

Basic Linux (ver. 2. 5. 1)

(기초 리눅스 명령어)

2012. 3. 25

박 근 홍

0. 들어가는 글

2009년 처음으로 천문관측 및 실험 1의 조교를 맡았습니다. 그간 천문학과에서 널리 사랑을 받아온 선배님들의 안내서들¹⁾(ver. 1. 0)을 바탕으로 하여 내용을 추가, 구조를 재편하여 두 개의 안내서²⁾(ver. 2. 0)를 만들었습니다.

그로부터 3년이 지났습니다. 그 사이 천문학과는 전산 실습실을 갖췄고, 덕분에 학생들은 관측 자료 처리를 그간 사용되어왔던 astro10 서버가 아닌 전산 실습실에서 할 수 있게 되었습니다. 그래서 astro10에 깔려 있던 c기반의 구버전 IRAF가 아닌, ecl기반의 최근 IRAF를 기준으로 IRAF 안내서를 다시 손보게 되었습니다. 안내서를 만들기에 앞서 6년 전, 복학하고 학부 천문 관측 및 실험 1 수업을 들었던 때, 제가 어떤 점이 어려웠는지를 떠올려 보았습니다. 아무래도 리눅스란 운영 체제를 처음으로 접했음에도 불구하고, 수업 실습은 리눅스 명령어를 기본적으로 알고 있다는 전제 아래 진행되었던 점이 큰 어려움으로 다가왔던 것이 기억에 남더군요. 그러한 이유에서 지난 안내서에 리눅스 부분을 일부 추가하였지만, 양이나 깊이에서 아쉬운 점이 많았습니다. 이에, 이번 안내서 판에서는 리눅스 명령어 부분을 보강하고 독립시켰습니다. vi 에디터 부분에서 SAGA에 올려주신 이준협 선배님의 정리본이 큰 도움이 되었습니다. 이 자리를 빌어 감사의 말씀을 올립니다. 아울러 많은 조언을 제기해주시는 김성진 선배님께도 감사의 말씀을 올립니다.

해당 안내서의 오류 및 건의 사항이 있으면 khpark@astro.snu.ac.kr로 언제든지 알려주시면 수정 혹은 적극 반영토록 하겠습니다. 부족하지만 천문 관측 및 실험 수강생들의 리눅스 입문에 도움이 되길 바랍니다.

2012년 3월, 서울대학교에서

박 근 홍

(※ SAGA 홈페이지³⁾에서 가장 최근 버전의 안내서를 받을 수 있습니다.)

1) IRAF ABC (황호성, 이명균, 2004), DAOPHOT을 이용한 PSF측광 (황호성, 이명균, 2004), IRAF/PHOT을 이용한 변광성 측광 (황호성, 이명균, 2006), PHOTCAL Tutorial (이명균, 이준협, 김상철, 2002)

2) Linux 명령어 및 IRAF를 이용한 전처리 (황호성, 이명균, 박근홍, 이광호, 이상각, 박홍수, 류진혁, 2009), IRAF를 이용한 구경측광, PSF측광 그리고 표준화 (황호성, 이명균, 이준협, 박근홍, 이광호, 이상각, 박홍수, 류진혁, 2009)

3) <http://astro.snu.ac.kr/~grad>

I. 사용자 정보 확인

1) who, finger, w

현재 시스템을 이용하는 사용자를 화면에 출력한다. (출력 정보는 서로 조금씩 다르다.)

```
[pkhinkorea@astro10 ~]$ who
root      :0                Feb 20 16:35
pkhinkorea pts/1          Mar 25 14:25
[pkhinkorea@astro10 ~]$ w
14:25:03 up 231 days, 53 min,  2 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
root      :0        -             20Feb12 ?xdm?  21:19   0.21s /usr/bin/gnome-
pkhinkor pts/1     -             14:25   0.00s  0.01s  0.00s w
[pkhinkorea@astro10 ~]$ finger
Login      Name      Tty      Idle  Login Time   Office   Office Phone
pkhinkorea          pts/1           Mar 25 14:25
root       root      *:0           Feb 20 16:35
```

2) whoami, who am I

whoami는 내가 로그인한 사용자를 보여준다. who am i는 추가로 접속 단말기와 접속 시간, 호스트 이름 등의 정보를 보여준다.

```
[pkhinkorea@astro10 ~]$ whoami
pkhinkorea
[pkhinkorea@astro10 ~]$ who am i
pkhinkorea pts/1          Mar  5 21:24
```

3) passwd

비밀번호를 변경하고 싶은 경우 passwd라고 입력한다. 현재 비밀번호, 변경할 비밀번호, 변경할 비밀번호 확인을 입력하면 해당 계정의 비밀번호를 변경할 수 있다.

```
[pkhinkorea@astro10 ~]$ passwd
Changing password for user pkhinkorea.
Changing password for pkhinkorea
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

II. 디렉토리 및 파일 이동/복사/생성/삭제 정보 보기

① 디렉토리 생성 및 이동하기

pwd(print name of current working directory)는 현재 자신이 있는 디렉토리의 위치를 나타낸다.

mkdir(make directory)은 디렉토리를 만드는데 사용한다.

용법 : \$ mkdir [디렉토리 명]

notice 1, 이 안내서에서 반드시 입력이 되어야하는 부분은 [괄호] 로 나타낸다, 이 부분이 없으면 에러 메시지가 출력된다, 입력이 되지 않아도 되는 옵션은 (괄호)로 나타내겠다.

cd(change directory) 명령어는 디렉토리를 변경하는 명령어다.

용법 : \$ cd (경로)

같은 명령어를 치더라도 현재 위치에 따라 이동하는 디렉토리가 달라질 수 있는 것을 상대적으로 (행정구역을 저장한 시스템이 있다고 가정할 때, 각각 /서울시/관악구/, /서울시/은평구/, /서울시/강남구/에서 **cd 신사동** 이란 같은 명령을 실행하더라도 이동되는 디렉토리는 각각 '/서울시/관악구/신사동/, /서울시/은평구/신사동/, /서울시/강남구/신사동/'으로 다르다.), 어느 위치에서 명령어를 치더라도 항상 같은 디렉토리로 이동하는 것을 절대경로(가령 **cd /서울시/중구/명동/** 이라고 입력하면 내가 어느 디렉토리에 있는지와 상관없이 전부 동일한 위치로 이동된다.)라고 한다.

상대경로 명령어로는

cd 경로 : 현 디렉토리의 하위 디렉토리로 이동한다.

cd ./경로 : 위의 명령어와 동일

cd ../경로 : ..은 현디렉토리보다 한단계 상위 디렉토리를 의미한다

등이 있고 절대경로 명령어는

cd /경로 : 경로에 해당하는 위치로 이동한다

cd ~/경로 : ~은 자신의 홈디렉토리를 의미한다.

등이 있다. 그 외에

cd : 홈디렉토리로 이동

Basic Linux

cd ~ : 역시 홈 디렉토리로 이동

cd / : 루트디렉토리로 이동

cd ~ (아이디): 해당 아이디의 홈디렉토리로 이동

```
[pkhinkorea@astro10 ~]$ pwd ; 내 위치 확인
/home/pkhinkorea
[pkhinkorea@astro10 ~]$ cd ~zknight ; home 아래 zknight 폴더로 이동
[pkhinkorea@astro10 zknight]$ pwd
/home/zknight
[pkhinkorea@astro10 zknight]$ cd ; 내 홈디렉토리 (home/pkhinkorea) 이동
[pkhinkorea@astro10 ~]$ mkdir M50 ; M50 폴더 생성
[pkhinkorea@astro10 ~]$ cd M50 ; M50 폴더로 이동
[pkhinkorea@astro10 M50]$ pwd
/home/pkhinkorea/M50
[pkhinkorea@astro10 M50]$ cd ../hw ; 한 단계 위 폴더(..)의 아래에 있는 hw 폴더로 이동
[pkhinkorea@astro10 hw]$ pwd
/home/pkhinkorea/hw
[pkhinkorea@astro10 hw]$ cd .. ; 한 단계 위 폴더로 이동
[pkhinkorea@astro10 ~]$ pwd
/home/pkhinkorea
[pkhinkorea@astro10 ~]$ cd /home/pkhinkorea/hw ; 현재 위치와 상관없이 /home/pkhinkorea/hw로 이동
[pkhinkorea@astro10 hw]$ cd ../../ ; 두 단계 위 폴더로 이동
[pkhinkorea@astro10 home]$ pwd
/home
[pkhinkorea@astro10 home]$ cd /home/zknight/Desktop/ ; 현재 위치와 상관없이 /home/zknight/Desktop로 이동
[pkhinkorea@astro10 Desktop]$ pwd
/home/zknight/Desktop
[pkhinkorea@astro10 home]$ cd ~/hw ; 현재 위치와 상관없이 /home/pkhinkorea/hw로 이동
[pkhinkorea@astro10 hw]$ pwd
/home/pkhinkorea/hw
[pkhinkorea@astro10 hw]$ cd / ; 현재 위치와 상관없이 최상위 폴더로 이동
[pkhinkorea@astro10 /]$ pwd
/
[pkhinkorea@astro10 /]$ cd ~ ; 내 홈 디렉토리로 이동
[pkhinkorea@astro10 ~]$ pwd
/home/pkhinkorea
```

② 파일 목록보기

현재 디렉토리의 파일 목록을 보는 명령어는 ls(list segments)명령어이다.

용법 : \$ ls (옵션) (경로 혹은 조건)

```
[pkhinkorea@astro10 ~]$ cd hw
[pkhinkorea@astro10 hw]$ ls
catalog.txt  n2041b.fits  n4621_IJ.fits  observation.txt  result1.txt  se          stdcat2.cfg  stdresult1.txt
galaxy       n2041v.fits  n4621_IR.fits  result1          result2.txt  stdcat1.cfg  stdresult.txt
```

다른 위치에서 파일의 목록을 보기 위해서는 `ls home/zknight`과 같이 `ls` 뒤에 보고자 하는 경로를 입력해주면 된다. 이 때 경로를 입력하는 방식은 앞의 `cd`의 입력 방식과 동일하다.

옵션 값은 다음과 같고, 중복하여 사용할 수 있다.

- a : 숨겨진 파일도 출력
- l : 파일의 상세한 정보 (권한, 용량, 수정날짜)등을 볼 수 있다.
 - lh: 파일의 용량을 단위와 함께 출력
- S: 크기순으로 출력
- r: 역순으로 출력
- t: 시간 순으로 출력
- F: 파일과 디렉토리를 구별해서 출력 (디렉토리 뒤에 /가 붙음)
- x: 정렬순서를 가로로 출력
- u: 최종적으로 사용된 시간 순서대로 출력

```
[pkhinkorea@astro10 hw]$ ls -tral
total 1176
-rw-r--r-- 1 pkhinkorea master 646 Apr 14 2008 catalog.txt
-rw-r--r-- 1 pkhinkorea master 1288 Apr 14 2008 observation.txt
-rw-r--r-- 1 pkhinkorea master 201 Apr 14 2008 stdcat1.cfg
...
```

또는, 파일명이 특정조건을 만족해주는 대상만 보고 싶을 때는 `ls dark*.fit`과 같이 `ls` 뒤에 해당 조건을 적어두면 된다. 또한 `ls`뒤에 다음과 같은 옵션을 붙여서 다양한 기능을 수행할 수 있다.

```
[pkhinkorea@astro10 hw]$ ls *txt -tla
-rw-r--r-- 1 pkhinkorea master 936 Jun 1 2010 stdresult1.txt
-rw-r--r-- 1 pkhinkorea master 1047 Mar 29 2009 stdresult.txt
-rw-r--r-- 1 pkhinkorea master 936 Apr 14 2008 result1.txt
...
```

tip 1, 와일드카드의 활용

와일드카드는 여러 파일을 한꺼번에 지정할 목적으로 사용하는 기호로, `*`과 `?` 두 종류가 있다. `?`는 임의의 한 문자를 의미하여 단 한 글자를 대체하며, `*`는 임의의 문자열을 의미하여 여러 글자를 대체해준다. 가령 세상의 모든 영어 단어에서 `a?e`를 검색하면 3글자에 `a`로 시작하고 `e`로 끝나는 단어를 찾아주며, `a*e`를 검색하면 글자 수와 상관없이 `a`로 시작하고 `e`로 끝나는 단어를 찾아준다. 도끼(`axe`)는 두 조건에서 다 검색이 되지만, 사과(`apple`)는 `a*e`에서는 검색이 되지만, `a?e`에는 검색되지 않는다. 다음의 예시를 보자,

```
[pkhinkorea@astro10 pkh]$ ls
a01 a02 a03 a10 a12 a123
[pkhinkorea@astro10 pkh]$ ls a*
a01 a02 a03 a10 a12 a123
```

```
[pkhinkorea@astro10 pkh]$ ls a?
ls: a?: No such file or directory
[pkhinkorea@astro10 pkh]$ ls a??
a01 a02 a03 a10 a12
[pkhinkorea@astro10 pkh]$ ls a???
a123
[pkhinkorea@astro10 pkh]$ ls ?0?
a01 a02 a03
[pkhinkorea@astro10 pkh]$ ls *3
a03 a123
```

③ 이동, 복사, 삭제

cp(copy)는 파일 및 디렉토리의 복사를 담당하는 명령이다.

용법 : \$ cp (옵션) [원본] [복사본]

cp의 옵션은 다음과 같은 항목들이 있다.

- a : 원본파일의 속성, 링크 정보를 그대로 유지
- b : 같은 이름의 파일이 있을 때 백업파일 생성 후 복사
- f : 같은 이름의 파일이 있을 때 삭제 후 복사 (덮어쓰기)
- u : 같은 이름의 파일이 있을 때 변경날짜가 같거나 최근이면 복사하지 않음 (업데이트)
- i : 같은 이름의 파일이 있으면 사용자에게 물어본다
- r : 하위 디렉토리까지 복사
- p : 권한까지 복사
- v : 복사 과정을 보여준다

```
[pkhinkorea@astro10 pkh]$ ls
a01 a02 a03 a10 a12 a123
[pkhinkorea@astro10 pkh]$ cp a01 b01 ; a01파일을 b01에 복사하라
[pkhinkorea@astro10 pkh]$ ls
a01 a02 a03 a10 a12 a123 b01
[pkhinkorea@astro10 pkh]$ cp -i a02 b01 ; a02파일을 b01에 복사하되, 중복파일은 물어보라.
cp: overwrite `b01'? yes
[pkhinkorea@astro10 pkh]$ mkdir test ; test폴더를 만들어라.
[pkhinkorea@astro10 pkh]$ ls
a01 a02 a03 a10 a12 a123 b01 test
[pkhinkorea@astro10 pkh]$ ls test ; test폴더에 있는 파일 및 폴더를 나타내라 (아무 것도 없음)
[pkhinkorea@astro10 pkh]$ cp a0? test ; a0? 조건을 만족하는 파일을 test폴더로 복사하라
[pkhinkorea@astro10 pkh]$ ls test ; test폴더에 있는 파일 및 폴더를 나타내라
a01 a02 a03
[pkhinkorea@astro10 pkh]$ mkdir test/test00 ; test아래 test00이라는 폴더를 만들어라
[pkhinkorea@astro10 pkh]$ ls test
a01 a02 a03 test00
[pkhinkorea@astro10 pkh]$ cp -r test test1 ; test의 하위 디렉토리를 포함해서 test1로 복사하라.
[pkhinkorea@astro10 pkh]$ ls
```

Basic Linux

```
a01 a02 a03 a10 a12 a123 b01 test test1
[pkhinkorea@astro10 pkh]$ ls test1
a01 a02 a03 test00
```

mv(move) 명령어는 파일 및 디렉토리를 옮겨주는 명령어이다. 또한 파일의 이름을 다른 이름으로 변경할 때도 쓰인다.

용법 : \$ mv (옵션) [원본] [이동할 위치, 혹은 새 파일명]

mv의 옵션은 다음과 같다.

- b: 같은 이름의 파일이 있을 때 백업파일 생성 후 이동
- f: 같은 이름의 파일이 있을 때 삭제 후 이동 (덮어쓰기)
- i: 같은 이름의 파일이 있으면 사용자에게 물어본다
- v: 이동 과정을 보여준다

```
[pkhinkorea@astro10 pkh]$ mv b01 test; b01을 test폴더로 이동하라
[pkhinkorea@astro10 pkh]$ ls
a01 a02 a03 a10 a12 a123 test test1
[pkhinkorea@astro10 pkh]$ ls test
a01 a02 a03 b01 test00
[pkhinkorea@astro10 pkh]$ mv a02 b02 ; a02란 파일을 b02란 파일로 이름을 바꿔라
[pkhinkorea@astro10 pkh]$ ls
a01 a03 a10 a12 a123 b02 test test1
[pkhinkorea@astro10 pkh]$ mv test1 test2 ; test1이란 폴더의 이름을 test2로 바꿔라
[pkhinkorea@astro10 pkh]$ ls
a01 a03 a10 a12 a123 b02 test test2
```

rm(remove) 명령어는 파일 및 디렉토리를 삭제할 때 사용한다.

용법 : \$ rm (옵션) [파일 혹은 디렉토리명]

rm의 옵션은 다음과 같다

- i: 삭제 여부를 확인한다.
- f: 확인작업 없이 삭제
- r: 하위 디렉토리까지 삭제
- v: 삭제 과정을 보여준다

```
[pkhinkorea@astro10 pkh]$ rm b02 ; b02파일을 삭제하라
[pkhinkorea@astro10 pkh]$ ls
a01 a03 a10 a12 a123 test test2
[pkhinkorea@astro10 pkh]$ rm * ; 모든 파일을 삭제하라
rm: cannot remove `test': Is a directory ; 시스템 : test는 디렉토리여서 삭제가 안 됩니다.
rm: cannot remove `test2': Is a directory ; 시스템 : test는 디렉토리여서 삭제가 안 됩니다.
[pkhinkorea@astro10 pkh]$ ls
test test2 ; 디렉토리들만 남았다.
```

Basic Linux

```
[pkhinkorea@astro10 pkh]$ rm -rf * ; 디렉토리도 강제로 삭제하고 묻지 마세요
[pkhinkorea@astro10 pkh]$ ls ; 아무 것도 남지 않음
```

④ vi(visual) 에디터 사용하기

vi 에디터는 텍스트 파일의 제작 및 수정을 할 수 있는 강력한 에디터이다. vi는 크게 명령 모드(파일을 편집 및 커서 이동), 입력 모드(파일에 글자를 입력), 실행모드(파일의 저장, 읽기, 외부 명령 실행, 종료 등)로 나뉜다.

1) vi 에디터 실행하기

기본적인 실행방법은 vi 이후 파일명을 입력하여, 그 파일명이 있다면 수정을, 없다면 생성을 한다.

용법 : \$ vi (옵션) [파일명]

```
[pkhinkorea@astro10 pkh]$ ls
login.cl
[pkhinkorea@astro10 pkh]$ vi haha.txt ; haha.txt란 파일이 없었으므로, 새로 생성
[pkhinkorea@astro10 pkh]$ vi login.cl ; 기존의 login.cl 파일을 수정
```

옵션으로는 vi 뒤에 +숫자를 써주면, 그 숫자에 해당하는 행에 커서를 옮긴 채로 시작한다. 또한 +/-후 문자를 적으면, 그 문자가 제일 처음 있는 곳부터 시작한다.

```
[pkhinkorea@astro10 pkh]$ vi +10 login.cl ; 10번째 줄에 커서가 위치
[pkhinkorea@astro10 pkh]$ vi +15 login.cl ; 15번째 줄에 커서가 위치
[pkhinkorea@astro10 pkh]$ vi +/set login.cl ; 가장 앞에 있는 set에 커서가 위치
```

2) 명령 모드

vi로 파일을 불러온 직후의 상태로, 입력 모드에서는 esc를 누를 시 진입한다.

h(=←back space), j(=enter), k, l(=space) : ← ↓ ↑ → 이동

w : 커서의 오른쪽 단어로 이동

e : 커서가 있는 단어의 오른쪽으로 이동

b : 커서가 있는 단어의 왼쪽으로 이동

^ : 행의 맨 왼쪽으로 이동

\$: 행의 맨 오른쪽으로 이동

H : 현재 화면의 맨 위로 이동

M : 현재 화면의 중간으로 이동

L : 현재 화면의 아래쪽으로 이동

Basic Linux

[{ : 단락의 맨 앞으로 이동

}

[+ [: 파일의 맨 앞으로 이동

] +] 또는 [G : 파일의 맨 끝으로 이동

숫자 + [G : 숫자에 해당하는 행으로 이동 (ex> 9G : 9행으로 이동)

ctrl + [f : 한 화면 아래로

ctrl + [b : 한 화면 위로

ctrl + [d : 반 화면 아래로

ctrl + [u : 반 화면 위로

ctrl + [e : 한 행 아래로

ctrl + [y : 한 행 위로

[x], delete : 한글자 삭제

숫자 + [x : 숫자만큼 글자를 삭제 (ex> 3x : 커서로부터 세 글자 삭제)

[d + [w : 한 단어 삭제

숫자 + [d + [w : 숫자만큼 단어 삭제

[d + [d : 한 행 삭제

숫자 + [d + [d : 숫자만큼 행을 삭제

[D 또는 [d + [\$: 현재 행에서, 커서 우측부분을 전부 삭제

[d + [0 또는 [d + [^ : 현재 행에서, 커서 좌측 부분을 전부 삭제

[y + [y 또는 [Y : 행 복사

숫자 + [y + [y 또는 숫자 + [Y : 숫자만큼 행 복사 (ex> 5Y : 다섯 줄 복사)

[P : 복사(삭제)한 행을 현재 행 위에 삽입

[p : 복사(삭제)한 행을 현재 행 아래에 삽입

이를 응용한 것으로, [x + [p : 커서 문자와 그 다음 문자의 위치를 교환

숫자 + [P 또는 [p : 숫자만큼 반복해서 삽입 (ex> 5p 현재 행 아랫줄에 복사한 행을 다섯 번 반복하면서 복사)

[u : 실행 취소

숫자 + [u : 숫자만큼 실행 취소 (3u : 마지막으로 적용한 세 단계의 실행을 취소한다)

[. : 마지막 명령 반복

[/문자열 : 해당 문자열을 아래 방향으로 검색

[?문자열 : 해당 문자열을 위 방향으로 검색

[n : 다음 찾기

[N : 이전 찾기

[r문자 : 커서 위치의 글자를 입력한 문자로 변경

Basic Linux

[r] + **[enter]** : 행분리

[J] : 아래 행을 현재 행 오른쪽으로 끌어 올림

[Z] + **[Z]** : 저장 후 종료

명령모드에서 **[:]**를 누르면 실행모드로 갈 수 있다.

3) 입력 모드

[insert] 또는 **[i]** : 현재 커서 앞에서 입력

[I] : 현재 행의 맨 앞에서 입력

[a] : 현재 커서의 뒤쪽에서 입력

[A] : 현재 행의 맨 마지막에서 입력

[o] : 아래 행에 입력

[O] : 위 행에 입력

[c] + **[w]** : 커서가 놓인 단어의 우측 부분을 삭제 후 입력

[c] + **[c]** 또는 **[S]** : 커서가 놓인 행을 삭제 후 입력

[C] : 커서가 놓인 행의 우측 부분을 삭제 후 입력

[s] : 커서가 놓인 문자 하나를 삭제 후 입력

입력 모드에서 **[esc]**를 누르면 명령모드로 빠져나갈 수 있다.

4) 실행 모드

명령모드에서 **[:]**를 눌러서 진입 할 수 있다.

: set nu : 행 번호 표시

: set nonu : 행 번호 표시 제거

: 숫자,숫자d : 입력한 숫자들 사이의 행 제거 (ex> 3,6d : 3~6 삭제)

: 숫자,숫자 co 숫자 : 입력한 숫자들 사이의 행을 복사하여 숫자 아래에 삽입 (ex> 1,3 co 5
1~3열을 복사하여 5열 아래 6열부터 삽입)

: 숫자,숫자 m 숫자 : 입력한 숫자들 사이의 행을 복사하여 숫자 아래로 이동 (ex> 1,3 co 5
1~3열을 현재 5열 아래로 이동)

:g/OLD/s//NEW/gc : 'OLD'를 찾아서 확인하고 'NEW'로 대체

:s/OLD/NEW/ : 현재 행의 OLD를 NEW로 대체

:3,5s/OLD/NEW/ : 3행에서 5행 사이의 OLD를 NEW로 대체

:2,,s/OLD/NEW/ : 2행부터 현재행 사이의 OLD를 NEW로 대체

Basic Linux

.:,\$s/OLD/NEW/ : 커서의 위치로부터 파일의 끝까지 있는 OLD를 NEW로 대체

:%s/OLD/NEW/g : 파일 전체에서 OLD를 NEW로 전부 대체

:r 파일명 : 커서 다음에 파일 삽입

:숫자 r 파일명 : 파일을 숫자에 해당하는 행 다음에 삽입

:e 파일명 : 파일을 불러와서 수정

:w : 변경사항 저장

:w 파일명 : 다른 이름으로 저장

:wq : 변경사항 저장 후 vi 종료

:q : (최종 변경사항이 저장된 후에) vi 종료

:q! : 변경사항 저장하지 않고 종료

:e! : 수정한 것을 무시하고 다시 편집 상태로

⑤ 파일 보기

1) cat (catenate)

주로 파일을 보는데 사용한다.

용법 : \$ cat (옵션) (> 혹은 >>) [파일명]

cat > (파일명) 으로 입력시에는 간단한 새 파일 역시 제작이 가능하다. (이 때 종료는 `ctrl` + `c`)

```
[pkhinkorea@astro10 pkh]$ cat star.txt
cat: star.txt: No such file or directory
[pkhinkorea@astro10 pkh]$ cat > star.txt
show me the money
```

black sheep wall

```
[pkhinkorea@astro10 pkh]$ cat star.txt
show me the money
```

black sheep wall

옵션으로는

-n : 행 번호를 출력한다

-b : 빈 행에는 번호를 출력하지 않는다

-e : 각 행 끝에 \$를 단다

Basic Linux

-s : 연속 되는 빈 행을 하나로 출력한다.

```
[pkhinkorea@astro10 pkh]$ cat -n star.txt ; star.txt 파일을 행번호와 함께 출력하라.
```

```
1 show me the money
2
3
4 black sheep wall
```

```
[pkhinkorea@astro10 pkh]$ cat -nbs star.txt ; star.txt.파일을 행번호와 함께 출력하되, 빈행의 번호는 빼고, 연속되는 빈행은 하나로 출력하라. 그리고 각 행의 끝에는 $를 달아라.
```

```
1 show me the money$
$
2 black sheep wall$
```

2) more

more 역시 텍스트 파일을 출력해주는 명령어이다.

용법 : \$ more (옵션) [파일명]

옵션은 아래와 같다.

-c : 내용을 보여주기 전에 화면을 청소
-lines : 한 번에 보여줄 줄 수를 정함
-d : 스페이스나 q키를 누르라는 프롬프트를 출력
[+숫자] : 지정한 행부터 보여줌

```
[pkhinkorea@astro10 pkh]$ cat terran.txt >> star.txt
[pkhinkorea@astro10 pkh]$ cat zerg.txt >> star.txt
[pkhinkorea@astro10 pkh]$ cat protoss.txt >> star.txt
[pkhinkorea@astro10 pkh]$ more -5 star.txt ; 한 번에 다섯 줄 씩 보이게 출력하라.
show me the money
```

```
black sheep wall
marine
--More--(31%)
```

```
[pkhinkorea@astro10 pkh]$ more +5 star.txt ; 5번째 행부터 출력하라.
```

3) head와 tail

head는 파일의 시작부분을, tail은 끝부분을 보여준다.

용법 : \$ head/tail (옵션) [파일명]

옵션으로는

Basic Linux

-숫자 : 처음(끝)부터 보여줄 줄 수를 지정한다.

```
[pkhinkorea@astro10 pkh]$ head -3 star.txt ; star.txt 파일의 시작 세 줄을 출력한다.  
show me the money
```

```
[pkhinkorea@astro10 pkh]$ tail -3 star.txt ; star.txt 파일의 끝 세 줄을 출력한다.  
zergling  
marine  
scv
```

Tip 2. 파이프라인(pipeline) | (shift+\)는 프로그램간 결과를 전송하는 명령어로, 두 명령어를 복합해서 명령을 내릴 수 있다. 앞의 명령어에서 나오는 출력을, 뒤의 명령어로 받아서 실행한다. 가령, `cat -n star.txt | tail -2` 이라고 입력하면, 앞의 `cat`명령어대로 줄번호를 적은 파일을 출력...할 것을 다시 입력으로 받아 `tail`의 명령을 수행하여, 뒤에서부터 2번째 행 까지 나타내게 된다,

```
[pkhinkorea@astro10 pkh]$ cat -n star.txt | tail -2  
19 marine  
20 scv
```

또한 리다이렉션(redirection) > 는 파일간의 결과 전송을 의미한다. 프로그램을 돌려 나온 결과를 화면에 출력하는 것이 아니라, 리다이렉션 다음에 나오는 파일명에 저장을 해달라는 것을 의미한다. 어펜드(append)는 리다이렉션의 파일 뒤에 붙이기 기능이 추가 된 것이다. 리다이렉션은 기존 같은 이름의 파일이 있을 경우, 실행이 되지 않거나, 혹은 지우고 새 파일을 만들지만, 어펜드는 기존 파일 뒤편에 합병해준다,

```
[pkhinkorea@astro10 pkh]$ cat terran.txt ; terran.txt를 출력하라  
marine  
scv  
[pkhinkorea@astro10 pkh]$ cat protoss.txt ; protoss.txt를 출력하라.  
zealot  
probe  
[pkhinkorea@astro10 pkh]$ cat zerg.txt ; zerg.txt를 출력하라.  
drone  
zergling  
[pkhinkorea@astro10 pkh]$ cat zerg.txt > protoss.txt ; zerg.txt를 protoss.txt에 덮어 씌워라  
[pkhinkorea@astro10 pkh]$ cat protoss.txt ; protoss.txt를 출력하라.  
drone ; 기존 내용이 없어지고, zerg.txt내용과 같  
zergling 은 것을 볼 수 있다.  
[pkhinkorea@astro10 pkh]$ cat terran.txt >> protoss.txt ; terran.txt 파일을 protoss.txt파일에 합병하라.  
[pkhinkorea@astro10 pkh]$ cat protoss.txt ; protoss.txt 파일을 출력하라  
drone ; 기존의 protoss.txt 파일에, terran.txt  
zergling ; 파일이 합병된 것을 확인 할 수 있다.  
marine  
scv  
[pkhinkorea@astro10 pkh]$ ls *txt ; txt로 끝나는 파일들을 출력하라
```

```
protoss.txt star.txt terran.txt zerg.txt
[pkhinkorea@astro10 pkh]$ ls *txt > starcraft.txt

[pkhinkorea@astro10 pkh]$ cat starcraft.txt
protoss.txt
star.txt
terran.txt
zerg.txt
```

; txt로 끝나는 파일을 출력하는 대신, starcraft.txt라는 파일을 만들고 그 안에 저장하라.

III. 검색하기

1) which

명령어를 검색하는 프로그램이다.

용법 : \$ which [명령어]

```
[pkhinkorea@astro10 pkh]$ which xgterm
/usr/local/bin/xgterm
```

; xgterm 실행 파일이 어딘는지 찾아라.

2) grep(global regular expression print)

파일 내용을 뒤져, 해당하는 단어나 문자를 찾는다.

용법 : \$ grep (옵션) [단어] [파일명]

옵션은 아래와 같다.

- c : 파일이 아닌, 총 숫자를 출력
- C 숫자 : 일치하는 행에서 위와 아래로 지정한 숫자만큼의 행을 추가로 더 보여준다.
- A 숫자 : 일치하는 행에서 아래로 지정한 숫자만큼의 행을 추가로 더 보여준다.
- i : 대소문자를 구분하지 않는다.
- n : 파일 내 행 번호를 함께 출력한다.
- v : 지정한 문구와 일치하지 않는 것들을 보여준다.
- r : 하위 디렉토리까지 검색
- b : 블록 번호 표시

```
[pkhinkorea@astro10 pkh]$ grep scv *
protoss.txt:scv
star.txt:scv
star.txt:scv
...
```

```
[pkhinkorea@astro10 pkh]$ grep -nC 1 scv star.txt
5:marine
6:scv
7-drone
--
9:marine
10:scv
11-drone
...
```

3) find

find는 크기나, 수정된 날짜 등 지정한 조건에 맞는 파일들을 찾아 준다.

용법 : \$ find (경로) [조건]

조건은 다음과 같다.

- name [파일명] : 이름으로 검색한다.
- user [아이디] : 소유자로 검색한다.
- amin -숫자 : 숫자 분 이내에 접근한 파일을 검색한다.
- atime -숫자 : 숫자 일 이내에 접근한 파일을 검색한다.
- ctime -숫자 : 숫자 일 이내에 상태가 변경된 파일을 검색한다.
- mtime -숫자 : 숫자 일 이내에 수정한 파일을 검색한다.
- perm [조건] : 권한이 조건에 맞는 파일을 찾는다.
- size [조건] : 파일 크기가 일치하는 것을 찾는다. 이 때 숫자 뒤에 +를 붙이면 해당 크기 이상을, -를 붙이면 해당크기 이하를 의미한다. 또한, 숫자뒤에 k를 붙이면 kb, m을 붙이면 메가 바이트를 의미한다.
- newer [파일명] : 해당파일보다 이후에 갱신된 모든 파일을 검색한다.
- exec [명령] {} \; : 원하는 검색 조건에 맞는 파일들을, 주어진 명령으로 실행한다.

```
[pkhinkorea@astro10 ~]$ find -name *fits -size -5M
./alpTrES-2-698_I.fits
[pkhinkorea@astro10 ~]$ find /home/pkhinkorea -amin -50 -name *txt
/home/pkhinkorea/pkh/protoss.txt
[pkhinkorea@astro10 ~]$ find /home/pkhinkorea -atime -5 -name *txt
/home/pkhinkorea/pkh/zerg.txt
/home/pkhinkorea/pkh/terran.txt
/home/pkhinkorea/pkh/star.txt
/home/pkhinkorea/pkh/protoss.txt
/home/pkhinkorea/pkh/starcraft.txt
[pkhinkorea@astro10 ~]$ find /home/pkhinkorea -atime -5 -name *txt -exec ls -la {} \;
-rw-r--r-- 1 pkhinkorea master 15 Mar  7 20:57 /home/pkhinkorea/pkh/zerg.txt
-rw-r--r-- 1 pkhinkorea master 63 Mar  8 21:49 /home/pkhinkorea/pkh/terran.txt
-rw-r--r-- 1 pkhinkorea master 141 Mar  8 21:50 /home/pkhinkorea/pkh/star.txt
-rw-r--r-- 1 pkhinkorea master 26 Mar  7 20:58 /home/pkhinkorea/pkh/protoss.txt
-rw-r--r-- 1 pkhinkorea master 41 Mar  8 21:44 /home/pkhinkorea/pkh/starcraft.txt
```

IV. 압축

Linux에서 압축을 담당하는 명령어는 tar와 gzip(GNU zip)이다. tar는 파일을 하나로 모으되 압축은 하지 않고, gzip은 파일을 모을 수는 없지만 압축을 담당한다. 따라서 여러 파일을 압축할 때는 tar로 파일을 묶고 gzip으로 tar파일을 압축하게 된다. tar의 옵션에서 gzip 실행 여부를 지정할 수 있기 때문에, gzip이 단독으로 쓰일 경우는 많지 않다.

용법 : \$ gzip [파일명] ; 압축 할 때 \$ gunzip [파일명] ; 압축 풀 때

용법 : \$ tar -c[옵션] [압축파일명] [압축할파일 혹은 디렉토리] ; 압축 할 때 \$ tar -x[옵션] [압축파일명] [압축할파일 혹은 디렉토리] ; 압축 풀 때

tar의 옵션은 다음과 같다

- c: 묶음 파일 생성
- x: 묶음 파일 해제
- v: 작업내용 출력
- p: 권한을 그대로 유지
- z: gzip형식으로 압축 혹은 압축해제
- f: 사용할 파일의 tar지정

```
[pkhinkorea@astro10 pkh]$ ls
protoss.txt  star.txt  starcraft.txt  terran.txt  zerg.txt
[pkhinkorea@astro10 pkh]$ tar -czvf blizzard.tar.gz *.txt
protoss.txt
star.txt
starcraft.txt
terran.txt
zerg.txt
[pkhinkorea@astro10 pkh]$ ls
blizzard.tar.gz  protoss.txt  star.txt  starcraft.txt  terran.txt  zerg.txt
[pkhinkorea@astro10 pkh]$ rm *.txt
[pkhinkorea@astro10 pkh]$ ls
blizzard.tar.gz
[pkhinkorea@astro10 pkh]$ tar -xzvf blizzard.tar.gz
protoss.txt
star.txt
starcraft.txt
terran.txt
zerg.txt
[pkhinkorea@astro10 pkh]$ ls
blizzard.tar.gz  protoss.txt  star.txt  starcraft.txt  terran.txt  zerg.txt
```


V. 권한

1) 권한

권한은 해당 파일에 대한 읽기(r, 4) · 쓰기(w, 2) · 실행(x, 1) 여부를 결정한다. 디렉토리나 파일의 권한 설정 여부는 **ls -l**을 입력할 때 제일 앞에 적히는 10자리의 기호로 알 수 있다.

10자리의 기호는

[디렉토리(d),파일 구분자(-) 1자리]+[소유자권한 3자리]+[그룹권한 3자리]+[전체권한 3자리]로 구성되어 있다.

가령 -rwxrwxr-x라고 표기된 것은, 해당 파일(-)은 소유자는 읽기 · 쓰기 · 실행이 전부 가능(rwx)하며, 마찬가지로 그룹 역시 전부 가능(rwx)하다. 하지만 그 외의 사람들은 읽고 실행은 가능하지만, 수정은 불가능(r-x)하다는 것을 의미한다.

drw-r-----라고 표기된 것은, 해당 디렉토리(d)는 소유자는 읽기 · 쓰기가 가능하지만(rw-) 그룹은 읽기만 가능하며 (r--), 그 외의 사람들은 접근조차 허용하지 않는다는 의미(---)다.

tip 3. 폴더의 경우 만들 때 -m 옵션값을 붙여주면 시작부터 권한을 설정해 줄 수 있다.
가령, 소유주는 읽고(4) 쓰고(+2) 실행(+1)이 가능하고, 그룹은 읽고(4), 실행(+1)만 가능하며, 그 외의 사람은 읽기(4)만 가능한 test라는 폴더를 만들고 싶을 경우 아래와 같이 입력한다.

```
$mkdir -m 754 test
```

2) chmod(change mode)

파일 및 디렉토리의 권한을 설정하는 명령어이다.

용법 : **\$ chmod [권한값] [파일명]**

```
[pkhinkorea@astro10 pkh]$ ls
protoss.txt terran.txt zerg.txt
[pkhinkorea@astro10 pkh]$ ll
total 12
-rw-r--r-- 1 pkhinkorea master 26 Mar  7 20:58 protoss.txt
-rw-r--r-- 1 pkhinkorea master 63 Mar  8 21:49 terran.txt
-rw-r--r-- 1 pkhinkorea master 15 Mar  7 20:57 zerg.txt
```

현재, protoss.txt, terran.txt, zerg.txt 라는 세 개의 파일이 있고, 이 파일들의 권한은 소유주는 읽기 · 쓰기, 그룹 및 그 외의 사람들은 읽기 전용으로 되어 있다. protoss.txt 파일에 대해

Basic Linux

소유주의 쓰기 권한을 제해, 읽기(4)만 가능하게 만들어보자.

```
[pkhinkorea@astro10 pkh]$ chmod 444 protoss.txt
```

이후 vi로 protoss.txt 파일을 수정하려고 하면, read only라고 적혀 있고 수정이 되지 않는 것을 알 수 있다. 모든 파일을 모든 사용자에게 대해 읽기(4) 쓰기(2)를 자유롭게 열어보자.

```
[pkhinkorea@astro10 pkh]$ chmod 666 *.txt
```

3) chown(change owner)

디렉토리나 파일의 권한을 다른 사용자나 그룹에 넘겨주는 명령어이다.

용법 : `$ chown (옵션) [소유자(:그룹)] [파일 혹은 디렉토리명]`

옵션값은 아래와 같다.

-R : 모든 하위 디렉토리 및 파일의 권한을 함께 변경한다.

-f : 권한 변경 실패 시, 에러 메시지를 출력하지 않는다.

-v : 파일에 대해 변경한 정보나, 변경되지 않은 정보를 출력한다.

iraf설치 시, /iraf/iraf/와 그 하위 폴더를 iraf 그룹의, iraf유저에게 권한을 넘겨주는 데, 그 때 명령은 아래와 같다.

```
# chown -R iraf:iraf /iraf
```

VI. 링크

링크에는 ln(link) 명령어가 사용된다.

1) 하드링크

하드링크는 동일한 데이터 블록을 참조하는 서로 다른 파일을 만드는 것이다. (디렉토리는 하드링크가 불가능) 원본의 이동이나 삭제와 상관없이 링크된 파일은 그대로 남아 있다.

용법 : `$ ln [원본 파일] [하드링크 파일]`

```
[pkhinkorea@astro10 pkh]$ ls
```

```
protoss.txt  terran.txt  zerg.txt
```

```
[pkhinkorea@astro10 pkh]$ mkdir linktest
```

```
[pkhinkorea@astro10 pkh]$ cd linktest
```

```
[pkhinkorea@astro10 linktest]$ ln ../protoss.txt hardlink.txt ; 한 폴더 위의 protoss.txt와 이 폴더의 hardlink.txt사이 하드링크를 생성
```

Basic Linux

```
[pkhinkorea@astro10 linktest]$ ls
hardlink.txt
[pkhinkorea@astro10 linktest]$ cat ../protoss.txt
drone
zergling
marine
scv
[pkhinkorea@astro10 linktest]$ cat hardlink.txt          ; 두 파일 사이의 내용은 동일
drone
zergling
marine
scv
```

cp를 통해 만든 파일과의 차이점으로는, 하드링크로 생성된 파일은 하드 공간을 차지하지 않으며, 원본 수정 시 하드링크도 역시 수정된다.

2) 심볼릭 링크

심볼릭 링크는, 윈도우의 바로가기 아이콘과 비슷한 역할을 수행한다. 하드링크와 달리, 디렉토리에 적용이 가능하기 때문에 긴 디렉토리로 이동할 때 유용하며, 원본 파일의 이동이나 삭제에 영향을 받는다.

용법 : `$ ln -s [원본 파일이나 디렉토리] [링크걸 파일이나 디렉토리]`

```
[pkhinkorea@astro10 pkh]$ mkdir 1
[pkhinkorea@astro10 pkh]$ mkdir 1/2
[pkhinkorea@astro10 pkh]$ mkdir 1/2/3
[pkhinkorea@astro10 pkh]$ mkdir 1/2/3/4          ; 1/2/3/4 까지 내려가는 디렉토리를 생성
[pkhinkorea@astro10 pkh]$ cp *txt 1/2/3/4
[pkhinkorea@astro10 pkh]$ ln -s 1/2/3/4 0          ; 1/2/3/4 디렉토리의 심볼릭 링크인 0이란 디렉토리 생성
[pkhinkorea@astro10 pkh]$ ls
0 1 protoss.txt terran.txt zerg.txt
[pkhinkorea@astro10 pkh]$ ls 0                    ; 0의 내용을 확인하면 1/2/3/4와 같은 내용이 있다.
protoss.txt terran.txt zerg.txt
```

VII. 기타

1) 용량

파일 및 디렉토리의 사용량을 확인할 때는 du(disk usage)라는 명령어를 사용한다.

용법 : \$ du (옵션) (파일이나 디렉토리)

```
[pkhinkorea@astro10 pkh]$ du ; 현재 위치 아래의 용량을 출력하라
16  ./1/2/3/4
20  ./1/2/3
24  ./1/2
28  ./1
60  .
[pkhinkorea@astro10 pkh]$ du *txt ; 현재 위치에 있는 txt파일들의 용량을 출력하라
4   protoss.txt
4   terran.txt
4   zerg.txt
```

옵션으로는 다음과 같다.

-h : 단위를 표시한다.

-s : 간단히 합산된 용량만 보여준다

```
[pkhinkorea@astro10 pkh]$ du -sh ~pkhinkorea
236M  /home/pkhinkorea
```

df(disk free)는 사용 중인 파일 시스템의 전체 용량, 사용한 용량, 사용 가능한 용량, 사용률 등을 보여주는 명령이다.

용법 : \$ df (옵션)

```
[pkhinkorea@astro10 pkh]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        77808904 14037512 59755144 20% /
tmpfs             515508      0    515508  0% /dev/shm
/dev/sda3        154791012 135511940 11289148 93% /home
/dev/sdb1        131033152 10302720 114074304 9% /packages
/dev/sdb2        109275268 108634052      0 100% /backup
```

옵션은 아래와 같다

-a : 크기가 0인 파일 시스템까지 나타낸다.

-i : 용량을 block 단위가 아닌, inode 단위로 나타낸다.

-T : 파일 시스템의 종류와 함께 표시한다.

-t [파일시스템] : 뒤에 적은 파일 시스템만 나타낸다.

-x [파일시스템] : 뒤에 적은 파일 시스템을 제외하고 나타낸다.

-h : 단위를 표시한다.

2) clear, history

현재 터미널의 창을 지울 때는 clear라는 명령어를, 기존 자신이 입력했던 명령어를 보고자 할 때는 history라는 명령어를 입력하면 된다.

용법 : `$ history (숫자) ; 숫자만큼의 명령어 내역을 출력`

3) ps(process status), kill

ps는 작동중인 프로세스를 나타낸다.

용법 : `$ ps (옵션)`

옵션값은 아래와 같다.

e: 모든 프로세스를 표시

f: 전체 경로로 프로세스를 표시

l: 긴 포맷으로 출력

u: 실행한 유저 및 실행 시간을 표시

a: 다른 유저의 프로세스도 표시

x: 터미널 제어 없이 프로세스 현황 보기

kill은 실행 중인 특정 프로세스를 종료하는데 쓰인다.

용법 : `$ kill -9 [PID 번호]`

`$ kill -l`을 입력하면 kill 명령어의 기타 옵션들을 볼 수 있는데, 이 중 9번째 옵션이 프로세스를 종료하는 옵션이다. 따라서 ps로 확인한 kill -9 이후 ps로 확인한 PID 번호를 입력하면 해당 프로세스가 강제 종료된다.

```
[khpark@beatrice ~]$ ps
  PID TTY          TIME CMD
 11024 pts/20    00:00:00 bash
  11123 pts/20    00:00:00 xgterm
  11144 pts/20    00:00:00 ps
[khpark@beatrice ~]$ kill -9 11123
```

Appendix. A. 리눅스 서버 접속 및 파일 전송 방법

1) 리눅스-> 리눅스 (ex 천문 전산 실습실에서 astro10에 접속 시도)

① 접속하기

```
$ ssh [아이디]@[서버]
```

접속을 종료할 때는 exit라고 입력하면 된다.

② 파일전송하기

```
$ sftp [아이디]@[서버]
```

이 때 ls나 cd와 같은 명령어는 서버의 컴퓨터에 적용되며, 내 컴퓨터에서 ls나 cd 등의 명령을 내리고 싶을 때는 앞에 l자를 하나 더 붙이면 된다. (lls, lcd, lcp, lrm, ...)

파일을 받을 때는 get (파일이름), 파일을 전송할 때는 put (파일이름)
접속을 종료할 때는 exit라고 입력하면 된다.

2) 윈도우-> 리눅스 (ex 개인 컴퓨터에서 astro10에 접속 시도)

① 프로그램 설치하기

서울대학교 정보화본부 홈페이지(it4u.snu.ac.kr) 이동 후, 캠퍼스 라이선스 S/W 다운로드 항목으로 들어간다. 11번 게시물의 Xmanager Enterprise 3.0을 설치한다.

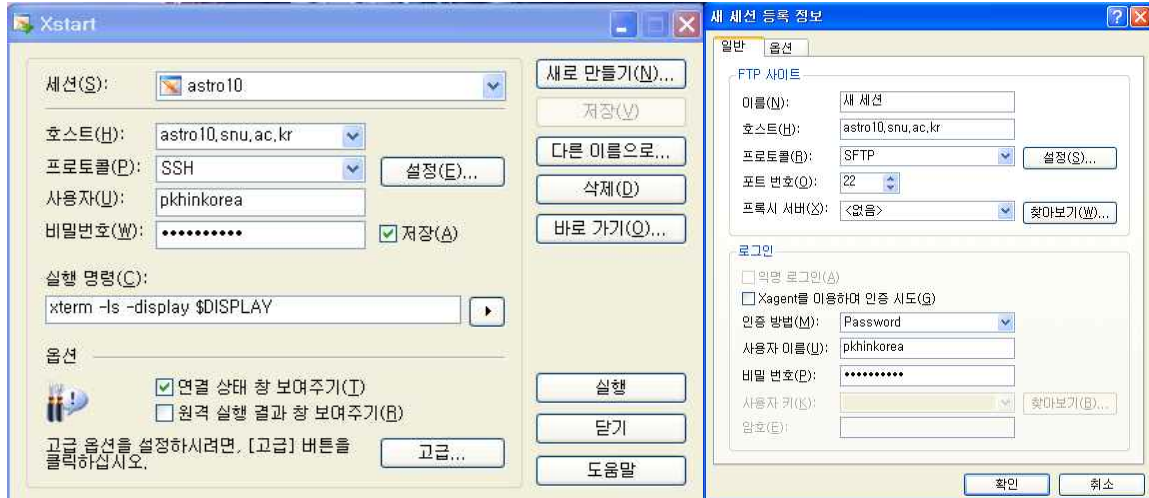
이 사이트는 교내에서만 이용이 가능한 것을 염두해두자.

② 접속하기

설치한 프로그램 중 Xstart를 시작한다. 우측에 새로만들기 클릭 후, 세션에 적당한 이름을 입력한다. 호스트는 접속하고자 하는 서버, 프로토콜은 SSH, 그리고 사용자 아이디와 비밀번호를 입력한다.

astro10의 경우에는 실행명령 창에 옆에 우측 화살표를 클릭 후, 리눅스를 선택한 다음 [그림 A.1]과 같이 xterm 뒷부분만 남기고 앞부분은 삭제한다.

실행을 누르면 서버에 접속할 수 있다.



[그림 A. 1] Xstart 설정

[그림 A. 2] Xftp 설정

③ 파일 전송하기

설치한 프로그램 중 Xftp를 시작한다. [그림 A.2]와 같이 세션에 적당한 이름을 적고 호스트에는 접속하고자 하는 서버를 적는다. 프로토콜은 SFTP, 인증방법은 Password 방식에, 본인의 아이디와 비밀번호를 입력한다.

3) VNC 설정하기

VNC는 처음 실행해 주기 전까지 과정이 복잡하지만, 실행 후에는 ‘원격 데스크탑 연결’과 유사하게 그래픽 형태로 서버의 화면을 띄워주기 때문에 유용하다.

① 서버 설정하기

여기서는 서버에 VNC가 설치되어, 기본적인 세팅이 완료되어 있는 상태에, 새로운 사용자를 추가하는 방법에 대해서 설명한다. 개인 컴퓨터를 서버로 사용하기 위해 처음부터 VNC를 설치하길 원하면 인터넷에 검색해서 찾아보자.

우선, 서버상의 VNC에, VNC를 사용할 사용자 계정과 포트를 등록하자. su 명령어로 관리자로 접속한 다음,

```
# vi /etc/sysconfig/vncserver
```

이후, 파일 아래에 포트와 아이디를 추가해준다. 이 때 포트는 기존 사용자들과 겹치지 않는 번호를 입력한다. 가령, 3번 포트에 khpark을 추가해주고 싶다면

```
VNCSERVERS="3:khpark"
VNCSERVERARGS[3]="-geometry 800x600"
```

Basic Linux

두 줄을 추가하자.

그 다음, 다른 터미널에서, 자기 계정 아래 .vnc 폴더를 만들고, 비밀번호를 설정하자

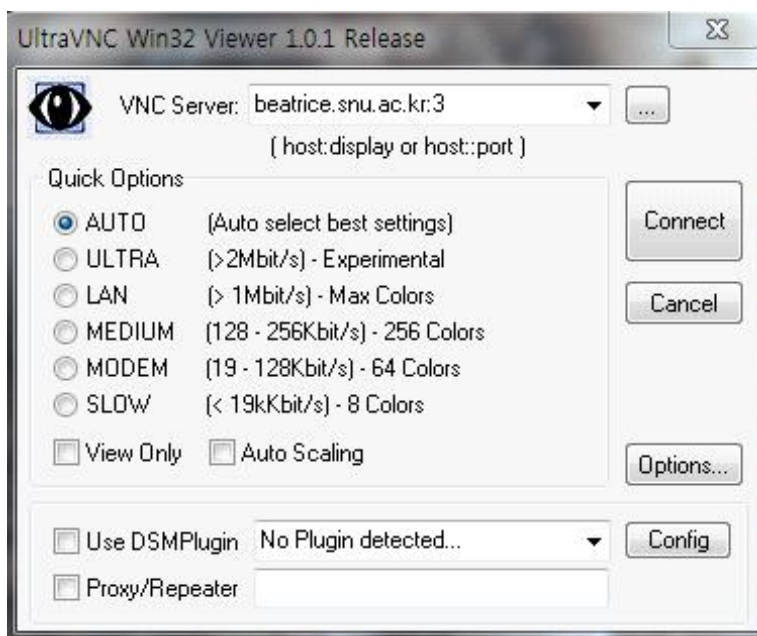
```
$ cd ~  
$ mkdir .vnc  
$ cd .vnc  
$ vncpasswd
```

마지막으로, 방화벽을 풀어주자. 관리자 권한으로

```
# iptables -I INPUT 6 -p tcp -m state --state NEW -m tcp --dport 5903 -j ACCEPT  
# service network restart  
# /etc/init.d/vncserver start
```

을 입력해주자. 이 때, 포트번호는 5900+(자기가 앞서 입력한 포트번호)이다.

자, 이제 준비는 끝났다. 인터넷에서 VNC Viewer 파일을 구한 후, 실행하자. 접속하기에서, (서버명 혹은 IP) : (포트번호)를 입력해서 실행해주자.



[그림 A.3] VNC Viewer 실행창

Appendix. B. 관리자 권한 활용하기

개인 컴퓨터나, 서버 관리자를 맡게 되면 프로그램 설치 등의 사유로 관리자 권한을 활용할 수 있다. 관리자 권한을 사용하는 방법으로는 su(substitute)와, sudo 두 가지가 있다.

\$ su (아이디) ; 해당 아이디로 계정을 전환. 입력하지 않으면 관리자 계정으로 전환한다.

sudo는 계정을 바꾸지 않은 상태에서 관리자 권한으로 명령을 실행하는 것이다. 가령 일반 계정으로 cat /etc/sudoers 라는 명령을 내리면 권한 문제로 기각된다. 하지만 관리자 비밀번호를 알고 있다면 sudo cat /etc/sudoers라는 명령으로 해당 파일을 열람할 수 있다. 단, 이 명령을 수행하기 이전에 /etc/sudoers 파일에 아래와 같은 부분에 계정을 추가해주어야 한다.

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
wskang  ALL=(ALL)        NOPASSWD: ALL
lingo   ALL=(ALL)        ALL
khpark  ALL=(ALL)        NOPASSWD: ALL
```

lingo라는 사용자는 sudo라는 명령을 사용할 수 있지만, 사용할 때마다 비밀번호를 확인하게 되고, wskang과 khpark 사용자는 비밀번호 확인을 생략하고 sudo를 실행 할 수 있다.