



2020 Fall System Programming

리눅스 개발환경 익히기

2020. 09. 18

권진세

Embedded System Lab.
Chungnam National University



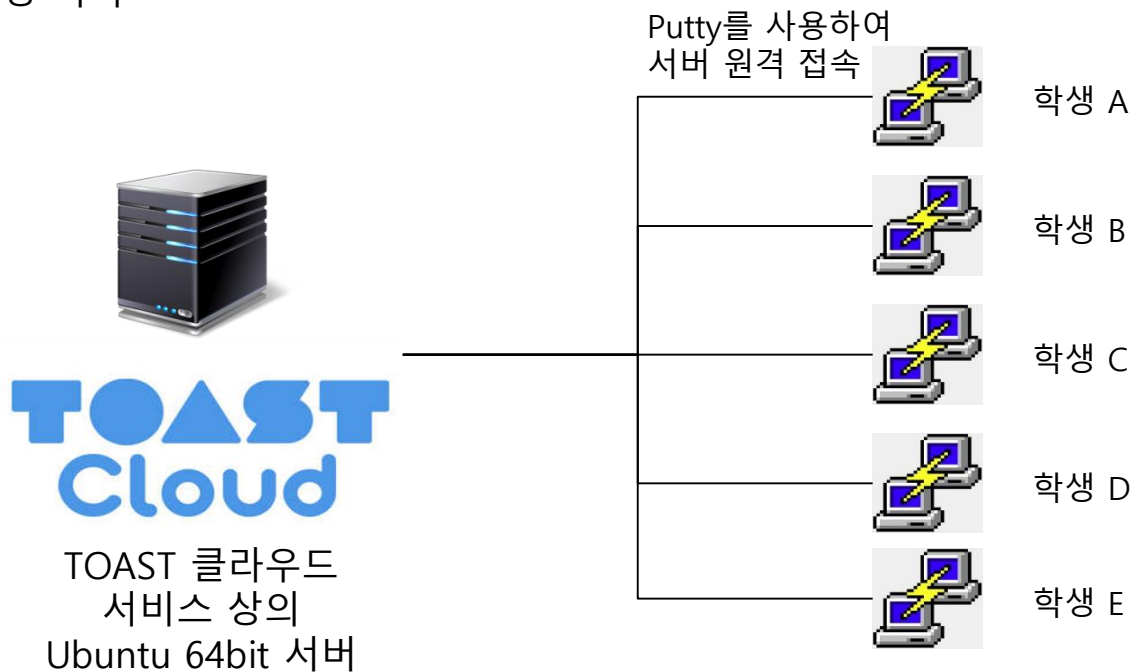
강의 일정

주	날짜	강의 (월)	날짜	실습 (금)
1			09월 04일	1.Intro (강의)
2	09월 07일	2.정수	09월 11일	2. 정수 + 3.부동소수점1
3	09월 14일	3.부동소수점2	09월 18일	리눅스 환경 익히기
4	09월 21일	4.어셈블리-MOVE	09월 25일	GCC & Make
5	09월 28일	5.어셈블리-제어문	10월 02일	Datalab
6	10월 05일	6.어셈블리-프로시저	10월 09일	휴강(추석)
7	10월 12일	7.어셈블리-프로시저	10월 16일	GDB
8	10월 19일	8. 프로세스 1	10월 23일	폭탄랩
9	10월 26일	중간고사(저녁 7시)	10월 30일	중간고사(저녁 7시)
10	11월 02일	9. 프로세스 2	11월 06일	셸랩 1
11	11월 09일	10. 시그널	11월 13일	셸랩 2
12	11월 16일	11. 동적메모리 1	11월 20일	셸랩 3
13	11월 23일	12. 동적메모리 2	11월 27일	말록랩 1
14	11월 30일	13. 링커	12월 04일	말록랩 2
15	12월 07일	보강기간	12월 11일	기말고사
16	12월 14일	Wrap-up/종강		



실습 소개

- 실습 환경
 - NHN 엔터테인먼트에서 제공하는 TOAST CLOUD 서비스 이용
 - Amazon AWS Cloud와 같은 서비스 제공
 - 가상 서버





금일 출석 방법

- 출결 시간 : 오전 10시~오전 12시 사이에 출석 체크함
 - 금일 실습 중 임의의 시간에 출결 공지함
- 출결 방법 : 해당 시간 기준으로 서버에 접속한 id로 자동 출결 처리함.
 - 실습 서버에 접속하여 실습 중이라면 자동 출석 인정
 - 서버는 10분간 아무 작업이 없을 경우 자동 Log-out 됨.



목차

I. 실습 명

1. 리눅스 개발환경 익히기

II. 목표

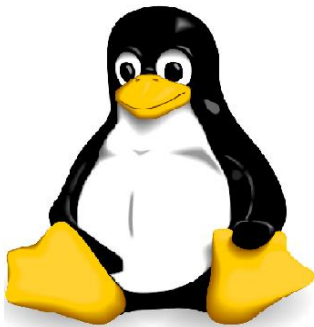
1. 리눅스를 사용하기 위한 기본 명령어 및 사용법 습득

III. 주제

1. 무엇을 하려 하는가?
2. 서버 접속
3. 기본 명령어
4. vi 에디터



1. 무엇을 하려 하는가?



Linux





2. 서버 접속 – Putty (windows/mac)

I. Putty 다운로드

- <https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

II. Host Name

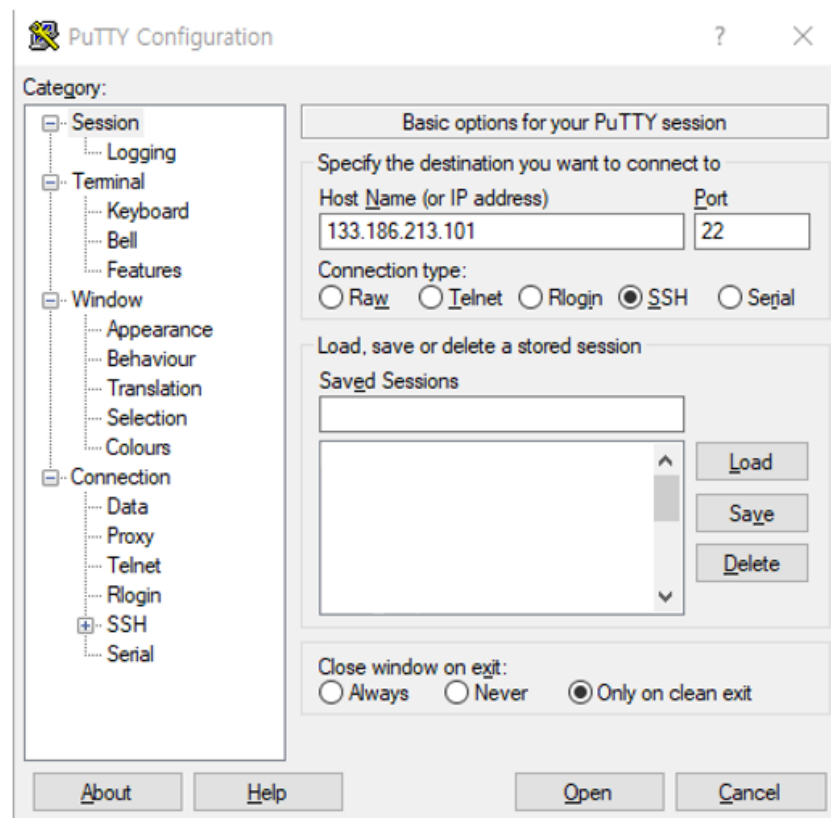
- IP : 133.186.221.214

III. 아이디 : c본인학번

- Ex) c201800000

IV. 비밀번호 : sys

- 비밀번호 입력 시에는 리눅스 보안상의 문제로
- 입력 문자가 보이지 않음.





2. 서버 접속 – Windows - PowerShell macOS - Terminal

I. PowerShell 실행 / Terminal 실행

- `ssh c본인학번@133.186.221.214`

II. 아이디 : c본인학번

- Ex) c201800000

III. 비밀번호 : sys

- 비밀번호 입력 시에는 리눅스 보안상의 문제로
- 입력 문자가 보이지 않음.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\WKWON> ssh c20152007@133.186.221.214
c20152007@133.186.221.214's password:
```


3. 기본 명령어

I. passwd

- 패스워드 변경

```
[root@localhost ~]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

II. pwd

- 현재 작업 디렉터리 절대경로 표시

```
[root@localhost ~]# pwd
/root
[root@localhost ~]#
```

III. clear

- 화면 지우기

```
anaconda-ks.cfg  root
install.log      root.tar.gz
install.log.syslog test.c
[root@localhost ~]# ls
anaconda-ks.cfg  root
install.log      root.tar.gz
install.log.syslog test.c
[root@localhost ~]# ls
anaconda-ks.cfg  root
install.log      root.tar.gz
install.log.syslog test.c
[root@localhost ~]#
```

실행 전



```
[root@localhost ~]#
```

실행 후



3. 기본 명령어

iv. mkdir

- Make directory: 디렉터리 생성
- 옵션
 - p: 만들고자 하는 디렉터리의 상위 디렉터리가 없는 경우 상위 디렉터리 까지 만들어 준다.
- 사용 예: mkdir [디렉터리명]
- 실습 : mkdir test

```
[root@localhost omega]# ls
[root@localhost omega]# mkdir protein
[root@localhost omega]# ls
protein
```

v. rmdir

- Remove directory: 디렉터리 삭제
- 사용 예: rmdir [디렉터리명]

```
[root@localhost omega]# mkdir protein
[root@localhost omega]# ls
protein
[root@localhost omega]# rmdir protein
[root@localhost omega]# ls
[root@localhost omega]#
```



3. 기본 명령어

VI. ls

- List: 디렉터리 목록 보기
- 사용법: ls [파일명]
- 옵션 (이어 붙여서 사용가능)
 - -a: 디렉터리 내의 모든 파일을 출력
 - -i: 파일의 inode 번호를 출력
 - -h: 파일의 크기를 쉬운 단위로 표시
 - -l :파일의 다양한 정보를 함께 출력(소유자, 권한, 크기, 날짜)
 - -m: 파일을 쉼표로 구분하여 가로로 출력
 - -s: KB 단위의 파일 크기를 출력
 - -t: 파일을 최근에 생성된 시간 순으로 출력
 - -F: 파일의 형태와 함께 출력
 - -R: 하위 디렉터리 내용을 모두 출력
 - -S: 파일 크기 순으로 출력
 - --help: 도움말 출력
- 사용 예: ls -al

```
[root@localhost ~]# ls
anaconda-ks.cfg      root
install.log          root.tar.gz
install.log.syslog   test.c
[root@localhost ~]#
```



3. 기본 명령어

VII. cd

- Change directory: 디렉터리 이동
- 사용법: cd [디렉터리명]
- 디렉터리 구조에 관한 명령어
 - cd / : root 디렉터리로 이동
 - cd ~ : home 디렉터리로 이동
 - cd .. : 현재 작업 디렉터리의 상위 디렉터리로 이동
- 사용 예:

```
[root@localhost omega]# pwd
/root/omega
[root@localhost omega]# cd ..
[root@localhost ~]# cd ./omega/
[root@localhost omega]# pwd
/root/omega
[root@localhost omega]# cd ..
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd ./omega
[root@localhost omega]# pwd
/root/omega
[root@localhost omega]# cd ~
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /root/omega/
[root@localhost omega]#
```

pwd	: 현재 위치 출력
cd ..	: 상위 디렉터리 이동
cd ./omega	: 상대주소로 이동
cd ~	: 본인 계정 디렉터리로 이동
cd /root/omega/	: 절대주소로 이동



참고) 상대주소, 절대주소

아래의 기호와 의미를 암기 한다 !

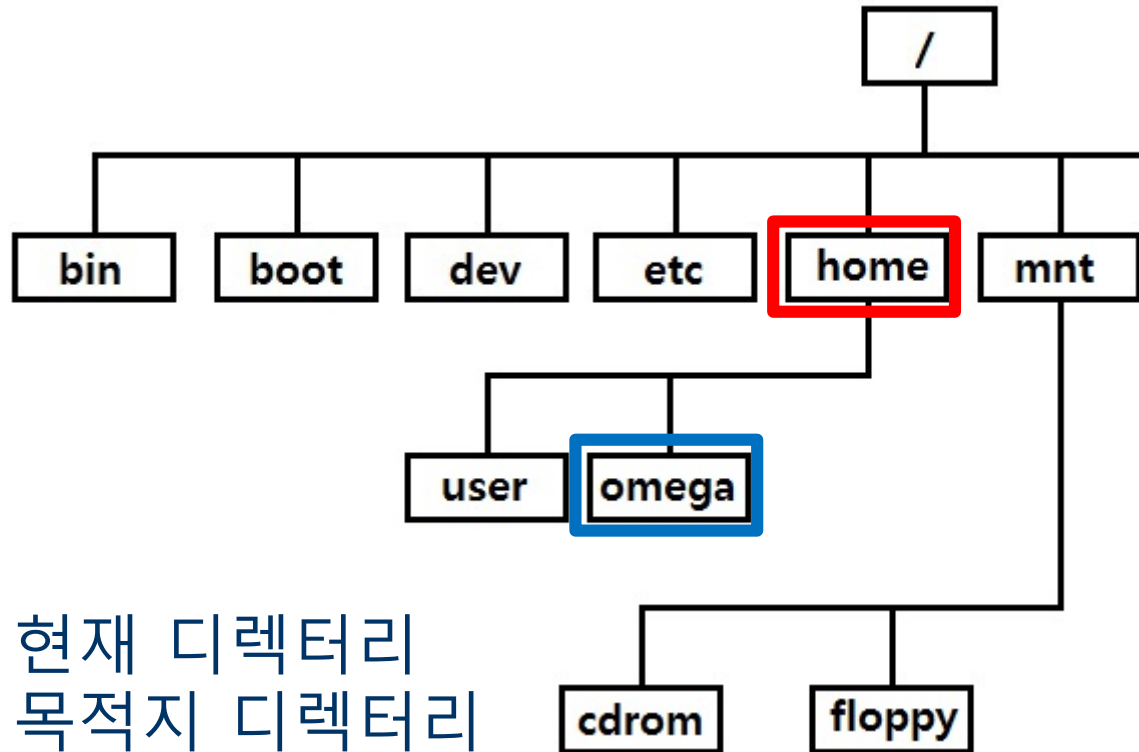
- 최상위 디렉터리(root directory) => /
- 현재 디렉터리 => .
- 상위 디렉터리 => ..
- 자신의 계정 디렉터리(home directory) => ~

절대주소: 최상위 디렉터리에서부터 목적지까지의 경로.

상대주소: 현재 디렉터리로부터의 경로.



참고) 상대주소, 절대주소



빨간색: 현재 디렉터리

파란색: 목적지 디렉터리

현재 home에서 omega의 절대 주소: /home/omega

현재 home에서 omega의 상대 주소: ./omega



3. 기본 명령어

VIII. cp

- Copy: 파일 및 디렉터리 복사
- 사용법: cp [원본 파일] [복사될 위치 디렉터리/복사될 파일]
- 옵션
 - -i : 파일 복사 시 동일한 파일명이 있을 경우 사용자에게 물어봄
 - -f : 동일 파일명 발생시에도 모두 강제로 복사
 - -p : 원본 파일의 시간 및 소유 권한 보존
 - -r : 포함된 자식 디렉터리 까지 모두 복사
- 사용 예

```
[root@root root] mkdir test
[root@root root] cp /home/sys00/sys00/data1.out ./test
[root@root root] cd test
[root@root test] ls
data1.out
```



3. 기본 명령어

IX. mv

- Move: 파일 및 디렉터리 이동
- 파일 이름을 변경할 때도 쓰임
- 사용법: [변경할 파일/디렉터리] [변경될 파일/디렉터리]
- 옵션
 - -f: 대상파일이 존재할 때 새로운 파일로 대체
 - -i: 대상파일이 있을 때 확인 후 작업
- 사용 예: mv data1.out data2.out



3. 기본 명령어 - 권한의 의미

- 리눅스에서의 권한

```
[d201153100@localhost syspro]$ cd /root/root  
bash: cd: /root/root: Permission denied
```

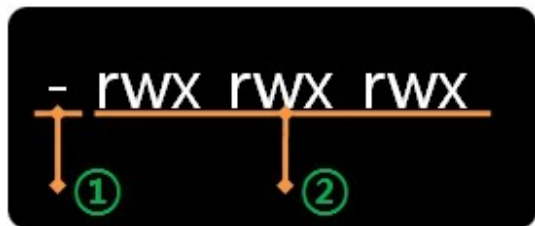
- **bash: cd: /root/root/: Permission denied**
- 위와 같이 본인 계정이 가지지 않는 권한의 디렉터리로 접근 하려고 하면 거부 된다.
- 권한은 해당 계정의 자료를 보호하는 기능을 한다.
- 따라서 권한에 대한 이해는 매우 중요하다.

3. 기본 명령어 – 권한 확인 방법

- ls -l을 통해서 파일 리스트와 함께 권한을 출력 시킨다.

```
[root@localhost tmp]# ls -l
total 16
-rw-r--r-- 1 root root 0 Mar 13 12:14 hello
srwxrwxrwx 1 wnn wnn 0 Mar 14 08:39 jd_sockU4
drwx----- 2 root root 4096 Mar 14 21:57 orbit-root
drwx----- 2 root root 4096 Mar 14 21:51 ssh-XXj7zmo7
drwx----- 2 root root 4096 Mar 13 09:01 ssh-XXZunr2T
-rw-r--r-- 1 root root 0 Dec 9 2012 te
-rw-r--r-- 1 root root 13 Mar 14 22:02 test
[root@localhost tmp]# _
```

- 권한 보는 방법



- 파일 유형
 - 첫 번째 문자(①)는 파일의 유형을 뜻한다.
 - : 일반적인 파일을 뜻한다.
 - > d : 디렉토리를 뜻한다. l : 링크 파일을 뜻한다. s : 소켓 파일을 뜻한다.
- 파일 유형
 - r: 읽기(read), w: 쓰기(write), x: 실행(execute)



3. 기본 명령어 – 권한 변경

XI. chmod

- chmod는 파일, 디렉터리의 권한을 변경 하는데 사용 한다.
- 사용법 : chmod [옵션] [권한] files directory..
- 옵션
 - c : 실제로 파일의 권한이 바뀐 파일만 자세히 기술한다.
 - f : 파일의 권한을 바꿀 수 없는 경우에도 에러 메시지를 출력하지 않는다.
 - v : 변경된 권한에 대해서 자세히 기술한다.
 - R : 디렉터리와 파일들의 권한을 재귀적으로 모두 바꾼다. ※**제일 많이 쓰이는 옵션이다.**
- 권한
 - r(read 읽기), w(write 쓰기), x(execute 실행)의 3가지 권한이 있다.
 - 읽기, 쓰기, 실행 등의 모든 권한을 주려면, 4+2+1, 따라서 합계 7이 된다. 만약 모든 그룹의 권한을 다 허용해 줄 경우, 777이 된다.

3. 기본 명령어 - 권한 변경

- 권한(문자모드)

Class	연산자	권한
u : user	= : 권한 설정	r : read
g : group	+ : 권한 추가	w : write
o : other	- : 권한 삭제	x : execute
a : all		

- 권한(숫자모드)

사용자(user)			그룹(group)			타인(other)		
r	w	x	r	w	x	r	w	x
400	200	100	40	20	10	4	2	1
700			70			7		



3. 기본 명령어 – 권한 변경

- chmod 따라 하기

- 1) mkdir을 이용하여 자신의 계정 디렉터리에서 chmodtest를 만들고 ls -l 명령을 통해 현재 권한을 확인한다.

```
a201111111@localhost:~  
[a20111111@localhost ~]$ mkdir chmodtest  
[a20111111@localhost ~]$ ls -l  
한계 4  
drwxrwxr-x. 2 a201111111 a201111111 4096  9월  6 13:36 chmodtest
```

- chmodtest 디렉터리
- user와 group 읽기쓰기실행 (rwx) 권한
- 다른 사용자(other) 읽기실행(r-x) 권한

3. 기본 명령어 - 권한 변경

- 2) chmodtest 디렉토리를 문자모드를 이용해서 그룹에게 쓰기권한을 제거한다.

- `# chmod g-w chmodtest`

```
a201111111@localhost:~  
[a20111111@localhost ~]$ chmod g-w chmodtest/  
[a20111111@localhost ~]$ ls -l  
한계 4  
drwxr-xr-x. 2 a20111111 a20111111 4096  9월  6 13:36 chmodtest
```

- chmod 명령어 옵션으로 그룹(g)에 쓰기 기능(w)을 제거(-)

- 3) 이번에는 숫자모드를 이용 하여 모든 권한 (rwx)을 추가한다.

- `#chmod 777 chmodtest`

```
a201111111@localhost:~  
[a20111111@localhost ~]$ chmod 777 chmodtest  
[a20111111@localhost ~]$ ls -l  
한계 4  
drwxrwxrwx. 2 a20111111 a20111111 4096  9월  6 13:36 chmodtest
```



3. 기본 명령어

XII. rm

- Remove: 파일 및 디렉터리 삭제
- 사용법: `rm [삭제할 파일]`
- 옵션
 - `-f`: 디렉터리 안의 파일을 삭제할 때 사용자에게 확인을 요구하지 않음
 - `-r`: 인수 list 에서 지정한 디렉터리 혹은 아래의 서브디렉터리 삭제
 - `-i`: write permission 이 없는 파일의 삭제를 위해 대화식으로 확인
 - `-p`: 디렉터리 `dir-name` 과 비어있는 부모 디렉터리를 사용자가 제거 할 수 있다.
 - `-s`: `-p` 선택항목 지정 시 표준오류에 출력되는 메시지를 삭제
- 사용 예

```
[root@root test] rm data1.out
[root@root test] rm data*.out
[root@root test] ls
```

XIII. cat

- Catenae: 텍스트 파일 내용 출력
- 사용법: `cat [파일명]`
- 사용 예: `cat data1.out`



3. 기본 명령어

XIV. tar

- 아카이브(.tar)를 만들거나 푼다
 - 아카이브란 여러 파일을 하나로 묶는 것
- 옵션
 - -c : 아카이브 생성
 - -x : 아카이브에 묶인 파일이나 디렉터리를 풀어 줌
 - -f : 파일 이름 지정
 - -v : 아카이브에 추가되거나 풀리고 있는 파일의 이름을 화면에 보여 줌
 - -z : *.gz 파일로 압축
- 사용법
 - 파일 묶고 압축 : tar cvfz [파일명.tar.gz] [압축할 파일/디렉터리]
 - 압축된 파일 풀기 : tar xvfz [파일명]
- 예제
 - \$ tar cvf tarfile .
 - \$ tar xvf tarfile



3. 기본 명령어

xv. find

- 디스크에 저장된 각종 파일/디렉터리 검색
- 사용법: `find [path..] [expression]`
- Path : 파일을 탐색할 경로의 리스트 (recursive search)
- Expression
 - `-name 'pattern'` : 파일명이 'pattern'과 일치이면 참 (*, [,], ? 포함가능)
 - `-perm 'oct'` : permission의 8진수 표현이 oct와 일치이면 참
 - `-type 'ch'` : 파일 유형이 ch 이면 참
 - `-user 'userId'` : 파일 소유자가 userId이면 참
 - `-group 'groupId'` : 파일 그룹이 groupId이면 참
 - `-atime 'count'` : 파일에 접근한 날 수가 count 이내 이면 참
 - `-mtime 'count'` : 파일을 수정한 날 수가 count 이내이면 참
 - `-ctime 'count'` : 파일이 수정되고 파일 속성이 바뀐 날수가 count 이내
 - `-exec 'command'` : 수행중인 command의 종료 값이 0이면 참 \;로 끝남, command의 argument가 {}이면, find가 찾은 파일을 의미
 - `-ls` : 현재 파일의 속성을 보여주고 참 값을 반환



3. 기본 명령어

- Expression

- name 'pattern' : 파일명이 'pattern'과 일치하면 참 (*, [,], ? 포함가능)

```
[eslab@localhost FindGrep]$ find -name test.c
./test.c
[eslab@localhost FindGrep]$ find test.c
test.c
[eslab@localhost FindGrep]$
```

- perm oct : permission의 8진수 표현이 oct와 일치이면 참

```
[eslab@localhost FindGrep]$ find -perm 644
./testfile
./test.c
./apple.c
./orange.txt
[eslab@localhost FindGrep]$ ls -al
total 28
drwxr-xr-x.  4 root  root  4096 Sep  6 21:40 .
drwx----- 24 eslab eslab 4096 Sep  6 21:40 ..
-rw-r--r--.  1 eslab root    26 Sep  6 20:45 apple.c
-rw-rw-r--.  1 eslab eslab    0 Sep  6 21:38 banana.java
drwxr-xr-x.  2 root  root  4096 Sep  6 21:28 directory1
drwxr-xr-x.  2 root  root  4096 Sep  6 05:09 directory2
-rw-rw-r--.  1 eslab eslab    0 Sep  6 21:40 kiwi.java
-rw-r--r--.  1 eslab root    0 Sep  6 05:08 orange.txt
-rwxrwxrwx.  1 eslab root    0 Sep  6 05:26 pineapple.c
-rw-r--r--.  1 eslab root    26 Sep  6 20:46 test.c
-rw-r--r--.  1 eslab root   234 Sep  6 05:36 testfile
[eslab@localhost FindGrep]$
```



3. 기본 명령어

- type term : 파일 유형이 term 이면 참

```
[eslab@localhost FindGrep]$ find -type f
./testfile
./test.c
./pineapple.c
./banana.java
./directory1/tomato.c
./directory1/melon.c
./apple.c
./orange.txt
[eslab@localhost FindGrep]$ find -type d
.
./directory1
./directory2
```

term:
f : 파일만 찾는다.
d : 디렉터리만 찾는다.

- user userId : 파일 소유자가 userId이면 참

```
[eslab@localhost FindGrep]$ find -user eslab
./testfile
./test.c
./pineapple.c
./banana.java
./kiwi.java
./apple.c
./orange.txt
[eslab@localhost FindGrep]$ ls -al
total 28
drwxr-xr-x. 4 root root 4096 Sep  6 21:40 .
drwx----- 24 eslab eslab 4096 Sep  6 21:40 ..
-rw-r--r--. 1 eslab root  26 Sep  6 20:45 apple.c
-rw-rw-r--. 1 eslab eslab  0 Sep  6 21:38 banana.java
drwxr-xr-x. 2 root root 4096 Sep  6 21:28 directory1
drwxr-xr-x. 2 root root 4096 Sep  6 05:09 directory2
-rw-rw-r--. 1 eslab eslab  0 Sep  6 21:40 kiwi.java
-rw-r--r--. 1 eslab root  0 Sep  6 05:08 orange.txt
-rwxrwxrwx. 1 eslab root  0 Sep  6 05:26 pineapple.c
-rw-r--r--. 1 eslab root  26 Sep  6 20:46 test.c
-rw-r--r--. 1 eslab root 234 Sep  6 05:36 testfile
[eslab@localhost FindGrep]$
```

- group groupId : 파일 그룹이 groupId이면 참



3. 기본 명령어

xvii. I/O Redirection

- I/O의 방향을 사용자가 원하는 대로 바꿀 수 있음

기호	기능
>	쓰기
<	읽기
>>	추가해서 쓰기

- 사용 예
 - `ls -al > ls.list`
 - `sort < ls.list > sorted.list`
 - `cat > test.txt`
 - > 탈출 시 `ctrl + d`
 - `cat >> test.txt`



3. 기본 명령어

xvi. grep

- 사용법: `grep [<option>] <pattern> [<file name>]`
- 옵션
 - `-i` : 영문의 대소문자를 구분하지 않음
 - `-v` : `pattern`을 포함하지 않는 라인 출력
 - `-n` : 검색 결과의 각 행의 선두에 행 번호를 넣음
 - `-l` : 파일명만 출력
 - `-c` : 패턴과 일치하는 라인의 개수만 출력
 - `-r` : 하위 디렉터리까지 검색
- 파일 이름에 `*`의 사용
 - 파일이름에서 여러 개의 문자를 대치할 수 있다.
 - 예 : `grep the *.txt`
- 사용 예시
`$ grep -r 'Hello' lab01`

```
sys00@2018-sp:~$ grep -r 'Hello' lab01
lab01/find/test.c:      printf("Hello I'm test.c\n");
lab01/findfind/findfindfind/test2.c:  printf("Hello I'm test2.c\n");
```



3. 기본 명령어

xviii. Pipe

- 기호: | (shift + \)
- 기호를 기준으로 왼쪽 명령어의 출력을 오른쪽 명령어의 입력으로 보낸다.

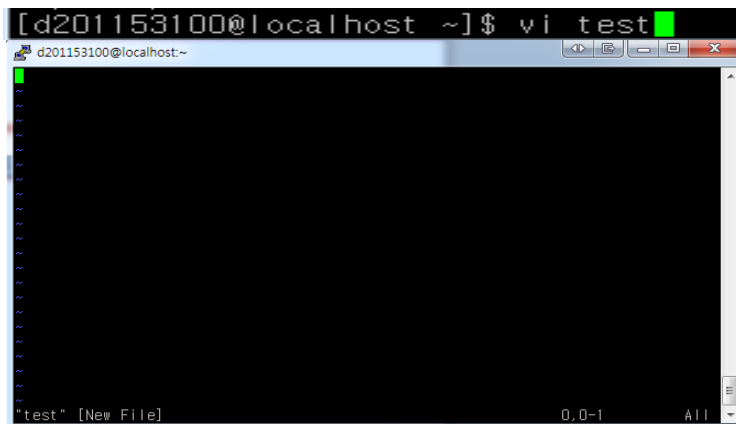
- 사용 예
 - `cat /etc/passwd | more`

4. vi 에디터 – vi 모드의 이해

I. vi 에디터는 코드를 작성할 때 쓰이는 편집 도구이다.

II. **표준모드**

- 실행 명령어: vi 파일명

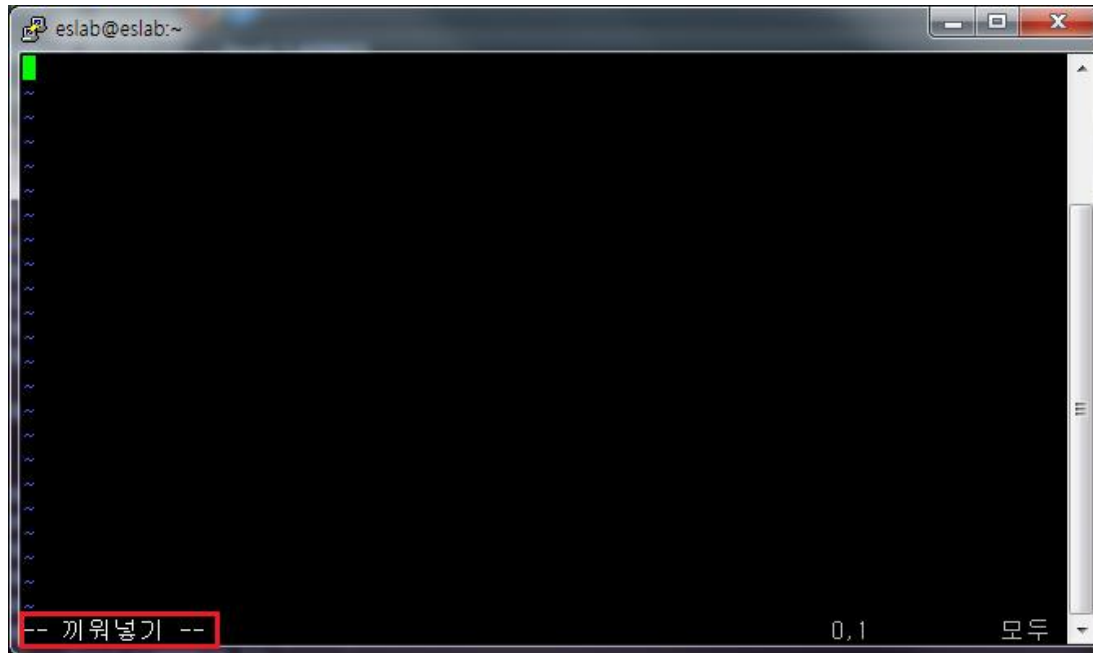


- 표준 모드는 키 입력을 통해 vi에게 명령을 내리는 모드다. 표준 모드에서 커서를 이동하거나, 삭제, 복사 붙이기 등의 작업을 수행할 수 있다.
- vi를 실행하면 표준 모드부터 시작하는데, 표준 모드에서는 아무리 타이핑해도 글자가 입력 되지 않는다.
- 표준 모드는 vi에게 명령을 내리기 위한 모드지 편집을 위한 모드가 아니라는 점을 기억하기 바란다.

4. vi 에디터 – vi 모드의 이해

III. 입력 모드

- vi 에디터 표준모드에서 'i', 'a', 'o', 's' 4개의 키 중 하나를 누른다.



- 입력 모드는 실제로 문서를 편집하기 위한 모드다. 따라서 타이핑 하면 실제로 화면에 출력되면서 글자의 입력이 가능하게 된다.



4. vi 에디터 – vi 모드의 이해

IV. 명령 모드

- vi 에디터 표준모드에서 :, /, ? 3개의 키 중 하나를 누른다.
- 입력 모드에서는 esc 키를 누른 후 입력.



- 명령 모드에서 수행할 수 있는 일에는 vi 설정을 바꾸거나, 파일을 저장하고 읽거나, 특정 패턴을 찾고 바꾸거나, 외부 명령을 실행하거나, vi를 종료 하는 등과 같은 일을 할 수 있다.
- 위 예제의 q!는 vi 에디터를 저장하지 않고 종료하는 명령어이다. enter를 치면 에디터가 종료 되는 것을 볼 수 있다.



4. vi 에디터

- 명령모드에서 텍스트 입력모드로 전환

명령어	설 명
i	텍스트가 커서 앞에 삽입
l	텍스트가 현재 줄의 맨 앞에 삽입
a	텍스트가 커서 뒤에 삽입
A	텍스트가 현재 줄의 맨 앞에 삽입
o	텍스트가 현재 줄 다음부터 삽입
O	텍스트가 현재 줄 앞에서 삽입
r	텍스트가 커서 위치에서 대치
R	텍스트가 현재 줄에서 대치

- 텍스트 입력 모드에서 명령 모드로의 전환
 - ESC



4. vi 에디터

• 커서 이동

키	설 명
↑ or k	한 줄 위
↓ or j	한 줄 아래
← or h	한 문자 왼쪽
→ or l	한 문자 오른쪽
G	파일의 마지막 행으로 이동
gg	파일의 첫 번째 행으로 이동
행 번호G or :행 번호	행 번호로 이동
^ or home	같은 행에서 행의 시작 지점으로 이동
\$ or end	같은 행에서 행의 마지막 지점으로 이동
b	띄어쓰는 단어들에서 한 단어 후퇴
w or W	띄어쓰는 단어들에서 한 단어 전진



4. vi 에디터

- 화면 이동

키	설 명
z + enter	현재 줄이 맨 위에 오도록
z.	현재 줄이 화면 가운데 오도록
z-	현재 줄이 화면의 맨 아래 오도록
ctrl + f or page up	화면 전체의 내용을 한 화면 위로 이동
ctrl + b or page down	화면 전체의 내용을 한 화면 아래로 이동
ctrl + e	화면 전체의 내용을 한 줄씩 위로 이동
ctrl + y	화면 전체의 내용을 한 줄씩 아래로 이동



4. vi 에디터

- 텍스트 삭제

키	설 명
x	문자 삭제
dw	단어 삭제
dd	행 삭제
D	커서 오른쪽 행 삭제
d^	행의 처음부터 커서까지 삭제
d\$	커서에서 행의 마지막까지 삭제
: 1,\$ d	첫 줄부터 마지막 줄까지 삭제
: .,-2 d	현재 줄부터 이전 두 줄까지 삭제



4. vi 에디터

- 텍스트 치환

키	설 명
CW	단어 변경
CC	행 변경
C	커서 오른쪽의 행 변경

- 탐색 및 치환

키	설 명
/문자열	현재 위치에서 파일의 뒤 쪽으로 문자열 검색 (n 반복)
?문자열	현재 위치에서 파일의 앞쪽으로 문자열 검색 (n 반복)
:<range>s/old/new	<range> 내의 문자열 old의 처음것만 new로 치환
:<range>s/old/new/g	<range> 내의 문자열 old를 new로 모두 치환
:%s/old/new	현재 행에서 old를 new로 치환
:%s/old/new/g	현재 파일에서 old를 new로 모두 치환



4. vi 에디터

- 파일 저장 및 불러오기

명령어	설 명
:q	텍스트 수정이 없을 때 종료
:q!	텍스트 수정이 있었을 때에도 저장하지 않고 종료
:w	저장
:wq or :x	저장 후 종료
:w <이름>	<이름>이라는 파일로 저장
:e <이름>	현재 파일이 아닌 다른 파일 <이름>을 편집
:n	현재 편집중인 파일의 다음 파일을 편집
:! <명령>	Shell 명령을 실행하고 vi로 되돌아 감
:r xx	다른 파일을 읽어 와 덧붙이기



4. vi 에디터

- 명령모드에서 버퍼 이용 붙이기

키	설 명
<n>yy	n개의 줄을 이름 없는 버퍼로 복사
<n>Y	n개의 줄을 이름 없는 버퍼로 복사
<n>p	버퍼에서 복사해서 현재 행 위에 n개 삽입
<n>P	버퍼에서 복사해서 현재 행 아래에 n개 삽입
:<range> y	<range>의 내용을 이름 없는 버퍼에 복사
:<line>pu	line 행 다음에 붙임

- 기타 명령어

키	설 명
u	실행취소
.	최근 명령 재실행
:sh	shell 수행, exit 복귀



실습

1. 디렉터리 생성, 권한 변경, 목록출력 실습

- 자신의 홈 디렉터리에 학번으로 디렉터를 생성
- 디렉터리 접근 권한을 소유자 전용으로 변경(drwx --- ---)
- 변경된 권한을 화면에 출력
- Hint: chmod [권한] [대상폴더] (권한에 700)

2. 복사, 압축해제, 이동(이름변경), 압축 실습

- /home/sys02/sys02/lab01.tar.gz를 자신의 홈 디렉터리로 복사
- 압축해제
- lab01 디렉터를 학번으로 이름 변경
- 변경된 학번 디렉터를 분반_학번.tar.gz로 압축



실습

3. 삭제 실습

- 자신의 홈 디렉터리(**a201700000@2019-sp : ~ \$**)에 영문이름으로 디렉터리를 생성
- 영문이름 디렉터리 안에 학번 디렉터를 생성
- 한번에 생성한 2개의 디렉터리 삭제
- **Hint : rm -rf [디렉토리명]**



실습

4. 검색 실습

- /home/sys02/sys02/lab01.tar.gz를 자신의 홈 디렉터리에 복사
- 압축 해제
- Lab01 디렉터리에서 **test.c** 파일이 어디에 위치하는지 검색
- Lab01 디렉터리에서 **test2.c** 파일이 어디에 위치하는지 검색
 - Hint: 반드시 Find 이용(-name)



실습

5. cat 명령어와 IO Redirection을 사용하여 grepfile.txt 을 만들어라

Hello, World!!
2020 System programming!!

리눅스에서 파일 만드는 법
cat > 파일명
내용 입력
입력 후 Ctrl + c

<Hint : 실습 grep 명령어 참고>

6. grepfile에서 Hello 글자가 있는 줄의 목록을 걸러 줄 번호와 함께 화면에 출력하라
7. grepfile에서 Hello 라는 단어가 없는 줄의 목록을 걸러 result 파일에 출력하라



실습

8. vi 에디터로 testfile.txt 파일을 만들어라

with a screen editor you can scroll
the page, move the cursor, delete lines,

12. 위 문서에서 모든 **you**를 **I**로 변경 한다.

조건: vi 에디터의 기능인 치환 기능을 이용해서 한번에 변경 (실습 탐색 및 치환
참고) Hint : :%s/old/new/g 는 현재 파일에서 old를 new로 모두 치환

13. 2번째 줄을 한번에 vi 명령어를 이용해서 삭제한다. (visual 모드 이용 or <range> d)