

시스템 프로그래밍

리눅스 개발환경 익히기

2014. 09. 04.

박시형

sihyeong@cnu.ac.kr

Embedded System Lab. Computer Engineering Dept.
Chungnam National University



실습 소개

■ 과목 홈페이지

- <http://eslab.cnu.ac.kr>

- ▶ Lectures -> undergraduate course -> 2014 Fall 25154 System Programming(2014 fall)
- ▶ http://eslab.cnu.ac.kr/board_docs/lecture/systemProgramming_14_fall.htm

- 실습 게시판 : 02반

- ▶ http://eslab.cnu.ac.kr/xe/index.php?mid=systemProgramming_14_fall_02_bul

■ 연락처

- 박시형
- 공대 5호관 533호 임베디드 시스템 연구실
- sihyeong@cnu.ac.kr

개요

■ 실습 명

- 리눅스 개발환경 익히기

■ 목표

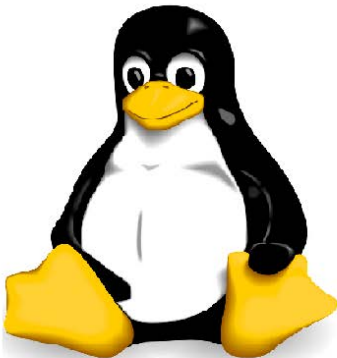
- 리눅스를 사용하기 위한 기본 명령어 및 사용법 습득


■ 주제

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



1. 무엇을 하려 하는가?



Linux 

2. 서버 접속 - Putty

■ Putty 다운로드

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

■ Host Name

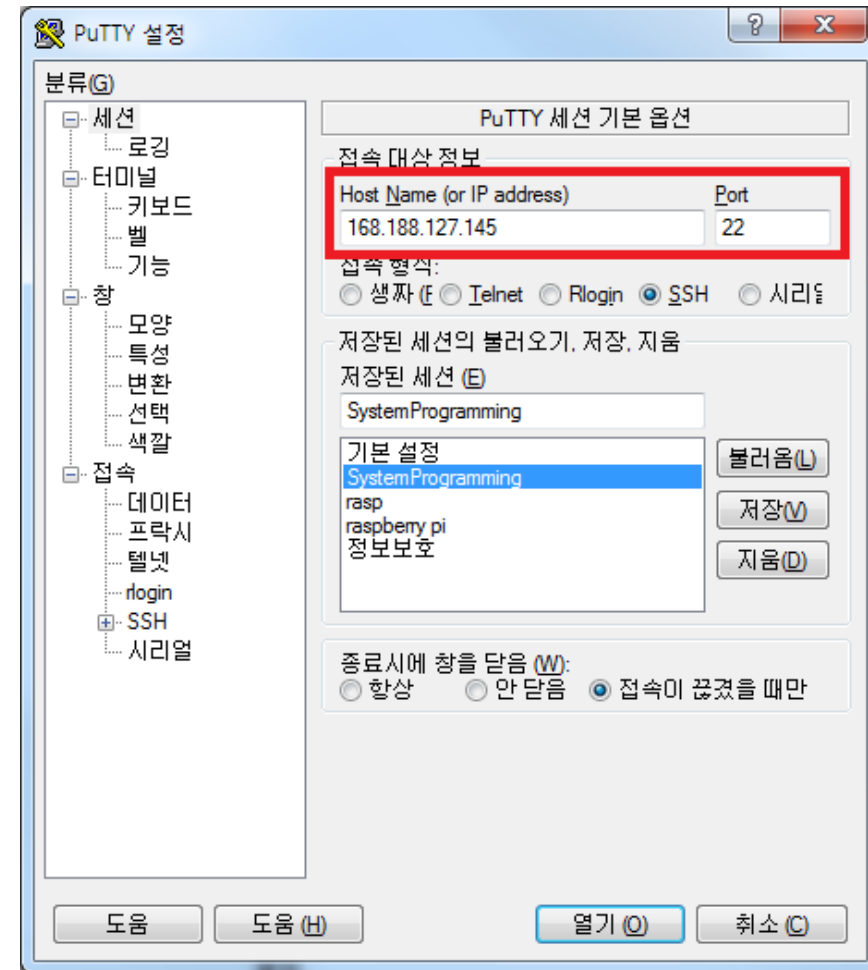
- IP : 168.188.127.145

■ 아이디 : b본인학번

- Ex) b201300000

■ 비밀번호 : eslab

- 비밀번호 입력 시에는 리눅스 보안상의 문제로 문자가 보이지 않음.

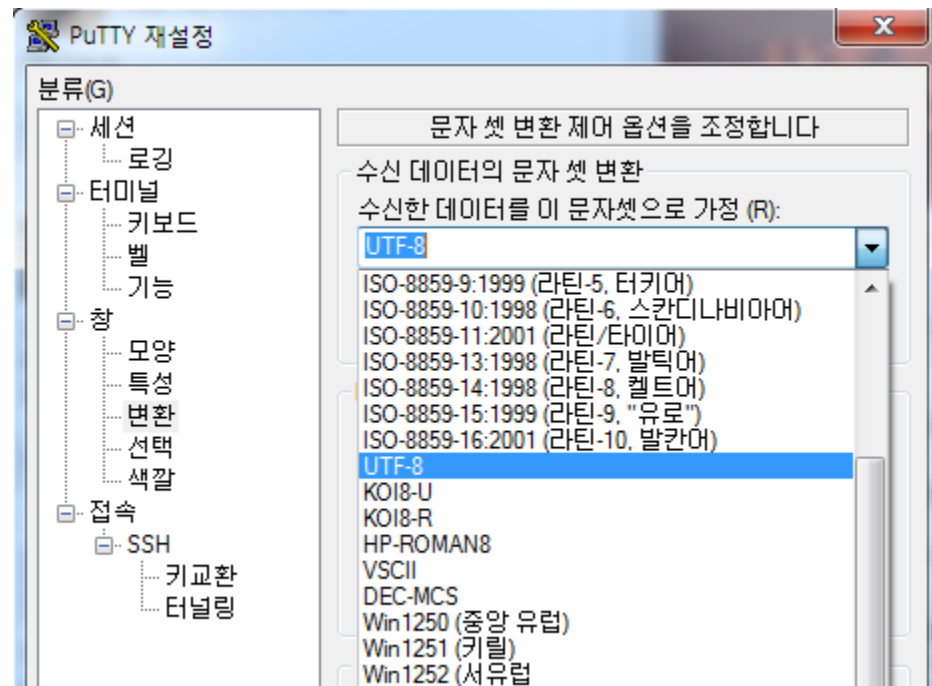


2. 서버 접속 – Putty 설정

- 서버의 언어가 한글로 설정되어 있어서, Putty 에서 글자가 깨질 수 있다.
 - Putty창 우 클릭 - '설정 변경 (G)' 클릭
 - '창' - '변환' - '수신한 데이터를 이 문자 셋으로 가정 (R)'
 - UTF-8 선택

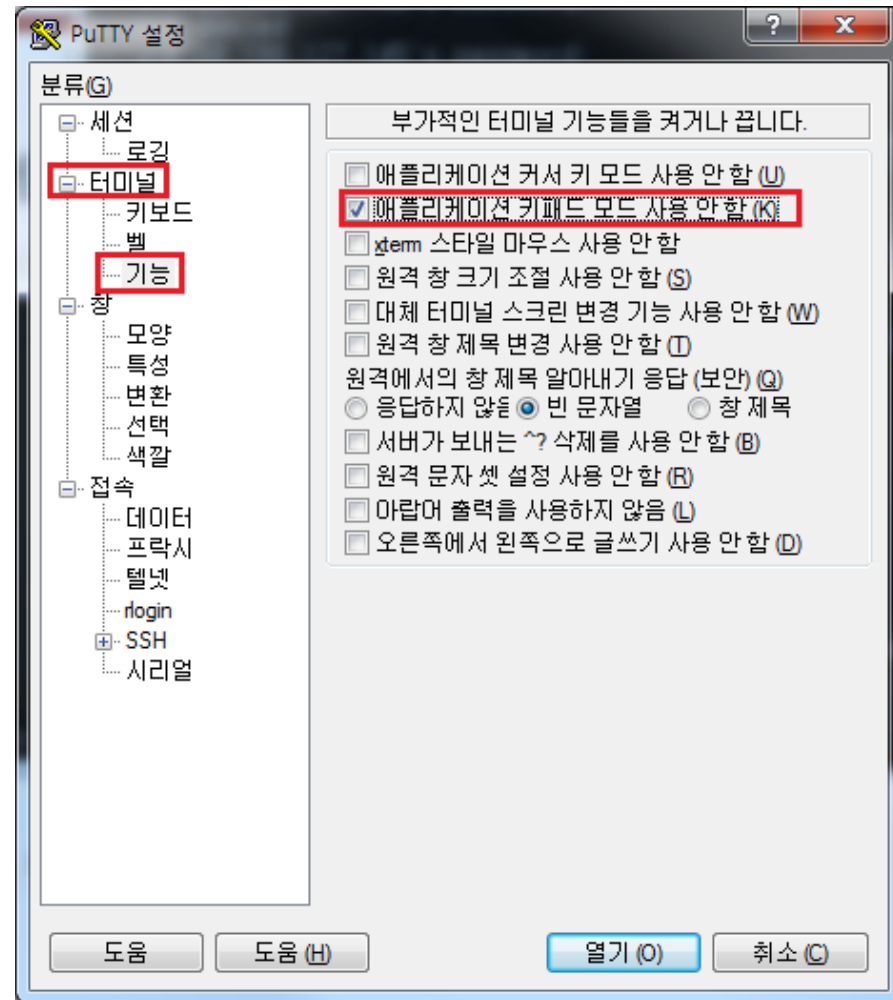
```

eslab@eslab:~$
login as: eslab
eslab@168.188.127.145's password:
Access denied
eslab@168.188.127.145's password:
Last failed login: Wed Sep 3 16:23:02 KST 2014 from 168.188.127.142 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Tue Sep 2 14:18:45 2014 from 168.188.127.142
[eslab@eslab ~]$ ls
Desktop  Downloads  Pictures  Templates  bomblab
Documents Music    Public    Videos    락월깃? 뵐(=)
[eslab@eslab ~]$ ls
Desktop  Downloads  Pictures  Templates  bomblab
Documents Music    Public    Videos    락월깃? 뵐(=)
[eslab@eslab ~]$ passwd
eslab ? ㄴ 兕? 땃췔 鑰궑? 躑궑샹 躑 冤? 以?
eslab? ? ? ? ? 뵐샹 躑 冤? 以?
(? 兕兕) UNIX ? 뵐샹:
    
```



2. 서버 접속 – Putty 설정

- Putty의 기본 설정은 키보드의 키 패드를 방향키로 사용
 - 키 패드를 사용하기 위해서는 아래의 설정을 적용
 - Putty창 우 클릭 - '설정 변경 (G)' 클릭
 - '터미널' - '기능' - '애플리케이션 키 패드 모드 사용 안 함' 을 체크하여 설정



3. 기본 명령어

■ passwd

● 패스워드 변경

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

■ pwd

● 현재 작업 디렉터리 절대경로 표시

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

■ 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

■ man

- Manual page: 명령어에 필요한 정보 출력
- 사용법: man [명령어]
- cf) 탈출시: q

```
[root@localhost ~]# man ls
LS(1)          User Commands          LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the
    FILES (the current directory
    by default).  Sort entries
    alphabetically if none of
    -cftuvSUX nor --sort.

    Mandatory arguments to long
    options are mandatory for
    short options too.

    -a, --all
        do not ignore entries
        starting with .

    -A, --almost-all
Manual page ls(1) line 1
```

3. 기본 명령어

■ ls

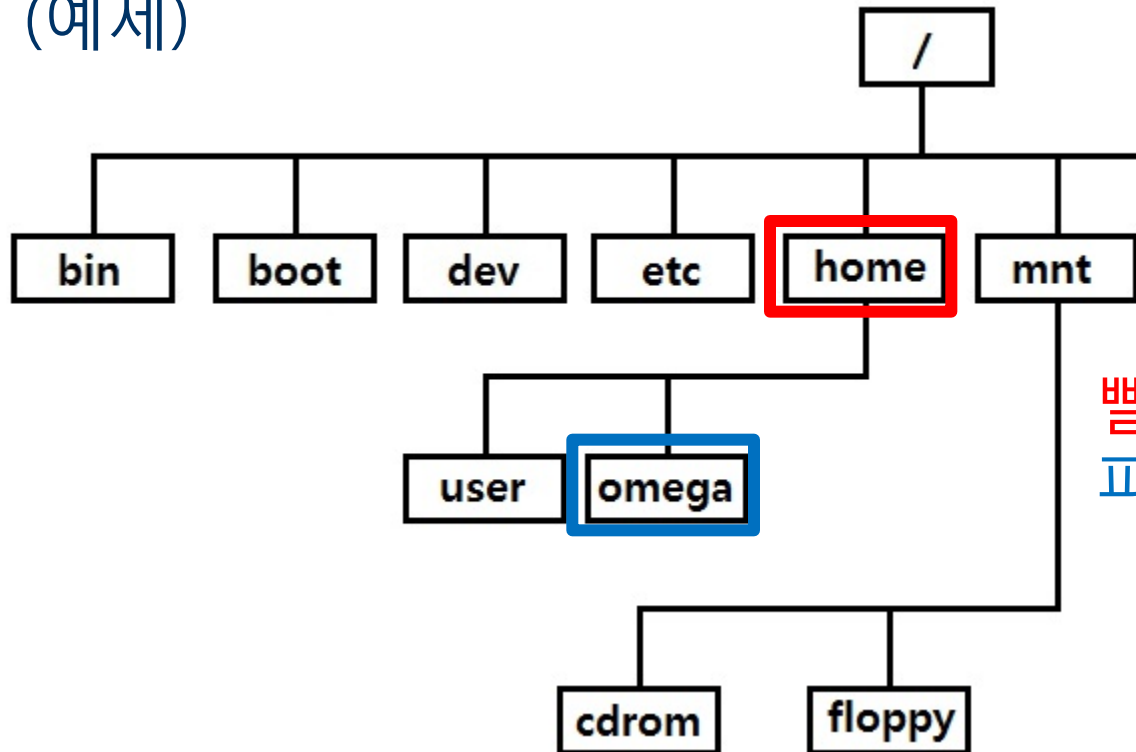
- List: 디렉터리 목록 보기
- 사용법: ls [파일명]
- 옵션
 - ▶ -a: 디렉터리 내의 모든 파일을 출력
 - ▶ -i: 파일의 inode 번호를 출력
 - ▶ -h: 파일의 크기를 쉬운 단위로 표시
 - ▶ -l :파일의 다양한 정보를 함께 출력(소유자, 권한, 크기, 날짜)
 - ▶ -m: 파일을 쉼표로 구분하여 가로로 출력
 - ▶ -s: KB 단위의 파일 크기를 출력
 - ▶ -t: 파일을 최근에 생성된 시간 순으로 출력
 - ▶ -F: 파일의 형태와 함께 출력
 - ▶ -R: 하위 디렉터리 내용을 모두 출력
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

참고) 상대주소, 절대주소

(예제)



빨간색: 현재 디렉터리
파란색: 목적지 디렉터리

현재 home에서 omega의 절대 주소: /home/omega

현재 home에서 omega의 상대 주소: ./omega

참고) 상대주소, 절대주소

■ 아래의 기호와 의미를 암기 한다 !

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

■ **절대주소**: 현재 어느 디렉터리에 있건 변하지 않는 경로로서 최상위 디렉터리에서부터 목적지까지의 경로.

■ **상대주소**: 현재 디렉터리로부터의 경로.

3. 기본 명령어

■ 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

3. 기본 명령어

■ mkdir

- Make directory: 디렉터리 생성

- 옵션

-p: 만들고자 하는 디렉터리의 상위 디렉터리가 없는 경우 상위 디렉터리 까지 만들어 준다.

- 사용 예: mkdir [디렉터리명]

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

■ 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

■ 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 date1.out date2.out
[root@root root] cp date1.out date3.out
[root@root root] cp date* ./test/
[root@root root] cp -i date* ./test/
[root@root root] cd test
[root@root test] ls
date1.out  date2.out  date3.out
```

3. 기본 명령어

■ 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 date.out date1.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**

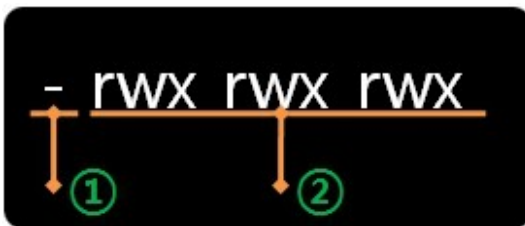
- 위와 같이 본인 계정이 가지지 않는 권한의 디렉터리로 접근 하려고 하면 거부 된다.
- 권한은 해당 계정의 자료를 보호하는 기능을 한다.
- 따라서 권한에 대한 이해는 매우 중요하다.

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_sock04
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-XXZuar2T
-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

■ 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)

권한을 표시하는 문자

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		

뒤에 따라하기를 해보면서 권한을 변경을 해보도록 한다.

3. 기본 명령어 – 권한 변경

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

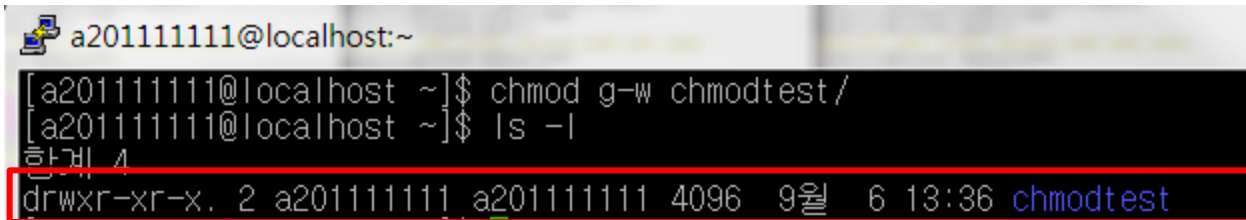
```
a201111111@localhost:~  
[a20111111@localhost ~]$ mkdir chmodtest  
[a20111111@localhost ~]$ ls -l  
한글 4  
drwxrwxr-x. 2 a201111111 a201111111 4096  9월  6 13:36 chmodtest
```

- 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**

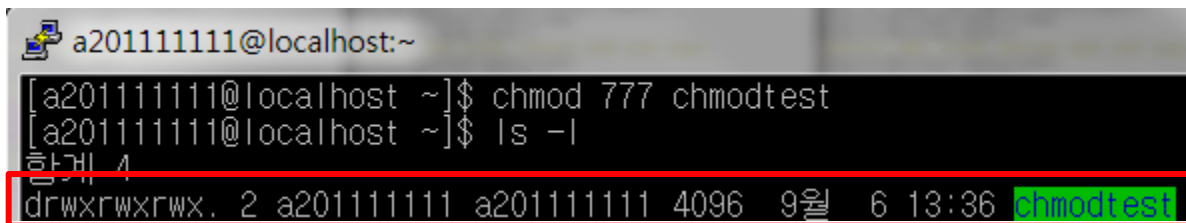


```
a201111111@localhost:~  
[a201111111@localhost ~]$ chmod g-w chmodtest/  
[a201111111@localhost ~]$ ls -l  
한글 4  
drwxr-xr-x. 2 a201111111 a201111111 4096 9월 6 13:36 chmodtest
```

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

3. 기본 명령어

■ 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 디렉터를 사용자가 제거 할 수 있다.
 - ▶ `-s`: `-p` 선택항목 지정 시 표준오류에 출력되는 메시지를 삭제
- 사용 예

```
[root@root test] rm date1.out
[root@root test] rm date*.out
[root@root test] ls
```

■ cat

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

3. 기본 명령어

■ 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
```


1. 디렉터리 생성, 권한 변경, 목록출력 실습

- 자신의 학번으로 디렉터를 생성
- 디렉터리 접근 권한을 소유자 전용으로 변경
- 변경된 권한을 화면에 출력
- Hint: `chmod [권한] [대상폴더]`

2. 복사, 압축해제, 이동(이름변경), 압축 실습

- `/home/syspro/lab01.tar.gz`을 복사
- 압축해제
- lab01 디렉터를 학번으로 이름 변경
- 변경된 학번 디렉터를 **분반_학번.tar.gz**로 압축

3. 삭제 실습

- 영문이름으로 디렉터리를 생성
- 영문이름 디렉터리 안에 학번 디렉터리를 생성
- 한번에 생성한 2개의 디렉터리 삭제

3. 기본 명령어

■ 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 일치하면 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]$
```

644
644
644
644
644
644

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

- ▶ -atime count : 파일에 접근한 날 수가 count 이내 이면 참

```
[eslab@localhost FindGrep]$ find -atime -10
.
./testfile
./test.c
./pineapple.c
./banana.java
./directory1
./directory1/tomato.c
./directory1/melon.c
./directory2
./apple.c
./orange.txt
```

-10의 의미:
10일 이내에 접근한 파일

- ▶ -mtime count : 파일을 수정한 날 수가 count 이내이면 참

```
[eslab@localhost FindGrep]$ find -mtime -3
.
./testfile
./test.c
./pineapple.c
./banana.java
./directory1
./directory1/tomato.c
./directory1/melon.c
./directory2
./apple.c
./orange.txt
```

-3의 의미:
3일 이내에 수정한 파일

- ▶ -ctime count : 파일이 수정되고 파일 속성이 바뀐 날수가 count 이내

3. 기본 명령어

- ▶ -exec command : 수행중인 command의 종료 값이 0이면 참 ㅍ로 끝남, command의 argument가 {}이면, find가 찾은 파일을 의미
- ▶ <http://byeonely.tistory.com/111>

```
[eslab@localhost FindGrep]$ find -name "*.java" -exec rm {} \;
[eslab@localhost FindGrep]$ ls
apple.c  directory1  directory2  orange.txt  pineapple.c  test.c  testfile
[eslab@localhost FindGrep]$
```

- ▶ -ls : 현재 파일의 속성을 보여주고 참 값을 반환

```
[eslab@localhost FindGrep]$ find -ls
4194359    4 drwxrwxrwx    4 root    root          4096 Sep  6 21:45 .
4196972    4 -rw-r--r--    1 eslab    root           234 Sep  6 05:36 ./testfile
4196975    4 -rw-r--r--    1 eslab    root            26 Sep  6 20:46 ./test.c
4194640    0 -rw-rwxrwx    1 eslab    root            0 Sep  6 05:26 ./pineapple.c
4194368    4 drwxr-xr-x    2 root    root          4096 Sep  6 21:28 ./directory1
4196971    4 -rw-rwxrwx    1 root    root           20 Sep  6 21:28 ./directory1/tomato.c
4196973    4 -rw-rwxrwx    1 root    root            6 Sep  6 06:11 ./directory1/melon.c
4196969    4 drwxr-xr-x    2 root    root          4096 Sep  6 05:09 ./directory2
4196976    4 -rw-r--r--    1 eslab    root            26 Sep  6 20:45 ./apple.c
4196968    0 -rw-r--r--    1 eslab    root            0 Sep  6 05:08 ./orange.txt
[eslab@localhost FindGrep]$
```

3. 기본 명령어

- ▶ -name 'pattern' : 파일명이 'pattern'과 일치이면 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

- ▶ -atime count : 파일에 접근한 날 수가 count 이내 이면 참

```
[eslab@localhost FindGrep]$ find -atime -10
.
./testfile
./test.c
./pineapple.c
./banana.java
./directory1
./directory1/tomato.c
./directory1/melon.c
./directory2
./apple.c
./orange.txt
```

-10의 의미:
10일 이내에 접근한 파일

- ▶ -mtime count : 파일을 수정한 날 수가 count 이내이면 참

```
[eslab@localhost FindGrep]$ find -mtime -3
.
./testfile
./test.c
./pineapple.c
./banana.java
./directory1
./directory1/tomato.c
./directory1/melon.c
./directory2
./apple.c
./orange.txt
```

-3의 의미:
3일 이내에 수정한 파일

- ▶ -ctime count : 파일이 수정되고 파일 속성이 바뀐 날수가 count 이내

3. 기본 명령어

- ▶ -exec command : 수행중인 command의 종료 값이 0이면 참 ㅍ로 끝남, command의 argument가 {}이면, find가 찾은 파일을 의미
- ▶ <http://byeonely.tistory.com/111>

```
[eslab@localhost FindGrep]$ find -name "*.java" -exec rm {} \;
[eslab@localhost FindGrep]$ ls
apple.c  directory1  directory2  orange.txt  pineapple.c  test.c  testfile
[eslab@localhost FindGrep]$
```

- ▶ -ls : 현재 파일의 속성을 보여주고 참 값을 반환

```
[eslab@localhost FindGrep]$ find -ls
4194359    4 drwxrwxrwx    4 root    root          4096 Sep  6 21:45 .
4196972    4 -rw-r--r--    1 eslab    root           234 Sep  6 05:36 ./testfile
4196975    4 -rw-r--r--    1 eslab    root            26 Sep  6 20:46 ./test.c
4194640    0 -rw-rwxrwx    1 eslab    root            0 Sep  6 05:26 ./pineapple.c
4194368    4 drwxr-xr-x    2 root    root          4096 Sep  6 21:28 ./directory1
4196971    4 -rw-rwxrwx    1 root    root           20 Sep  6 21:28 ./directory1/tomato.c
4196973    4 -rw-rwxrwx    1 root    root            6 Sep  6 06:11 ./directory1/melon.c
4196969    4 drwxr-xr-x    2 root    root          4096 Sep  6 05:09 ./directory2
4196976    4 -rw-r--r--    1 eslab    root            26 Sep  6 20:45 ./apple.c
4196968    0 -rw-r--r--    1 eslab    root            0 Sep  6 05:08 ./orange.txt
[eslab@localhost FindGrep]$
```

3. 기본 명령어

■ 사용 예

- 확장자가 c인 파일을 현재 디렉터리 이하에서 찾기
 - ▶ `find . -name '*.c'`
- 확장자가 bak인 파일을 찾아 속성을 표시한 후 삭제
 - ▶ `find . -name '*.bak' -ls -exec rm {} \;`
- 현재 디렉터리 이하에서 user가 syspro인 디렉터리 찾기
 - ▶ `find . \((-type d -a -user syspro \)`
 - ▶ AND: `expression -a expression`
 - ▶ OR: `expression -o expression`
 - ▶ NOT: `! expression`

3. 기본 명령어

■ grep

- 사용법: `grep [<option>] <pattern> [<file name>]`

- 옵션

- ▶ -i : 영문의 대소문자를 구분하지 않음
- ▶ -v : pattern을 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`

3. 기본 명령어

■ 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`
- ▶ `find / -name '*.c' 2>/dev/null`
 - `2>/dev/null` 는 표준 에러를 표시하지 않는다는 명령어

3. 기본 명령어

■ Pipe

- 기호: | (shift + ₩)
- 기호를 기준으로 왼쪽 명령어의 출력을 오른쪽 명령어의 입력으로 보낸다.
- 사용 예
 - ▶ `cat /etc/passwd | more`
 - ▶ `ls /etc/rc5.d | sort -r | grep S > result`

4. 검색 실습

- /home/syspro/lab01.tar.gz을 복사
- 압축 해제
- lab01 디렉터리에서 **test.c** 파일이 어디에 위치하는지 검색
- lab01 디렉터리에서 **test2.c** 파일이 어디에 위치하는지 검색
 - ▶ 주의: 반드시 Find 이용

5. 백업 실습

- 백업 디렉터리(/backup)를 생성
- 본인 계정의 모든 파일을 백업
 - ▶ 주의: 반드시 tar 이용

6. Pipe 실습

- lab01의 List 명령을 사용하여 최근에 생성된 순서대로 파일의 inode 번호를 포함한 결과를 **I/O Redirection**을 이용해서 자신의 계정 디렉터리에 list.txt로 저장
 - ▶ Hint: 실습자료 10쪽 참조, I/O redirection 명령 사용

7. cat 명령어를 사용하여 grepfile 을 만들어라

Well you know it's your bedtime,
So turn off the light,
Say all your prayers and then,
Oh you sleepy young heads dream of wonderful things,
Beautiful mermaids will swim through the sea,
And you will be swimming there too.

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

8. grepfile에서 the 글자가 있는 줄의 목록을 걸러 줄 번호와 함께 화면에 출력하라

9. grepfile에서 your 라는 단어가 없는 줄의 목록을 걸러 result 파일에 출력하라

10. grepfile에서 your 라는 단어가 없는 줄의 수를 출력하라. (cont 값)

4. vi 에디터 – vi 모드의 이해

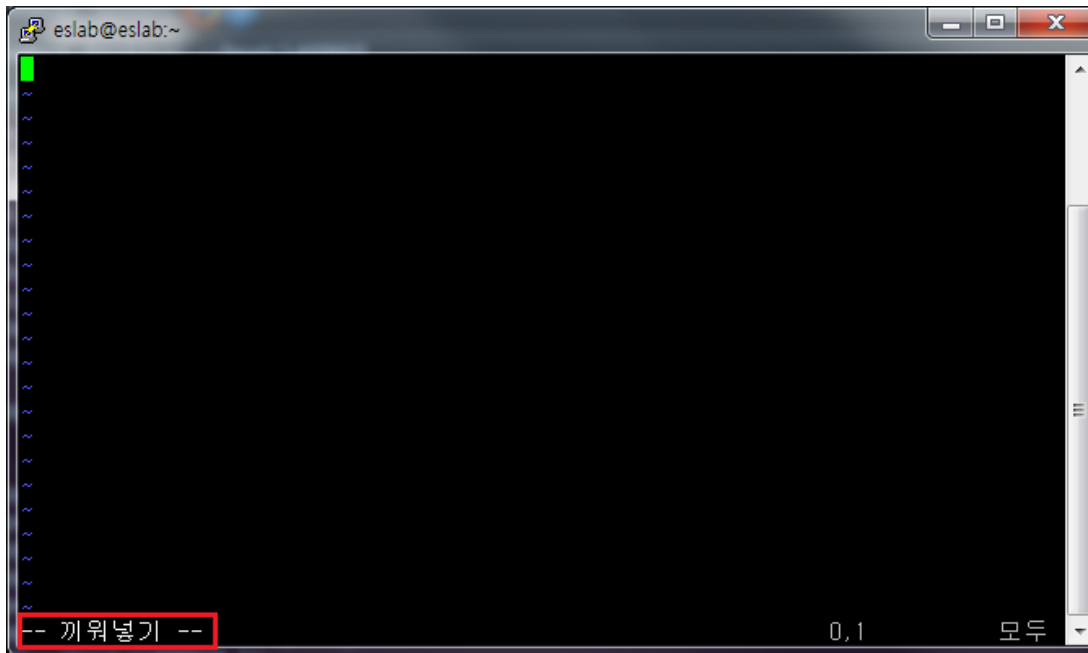
- vi 에디터는 코드를 작성할 때 쓰이는 편집 도구이다.
- **표준모드**
 - 실행 명령어: vi 파일명



- 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

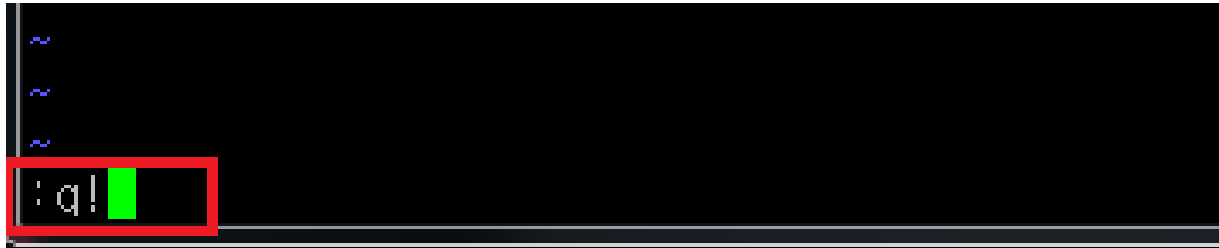
- **입력 모드**
- vi 에디터 표준모드에서 'i', 'a', 'o', 's' 4개의 키 중 하나를 누른다.



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

4. vi 에디터 – vi 모드의 이해

- 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

설 명

i	텍스트가 커서 앞에 삽입
I	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


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

4. vi 에디터

■ 화면 이동

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

- 블록 지정 방법 따라하기
 - 표준 모드에서 **v** 키를 눌러 블록 설정을 한 후 삭제.



1 hello world
2 hello world
3 hello world
4 hello world
5 hello world
6 hello world
7 hello world
8 hello world
9 hello world
10 hello world
11 hello world
12 hello world
13 hello world
14 hello world
15 hello world
16 hello world
17 hello world
18 hello world
19 hello world
20 hello world
21 hello world
22 hello world
23 hello world
24 hello world
25 hello world
26 hello world
27 hello world
28 hello world

~ 삭제하고 싶은 영역을 커서
~ 를
~ 이용해서 블록 잡는다.



```
d201153100@localhost:~  
1 hello world  
2 hello world  
3 hello world  
4 hello world  
5 hello world  
6 hello world  
7 hello world  
8 hello world  
9 hello world  
10 hello world  
11 hello world  
12 hello world  
13 e llo world  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
d 키를 입력해서 블록 영역을  
삭제 한다.  
~  
~  
15 fewer lines
```


4. vi 에디터

■ 텍스트 삭제

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
:<range> d	<range> 삭제
: 1,\$ d	첫 줄부터 마지막 줄까지 삭제
: .,-2 d	현재 줄부터 이전 두 줄까지 삭제



4. vi 에디터

■ 텍스트 치환

설 명	
CW	단어 변경
CC	행 변경
C	커서 오른쪽의 행 변경

■ 탐색 및 치환

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

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

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

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

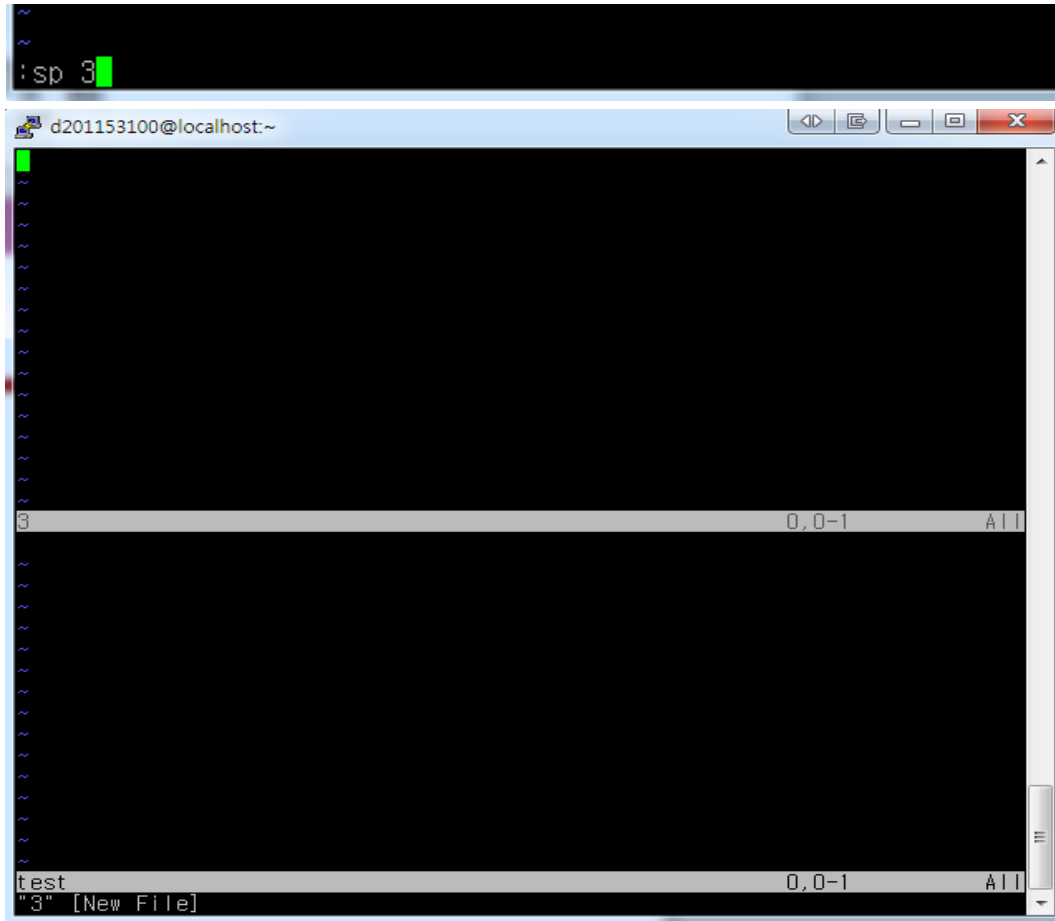
4. vi 에디터

■ 다중 창 관련 명령

설 명	
ctrl+w s or :sp[plit]	현재 파일을 두 개의 수평 창으로 나눔
ctrl+w v or :[N]vs[plit]	새로운 수직 창 생성, N이 붙으면 N칸 크기의 창 분할
ctrl+w n or :new	새로운 수평 창 생성
ctrl+w q	현재 커서의 창을 종료
ctrl+w 방향키	원하는 창으로 이동
ctrl+w =	창의 크기를 모두 균등하게 함
:20vs ./ or 20sp ./	탐색기 열기

4. vi 에디터

■ 다중 창 따라하기

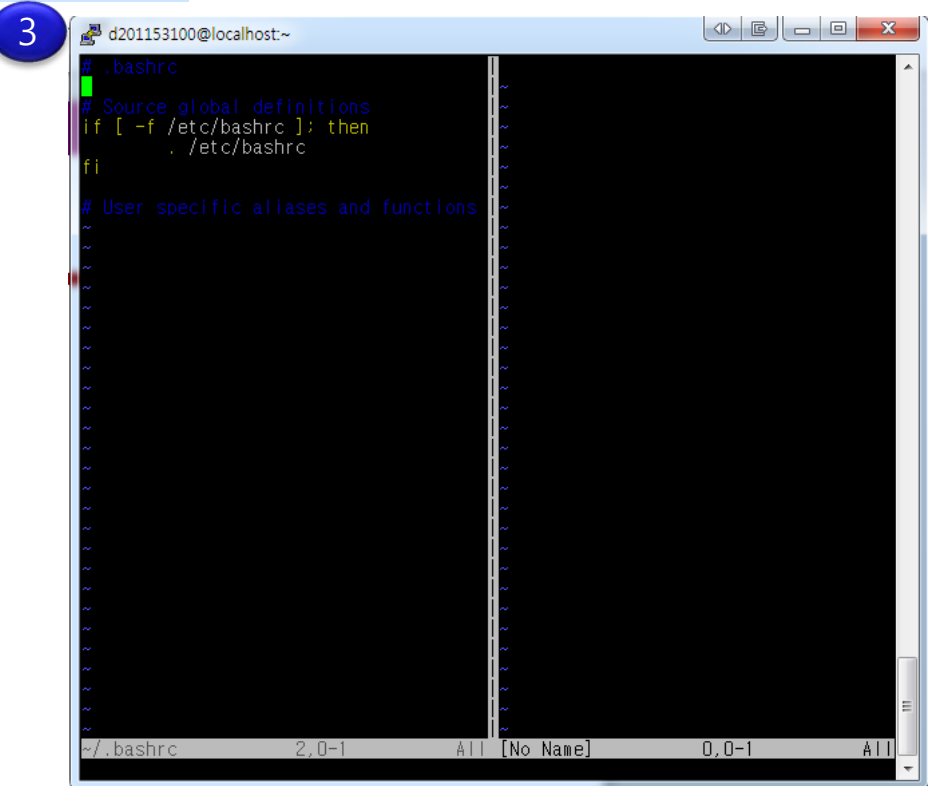
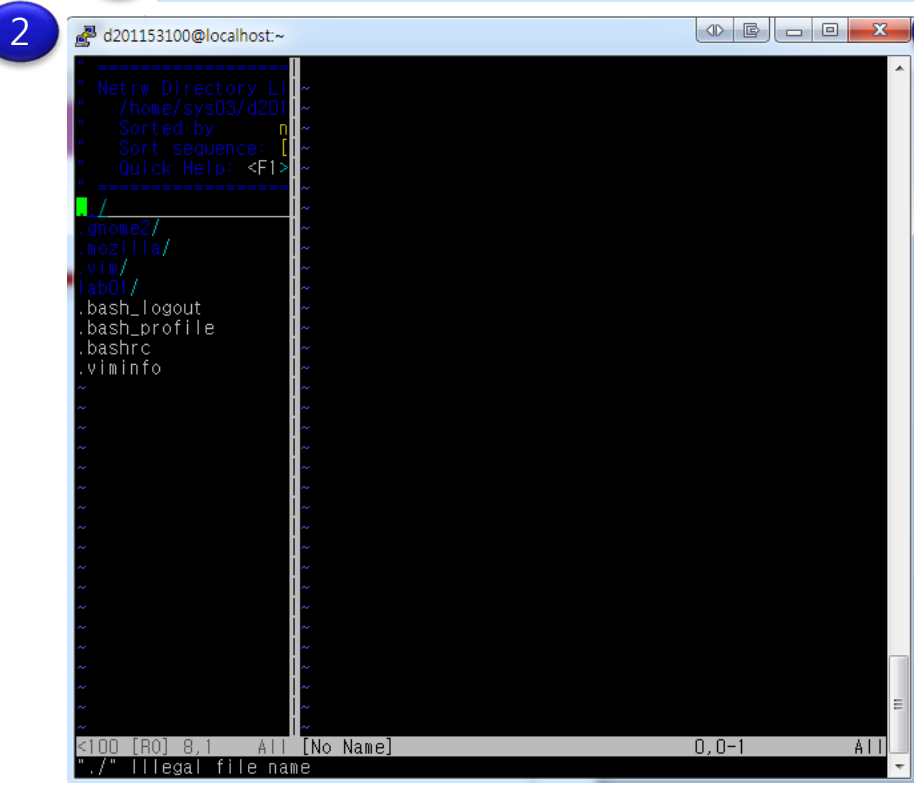


■ 창을 가로로 분할하고 3이라는 파일을 없으면 생성해서 연다.

4. vi 에디터

■ 따라하기

1 :20vs ./



■ 명령라인 모드에서 ./20vs를 하면 세로로 20정도의 크기의 탐색기 창이 열린다. 탐색기에서 특정 파일을 선택하면 탐색기는 현재 vi창에 해당 파일의 내용을 불러 온다.

■ vi 에디터로 testfile을 만들어라

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

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

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

■ **copy and paste** 명령어를 사용해서 상위 2줄의 내용을 복사해서 2번 붙여 넣기 한다.

11. 같은 행에서 처음과 마지막으로 이동하는 명령어 조사

12. 문서의 시작과 끝으로 한번에 이동 하는 명령어 조사

13. 위 문서에서 모든 **you**를 **I**로 변경 한다.

조건: vi 에디터의 기능인 치환 기능을 이용해서 한번에 변경

14. 두 번째 줄 뒤에 “replace worlds, delete worlds” 라인을 추가

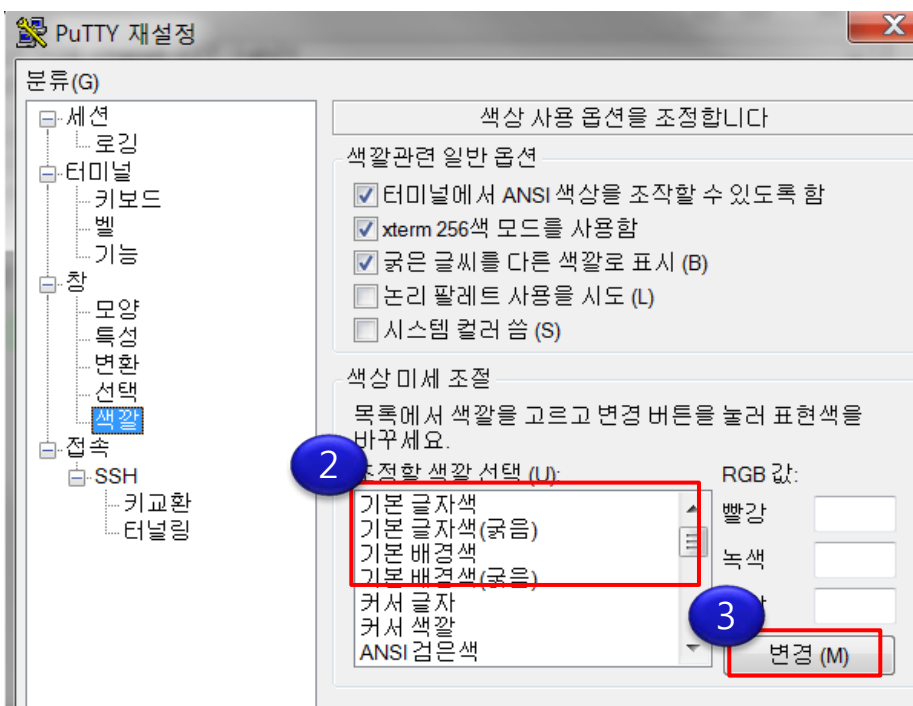
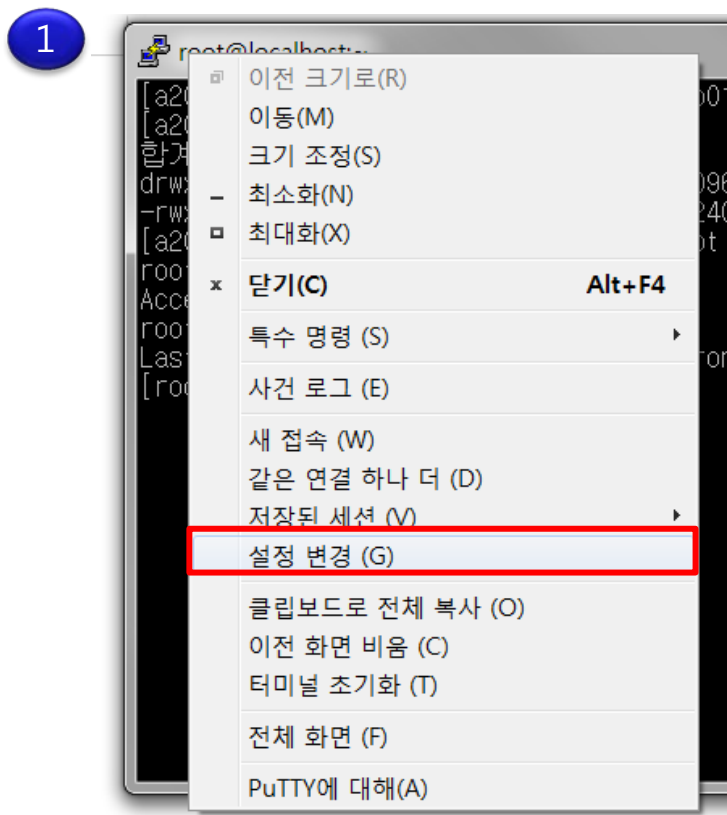
15. 3번째 단락을 한번에 vi 명령어를 이용해서 삭제한다.

제출사항

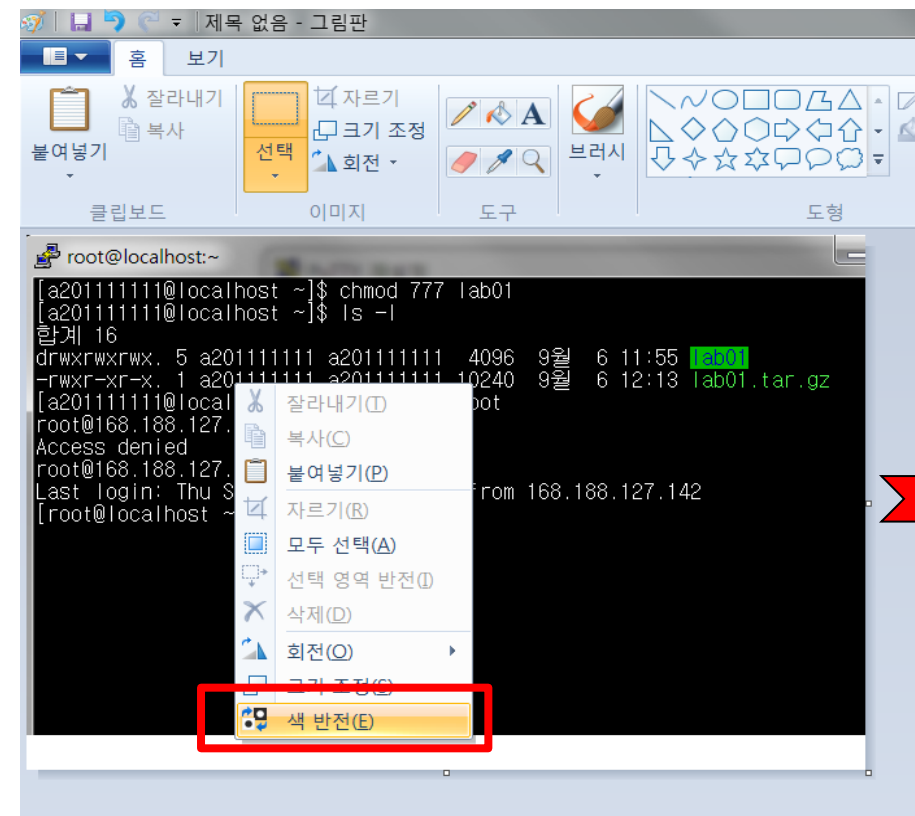
- 오늘 실습한 15개의 과정들을 문서로 작성하여 이메일과 서면으로 제출
 - sihyeong@cnu.ac.kr
 - 제목 양식: [sys02]HW01_학번_이름
 - 파일 제목: [sys02]HW01_학번_이름
 - 반드시 메일 제목과 파일 제목 양식을 지켜야 함. (위반 시 감점)
 - 보고서는 제공된 양식 사용
- 자신이 실습한 내용을 증명할 것 (자신의 학번이 항상 보이도록)
- 제출일자: 2014년 9월 10일 수요일 23시 59분 59초까지...
 - 서면 제출 : 2014년 9월 11일 수업시간

참고) putty 배경 및 글자 변경 방법

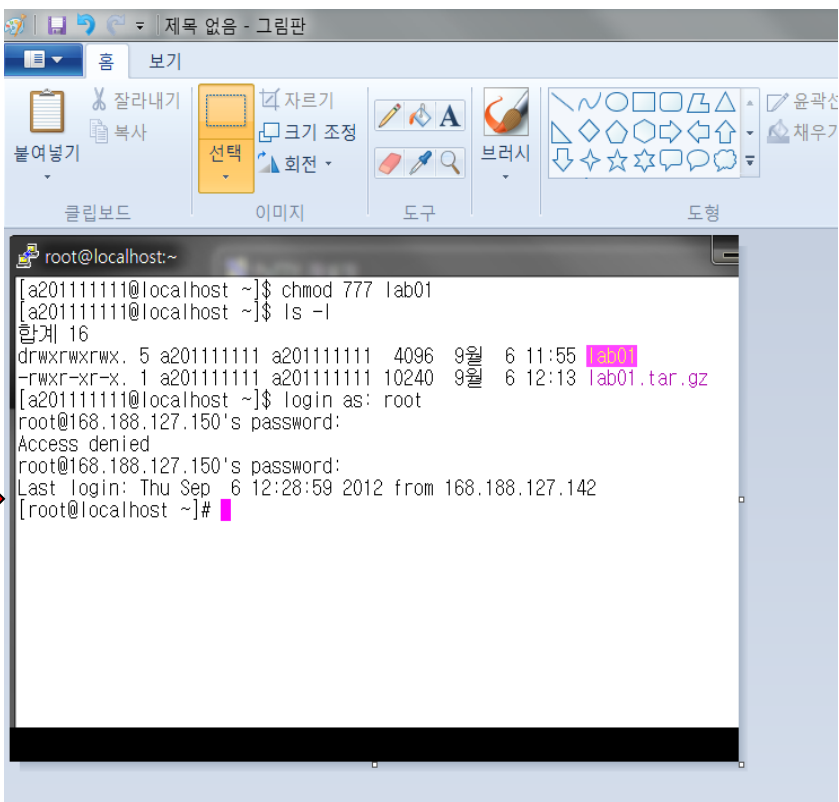
- 글자 색은 검은색 배경색은 흰색으로 변경 그 외 설정 변경도 이곳에서..
- 접속 할 때 왼쪽 메뉴 또는 접속 후 상태 창 '우 클릭' - 설정 변경
- 기본 글자 색 : 0/0/0, 기본 배경색 : 255/255/255



참고) 그림판 색 반전



캡처하여 그림판으로 복사



색 반전 후 저장