제 70. /home/oracle 밑에 있는 확장자 .txt 파일들 전부를 /home/oracle/backup 이라는 폴더를 만들고 거기에 복사 하시오.

```
[orcl:~]$ mkdir backup
[orcl:~]$ cp /home/oracle/*.txt /home/oracle/backup/
[orcl:~]$ cd backup/
[orcl:backup]$ ls
a.txt  c.txt  dept.txt  emp.txt  emp50.txt  emp9.txt  sar_0622.txt  winter_head.txt
b.txt  d.txt  e.txt  emp40.txt  emp7.txt  ls.txt  winter.txt
```

2. mv 명령어

" 파일의 이름을 바꾸거나 파일을 다른 디렉토리로 이동하는 명령어 "

옵션	Long 옵션	설명
	--backup[= CONTROL]	대상 파일이 존재하면 백업 파일을 만들
-b		--backup과 비슷하게 백업 파일을 생성
-f	--force	사용자에게 묻지 않고 파일을 강제로 덮어쓰
-i	--interactive	존재하는 파일을 덮어 쓸 경우 확인
-n	--no-clobber	존재하는 파일을 덮어쓰지 않음
-S	--suffix= SUFFIX	지정한 접미사로 백업을 생성
-t	--target-directory= DIRECTORY	전체 원본 파일을 대상 디렉토리로 이동
-T	--no-target-directory	파일을 대상 디렉토리로 취급하지 않고 원본 파일로 취급
-u	--update	파일이 업데이트된 경우에만 이동
-v	--verbose	진행 상황에 대한 상세한 정보를 출력

Ex. \$ mv 기존파일명 새로운파일명

\$ mv emp400.txt emp500.txt

\$ mv emp.txt ./backup/

문제 71. 다시 /home/oracle/backup 밑에 있는 emp.txt를 지금 현재 디렉토리인 /home/oracle 밑으로 이동 시키시오.

```
[orcl:~]$ mv ./backup/emp.txt .
[orcl:~]$ ls
Adobe      b.txt      deptno2.sh  emp7.txt    labs        test200     winter_head.txt
Desktop    backup     diff.sh     emp9.txt    ls.txt      test5       year.sh
R-3.2.3    c.txt      e.txt       ename2.sh   oradiag_oracle test64      year_sum.sh
R-3.2.3.tar.gz d.txt     emp.txt     find_file.sh sar_0622.txt test66
a.txt      datamodeler.log emp40.txt   job2.sh     size.sh     txttar.tar
all.tar     dept.txt   emp50.txt   job_sumsal.sh test100     winter.txt
```

변수

2018년 6월 28일 목요일 오후 8:31

■ 변수 사용법

1. 모든 변수는 문자열(string)로 취급된다.
2. 변수이름은 대소문자를 구분한다.
3. 변수에 값을 대입할 때는 '=' 좌우에 공백이 없어야 한다!!!
4. 변수의 값(문자열)을 출력하려면 \$를 붙이고 echo 명령어로 출력하면 된다.

```
ex. $ my_var='Hi ~~~'
    $ echo $my_var
```

■ 변수 산술연산 하는 방법

1. 변수에 대입한 값은 모두 문자열로 취급되기 때문에 변수에 들어있는 값을 숫자로 사칙연산 하려면 expr을 사용해야 한다.
2. 수식에 괄호 또는 곱하기를 사용하려면 그 앞에 꼭 역슬래쉬(\)를 붙여줘야한다.

```
ex. $ num1=100
    $ num2=200

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

[orcl:~]$ expr $num1 + $num2   (산술연산 사이에 한칸씩 띄어줘야한다)
300
```

■ 파라미터 변수

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

```
ex.
    $ vi b.sh
        echo "실행파일 이름은 $0 입니다. "
        echo "첫 번째 파라미터 값은 $1 입니다. "
        echo "두 번째 파라미터 값은 $2 입니다. "
        echo "전체 파라미터 값은 $* 입니다. "
```

```
$ sh b.sh 10 20
```

```
[orcl:~]$ vi b.sh
[orcl:~]$ sh b.sh
실행파일 이름 : b.sh
첫 번째 파라미터 :
두 번째 파라미터 :
전체 파라미터 값 :
[orcl:~]$ sh b.sh 10 20
실행파일 이름 : b.sh
첫 번째 파라미터 : 10
두 번째 파라미터 : 20
```

전체 파라미터 값 : 10 20