리눅스(Linux)

동의과학대학교 컴퓨터정보과

학습내용

- 로그 기록 – script
- 파일/파일 내용 검색 – find, whereis, which, grep
- 유용한 명령어와 기능
- root 권한 명령
- 사용자 관리
- 파일 권한 관리
- 파일 압축/풀기
- 파일 묶기
- 패키지 설치

로그 기록

- 로그 기록 파일로 남기기
  - 로그 시작
    - $ **script** <파일명>
  - 로그 종료
    - $ **exit**

파일명이 없으면 기본 파일명 : typescript가 생성됨

또는 Ctrl + D 키

```
jinsook@DESKTOP-VG22TUO:~$ ./hello.sh
오늘의 이야기를 듣고 싶으신가요? (y/n) : y
당신에게 오늘은 엄청난 행운이 있을거예요! 즐거운 하루 보내세요^^
jinsook@DESKTOP-VG22TUO:~$ ./hello.sh
오늘의 이야기를 듣고 싶으신가요? (y/n) : n
당신은 다른 계획이 있으시군요^^ 즐거운 하루 되길 바랍니다~
```

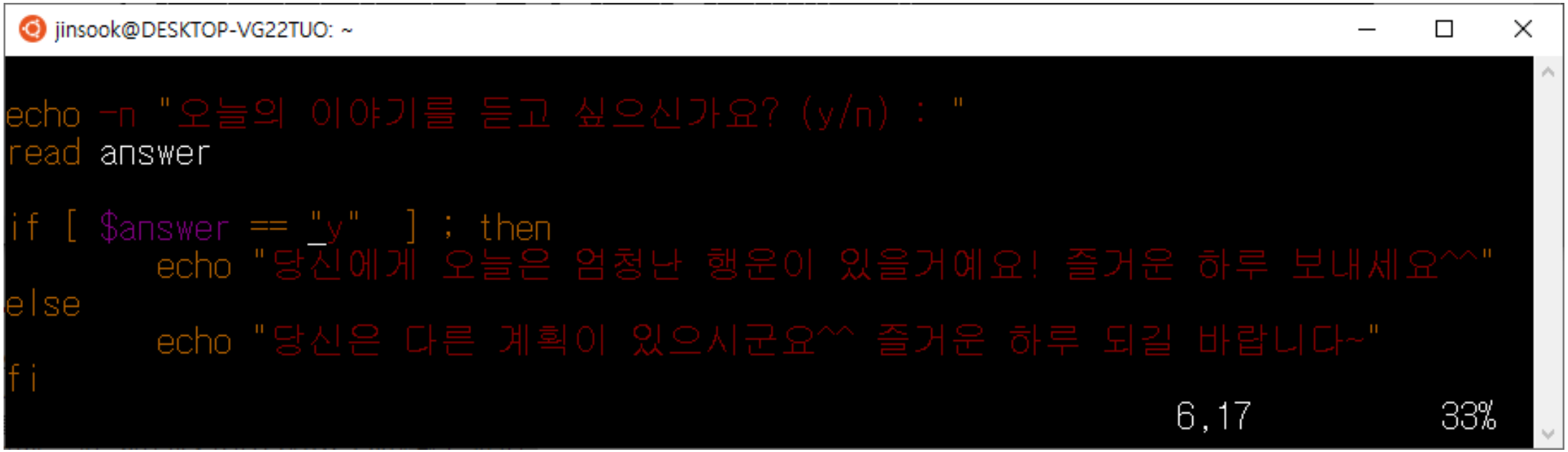
실습

1. 오늘의 날짜로 다음의 작업을 저장하는 로그파일 생성
2. 홈디렉토리에 디렉토리 mystuff를 만들기
3. mystuff 내에 vi로 shell script인 hello.sh을 작성
   - 사용자가 “y”를 입력하거나 “n”를 입력하면 다른 메시지 출력
4. 로그 종료
5. 로그 파일 내용 확인

실습

- 다음 결과를 출력하는 shell script를 작성하시오

```
jinsook@DESKTOP-VG22TUO:~$ ./hello.sh
오늘의 이야기를 듣고 싶으신가요? (y/n) : y
당신에게 오늘은 엄청난 행운이 있을거예요! 즐거운 하루 보내세요^^
jinsook@DESKTOP-VG22TUO:~$ ./hello.sh
오늘의 이야기를 듣고 싶으신가요? (y/n) : n
당신은 다른 계획이 있으시군요^^ 즐거운 하루 되길 바랍니다~
```



파일 위치, 경로 검색

- **find** : 실제 파일 시스템에서 파일을 검색
  - $find <검색디렉토리> -name <검색대상> **파일명으로 검색 옵션**
    - 하위 디렉토리, 숨겨진 파일도 표시

```
$ find /etc -name "*.conf"
$ find . –name “test*” –type d          #디렉토리만 검색
$ find /bin -size +1000k                #size 옵션으로 파일크기 지정
$ find . –empty                         #빈파일을 검색
$ find . –newer <파일명>                #<파일명>보다 최근에 변경된 파일 검색
$ find . –empty –exec ls –l {} ₩;       #검색한 파일로 부가적인 작업 수행
```

실습

```
$ find /lib -name "vars.sh"
```

1. /lib 디렉토리와 그 하위 디렉토리에서 “vars.sh”파일 이름으로 찾기
2. /usr 에서 “games” 디렉토리 찾기

```
$ find /usr -name "games" -type d
```

3. hello.sh 파일보다 최근에 만들어진 파일 검색

```
$ find . -newer hello.sh
```

파일 검색

- **whereis** : 명령어에 해당되는 바이너리와 매뉴얼 위치, 경로 출력
  - $whereis <명령어>

```
$ whereis ls          #ls의 실행파일, 소스파일, 매뉴얼 출력
$ whereis –b ls       #옵션 –b : 바이너리 파일
```

-b : 바이너리
-m : 매뉴얼
-s : 소스파일 등

파일 검색

#환경변수 확인하기

$ echo $PATH

- **which** : 특정 명령어의 위치 출력
  - 환경변수 PATH에 설정된 디렉터리만 검색
  - 현재 사용하고 있는 명령어의 실행파일의 위치를 알 수 있음

$ which find #find 명령어 위치 검색

실습

- cd, script, mkdir, rm, ls, man 등 이제까지 학습한 명령어들의 위치와 매뉴얼의 위치를 whereis와 which로 파악 하시오.

내용 검색

- **grep** : 파일 내에서 지정한 문자열을 찾아 문자열을 포함한 모든 행 출력
  - 텍스트 파일에서 특정 패턴의 문자열을 찾아 출력
  - $ grep [-옵션] 패턴 파일명(들)

```
$ grep date test01.sh
$ grep –n date  test02.sh
$ grep –c “echo” test01.sh test02.txt
```

[주요 옵션]
-c : 패턴이 일치하는 행의 수를 출력
-i : 비교 시 대소문자를 구별 안함
-v : 지정한 패턴과 일치하지 않는 행만 출력
-n : 행의 번호를 함께 출력
-l : 패턴이 포함된 파일의 이름을 출력
-w : 패턴이 전체 단어와 일치하는 행만 출력
-r : 현재 및 서브디렉토리 모든 파일에서 일치하는 문자열 출력

```
jinsook@DESKTOP-VG22TUO:~$ grep -n echo hello.sh
3:echo -n "오늘의 이야기를 듣고 싶으신가요? (y/n) : "
7:      echo "당신에게 오늘은 엄청난 행운이 있을거예요! 즐거운 하루 보내세요^^"
9:      echo "당신은 다른 계획이 있으시군요^^ 즐거운 하루 되길 바랍니다~"
jinsook@DESKTOP-VG22TUO:~$
```

참고 – 정규식(Regular Expression)

메타문자	기능
^	행의 시작 지시자
$	행의 끝 지시자
.	하나의 문자와 대응
?	앞 문자가 0 또는 한 개
.*	0이거나 그 이상의 문자
*	앞 문자가 하나이거나 반복된 것 의미
|	또는
[ ]	리스트 중의 한 문자
[0-9]	0,1,2,3,4,5,6,7,8,9 중 하나
[^]	[]안에서 ^는 제외의 의미
₩	지정된 문자의 특징 무시
₩<	단어의 시작 지시자
₩>	단어의 끝 지시자
x₩{m₩}	문자 x를 m번 반복
x₩{m,₩}	문자 x를 적어도 m번 반복

실습

명령어	설명
$ grep '^a' 파일명	^는 파일의 시작을 나타냄. 파일에서 a로 시작하는 행을 찾는다.
$ grep 'apple$' 파일명	$는 파일의 끝을 나타냄. 파일에서 e로 끝나는 행을 찾는다.
$ grep 'app*' 파일명	파일에서 app로 시작하는 모든 단어를 찾는다.
$ grep 'a.....e' 파일명	파일에서 a로 시작하고 e로 끝나는 7자리 단어를 찾는다.
$ grep [a-d] 파일명	파일에서 a,b,c,d 로 시작하는 단어를 모두 찾는다.
$ grep [aA]pple 파일명	파일에서 apple 또는 Apple로 시작하는 단어를 모두 찾는다.
$ grep 'apple' d*	d로 시작하는 모든 파일에서 apple 를 포함하는 모든 행을 찾는다.
$ grep 'apple' 파일명1 파일명2	지정된 두개의 파일에서 apple 를 포함하는 모든 행을 찾는다.
$ grep '^[ab]' 파일명	파일에서 a나 b로 시작되는 모든 행을 찾는다.

실습

- 다음의 텍스트에서 검색하시오.

1. a가 포함된 test03의 문장 검색
```
$ grep a* test03
```
2. a, b 또는 c로 시작하는 test03의 문장 검색
```
$ grep [a,b,c,]* test03
```
3. test03에서 p로 시작되는 단어와 번호가 출력
```
$ grep -n ^[p]* test03
```
4. test03에서 '.응'으로 시작하는 문장 검색
```
$ grep ₩.응 test03
```
5. 홈디렉토리에서 ls –la 명령의 결과를 가지고 'bash' 문장 검색(| 사용)
```
$ ls -la | grep bash
```
6. 컴퓨터에서 돌아가는 프로세스(명령어 : ps aux) 중tomcat 프로세서 출력
```
$ ps aux | grep tomcat
```

jinsook@klein: ~
```
2020-9-28
apple
banana
cherry
coconut
2550000원
pear
grape
potato
peach
melon
tomato
즐거운 추석
보람된 한가위
해피 추석
.응답하라 1988
.응답하라 1994
.응답하라 1997
```

유용한 명령어들

- **head, tail**

  텍스트로 작성된 파일의 앞 10행 또는 마지막 10행만 출력(행 개수 조절 가능 : -5 등등)

  예) $ head /etc/passwd

  예) $ tail -5 /etc/passwd

- **more**

  텍스트로 작성된 파일을 화면에 페이지 단위로 출력

  예) $ more /etc/passwd

- **less**

  more와 용도가 비슷하지만 기능이 더 확장된 명령

  예) $ less /etc/passwd

- **file**

  File이 어떤 종류의 파일인지를 표시

  예) $ file hello.sh

history와 자동완성

- **history** : 현재 세션에서 실행한 명령어 목록 출력
  - $ history
  - $!3 : history에서 3번째 명령 실행하기
  - $!! : 직전 명령 실행하기

- 자동 완성
  - 파일명의 일부만 입력한 후에 **Tab**키를 눌러 나머지 파일명을 자동으로 완성하는 기능
  - cd /etc/sysconfig/network-scripts/ 를 입력하려면
    cd /et[**Tab**키]sysco[**Tab**키]network[**Tab**키]

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

순차적 명령 수행

- 세미콜론(;)으로 다수의 명령어를 순차적으로 한 번에 실행하기
-

```
jinsook@klein:~$ mkdir dir1
jinsook@klein:~$ cd dir1
jinsook@klein:~/dir1$ pwd
/home/jinsook/dir1
jinsook@klein:~/dir1$
```

```
jinsook@klein:~$ mkdir dir1;cd dir1;pwd
/home/jinsook/dir1
jinsook@klein:~/dir1$
```

Pipe(|)

- 둘 이상의 명령을 함께 묶어 출력의 결과를 다른 명령의 입력으로 전환하는 기능
  - '|' 앞의 명령 결과가 '|' 뒤의 명령의 입력 데이터로 사용

```
jinsook@klein:~$ ls --help | grep access
                             modification time: atime or access or use (-u);
  -u                        with -lt: sort by, and show, access time;
                             with -l: show access time and sort by name;
                             otherwise: sort by access time, newest first
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).
jinsook@klein:~$ ls --help | grep access | grep time
                             modification time: atime or access or use (-u);
  -u                        with -lt: sort by, and show, access time;
                             with -l: show access time and sort by name;
                             otherwise: sort by access time, newest first
jinsook@klein:~$
```

IO Redirection

- redirection : standard stream의 흐름을 바꿈
  - 사용 기호 : <, >, <<, >>
    - < filename : 표준**입력**으로 filename 사용
    - > filename : 표준**출력**으로 filename 사용
      - 1> : standard output
      - 2> : standard error
    - << string : **string 문자열**이 입력될 때까지 입력을 받는다.
    - >> filename : 기존에 filename이 존재하면 그 뒤에 추가 한다.

- 휴지통 기능
$ ls –a > /dev/null :

list 파일 생성
$ ls –a > lista.txt
$ cat lista.txt

```
jinsook@klein:~$ rm file.txt 1>result.txt 2>error.log
jinsook@klein:~$ cat result.txt
jinsook@klein:~$ cat error.log
rm: cannot remove 'file.txt': No such file or directory
jinsook@klein:~$
```

```
jinsook@klein:~$ cat << hi > out.txt
> Hello, I am very happy
> to see you!
> he
> hi
jinsook@klein:~$ cat out.txt
Hello, I am very happy
to see you!
he
jinsook@klein:~$
```

실습

- 다음의 지시사항에 따라 실습하시오.

1. ls 명령어의 결과를 ls.txt에 저장
2. ls.txt의 내용을 화면에 출력하여 확인
3. ls.txt를 head 명령어의 입력 스트림으로 받아 처음 5행을 출력하여 head.txt 파일에 저장
4. head.txt 파일 화면 출력

```
jinsook@klein:~$ ls >ls.txt
jinsook@klein:~$ cat ls.txt
dir1
error.log
ex04
first.txt
jinsook
ls.txt
out.txt
result.txt
second.txt
seconde.txt
test
typescript
별헤는밤.txt
서시.txt
jinsook@klein:~$ head <ls.txt >head.txt
jinsook@klein:~$ cat head.txt
dir1
error.log
ex04
first.txt
jinsook
ls.txt
out.txt
result.txt
second.txt
seconde.txt
jinsook@klein:~$
```

sudo(substitute User Do) 명령어

- 현재 사용자 계정에서 일시적으로 root권한으로 명령어를 실행
  - sudo useradd –m user01 # user01을 만들면서 홈디렉토리도 생성(-m)
  - sudo userdel –r user01 # user01을 삭제하면서 홈디렉토리도 삭제(-r)
  - sudo passwd [root] # root의 비밀번호를 변경
- sudo 명령으로 root 계정으로 전환
  - sudo -s : 현재 디렉토리 유지하면서 관리자 권한 획득
  - sudo –i : /root 디렉토리로 이동. 계정 자체가 관리자로 변경

```
jinsook@DESKTOP-VG22TU0:~$ useradd -m user02
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
jinsook@DESKTOP-VG22TU0:~$ sudo useradd -m user02
[sudo] password for jinsook:
jinsook@DESKTOP-VG22TU0:~$ cd /home
jinsook@DESKTOP-VG22TU0:/home$ ls
jinsook  ubuntu  user01  user02
jinsook@DESKTOP-VG22TU0:/home$
```

```
jinsook@DESKTOP-VG22TU0:~$ sudo -i
root@DESKTOP-VG22TU0:~# pwd
/root
root@DESKTOP-VG22TU0:~# exit
logout
jinsook@DESKTOP-VG22TU0:~$ sudo -s
Hi, root nice to meet you!!
root@DESKTOP-VG22TU0:~# pwd
/home/jinsook
root@DESKTOP-VG22TU0:~# exit
exit
jinsook@DESKTOP-VG22TU0:~$
```

- 프롬프트 기호
$: 일반사용자
: root 사용자

사용자 관련 명령

- su [사용자] : 로그아웃하지 않고 다른 사용자의 계정으로 전환
  - [사용자]가 없으면 su root 와 같음
- su – [사용자] : 사용자 계정으로 전환하고 해당 사용자 환경으로 쉘을 실행

구분		su	su -
환경변수	TERM	변경	변경
	HOME	변경	변경
	SHELL	변경	변경
	USER	변경	변경
	LOGNAME	변경	변경
	PATH	유지	Default
	기타	유지	초기화
워킹디렉토리		유지	변경

su 또는 su root는 암호를 알아야 사용할 수 있으나 sudo –s는 허가된 사용자라면 본인의 암호를 사용하여 관리자 권한을 얻을 수 있음

사용자 관리

- 리눅스는 다중 사용자 시스템(Multi-User System)
- root : 슈퍼 유저(superuser)로 모든 작업을 할 수 있는 권한을 가짐
  - 사용자 관리는 root 계정으로 실행해야 함
- 모든 사용자는 하나 이상의 그룹에 소속됨
- 관련 파일
  - **/etc/passwd** : 사용자 정보
  - **/etc/shadow** : 사용자 비밀번호 정보
  - **/etc/group** : 그룹 정보 파일

```
jinsook@DESKTOP-VG22TU0:~$
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

1. /etc 폴더의 passwd 파일의 앞에서 5개의 정보 출력

사용자명:암호:사용자 ID:사용자가 소속된 그룹 ID:전체 이름:홈 디렉터리:기본 셸

```
root@DESKTOP-VG22TU0:/home/jinsook#
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,jinsook
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
x:9:
```

2. /etc 폴더의 group 파일의 앞에서 10개의 정보 출력

그룹명:비밀번호:그룹 id:그룹에 속한 사용자명

3. /etc 폴더의 shadow 파일을 한 페이지씩 정보 출력

사용자 관리

- root의 권한이 필요

명령어	설명	예시
useradd	새로운 사용자를 추가	# useradd –m newuser
passwd	사용자의 비밀번호를 지정하거나 변경	# passwd newuser
usermod	사용자의 속성을 변경	# usermod -g root newuser # usermod –e 2020-10-07 user01
userdel	사용자를 삭제	# userdel –r newuser
chage	사용자의 암호를 주기적으로 변경하도록 설정	# chage -m 2 newuser
groups	현재 사용자가 속한 그룹을 보여줌	# groups

- 유사한 명령어로 adduser, deluser 등이 있다.

실습

- useradd와 userdel 명령어의 매뉴얼을 확인하시오.
  - 홈디렉토리를 함께 생성하는 useradd의 옵션은?
  - 홈디렉토리를 함께 삭제하는 userdel의 옵션은?

- user01, user02, user03을 추가하면서 홈디렉토리도 함께 생성하시오.

- user01를 'Oct 7, 2020'에 폐기되도록 설정하고 그 정보를 확인하시오.
  - chage 명령/ 옵션 –l , -E , -m 등을 매뉴얼에서 확인하시오.
  - usermod 명령 / 옵션 –e

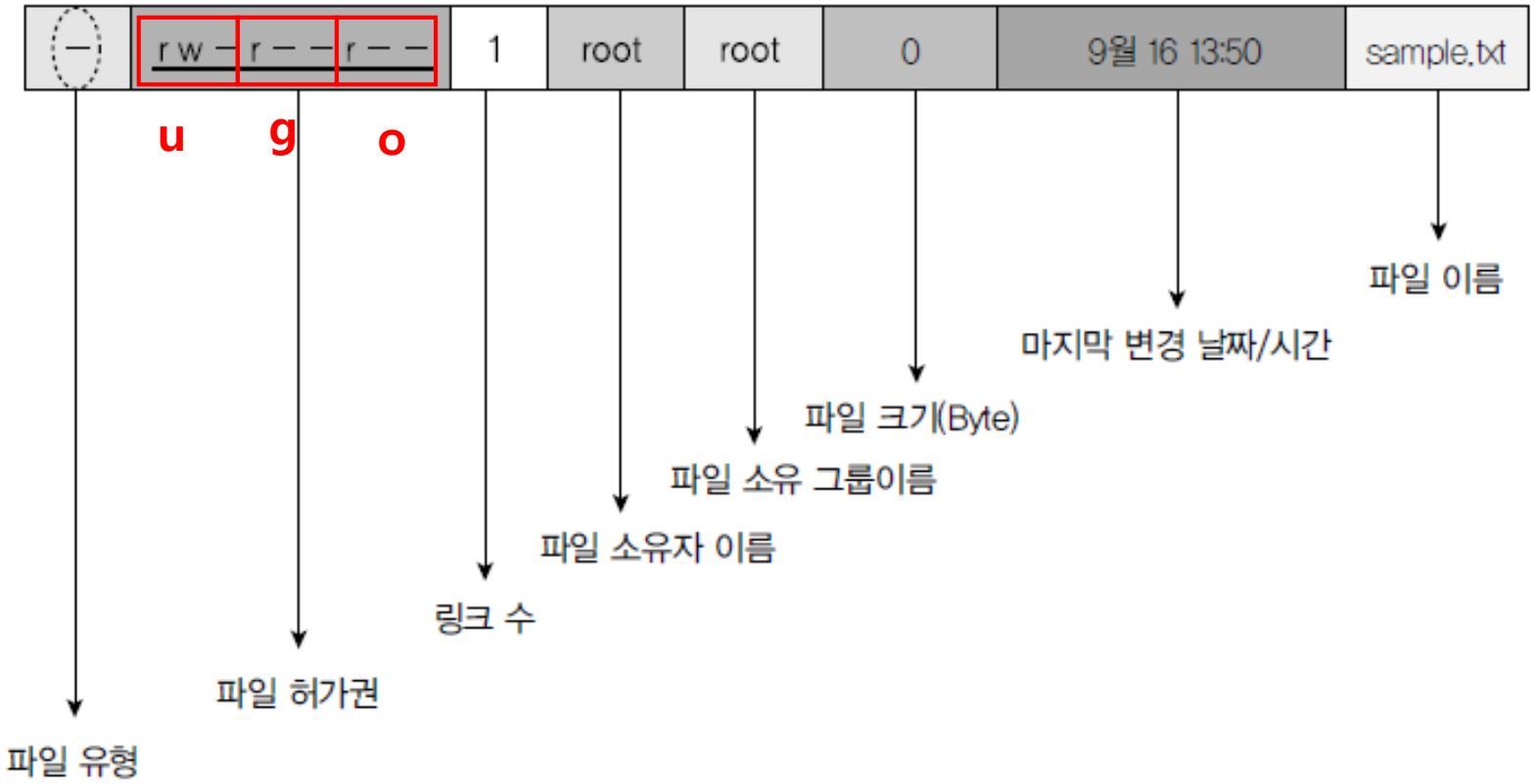
- user03를 홈디렉토리와 함께 삭제하시오.

```
jinsook@klein:~$ sudo useradd -m user01
jinsook@klein:~$ ls /home
jinsook  user01
jinsook@klein:~$ sudo -i
root@klein:~# sudo useradd -m user02
root@klein:~# ls /home
jinsook  user01  user02
root@klein:~# useradd -m user03
root@klein:~# ls /home
jinsook  user01  user02  user03
root@klein:~# chage -E 'Oct 30, 2019' user01
root@klein:~# chage -l user01
Last password change                                    : Oct 09, 2019
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Oct 30, 2019
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@klein:~# userdel -r user03
userdel: user03 mail spool (/var/mail/user03) not found
root@klein:~# ls /home
jinsook  user01  user02
root@klein:~#
```

파일 권한 관리

- 파일 속성 구조

```
-rw-rw-rw-  1 jinsook jinsook 67284 Oct  5 22:23 20191011.log
-rw-rw-rw-  1 jinsook jinsook     0 Oct  3 17:50 a.log
-rw-rw-rw-  1 jinsook jinsook     0 Oct  3 17:51 b.log
drwxrwxrwx  1 jinsook jinsook   512 Oct  5 20:09 bak
```



파일유형

문자	설명
-	일반파일
d	디렉토리
b	블록 디바이스
c	문자 디바이스
l	심볼릭 링크

파일 권한 관리

- 파일 유형
  - 디렉터리일 경우에는 **d**, 일반적인 파일일 경우에는 **-**가 표시

디렉토리의 경우 실행권한(x)이 있어야 디렉토리 진입 가능

- 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) u			그룹(Group) g			그 외 사용자(Other) o		
r	w	–	r	–	–	r	–	–
4	2	0	4	0	0	4	0	0
6			4			4		

파일 권한 관리

- **chmod** : 파일, 디렉토리의 권한(퍼미션, 허가권)을 변경
- 다음 명령을 설명하시오.
  - $ chmod g+w test.c
  - $ chmod o-r test.c
  - $ chmod u+x file.txt
  - $ chmod g+rwx file.txt
  - $ chmod go+r file.txt
  - $ chmod 000 test.c
  - $ chmod 777 test.c
  - $ chmod 744 test.c
  - $ chmod -R 777 backup

u : user
g : group
o : other
+ : 권한 부여
- : 권한 뺏기
-R : 하위 디렉토리 포함

파일 압축/풀기 명령

- 파일 압축 관련 명령
  - xz : 확장명 xz로 압축을 하거나 풀어준다
    예) xz 파일명
    xz -d 파일명.xz
  - bzip2 : 확장명 bz2로 압축을 하거나 풀어준다
    예) bzip2 파일명
    bzip2 -d 파일명.bz2
  - bunzip2 : “bzip2 -d”옵션과 동일한 명령어
  - gzip : 확장명 gz으로 압축을 하거나 풀어준다
    예) gzip 파일명
    gzip -d 파일명.gz
  - gunzip : “**gzip -d**”옵션과 동일한 명령어

실습

- xz, bzip2, bunzip2, gzip 의 매뉴얼을 찾아보고 다양한 옵션을 확인하세요.
- xz, bzip2, bunzip2, gzip 으로 파일을 압축해보고 가장 압축률이 높은 명령어를 확인하세요.

파일 묶기

- 파일 묶기(**tar**)
  - 리눅스에서 '파일 압축'과 '파일 묶기'는 원칙적으로 별개의 프로그램
  - 파일 묶기의 명령어는'tar'이며, 묶인 파일의 확장명도'tar'
- **tar** : 묶음 파일을 만들어 주거나 묶음을 풀어 줌
  - 동작 : c(묶기), x(풀기), t(경로확인)
  - 옵션 : f(파일), v(과정보이기), J(tar+xz), z(tar+gzip), j(tar+bzip2)
- 사용 예

```
# tar cvf my.tar /etc/sysconfig/ → 묶기
# tar cvfJ my.tar.xz /etc/sysconfig/ /etc/sysconfig/ → 묶기 + xz 압축
# tar xvf my.tar → tar 풀기
# tar xvfJ my.tar.xz /etc/sysconfig/ → xz 압축 해제 + tar 풀기
```

패키지 설치

- 패키지 : 배포된 리눅스에서 사용할 수 있는 소프트웨어
- 패키지 매니저 : 패키지를 다운로드, 설치, 삭제, 관리하는 소프트웨어
- **apt** : advanced Package
- 패키지 설치는 관리자 권한 필요
- git 패키지 설치
  - `$ sudo apt install git`

설치 명령

명령	설명
apt **install**	패키지 설치
apt **remove**	패키지 삭제
apt **purge**	패키지와 관련 설정 제거
apt **update**	레파지토리 인덱스 갱신
apt **upgrade**	업그레이드 가능한 모든 패키지 업그레이드
apt **autoremove**	불필요한 패키지 제거
apt full-upgrade	의존성 고려한 패키지 업그레이드
apt **search**	프로그램 검색
apt **show**	패키지 상세 정보 출력
apt **list**	apt-get install
apt **edit-sources**	소스 리스트 편집

gcc

```
jinsook@klein:~$ gcc

Command 'gcc' not found, but can be installed with:

sudo apt install gcc

jinsook@klein:~$
jinsook@klein:~$ sudo apt install gcc
[sudo] password for jinsook:
Sorry, try again.
[sudo] password for jinsook:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc-7 gcc-7-base libasan4 libatomic
  libcilkrts5 libgcc-7-dev libgomp1 libisl19 libitm1 liblsan0 libmpc3 libmpx2 libquadmath0 libtsan
Suggested packages:
  binutils-doc cpp-doc gcc-7-locales gcc-multilib make autoconf automake libtool flex bison gdb gc
  libgomp1-dbg libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg lib
  glibc-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc gcc-7 gcc-7-base libasan4 libat
  libcilkrts5 libgcc-7-dev libgomp1 libisl19 libitm1 liblsan0 libmpc3 libmpx2 libquadmath0 libtsan
0 upgraded, 27 newly installed, 0 to remove and 2 not upgraded.
Need to get 27.0 MB of archives.
After this operation, 115 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 binutils-common amd64 2.30-21ubun
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libbinutils amd64 2.30-21ubuntu1~
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 binutils-x86-64-linux-gnu amd64 2
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 binutils amd64 2.30-21ubuntu1~18.
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 gcc-7-base amd64 7.4.0-1ubuntu1~1
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 libisl19 amd64 0.19-1 [551 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/main amd64 libmpc3 amd64 1.1.0-1 [40.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 cpp-7 amd64 7.4.0-1ubuntu1~18.04.
25% [8 cpp-7 3637 kB/6742 kB 54%]
```