

시작과 종료

종료하는 방법

- ① 바탕 화면의 [사용자 이름] → [컴퓨터 끄기] → <컴퓨터 끄기>
- ② 터미널/콘솔에서 시스템 종료 명령 입력
"shutdown -P now", "halt -p", "init 0"

> # 프롬프트는 root 사용자
> \$ 프롬프트는 일반 사용자

시스템 재부팅

- ① 바탕 화면의 [사용자 이름] → [컴퓨터 끄기] → <다시 시작>
- ② 터미널/콘솔에서 시스템 재부팅 명령 입력
"shutdown -r now", "reboot", "init 6"

로그아웃

- ① 바탕 화면의 [사용자 이름] → [로그아웃]
- ② 터미널/콘솔에서 시스템 종료 명령 입력
"logout" 또는 "exit"

자동 완성과 히스토리

- 자동 완성이란 파일명의 일부만 입력한 후에 Tab키를 눌러 나머지 파일명을 자동으로 완성하는 기능

예) cd /etc/sysconfig/network-scripts/ 를 입력하려면
cd /et[Tab키]sysco[Tab키]networ[Tab키]

> 자동 완성기능은 빠른 입력효과도 있지만, 파일명이나 디렉터리가 틀리지 않고 정확하게 입력되는 효과도 있으므로 자주 활용된다.

- 도스 키란 이전에 입력한 명령어를 상/하 화살표 키를 이용해서 다시 나타내는 기능을 말함.



<실습1> 자동 완성과 도스키

실습목표

- 자동 완성 기능과 도스 키 기능을 익힌다.
- history 명령어의 기능을 확인한다.

실습화면(history 명령어)



파일 편집기 사용

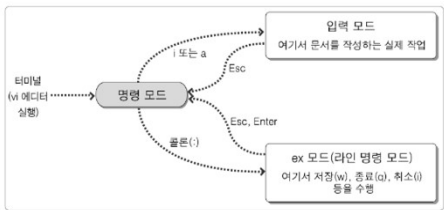
<실습2> 편집기를 사용하자

실습목표

- vi의 사용법을 연습한다.

> vi는 자주 사용해야 할 기능이므로 반드시 익혀야 한다.

vi 에디터 사용법 개요도



vi 기능 요약

- 문자열을 치환은 “:%s/기존문자열/새문자열”
- 행번호 표시는 “:set nu”

- 명령모드 → 입력모드

| | | | |
|---|---------------------------------|---|--|
| i | 현재 커서의 위치부터 입력(I) | I | 현재 커서 줄의 맨 앞에서부터 입력(Shift + I) |
| a | 현재 커서의 위치 다음 칸부터 입력(A) | A | 현재 커서 줄의 맨 마지막부터 입력(Shift + A) |

- 명령 모드에서 커서를 이동

| | | | |
|---|---|---|---|
| h | 커서를 왼쪽으로 한 칸 이동(H)과 같은 의미 (H) | j | 커서를 아래로 한 칸 이동(J)과 같은 의미. (J) |
| k | 커서를 위로 한 칸 이동(K)과 같은 의미. (K) | l | 커서를 오른쪽으로 한 칸 이동(L)과 같은 의미. (L) |

vi 기능 요약

- 문자열을 치환은 “:%s/기존문자열/새문자열”
- 행번호 표시는 “:set nu”

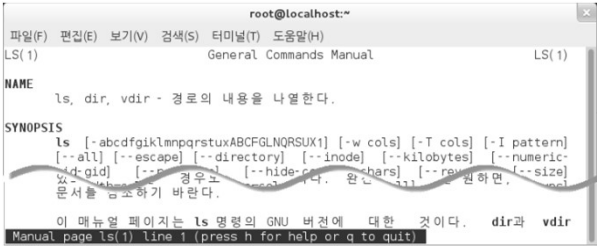
- 명령 모드에서 삭제, 복사, 붙여넣기

| | | | |
|----|---|----|--|
| x | 현재 커서가 위치한 글자 삭제([Del]과 같은 의미, [X]) | X | 현재 커서가 위치한 앞 글자 삭제([BackSpace]와 같은 의미, [Shift] + [X]) |
| dd | 현재 커서의 행 삭제([D] 연속 두 번 입력) | sd | 현재 커서부터 숫자만큼의 행 삭제(숫자 다음 [D] 연속 두 번 입력) |
| yy | 현재 커서가 있는 행을 복사([Y] 연속 두 번 입력) | yy | 현재 커서부터 숫자만큼의 행을 복사(숫자 다음 [Y] 연속 두 번 입력) |
| p | 복사한 내용을 현재 행 이후에 붙여 넣기([P]) | P | 복사한 내용을 현재 행 이전에 붙여 넣기([Ctrl] + [P]) |

도움말 사용법

- man 페이지는 섹션 1~9까지로 나뉨

- "man 명령어"를 사용하면 도움말 출력



리눅스 기본 명령어 (1)

- ls
Windows의 "dir"과 같은 역할로, 해당 디렉터리에 있는 파일의 목록을 나열
예) # ls /etc/sysconfig

- cd
디렉토리를 이동
예) # cd ../etc/sysconfig

- pwd
현재 디렉터리의 전체 경로를 출력

- rm
파일이나 디렉터리를 삭제
예) # rm -rf abc

리눅스 기본 명령어 (2)

- cp
파일이나 디렉터리를 복사
예) # cp abc.txt cba.txt
- touch
크기가 0인 새 파일을 생성, 이미 존재하는 경우 수정 시간을 변경
예) # touch abc.txt
- mv
파일과 디렉터리의 이름을 변경하거나 위치 이동 시 사용
예) mv abc.txt www.txt
- mkdir
새로운 디렉터리를 생성
예) # mkdir abc

리눅스 기본 명령어 (3)

- rmdir
디렉터리를 삭제. (단, 비어 있어야 함)
예) # rmdir abc
- cat
텍스트로 작성된 파일을 화면에 출력
예) # cat a.txt b.txt
- head, tail
텍스트로 작성된 파일의 앞 10행 또는 마지막 10행만 출력
예) # head anaconda-ks.cfg
- more
텍스트로 작성된 파일을 화면에 페이지 단위로 출력
예) # more anaconda-ks.cfg

리눅스 기본 명령어 (4)

- less
more와 용도가 비슷하지만 기능이 더 확장된 명령
예) # less anaconda-ks.cfg
- file
File이 어떤 종류의 파일인지를 표시
예) # file anaconda-ks.cfg
- clear
명령창을 깨끗하게 지워줌
예) # clear

파일과 디렉터리의 소유와 허가권 (1)

- 파일의 리스트와 파일 속성

The image shows a terminal window with the command `ls -l` executed. The output lists two files: `anaconda-ks.cfg` and `sample.txt`. Annotations with arrows point to specific parts of the output:

- `rwxr-xr-x`: 파일 유형 (File type)
- `1`: 링크 수 (Link count)
- `root`: 파일 소유자 이름 (File owner name)
- `root`: 파일 소유 그룹 이름 (File owner group name)
- `0`: 파일 크기(Byte) (File size in bytes)
- `9월 16 13:50`: 마지막 변경 날짜/시간 (Last modification date/time)
- `sample.txt`: 파일 이름 (File name)

파일과 디렉터리의 소유와 허가권 (2)

- 파일 유형
 - 디렉터리일 경우에는 d, 일반적인 파일일 경우에는 -가 표시
- 파일 허가권(Permission)
 - "rw-", "r--", "r--" 3개씩 끊어서 읽음 (r은 read, w는 write, x는 execute의 약자)
 - 첫 번째 "rw-"는 소유자(User)의 파일접근 권한
 - 두 번째의 "r--"는 그룹(Group)의 파일접근 권한
 - 세 번째의 "r--"는 그 외의 사용자(Other)의 파일접근 권한
 - 숫자로도 표시 가능 (8진수)

| 소유자(User) | | | 그룹(Group) | | | 그 외 사용자(Other) | | |
|-----------|---|---|-----------|---|---|----------------|---|---|
| r | w | - | r | - | - | r | - | - |
| 4 | 2 | 0 | 4 | 0 | 0 | 4 | 0 | 0 |
| 6 | | | 4 | | | 4 | | |

파일과 디렉터리의 소유와 허가권 (3)

- chmod 명령
 - 파일 허가권 변경 명령어
 - 예) # chmod 777 sample.txt
- 파일 소유권(Ownership)
 - 파일을 소유한 사용자와 그룹을 의미
- chown/chgrp 명령
 - 파일의 소유권을 바꾸는 명령어
 - 예) # chown centos.centos sample.txt 또는 # chown centos sample.txt 및 # chgrp centos sample.txt



<실습3> 파일의 허가권 및 소유권 연습

- 실습목표
 - 파일의 허가권에 대한 개념과 명령어를 익힌다.
 - 파일의 소유권에 대한 개념과 명령어를 익히고, 보안에 관련된 내용도 확인해 본다.
- 실습 화면 (소유권 변경 실패)

```
fedora@localhost:~  
[fedora@localhost ~]$ whoami  
fedora  
[fedora@localhost ~]$ chown root.root test  
chown: changing ownership of 'test': 명령을 허용하지 않음  
[fedora@localhost ~]$
```

프로그램 설치를 위한 RPM (1)

- RPM(Redhat Package Manager)
 - Windows의 "setup.exe"와 비슷한 설치 파일
 - 확장명은 *.rpm이며, 이를 '패키지(Package)'라고 부름.
- 파일의 의미

```
root@localhost:run/media/root/CentOS 7 x86_64/Packages  
[root@localhost Packages]# pwd  
/run/media/root/CentOS 7 x86_64/Packages  
[root@localhost Packages]# ls -l gedit-3*  
-rw-rw-r-- 2 root root 2706324 7월 4 10:27 gedit-3.8.3-6.el7.x86_64.rpm  
[root@localhost Packages]#
```

- > x86_64 : 64비트 CPU
- > i386~686 : 32비트 CPU
- > src : 소스
- > noarch : 모든 CPU

패키지이름-버전-릴리즈번호.CentOS버전.아키텍처.rpm

- 패키지이름 : gedit → 패키지(프로그램)의 이름
- 버전 : 3.8.3 → 대개 3자리수로 구성. 주버전, 부버전, 패치버전
- 릴리즈번호 : 6 → 문제점을 개선할 때마다 붙여지는 번호
- CentOS 버전 : el7 → CentOS에서 배포할 경우에 붙여짐
- 아키텍처 : x86_64 → 64비트 CPU를 의미

프로그램 설치를 위한 RPM (2)

- 자주 사용하는 RPM 명령어 옵션
 - 설치 : rpm -Uvh 패키지파일이름.rpm
 - U → (대문자) 패키지가 설치/업그레이드
 - v → 설치과정의 확인
 - h → 설치진행과정을 "#"마크로 화면에 출력
 - 삭제 : rpm -e 패키지이름
 - 이미 설치된 패키지 질의
 - rpm -qa 패키지 이름 → 패키지가 설치되었는지 확인
 - rpm -qf 파일의절대경로
 - 파일이 어느 패키지에 포함된 것인지 확인
 - 아직 설치되지 않은 rpm 파일에 대한 질의
 - rpm -qlp 패키지파일이름.rpm
 - 패키지 파일에 어떤 파일들이 포함되었는지 확인
 - rpm -qip 패키지파일이름.rpm → 패키지 파일의 상세정보

프로그램 설치를 위한 RPM (3)

- RPM 단점
 - '의존성' 문제
 - A패키지가 설치되기 위해서 B패키지가 필요할 경우, RPM으로 는 해결이 까다로움.
 - 이를 해결하기 위해 YUM이 등장함



<실습4> rpm 패키지 설치 연습

- 실습목표
 - RPM 명령어의 사용법을 익힌다.
 - RPM 파일의 의존성 문제를 파악한다.
- 실습화면 (mc 패키지 설치)

```
root@localhost/run/media/root/CentOS 7 x86_64/Packages
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost Packages]# rpm -Uvh mc-4.8.7-8.el7.x86_64.rpm
경고: mc-4.8.7-8.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5
: NOKEY
준비 중... ##### [100%]
Updating / installing...
 1:mc-1:4.8.7-8.el7 ##### [100%]
[root@localhost Packages]#
```

편리한 패키지 설치, YUM (1)

- YUM(Yellowdog Updater Modified) 개념
 - "rpm"명령의 패키지 의존성 문제를 완전하게 해결됨.
 - 인터넷을 통하여 필요한 파일을 저장소(Repository)에서 자동으로 모두 다운로드해서 설치하는 방식
- YUM 기본적인 사용법

➤ 저장소의 URL은 "/etc/yum.repos.d/"디렉터리

 - 기본 설치 : yum install 패키지이름
 - 주로 "yum -y install 패키지이름"으로 사용
 - "-y"는 사용자의 확인을 모두 "yes"로 간주하고 설치를 진행한다는 옵션
 - RPM 파일 설치 : yum localinstall rpm파일이름.rpm
 - 업데이트 가능한 목록 보기 : yum check-update
 - 업데이트 : yum update 패키지이름
 - 삭제 : yum remove 패키지이름
 - 정보 확인 : yum info 패키지이름



<실습5> yum 기본 연습

- 실습목표
 - 의존성 문제가 있는 패키지를 “yum”명령으로 설치해 본다.
- 실습 화면 (의존성 패키지 자동 다운로드)

```
> Finished Dependency Resolution
Dependencies Resolved

=====
Package                        Arch      Version      Repository    Size
=====
Installing:
mysql-connector-odbc           x86_64    5.2.5-6.el7  base          146 k
Installing for dependencies:
unixODBC                       x86_64    2.3.1-10.el7 base          413 k
=====

Transaction Summary
=====
Install 1 Package (+1 Dependent package)

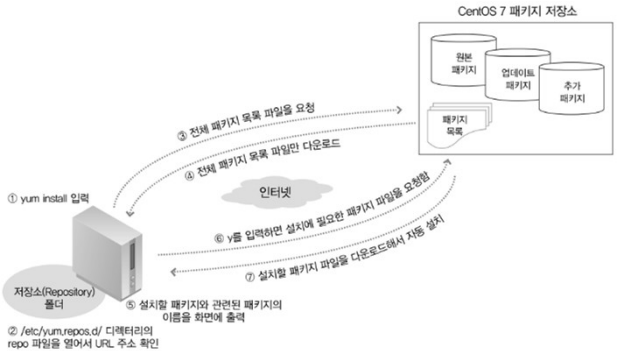
Total download size: 559 k
Installed size: 1.6 M
Is this ok [y/d/N]:
```

편리한 패키지 설치, YUM (2)

- YUM 고급 사용법
 - 패키지 그룹 설치
 - yum groupinstall “패키지그룹이름”
 - 패키지 리스트 확인
 - yum list 패키지이름
 - 특정 파일이 속한 패키지 이름 확인
 - yum provides 파일이름
 - GPG 키 검사 생략
 - yum install --nogpgcheck rpm파일이름.rpm
 - CentOS 19에서 검증되지 않은 패키지를 강제로 설치할 때 사용
 - 기존 저장소 목록 지우기
 - yum clean all

편리한 패키지 설치, YUM (3)

- YUM 작동 방식 설정 파일
 - ‘yum install 패키지이름’ 명령이 작동하는 방식

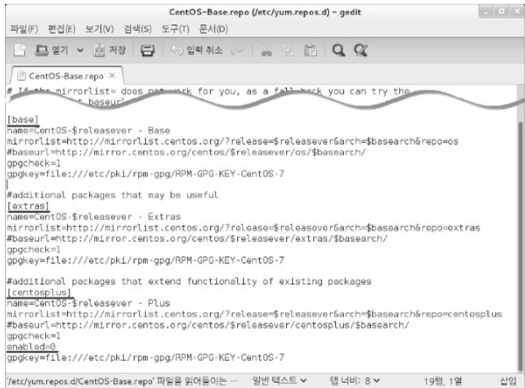


편리한 패키지 설치, YUM (4)

- YUM 작동 방식 설정 파일
 - /etc/yum.conf 파일 : 특별히 변경할 필요 없음
 - /etc/yum.repos.d/ 디렉터리
 - yum 명령을 입력했을 때 검색하게 되는 네트워크의 주소가 들어 있는 여러 개의 파일이 있음
 - /etc/yum.repos.d/ 디렉터리의 *.repo 파일
 - CentOS-Base.repo : [base], [extra]만 남기고 [updates] 부분은 삭제했음. 즉, 출시 시점의 원본 패키지만 설치됨.

편리한 패키지 설치, YUM (5)

- CentOS-Base.repo 파일



```
CentOS-Base.repo (/etc/yum.repos.d) - gedit
파일(F) 편집(E) 보기(V) 검색(S) 도구(T) 문서(D)
CentOS-Base.repo x
# If the mirrorlist= does not work for you, as a fallback you can try the
# baseurl= line
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#Additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#Additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
/etc/yum.repos.d/CentOS-Base.repo 파일을 읽어들이는 ...  일반 텍스트  탭 너비: 8  19행, 1열  삽입
```