



리눅스 명령 사용

- 프롬프트
 - » 사용자의 명령 입력을 기다리는 표시
 - » 사용하는 셸에 따라 다르게 나타남
 - › (배시셸: \$로 표시, 시스템 관리자의 경우에는 #로 표시)
 - » 프롬프트는 사용자 가 다른 모양으로 바꿀 수 있음

- 홈 디렉터리
 - » user1@localhost:~
 - › user1은 사용자 계정 이름
 - › localhost는 호스트 이름
 - › ~는 user1 사용자의 홈 디렉터리
 - » 사용자의 홈 디렉터리는 사용자 계정을 등록할 때 결정
 - » 홈 디렉터리 아래에 자신이 원하는 파일이나 디렉터리를 생성하며 작업 할 수 있음

리눅스 명령 사용

■ 문자 지우기

- » 리눅스의 명령행에서 문자를 지울 때 [backspace] 또는 [delete] 를 사용
- » 리눅스에 따라 둘 중 하나가 기본 값으로 설정 → CentOS는 [backspace]

■ 단어 지우기

- » 리눅스의 명령 행에서 단어를 지울 때는 [ctrl]+ w를 사용
- » 이때 단어는 공백으로 구분

■ 행 지우기

- » 리눅스의 명령 행에서 행을 지울 때는 [ctrl]+ u를 사용
- » 해당행의 내용을 모두 삭제 할 수 있음

리눅스 명령 사용

▪ 명령 구조

명령의 구조

- **형식** 명령 [옵션] [인자]
- **명령** 리눅스를 사용하기 위해 사용자가 입력하는 다양한 명령은 date, man, ls, cp, mv 등 수백 가지가 있다.
- **옵션** 옵션을 사용하여 명령의 세부 기능을 선택할 수 있다. 리눅스의 기능을 풍부하게 하는 중요한 특징으로 명령에 따라 있는 경우도 있고 없는 경우도 있다. 옵션은 -나 -- 기호로 시작하며 영문 소문자나 대문자로 구성된다. 명령에 따라 어떤 옵션이 있고 그 기능이 무엇인지는 해당 명령의 사용법을 참조해야 한다.
- **인자** 인자는 명령으로 전달되는 값이며 주로 파일명이나 디렉터리명이 사용된다. 명령에 따라 없을 수도 있고 각 명령에 필요한 인자도 각기 다르므로 사용법을 참조해야 한다.



시스템 종료

- 종료

- » shutdown -P now, shutdown -h now
- » halt -p, poweroff, init 0
- » ...

- 재부팅

- » shutdown -r now
- » reboot, init 6
- » ...

- 로그아웃

- » exit
- » logout

자동 완성과 히스토리

- 자동 완성

- » 파일명의 일부만 입력한 후에 Tab 키를 눌러 나머지 파일명을 자동으로 완성

- 히스토리

- » 방향키를 사용해서 이전에 입력한 명령을 화면에 표시

- 이전에 실행한 명령 표시

- » history 명령

- 명령 재실행

사용법	기능
!!	바로 직전에 실행한 명령을 재실행한다.
!번호	히스토리에서 해당 번호의 명령을 재실행한다.
!문자열	히스토리에서 해당 문자열로 시작하는 마지막 명령을 재실행한다.



문서 편집기

■ 리눅스의 편집기 종류

- » GUI 환경인 GNOME에서 제공하는 gedit
- » 유닉스에서부터 사용했던 행 편집기(라인 편집기)와 화면 편집기

구분	종류
행 단위 편집기	ed, ex, sed
화면 단위 편집기	vi, emacs(이맥스)
GUI 편집기	gedit

■ 행 단위 편집기

- » ed : 유닉스 초기의 행 편집기로 사용이 불편하여 거의 사용 않음
- » ex : 행 편집기이지만 단독으로 사용하기보다는 vi에 연결하여 vi를 더욱 강력하게 하는 다양한 기능을 제공
- » sed : 스트림 편집기로, 일반 편집기와 달리 지시된 명령에 따라 파일의 내용을 일괄적으로 변경해서 출력

문서 편집기

- 화면 단위 편집기
 - » vi → 리눅스에서 일반적으로 사용할 수 있는 화면 편집기
 - » emacs(이맥스) → 제공하는 기능이 매우 다양하지만 사용법이 어렵고 복잡하여 전문적인 애호가 위주로 사용

리눅스의 문서 편집기

■ 모드형

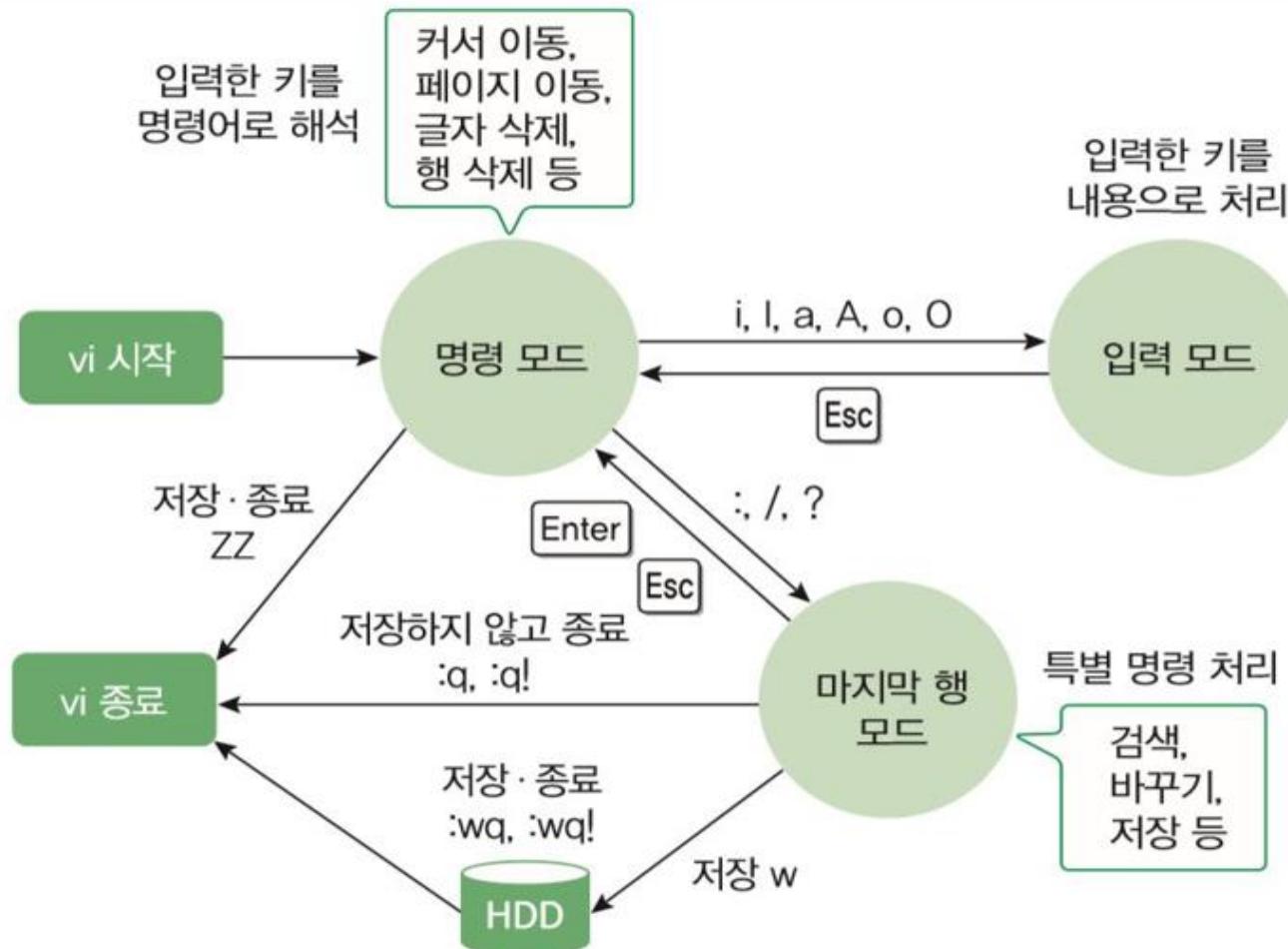
- » 입력 모드와 명령 모드가 구분
- » 입력 모드 : 텍스트를 입력할 수 있는 모드
- » 명령 모드 : 텍스트를 수정하고, 삭제하고, 복사와 붙이기 등 편집을 하는 모드
- » 같은 글자라도 입력 모드에서는 텍스트로 처리하여 입력되고, 명령 모드에서는 편집 명령으로 사용
- » vi는 모드형 편집기

■ 비모드형

- » 입력 모드와 명령 모드가 구분되어 있지 않음
- » 편집 기능을 Ctrl이나 Alt 같은 특수 키와 함께 사용
- » 한글과 워드는 비모드형 편집기

vi 사용법

■ vi 동작 모드



vi 사용법

■ vi 시작

- » 파일을 지정할 경우: 해당 파일이 있으면 파일의 내용이 보이고, 없는 파일이면 빈 파일이 열린다.
- » 파일을 지정하지 않을 경우: 빈 파일이 열린다(파일명은 저장할 때 지정 가능)

■ vi 파일 저장하고 종료

- » 명령 모드나 마지막 행 모드에서 저장하고 종료 가능
- » 파일의 저장과 종료를 동시에 수행할 수 있는 명령키 [shift+zz]

모드	명령키	기능
마지막 행 모드	:q	vi에서 작업한 것이 없을 때 그냥 종료한다.
	:q!	작업한 내용을 저장하지 않고 종료한다.
	:w 파일명	작업한 내용을 저장만 한다. 파일명을 지정하면 다른 파일로 저장한다.
	:wq, :wq!	작업한 내용을 저장하고 vi를 종료한다.
명령 모드	ZZ(<small>Shift</small> +zz)	작업한 내용을 저장하고 vi를 종료한다.

vi 사용법

■ vi 모드 전환

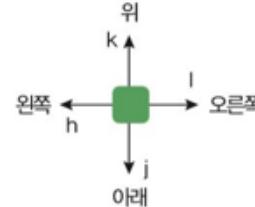
» 처음 실행 할 때 명령 모드로 시작

명령키	기능
i	커서 앞에 입력한다(현재 커서 자리에 입력한다).
a	커서 뒤에 입력한다(현재 커서 다음 자리에 입력한다).
o	커서가 위치한 행의 다음 행에 입력한다.
I(대문자 i)	커서가 위치한 행의 첫 칼럼으로 이동하여 입력한다.
A	커서가 위치한 행의 마지막 칼럼으로 이동하여 입력한다.
O	커서가 위치한 행의 이전 행에 입력한다.

vi 사용법

■ 커서 이동

명령키	기능
k / j	커서를 한 행 위 / 아래로 이동한다.
l / h	커서를 한 글자 오른쪽 / 왼쪽으로 이동한다.
^ 또는 0 / \$	커서를 현재 행의 처음 / 마지막으로 이동한다.
- / + 또는 Enter	커서를 앞 / 다음 행의 처음으로 이동한다.
H / M / L	커서를 화면의 맨 윗 / 중간 / 맨 아랫 행으로 이동한다.
w / e	커서를 다음 단어의 첫 / 마지막 글자로 이동한다.
b	커서를 앞 단어의 첫 글자로 이동한다.



■ 화면 이동

기존 명령키	기능	추가 명령키	명령키	기능
^u (Ctrl +u)	반 화면 위로 이동한다.		G(Shift +g)	파일의 마지막 행으로 커서가 이동한다.
^d (Ctrl +d)	반 화면 아래로 이동한다.		행 번호G(Shift +g)	지정한 행 번호로 커서가 이동한다.
^b (Ctrl +b)	한 화면 위로 이동한다.	Page Up	:행 번호	지정한 행 번호로 커서가 이동한다(마지막 행 모드).
^f (Ctrl +f)	한 화면 아래로 이동한다.	Page Down	:\$	파일의 마지막 행으로 커서가 이동한다(마지막 행 모드).
^y (Ctrl +y)	화면을 한 행만 위로 이동한다.			
^e (Ctrl +e)	화면을 한 행만 아래로 이동한다.			

vi 사용법

■ 내용 수정

- » 수정을 마치면 [esc] 를 눌러서 명령 모드로 변환 후 저장 명령으로 수정 완료

■ 내용 삭제

- » 입력 모드에서 내용을 입력하는 도중에 틀린 글자를 삭제할 때는 [backspace] 키나 [delete] 키 사용
- » 명령 모드에서 글자나 행을 삭제하려면 다음과 같은 명령키 사용

명령키	기능
x, #x	커서 위치의 글자를 삭제한다. #에는 삭제할 글자 수를 지정한다.
dw, #dw	커서 위치의 단어를 삭제한다. #에는 삭제할 단어 수를 지정한다.
dd, #dd	커서 위치의 행을 삭제한다. #에는 삭제할 행의 수를 지정한다.
D(Shift +d)	커서 위치부터 행의 끝까지 삭제한다.

vi 사용법

■ 명령 취소

명령키	기능
u	명령을 취소한다.
U	해당 행에서 한 모든 명령을 취소한다.
:e!	마지막으로 저장한 내용 이후의 것을 버리고 새로 작업한다.

■ 복사 / 자르기 → 붙이기

명령키	기능
yy, #yy	커서가 위치한 행을 복사한다. #에는 복사할 행의 수를 지정한다.
p	커서가 위치한 행의 아래쪽에 붙인다.
P	커서가 위치한 행의 위쪽에 붙인다.
dd, #dd	커서가 위치한 행을 잘라둔다. 삭제와 같은 기능이다. #에는 잘라둘 행의 수를 지정한다.

- » yy 명령키로 행 복사 → 원하는 행으로 커서 이동 후 p 명령키로 붙이기
- » dd 명령키로 행 삭제 → 원하는 행으로 커서 이동 후 p 명령키로 붙이기

vi 사용법

■ 검색

- » 명령 모드에서 /이나 ? 다음에 찾고자 하는 문자열 입력 후
- » [Enter]를 누르면 검색 진행 → 해당 문자열을 찾으면 그 문자열의 시작 위치로 커서 이동

명령키	기능
/문자열	문자열을 아래 방향으로 검색한다.
?문자열	문자열을 위 방향으로 검색한다.
n	원래 찾던 방향으로 다음 문자열을 검색한다.
N	역방향으로 다음 문자열을 검색한다.

vi 사용법

■ 바꾸기

명령어	기능
:s/문자열1/문자열2/	커서가 위치한 행에서 첫 번째로 나오는 문자열1을 문자열2로 바꾼다.
:%s/문자열1/문자열2/g	파일 전체에서 모든 문자열1을 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/	범위 내 모든 각 행에서 첫 번째로 나오는 문자열1을 찾아 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/g	범위 내 모든 행에서 문자열1을 문자열2로 바꾼다.
:<범위>s/문자열1/문자열2/gc	범위 내 모든 행에서 문자열1을 문자열2로 바꿀 때 수정할지 여부를 묻는다.

vi 사용법

■ vi 환경 설정

set 명령과 옵션	기능
set nu	파일 내용의 각 행에 행 번호를 표시한다(보이기만 할 뿐 저장되지는 않는다).
set nonu	행 번호를 감춘다.
set list	눈에 보이지 않는 특수문자를 표시한다(tab:^\t, eol:\$ 등).
set nolist	특수문자를 감춘다.
set showmode	현재 모드를 표시한다.
set noshowmode	현재 모드를 감춘다.
set	set으로 설정한 모든 vi 환경 설정 값을 출력한다.
set all	모든 vi 환경 변수와 현재 값을 출력한다.



리눅스 파일의 종류

■ 파일

- » 관련 있는 정보들의 집합
- » 리눅스는 파일을 효과적으로 관리하기 위해 디렉터리를 사용
- » 파일 시스템 → 디렉터리와 파일로 구성된 전체 집합
- » 리눅스에서 파일은 사용 목적에 따라 일반 파일, 디렉터리, 심볼릭 링크, 장치 파일로 구분

■ 일반 파일

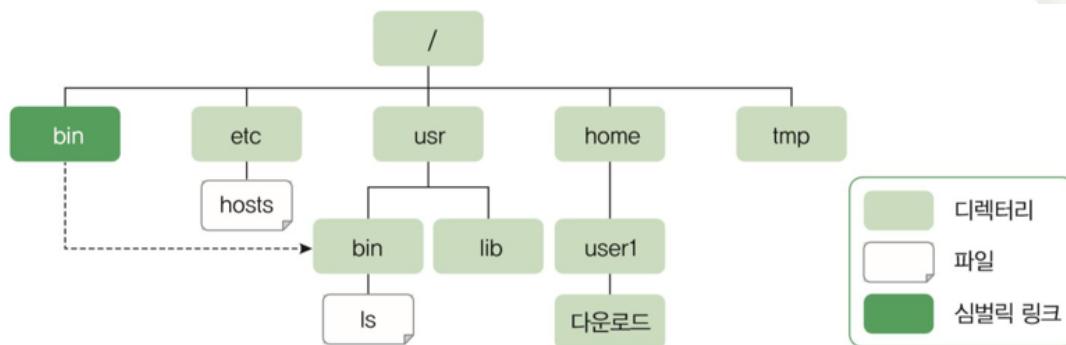
- » 데이터를 저장하는 데 주로 사용
- » 리눅스에서 사용하는 대부분의 파일은 일반 파일에 해당
- » 바이너리 파일 → 실행 파일이나 이미지 파일의 경우 바이너리 형태로 데이터가 저장되어 사용

리눅스 파일의 종류

- 디렉터리
 - » 리눅스에서는 디렉터리도 파일로 취급
 - » 디렉터리 파일에는 해당 디렉터리에 저장된 파일이나 하위 디렉터리에 대한 정보 저장
- 심볼릭 링크
 - » 원본 파일을 대신하여 다른 이름으로 파일명을 지정한 것(윈도의 바로 가기와 유사)
- 장치 파일
 - » 하드디스크, 키보드 등 리눅스 시스템에 부착된 장치들을 관리하기 위한 특수 파일

디렉터리 계층 구조

- 트리 구조
 - » 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성
 - » 모든 디렉터리의 출발점은 루트(root) 디렉터리이며 슬래시(/)로 표시
- 루트 디렉터리와 서브 디렉터리
 - » 서브 디렉터리(하위 디렉터리) → 최상단 루트 디렉터리(/) 아래의 디렉터리 (etc, usr, home, tmp)
 - » 부모 디렉터리(상위 디렉터리) → 서브 디렉터리 위에 자신을 포함하고 있는 디렉터리 (".."으로 표시)
 - » 루트 디렉터리를 제외하고 모든 디렉터리에는 부모 디렉터리가 있음



디렉터리 계층 구조

- 작업 디렉터리
 - » 현재 사용 중인 디렉터리
 - » .(마침표) 로 표시
 - » 작업 디렉터리의 위치는 pwd 명령으로 확인
- 홈 디렉터리
 - » 각 사용자에게 할당된 디렉터리로 처음 사용자 계정을 만들 때 지정
 - » 사용자는 자신의 홈 디렉터리 아래에 파일이나 서브 디렉터리를 생성하며 작업할 수 있음
 - » ~로 표시

절대 경로명과 상대 경로명

▪ 경로명

- » 파일 시스템에서 디렉터리 계층 구조에 있는 특정 파일이나 디렉터리의 위치 표시 (구분자 : /)
- » 가장 앞에 있는 / 는 루트 디렉터리를 뜻하지만 경로명 중간에 있는 / 는 구분자
 - › /usr/bin/ls에서 맨 앞의 /는 루트 디렉터리를 의미하고 중간에 있는 / 두 개는 디렉터리명과 파일명을 구분하는 구분자

▪ 절대 경로명과 상대 경로명

- » 절대 경로명 : 항상 루트(/) 디렉터리부터 시작
- » 상대 경로명 : 현재 디렉터리를 기준으로 시작

디렉터리 관련 명령

- 현재 디렉터리 확인

- » pwd 명령

```
[root@centos-server local]#  
[root@centos-server local]# pwd  
/usr/local  
[root@centos-server local]#
```

- 디렉터리 이동

- » cd 명령

- › cd ~ : ~를 사용하여 홈 디렉터리로 이동

- › cd : 목적지를 지정하지 않고 cd 명령만 사용하면 해당 계정의 홈

```
[root@centos-server local]# cd /usr/local  
[root@centos-server local]#  
[root@centos-server local]# pwd  
/usr/local  
[root@centos-server local]#  
[root@centos-server local]# cd ~  
[root@centos-server ~]#  
[root@centos-server ~]# pwd  
/root  
[root@centos-server ~]# █
```

디렉터리 내용 확인

- 디렉터리에 있는 파일이나 서브 디렉터리 등 디렉터리의 내용 표시
» ls 명령

```
[root@centos-server ~]#  
[root@centos-server ~]# ls  
anaconda-ks.cfg  
[root@centos-server ~]#  
[root@centos-server ~]# ls -a /usr  
. .. bin etc games include lib lib64 libexec local sbin share src tmp  
[root@centos-server ~]#  
[root@centos-server ~]# ls -a -l /usr  
합계 116  
drwxr-xr-x. 13 root root 155 2월 11 22:11 .  
dr-xr-xr-x. 17 root root 224 2월 11 22:17 ..  
dr-xr-xr-x. 2 root root 24576 2월 11 22:50 bin  
drwxr-xr-x. 2 root root 6 4월 11 2018 etc  
drwxr-xr-x. 2 root root 6 4월 11 2018 games  
drwxr-xr-x. 35 root root 4096 2월 11 22:50 include  
dr-xr-xr-x. 28 root root 4096 2월 11 22:50 lib  
dr-xr-xr-x. 42 root root 24576 2월 11 22:50 lib64  
drwxr-xr-x. 24 root root 4096 2월 11 22:50 libexec  
drwxr-xr-x. 13 root root 162 2월 11 23:37 local  
dr-xr-xr-x. 2 root root 12288 2월 11 22:50 sbin  
drwxr-xr-x. 87 root root 4096 2월 11 22:50 share  
drwxr-xr-x. 4 root root 34 2월 11 22:11 src  
lrwxrwxrwx. 1 root root 10 2월 11 22:11 tmp -> ../../var/tmp  
[root@centos-server ~]#
```

디렉터리 내용 확인

▪ ls -l 명령의 출력 정보

필드 번호	필드 값	의미
1	d	다음과 같은 파일 종류를 나타낸다. -: 일반(정규) 파일 d: 디렉터리 파일 l: 심볼릭 링크 파일 b: 블록 단위로 읽고 쓰는 블록 장치 파일 c: 섹터 단위로 읽고 쓰는 문자 장치 파일 p: 파이프 파일(프로세스 간 통신에 사용되는 특수 파일) s: 소켓(네트워크 통신에 사용되는 특수 파일)
2	rwxr-xr-x	파일 접근 권한. 파일의 소유자, 그룹, 기타 사용자가 읽고 수정하고 실행할 수 있는 권한이 어떻게 부여되어 있는지를 보여준다.
3	2	하드 링크의 개수
4	user1	파일 소유자
5	user1	파일이 속한 그룹
6	6	파일 크기(바이트 단위)
7	10월 27 21:11	파일이 마지막으로 수정된 시간
8	공개	파일명

디렉터리 관련 명령

- 디렉터리 생성
 - » `mkdir` 명령
 - › 디렉터리명은 상대 경로명이나 절대 경로명으로 지정
 - › 생성할 디렉터리를 여러 개 지정하면 동시에 생성 → 이때 디렉터리는 공백문자로 구분
 - › 디렉터리명으로 지정한 경로 중 중간 단계의 디렉터리가 없을 경우 중간 단계 디렉터리 생성 후 최종 디렉터리 생성 → -p 옵션 사용
- 디렉터리 삭제
 - » `rmdir` 명령
 - › 해당 디렉터리가 비어 있지 않은 경우 삭제 실패

파일 관련 명령

▪ 파일 내용 출력

» cat 명령 → 파일 내용을 연속으로 출력

› -n 옵션 : 행 번호 출력

» more 명령 사용 → 화면 단위로 파일 내용 출력

내부 명령	설명
space bar	다음 화면 출력
enter	다음 행 출력 (한 줄 단위 스크롤)
/문자열	해당 문자열 검색
q	종료

» less 명령 → 파일 내용을 화면 단위로 출력 (more 명령 개선)

내부 명령	설명
j	다음 행 출력 (한 줄 단위 스크롤)
k	이전 행 출력 (한 줄 단위 스크롤)
ctrl + b	이전 화면 출력

파일 관련 명령

- 파일 내용 출력
 - » head 명령 사용 → 파일 내용의 앞부분 출력
 - › -숫자 옵션 : 지정된 숫자의 행 출력
 - » tail 명령 사용 → 파일 내용의 뒷부분 출력
 - › -숫자 옵션 : 지정된 숫자의 행 출력
 - › -f 옵션 : 파일 내용을 주기적으로 반복 출력 → 파일 내용의 변화를 확인할 때 활용

파일 관련 명령

- 파일 복사

- » cp 명령 사용 → 파일 또는 디렉터리 복사

- » 인자를 여러 개 지정할 경우

- › cp명령에서 첫 번째 인자의 자리에 파일명 여러 개 지정 가능
 - › 마지막 인자는 반드시 디렉터리여야 함
 - › 마지막에 지정한 디렉터리로 앞서 지정한 파일들이 모두 복사됨

- » 디렉터리 복사

- › 첫 번째 인자와 두 번째 인자에 모두 디렉터리를 지정하고 -r 옵션 사용

파일 관련 명령

■ 파일 이동과 파일명 변경

» mv 명령

- › 첫 번째 인자로는 원본 파일이나 디렉터리를 지정
- › 두 번째 인자는 목적지 파일이나 디렉터리를 지정
- › 두 인자 모두 동일 경로의 파일 또는 디렉터리인 경우 이름 변경 효과
- › 여러 파일을 디렉터리로 이동
 - 첫 번째 인자에 파일 또는 디렉터리를 여러 개 지정
 - 마지막 인자는 반드시 디렉터리

■ 파일 삭제

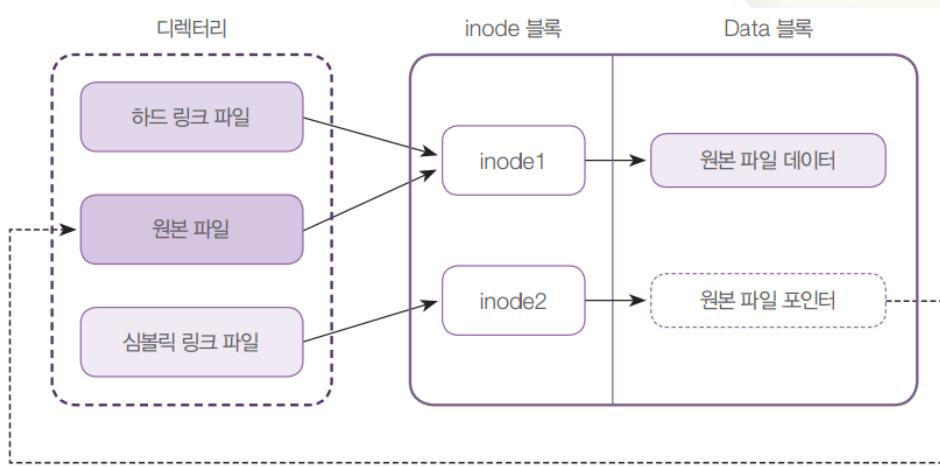
» rm 명령

- › 삭제할 파일을 인자로 지정하면 해당 파일 삭제
- › -r 옵션을 지정하면 디렉터리 삭제
 - 비어 있지 않은 디렉터리도 삭제 가능

파일 관련 명령

▪ 파일 링크

- » 기존에 있는 파일에 새로운 파일명을 붙이는 것
- » 종류
 - › 하드링크 : 기존 파일에 새로운 파일명을 추가로 생성
 - › 심볼릭 링크 : 원본 파일을 가리키는 새로운 파일을 생성



» inode

- › 파일에 대한 정보를 가진 특별한 구조체
- › 외부에는 번호로 표시
- › 내부에는 파일의 종류, 크기, 소유자, 파일 변경 시간, 파일명 등 파일 상세 정보와 데이터 블록의 주소 저장

파일 관련 명령

- 파일 링크 (계속)

- » 심볼릭 링크

- › 원도의 바로가기처럼 원본 파일을 가리키는 파일
 - › -s 옵션 사용해서 생성
 - › ls -l 명령으로 확인
 - 파일의 종류가 'l(소문자 L)'로 표시
 - 파일명에 '->'를 사용하여 원본 파일이 무엇인지 표시
 - › 원본 파일이 삭제되면 심볼릭 링크로 연결 불가

파일 관련 명령

- 빈 파일 만들기
 - » touch 명령
 - › 존재하지 않는 파일을 지정하면 빈 파일 생성
 - › 이미 있는 파일에 대해 touch 명령을 옵션 없이 사용하면 파일의 수정 시간을 현재 시간으로 변경
- 파일 내용 검색
 - » grep 명령
 - › 형식 : grep [옵션] [패턴] [파일]
- 파일 탐색
 - » find 명령
 - › 형식 : find [경로] [검색조건] [동작]

파일 관련 명령

- 명령의 위치 찾기

- » `whereis` 명령

- › 지정한 명령을 환경 변수 \$PATH와 \$MANPATH에 지정된 디렉터리에서 검색
 - › 관련된 파일 검색

- » `which` 명령

- › Alias나 PATH 환경 변수로 지정된 경로에서 검색
 - › 실행파일만 검색
 - › 절대 경로 출력



사용자 계정 관련 파일

- /etc/passwd 파일

- » 사용자 계정 정보가 저장된 기본 파일

- » 한 행에 사용자 한 명에 대한 정보가 기록되며 :으로 구분되는 일곱 개의 항목으로 구성

```
root:x:0:0:root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcscd daemon:/dev/null:/sbin/nologin
```

사용자 계정 관련 파일

- /etc/group 파일

- » 그룹의 정보가 저장된 파일
- » 네 개의 항목으로 구성되며 각 항목은 :으로 구분

```
root:x:0:  
bin:x:1:  
daemon:x:2:  
sys:x:3:  
adm:x:4:  
tty:x:5:  
disk:x:6:  
lp:x:7:  
mem:x:8:  
kmem:x:9:  
wheel:x:10:  
cdrom:x:11:  
mail:x:12:postfix  
man:x:15:  
dialout:x:18:  
floppy:x:19:  
games:x:20:  
tape:x:33:  
video:x:39:  
ftp:x:50:  
lock:x:54:  
audio:x:62:
```

사용자 계정 관리 명령

- 사용자 계정 생성
 - » useradd 명령
 - » CentOS의 adduser 명령은 useradd에 대한 심볼릭 링크
- 사용자 계정 정보 수정
 - » usermod 명령
- 사용자 계정 삭제
 - » userdel 명령
- 그룹 정보 수정
 - » groupmod 명령
- 그룹 삭제
 - » groupdel 명령

사용자 정보 관리 명령

- 사용자 로그인 정보 확인

- » who, w 명령

- » 현재 시스템에 로그인한 사용자의 이름과 로그인한 단말기 번호, 로그인 시간 등의 정보를 출력

- 로그인 이력 확인

- » last 명령

- » 사용자 이름 로그인 시간, 로그아웃 시간, 터미널 번호, IP 주소 등 출력

- 사용자 UID 확인

- » whoami, who am i, id 명령

- 소속 그룹 확인

- » groups 명령

- » 현재 사용자 계정이 속해 있는 그룹 확인

사용자 정보 관리 명령

- root 권한 사용

- » su 명령을 사용하여 root 계정으로 전환
- » sudo 명령 사용

- sudo 권한 설정

```
[root@centos-server ~]#  
[root@centos-server ~]# visudo  
[root@centos-server ~]#  
[root@centos-server ~]#  
  
## The COMMANDS section may have other options added to it.  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)      ALL  
ssac    ALL=(ALL)      ALL  
  
## Allows members of the 'sys' group to run networking, software,  
## service management apps and more.  
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE  
, DRIVERS
```

- sudo 명령 사용

- » sudo [실행할 명령문]

사용자 정보 관리 명령

- 암호 관리
 - » passwd 명령
- 파일 및 디렉터리의 소유자 변경
 - » chown 명령
- 파일과 디렉터리의 소유 그룹 변경
 - » chgrp 명령



파일 속성

■ 파일 접근 권한 보호

- » 리눅스는 파일에 무단으로 접근하는 것을 방지하고 보호하는 기능을 제공
- » 사용자는 자신의 파일과 디렉터리 중에서 다른 사용자가 접근해도 되는 것과 그렇지 않은 것을 구분하여 접근 권한 제한

```
[root@centos-server ~]#  
[root@centos-server ~]# ls -l /etc/hosts  
-rw-r--r--. 1 root root 75 2월 11 22:29 /etc/hosts  
[root@centos-server ~]#
```

속성 값	설명
-	파일의 종류(- : 일반 파일, d : 디렉터리)
rw-r-r--	파일에 대한 읽기, 쓰기 실행 권한 표시 (소유자, 소유자그룹, 기타 사용자)
1	하드 링크 개수
root	파일 소유자의 로그인 ID
root	파일 소유자의 그룹
158	파일 크기
2월 11 22:09	마지막 수정 일자
/etc/hosts	파일 이름

파일 접근 권한

▪ 접근 권한 표기

- » 사용자 카테고리별로 세 가지 권한을 rwx 세 문자를 끌어서 표기
- » 읽기 권한은 r, 쓰기 권한은 w, 실행 권한은 x로 표시.
- » 해당 권한이 없는 경우에는 -로 표기

접근 권한	의미
rwxr-xr-x	소유자는 읽기·쓰기·실행 권한을 모두 가지고 그룹과 기타 사용자는 읽기·실행 권한을 가지고 있다.
r-xr-xr-x	소유자, 그룹, 기타 사용자 모두 읽기·실행 권한을 가지고 있다.
rw-----	소유자만 읽기·쓰기 권한을 가지고 그룹과 기타 사용자는 아무 권한이 없다.
rw-rw-rw-	소유자, 그룹, 기타 사용자 모두 읽기·쓰기 권한을 가지고 있다.
rwxrwxrwx	소유자, 그룹, 기타 사용자 모두 읽기·쓰기·실행 권한을 가지고 있다.
rwx-----	소유자만 읽기·쓰기·실행 권한을 가지고 그룹과 기타 사용자는 아무 권한이 없다.
r-----	소유자만 읽기 권한을 가지고 있다.

접근 권한을 숫자로 표시

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

파일 접근 권한

■ 접근 권한 변경

» chmod 명령

» 기호 모드 또는 숫자 모드를 사용해서 변경

» 기호 모드

구분	문자/기호	의미	권한 표기	의미
사용자 카테고리 문자	u	파일 소유자	u+w	소유자(u)에게 쓰기(w) 권한 부여(+)
	g	파일 소유 그룹	u-x	소유자(u)의 실행(x) 권한 제거(-)
	o	소유자와 그룹 이외의 기타 사용자	g+w	그룹(g)에 쓰기(w) 권한 부여(+)
	a	전체 사용자	o-r	기타 사용자(o)의 읽기(r) 권한 제거(-)
연산자 기호	+	권한 부여	g+wx	그룹(g)에 쓰기(w)와 실행(x) 권한 부여(+)
	-	권한 제거	+wx	모든 사용자에게 쓰기(w)와 실행(x) 권한 부여(+)
	=	접근 권한 설정	a+rwx	모든 사용자에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(=)
접근 권한 문자	r	읽기 권한	u=rwx	소유자(u)에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(=)
	w	쓰기 권한	go+w	그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)
	x	실행 권한	u+x, go+w	소유자(u)에게 실행(x) 권한을 부여하고(+) 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(=)

파일 접근 권한

- 접근 권한 변경

- » chmod 명령

- 숫자 모드

접근 권한	환산	숫자	의미
rwx	111 → 4+2+1	7	읽기, 쓰기, 실행
rw-	110 → 4+2+0	6	읽기, 쓰기
r-x	101 → 4+0+1	5	읽기, 실행
r--	100 → 4+0+0	4	읽기
-wx	011 → 0+2+1	3	쓰기, 실행
-w-	010 → 0+2+0	2	쓰기
--x	001 → 0+0+1	1	실행
---	000 → 0+0+0	0	권한이 없음

기본 접근 권한 설정

▪ 기본 접근 권한

- » 파일이나 디렉터리를 생성할 때 기본 접근 권한 자동 설정
- » 일반 파일의 경우 소유자와 그룹은 읽기와 쓰기 권한이 설정되고 기타 사용자는 읽기 권한만 설정
- » 디렉터리의 경우 소유자와 그룹은 읽기, 쓰기, 실행 권한이 설정되고 기타 사용자는 읽기, 실행 권한만 설정

▪ 기본 접근 권한 확인 및 변경

- » mask → 파일이나 디렉터리 생성 시 부여하지 않을 권한 정보
- » mask 적용
 - › 파일 : 666 - mask
 - › 디렉터리 : 777 - mask

마스크 값	일반 파일	디렉터리	의미
022	644	755	그룹과 기타 사용자는 읽기만 가능하다.
077	600	700	그룹과 기타 사용자의 접근 권한을 모두 제거한다.
027	640	750	그룹은 읽기와 실행만 가능하고 기타 사용자의 접근 권한을 모두 제거한다.

기본 접근 권한 설정

- 기본 접근 권한 확인 및 변경 (계속)
 - » umask 명령 → mask 확인 또는 수정

```
[root@centos-server ~]#  
[root@centos-server ~]# umask  
0022  
[root@centos-server ~]#
```

- -----w--w-
- 소유자 그룹과 기타 사용자에게 쓰기 권한은 부여하지 않도록 설정



RPM 패키지의 특징과 구조

- RPM

- » Redhat의 패키지 관리 도구

- RPM의 특징

- » 패키지의 파일들을 컴파일 없이 관련 디렉터리로 바로 설치
 - » 설치된 패키지 파일의 일괄 삭제 가능
 - » 기존에 설치된 패키지를 삭제하지 않고 업그레이드 가능
 - » 패키지의 설치 상태 검증 가능
 - » 패키지에 대한 정보 제공

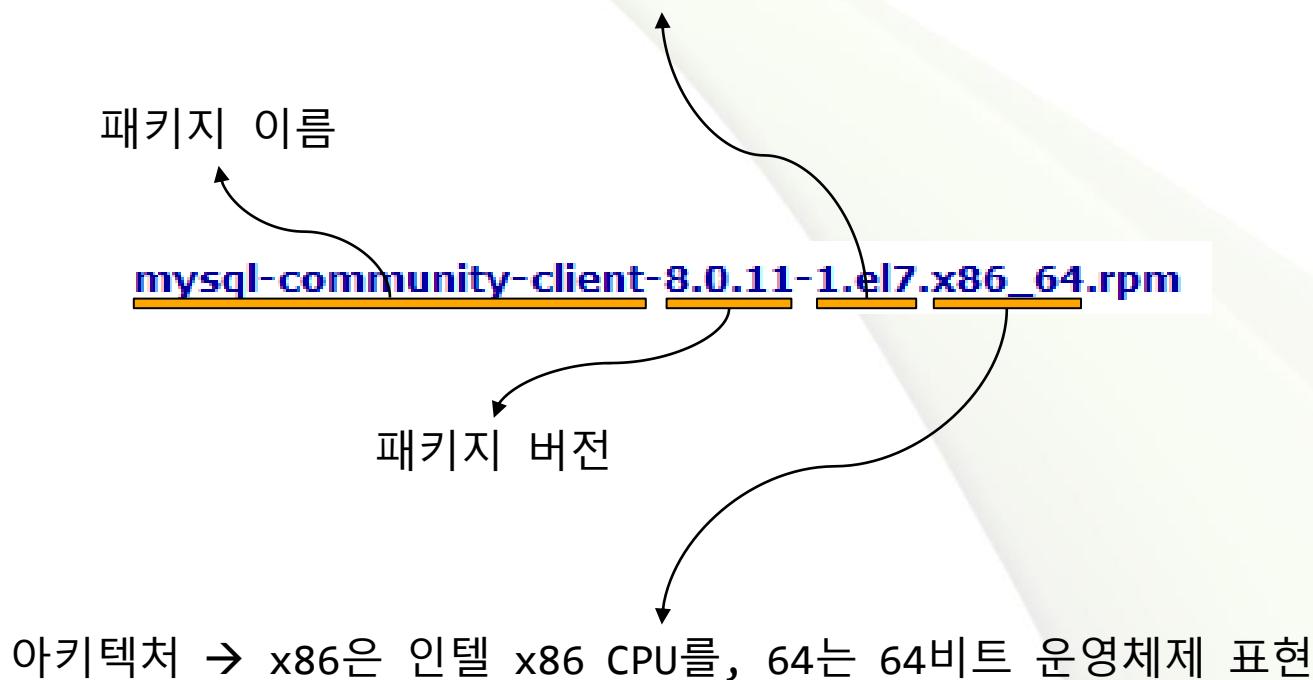
- RPM의 단점

- » 패키지 의존성에 따라 관련 패키지가 먼저 설치되어 있지 않으면 설치할 수 없음
 - » RPM의 단점을 해결하기 위해 yum 등장

RPM 패키지의 이름 구성

패키지 릴리즈

- 릴리스는 해당 버전에서 몇 번째 만들어진 것인지 표현
- '1.el7': 1은 첫 번째 만들어진 패키지 표현
- el7은 CentOS 7용도의 패키지 표현



RPM 패키지 관리

■ RPM 패키지 설치

```
rpm -i [설치옵션] [패키지명]
```

- **기능** RPM 패키지를 설치한다.
- **설치옵션** -h: #(해시)를 출력한다.
 - v: 설치 과정에 대한 메시지를 출력한다.
 - replacefiles: 이미 설치된 다른 패키지의 파일을 덮어쓰면서라도 패키지를 강제로 설치한다.
 - replacepkgs: 패키지가 이미 설치되어 있어도 다시 설치한다.
 - test: 설치하지는 않고 충돌 사항이 있는지만 점검하고 보고한다.

■ RPM 패키지 업그레이드 설치

» 패키지를 업그레이드할 때는 -Uvh 옵션을

» -U 옵션 → 기존에 설치되어 있던 패키지는 업그레이드 설치, 새로운 패키지는 그대로 설치

```
rpm -U [설치옵션] 패키지명
```

- **기능** RPM 패키지를 업그레이드한다.
- **설치옵션** -h: #를 출력한다.
 - v: 설치 과정에 대한 메시지를 출력한다.
 - U: 패키지를 업그레이드하여 설치한다.

RPM 패키지 관리

■ RPM 패키지 정보 검색

» rpm 명령으로 패키지의 정보를 검색할 때는 -q 옵션 사용

rpm -q [질의옵션]

- **기능** RPM 패키지의 정보를 검색한다.
- **질의옵션** -a: 전체 패키지의 목록을 출력한다.
 - f 파일명: 파일명을 포함하고 있는 패키지명을 출력한다.
 - p 패키지명: 지정한 패키지의 상세 정보를 출력한다.
 - i: 자세한 정보를 출력한다.
 - R: 의존하고 있는 패키지의 목록을 출력한다.
 - l: 패키지 내의 파일을 출력한다.
 - s: 패키지 안에 들어 있는 파일의 상태를 출력한다.
 - d: 문서 파일만 출력한다.
 - c: 설정 파일만 출력한다.

■ RPM 패키지 삭제

» rpm 명령으로 패키지를 삭제할 때는 -e 옵션을 사용

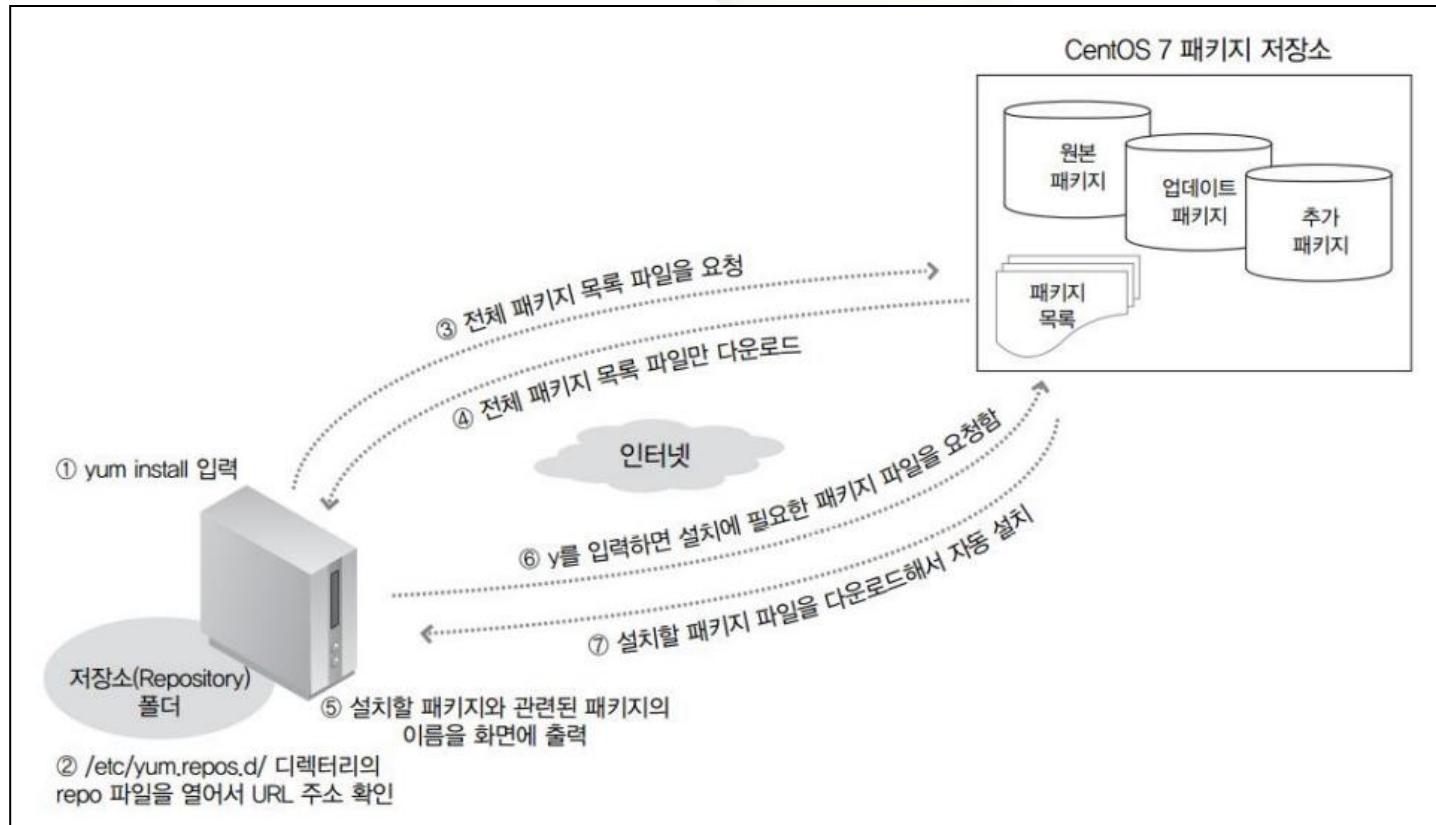
rpm -e 패키지명

- **기능** RPM 패키지를 삭제한다.

yum을 사용해서 패키지 설치

■ yum (Yellowdog Updater Modified)

- » rpm의 패키지 의존성 문제 해결
- » 인터넷을 통해 필요한 파일을 저장소에서 자동으로 다운로드하고 설치
- » centOS 8 부터는 dnf 명령으로 대체



yum을 사용해서 패키지 설치

- 패키지 목록 확인

- » `yum list [패키지이름]`

- » 패키지이름을 지정하지 않으면 전체 패키지 목록 표시

- 패키지 정보 확인

- » `yum info 패키지이름`

- 패키지 설치

- » `yum install 패키지_이름`

- » 패키지 이름에 .rpm 파일을 지정하면 .rpm 파일을 로컬 설치

- » 패키지를 설치할 때 의존성이 있는 패키지는 자동으로 함께 설치

- 패키지 그룹 설치

- » `yum groupinstall "패키지그룹이름"`

yum을 사용해서 패키지 설치

- 업데이트 확인
 - » `yum check-update` 명령
- 패키지 업데이트
 - » `yum update [패키지이름]`
- 패키지 삭제
 - » `yum remove 패키지이름`

파일 아카이브와 압축

■ 파일 아카이브

- » 리눅스에서 '파일을 묶어서 하나로 만든 것'의 의미로 사용
- » tar 명령을 사용해서 다른 시스템과 파일을 주고받거나, 백업을 하기 위해 여러 파일이나 디렉터리를 하나의 아카이브 파일로 생성하거나, 기존 아카이브에서 파일 추출

■ 아카이브 생성

- » tar cvf
 - › c옵션 : 아카이브 생성
 - › v옵션 : 처리 중인 파일의 이름을 화면에 출력
 - › f옵션 : f 다음에 오는 이름을 생성될 아카이브 이름으로 지정

■ 아카이브 내용 보기

- » tar tvf
 - › t옵션 : 아카이브의 내용 확인
 - › v옵션, f옵션은 생성과 동일

파일 아카이브와 압축

- 아카이브 내용 보기

- » `tar tvf`

- › t옵션 : 아카이브의 내용 확인
 - › v옵션, f옵션은 생성과 동일

- 아카이브 풀기

- » `tar xvf`

- › x옵션 : 아카이브 풀기
 - › v옵션, f옵션은 생성과 동일

파일 아카이브와 압축

- 압축 파일 종류
 - » xz, bz2, gz, zip 등
- 파일 압축 1
 - » gzip 명령
 - » 지정한 파일을 압축하여 파일명.gz 파일 생성
- 압축 파일 내용 보기 1
 - » zcat 명령
- 압축 풀기 1
 - » gunzip 명령
 - » 지정한 파일의 압축을 풀어서 원본 파일 복원

파일 아카이브와 압축

- 파일 압축 2
 - » bzip2 명령
 - » 지정한 파일을 압축해서 파일명.bz2 파일 생성
 - » bzip2는 gzip에 비해 대체로 압축률은 좋지만 속도가 다소 느림
- 압축 파일 내용 보기 1
 - » bzcat 명령
- 압축 풀기 2
 - » bunzip2 명령
 - » 지정한 파일의 압축을 풀어서 원본 파일 복원



출력 리다이렉션

▪ 출력 방향 변경

표현	설명
>	<ul style="list-style-type: none">▪ 기존 파일의 내용을 삭제하고 새로 결과를 저장▪ 지정한 이름의 파일이 없으면 파일을 생성해서 명령의 수행 결과를 저장하고, 파일이 있으면 이전의 내용을 지우고 명령의 수행 결과로 대체
>>	<ul style="list-style-type: none">▪ 기존 파일의 내용 뒤에 출력 내용 추가▪ 지정한 파일명의 파일이 없으면 파일을 생성하고, 해당 파일이 있으면 그 파일의 끝에 명령의 실행 결과를 추가

파이프

- 명령과 명령을 연결

- » 왼쪽 명령의 실행 결과를 오른쪽 명령의 입력으로 사용

```
[root@centos-server ~]#  
[root@centos-server ~]# ls -l /etc | less  
합계 1064  
-rw-r--r--. 1 root root      5090 11월 16 23:40 DIR_COLORS  
-rw-r--r--. 1 root root      5725 11월 16 23:40 DIR_COLORS.256color  
-rw-r--r--. 1 root root      4669 11월 16 23:40 DIR_COLORS.lightbgcolor  
-rw-r--r--. 1 root root       94  3월 25  2017 GREP_COLORS  
drwxr-xr-x. 7 root root     134   2월 11 22:37 NetworkManager  
drwxr-xr-x. 5 root root      57   2월 11 22:12 X11  
-rw-r--r--. 1 root root      16   2월 11 22:17 adjtime  
-rw-r--r--. 1 root root     1529  4월   1  2020 aliases  
-rw-r--r--. 1 root root    12288  2월 11 22:20 aliases.db  
drwxr-xr-x. 2 root root     236   2월 11 22:37 alternatives  
-rw-----. 1 root root      541   8월   9  2019 anacrontab  
-rw-r--r--. 1 root root      55   8월   8  2010 asound.conf
```

필터

- 조건에 따라 결과 필터링
 - » grep, head, tail, ...
 - » 주로 pipe와 연결해서 사용

```
[root@centos-server ~]#  
[root@centos-server ~]# ls -l /etc | grep shadow  
-----. 1 root root 400 2월 11 22:50 gshadow  
-----. 1 root root 388 2월 11 22:50 gshadow-  
-----. 1 root root 604 2월 11 22:50 shadow  
-----. 1 root root 585 2월 11 22:17 shadow-  
[root@centos-server ~]#
```

프로세스

- 현재 시스템에서 실행 중인 프로그램
- 모든 프로세스는 부모-자식 관계를 가지고 있음
- 부모 프로세스는 자식 프로세스를 생성하고, 자식 프로세스는 또 다른 자식 프로세스를 만들 수 있음
 - » 자식 프로세스는 할 일이 끝나면 부모 프로세스에 결과를 돌려주고 종료
 - › 사용자가 vi를 실행하여 셸이 vi 프로세스를 생성할 경우, 셸은 부모 프로세스가 되고 vi는 자식 프로세스
 - › 사용자가 vi를 종료하면 다시 부모 프로세스인 셸로 전환
- 프로세스의 번호 (PID)
 - » 각 프로세스에는 부여된 고유한 식별 번호
 - » PID는 1번부터 시작하고 프로세스가 실행되면서 하나씩 증가하여 부여

프로세스 관리 명령

- 실행 중인 프로세스 목록 확인

- » ps 명령

```
[root@centos-server ~]#  
[root@centos-server ~]# ps -ef  
UID      PID  PPID  C STIME TTY          TIME CMD  
root        1      0  0 15:10 ?          00:00:02 /usr/lib/systemd/systemd --switched-  
root        2      0  0 15:10 ?          00:00:00 [kthreadd]  
root        4      2  0 15:10 ?          00:00:00 [kworker/0:0H]  
root        5      2  0 15:10 ?          00:00:00 [kworker/u2:0]  
root        6      2  0 15:10 ?          00:00:01 [ksoftirqd/0]
```

- 프로세스 정보 검색

- » ps 명령과 grep 명령을 |(파이프)로 연결하여 특정 프로세스에 대한 정보 검색

```
[root@centos-server ~]#  
[root@centos-server ~]# ps -ef | grep k  
root        2      0  0 15:10 ?          00:00:00 [kthreadd]  
root        4      2  0 15:10 ?          00:00:00 [kworker/0:0H]  
root        5      2  0 15:10 ?          00:00:00 [kworker/u2:0]  
root        6      2  0 15:10 ?          00:00:01 [ksoftirqd/0]  
root       13      2  0 15:10 ?          00:00:00 [kdevtmpfs]  
root       15      2  0 15:10 ?          00:00:00 [khungtaskd]
```

프로세스 관리 명령

■ 프로세스 종료

- » kill 명령 사용 : 시그널을 보내 프로세스를 종료
- » kill [signal] PID

시그널	번호	기본 처리	의미
SIGHUP	1	종료	터미널과의 연결이 끊어졌을 때 발생한다.
SIGINT	2	종료	인터럽트로 사용자가 <code>Ctrl+C</code> 를 입력하면 발생한다.
SIGQUIT	3	종료, 코어덤프	종료 신호로 사용자가 <code>Ctrl+\</code> 을 입력하면 발생한다.
SIGKILL	9	종료	이 시그널을 받은 프로세스는 무시할 수 없으며 강제로 종료된다.
SIGALRM	14	종료	알람에 의해 발생한다.
SIGTERM	15	종료	kill 명령이 보내는 기본 시그널이다.