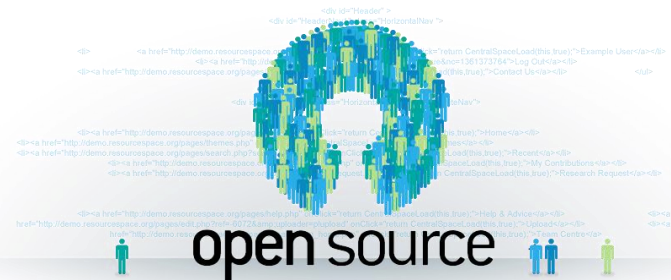




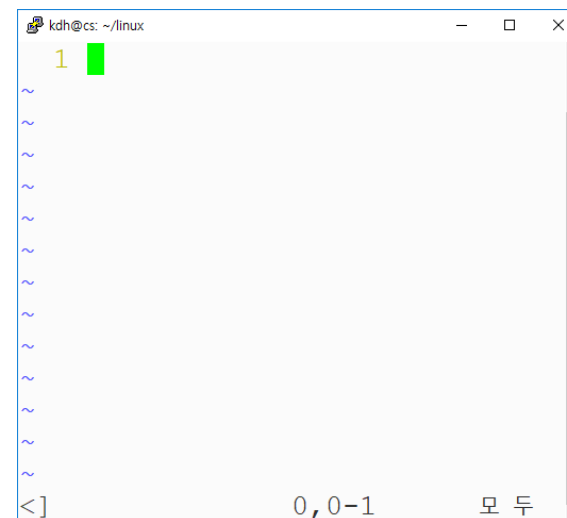
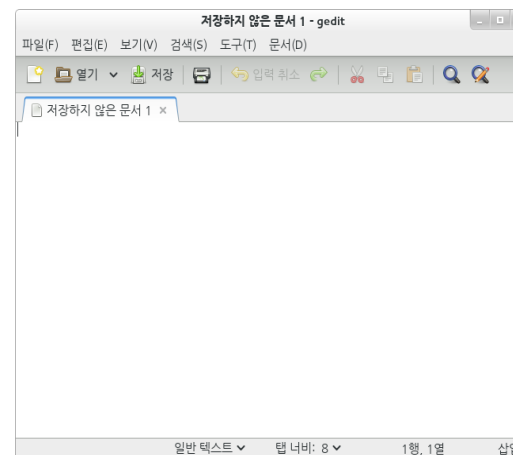
# 오픈소스 리눅스 실무





## 리눅스 문서편집기

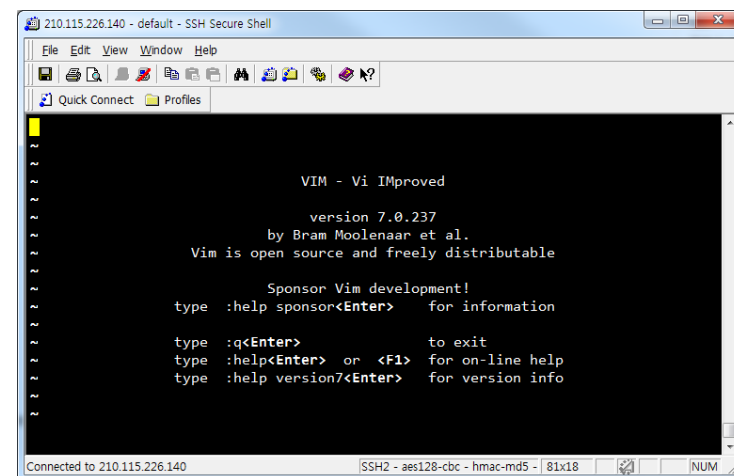
- 리눅스 대표 문서 편집기
- GUI 환경
  - gedit
- 터미널 환경
  - 행 단위 편집기 : ed , ex , sed
  - 화면 단위 편집기
    - vi – 리눅스에서 가장 일반적이면서도 대표적인 문서 편집기 ( 기본 설치 )
    - emacs – 기능은 매우 다양하지만 사용이 어렵고 복잡 ( 별도 설치 필요 )





## VI Editor 개요

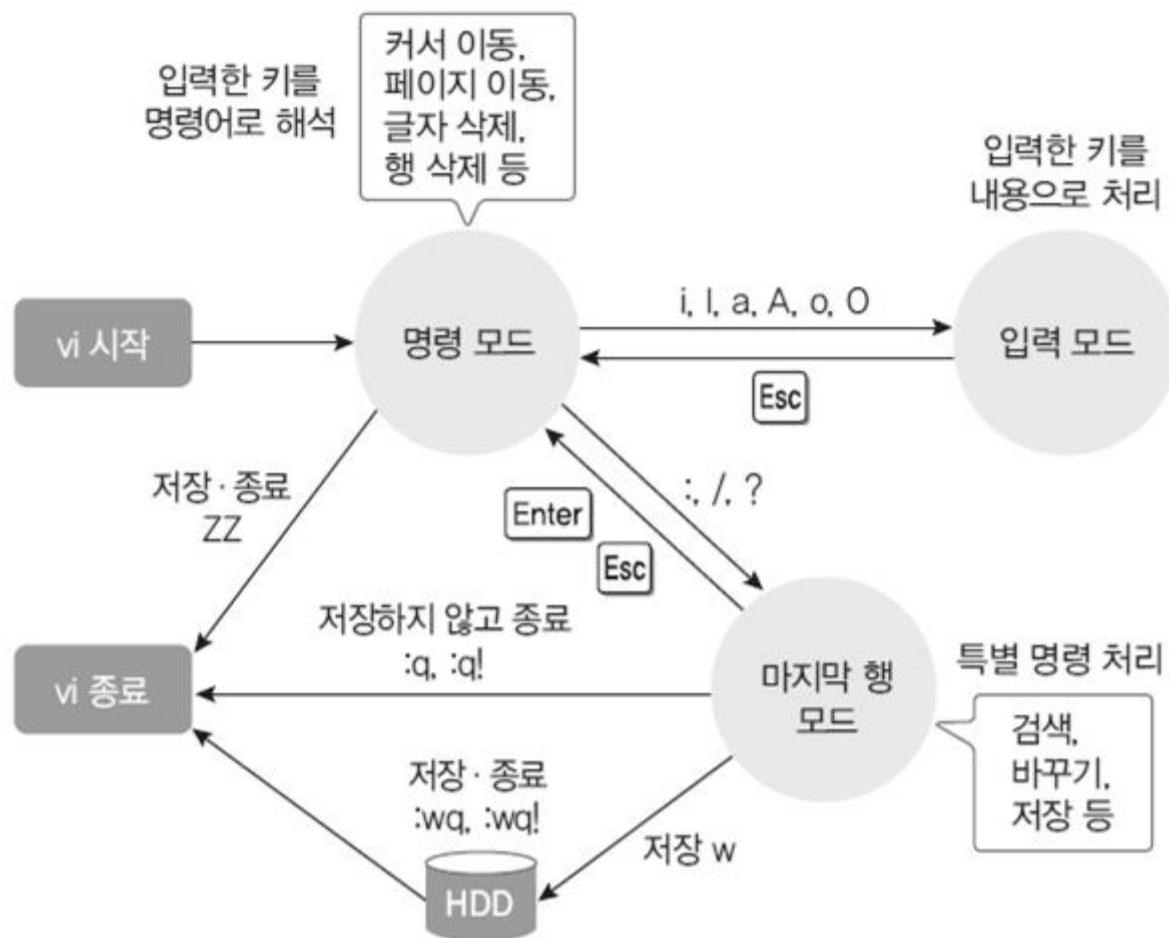
- **Visual edit** 의 약자.
  - 1976년에 BSD의 C shell을 만든 빌조이가 ed의 기능을 확장시킨 ex 편집기를 개발하고 이를 확장해 만듦.
  - 최대장점 - 키보드로 모든 것이 가능.
- Vim ( vi improved )
  - vi 클론중의 하나이다.
  - 여러 개의 파일을 동시에 편집
  - syntax highlighting





## 리눅스 문서편집기

- vi 동작 모드





- vi 시작
  - 지정한 파일을 편집
  - 파일명이 없는 경우 빈 파일 생성 (종료 시 저장)

vi [파일]





Linux

# 리눅스 문서편집기

- vi 시작
  - vi filename
    - 주어진 파일이름으로 편집을 시작한다. 파일이 존재하지 않으면 새로 만들게 된다.
  - vi +15 filename
    - 파일을 열 때 15 번째 줄로 커서를 보낸다.
  - vi -R filename
    - view filename 파일을 읽기 전용으로 읽는다. vi -r filename 깨진 파일을 복구한다.



## vi 환경 셋팅

- 마지막행 모드에서 실행

:set autoindent " 자동으로 들여쓰기를 한다.

:set cindent " C 프로그래밍을 할 때 자동으로 들여쓰기를 한다.

:set smartindent " 좀더 똑똑한 들여쓰기를 위한 옵션이다.

:set textwidth=79 " 만약 79번째 글자를 넘어가면 ₩

:set wrap " 자동으로 <CR>를 삽입하여 다음 줄로 넘어간다.

:set nowrapscan " 검색할 때 문서의 끝에서 다시 처음으로 돌아가지 않는다.

:set nobackup " 백업 파일을 만들지 않는다.

:set visualbell " 키를 잘못 눌렀을 때 뽁 소리를 내는 대신 번쩍이게 한다.

:set ruler " 화면 우측 하단에 현재 커서의 위치(줄,칸)를 보여준다.

:set tabstop=4 " Tab을 눌렀을 때 8칸 대신 4칸 이동하도록 한다.

:set shiftwidth=4 " 자동 들여쓰기를 할 때 4칸 들여 쓰도록 한다.

:set list "행의 끝이나 탭 같은 특수 문자를 보이도록 한다.

:set nu : 라인번호를 붙여준다.



## 리눅스 문서편집기

- vi 저장과 종료
  - 명령 모드로 종료
    - ZZ
  - 마지막행 모드에서 저장과 종료
    - :q - 파일이 저장되지 않았으면 종료 불가
    - :q! - 변경된 내용을 저장하지 않고 종료
    - :w 파일명 - 파일명으로 저장
    - :wq 또는 :wq! - 저장 후 종료





## 리눅스 문서편집기

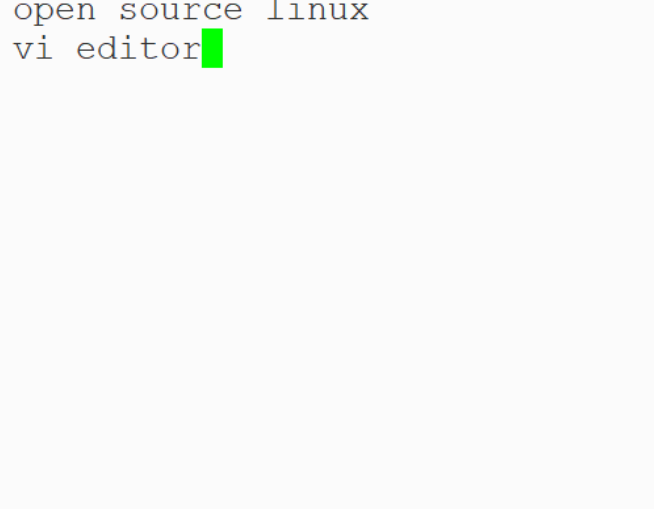
- vi 입력 모드
  - 명령모드에서 편집 명령키를 누르면 입력 모드로 변경
  - ESC 키를 누르면 다시 명령모드로 변경

| 명령 키     | 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.         |



- 문서 편집 예

- 문서 편집 예



```
kdh@cs: ~/linux
1 open source linux
2 vi editor
3 gcc -o test test.c
4 ./test
5 gcc -o test test.c -lm
6 ./test
7 gcc -o test test.c -lm -D_FORTIFY_SOURCE=0
8 ./test
9 gcc -o test test.c -lm -D_FORTIFY_SOURCE=0 -fcommon
10 ./test
```



## 리눅스 문서편집기

- i , a , o 명령키의 차이
  - i 명령키는 현재 커서 위치에서 문자 삽입
  - a 명령키는 현재 커서 위치 다음에 문자 추가
  - o 명령키는 현재 커서 다음 라인에 문자 삽입

```
kdh@cs: ~/linux
1 open source linux
2 vi editor
~
~
1, 10 모 두
```



## 리눅스 문서편집기

- I, A, O 명령키의 차이
  - I 명령키는 현재 라인 제일 앞에 문자 삽입
  - A 명령키는 현재 라인 제일 뒤에 문자 추가
  - O 명령키는 현재 라인 이전 라인에 문자 삽입

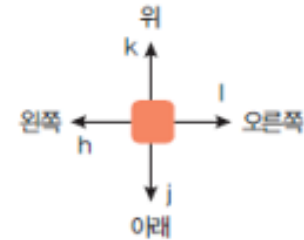
```
kdh@cs: ~/linux
1 open source linux
2 vi editor
~
~
1, 10 모 두
```



Linux

# 리눅스 문서편집기

- 커서의 이동
  - k 명령키는 커서를 한 행 위로 이동
  - J 명령키는 커서를 한 행 아래로 이동
  - l 명령키는 커서를 한 글자 앞으로 이동
  - h 명령키는 커서를 한 글자 뒤로 이동
  - 리눅스는 키보드의 방향키로도 커서 이동 가능
- 응용
  - 3k - 3줄 위로
  - 10h - 10글자 뒤로





## 리눅스 문서편집기

- 커서의 이동
  - `^`, `0` : 커서를 현재 행의 처음으로 이동
  - `$` : 커서를 현재행의 마지막으로 이동
  - `-` : 커서를 앞 행의 처음으로 이동
  - `+` : 커서를 다음행의 처음으로 이동
  - `H` : 커서를 화면의 맨 위행으로 이동
  - `M` : 커서를 화면의 중간 행으로 이동
  - `L` : 커서를 화면의 맨 아래 행으로 이동
  - `w` : 커서를 다음 단어의 첫 글자로 이동
  - `b` : 커서를 이전 단어의 첫 글자로 이동
  - `E` : 커서를 다음 단어의 끝 글자로 이동



## 리눅스 문서편집기

- 화면의 이동
  - 파일 내용이 터미널의 화면 크기보다 클 경우 화면을 이동시키기 위한 명령
  - ^u : 반 화면 위로 이동
  - ^d : 반 화면 아래로 이동
  - ^b : 한 화면 위로 이동
  - ^f : 한 화면 아래로 이동
  - ^y : 화면을 한 행만 위로 이동
  - ^e : 화면을 한 행만 밑으로 이동

```
#include <stdio.h>
main()
{
    printf("linux 01\n");
    printf("linux 02\n");
    printf("linux 03\n");

    printf("linux 04\n");
    printf("linux 05\n");
    printf("linux 06\n");
    printf("linux 07\n");

    printf("linux 08\n");
}
```



## 리눅스 문서편집기

- 지정한 행으로의 이동
  - 커서를 원하는 행으로 이동하는 명령
  - G : 파일의 마지막 행으로 커서 이동
  - nG : n의 값에 해당하는 행으로 커서 이동 예) 3G
  - :행 번호 : 마지막행 모드에서 지정한 행번호로 이동
  - :\$ : 파일의 마지막 행으로 커서 이동

```
kdh@cs: ~/linux
1 open 1G 또는 :1
2 source
3 linux
4 vi
5 editor G 또는 :$
```

3,6 모 두





## 리눅스 문서편집기

- 마킹을 이용한 이동
  - `ma ( .... mz )` : 현재 위치를 a로 마크
  - ``a` : a 마크 위치로 이동
  - `'a` : 마크된 ``a` 가 있는 줄의 처음으로 이동
  - ```` : 직전 커서 위치로 이동
  - `''` : 직전에 커서가 위치한 줄의 처음으로 이동



## 리눅스 문서편집기

- 명령모드에서 글 수정하기
  - r : 커서가 위치한 글자를 다른 글자로 수정 ( 한 글자 수정 후 명령 모드 유지)
  - cw , #cw : 커서 위치에서 부터 현재 단어 끝까지 수정 ( 명령모드 유지 )
  - s , #s : 커서 위치에서 한 글자 또는 #글자 수 만큼 지우고 ESC를 입력할 때까지 입력
  - R : 커서 위치에서 ESC를 입력할 때 까지 수정
  - cc : 커서가 위치한 행의 내용 모두를 지우고 입력
  - C : 현재 커서 위치부터 행 끝 까지를 지우고 입력



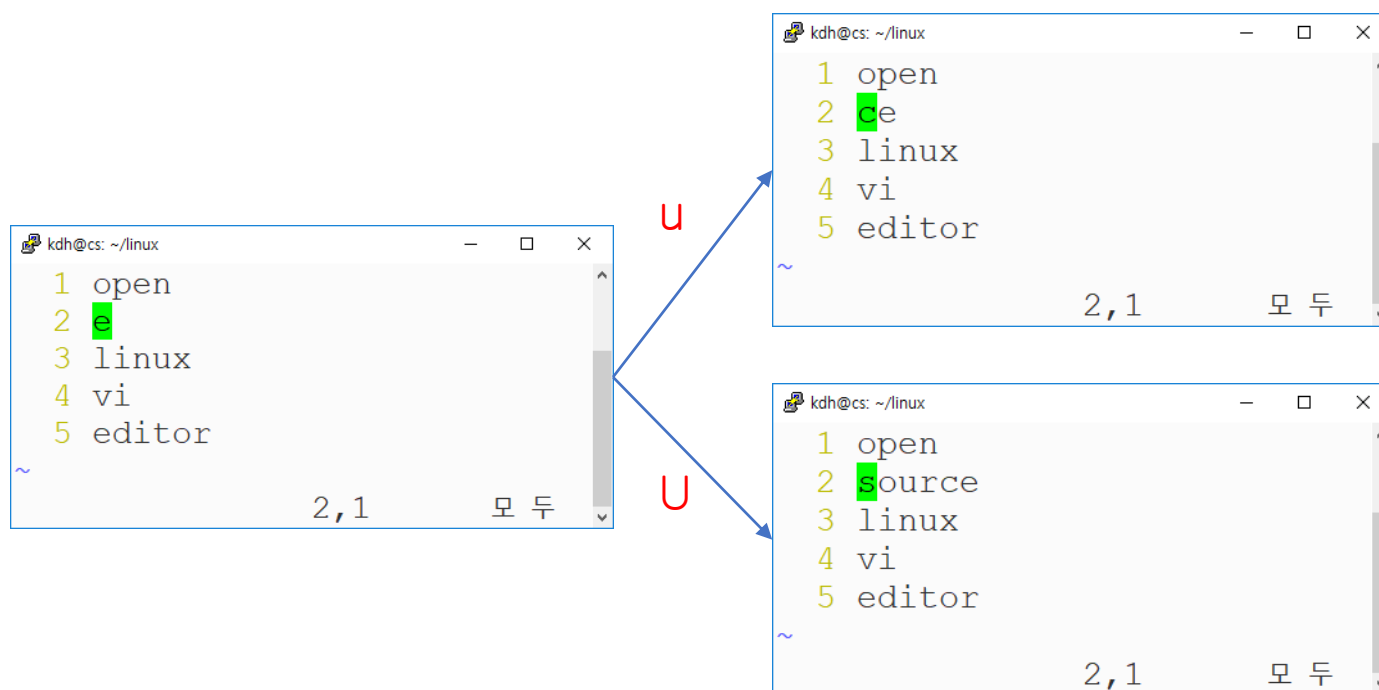
## 리눅스 문서편집기

- 명령모드에서 내용 삭제하기
  - `x` , `#x` : 커서 위치에서 지정한 글자 수 만큼 삭제 `x` , `3x`
  - `dw` , `#dw` : 커서 위치의 지정한 단어 수 만큼 삭제 `dw` , `3dw`
  - `dd` , `#dd` : 커서 위치에서 지정한 줄 수 만큼 삭제 `dd` , `3dd`
  - `D` : 커서 위치에서 행의 마지막까지 삭제



## 리눅스 문서편집기

- 이전 명령 취소하기
  - u : 커서 위치의 행에서 작업한 방금 전 명령을 취소
  - U : 커서 위치의 행에서 작업한 모든 명령 취소
  - :e! : 마지막으로 저장한 내용 이후의 변경내용을 버리고 새로 작업





## 리눅스 문서편집기

- 복사하기 , 잘라 내기 , 붙이기
  - yy , #yy : 커서 위치의 행에서 지정한 숫자만큼의 행을 복사
  - dd , #dd : 커서 위치의 행에서 지정한 숫자만큼의 행을 삭제
  - p : 커서가 위치한 행 다음에 복사된 내용을 붙임
  - P : 커서가 위치한 이전 행에 복사된 내용을 붙임

```
kdh@cs: ~/linux
1 open
2 source
3 linux
4 vi
5 editor
~
2,1 모 두
```

2yy : 2줄 복사  
P : 이전 행에 붙임

```
kdh@cs: ~/linux
1 open
2 source
3 linux
4 source
5 linux
6 vi
2,1 꼭 대 기
```



## 리눅스 문서편집기

- 버퍼에 복사
  - 언네임드 버퍼 : 이름이 없는 버퍼로 하나의 내용을 저장
  - 네임드 버퍼 : 이름이 다른 버퍼에 서로 다른 내용을 저장
    - "a , "b .....
    - "1 , "2 .....
    - 네임드 버퍼에 복사 : 복사 시 버퍼 이름을 표기 "ayy    "a3yy
    - 네임드 버퍼 붙이기 : 붙일 버퍼 이름을 표기 "ap



## 리눅스 문서편집기

- 범위 지정
  - 1, \$ 또는 1, % : 1행부터 마지막 행까지 범위 지정
  - 1, . : 1행부터 현재 커서가 위치한 행까지 범위 지정
  - ., \$ : 현재 행부터 마지막 행까지 범위의 지정
  - , -3 : 현재 행과 이전 세 행까지 지정
  - 10, 20 : 10번 행부터 20번 행까지 범위 지정
- 마지막행 모드 복사, 잘라내기, 붙이기 명령
  - :#y : 지정된 행을 복사 - :3y : 3번 행 복사
  - :<범위>y : 지정된 범위를 복사
  - :#d : 지정된 행을 삭제 - :3y : 3번 행 삭제
  - :<범위>d : 지정된 범위를 삭제
  - :pu : 복사된 내용을 현재 행 다음 위치에 붙임
  - :#pu : 복사된 내용을 지정한 행 다음 위치에 붙임



# Linux

## 리눅스 문서편집기

- 검색
  - 검색은 마지막행 모드의 하나인 ? 또는 / 를 입력
    - /문자열 : 문자열을 문서의 아래 방향으로 검색
    - ?문자열 : 문자열을 문서의 위쪽 방향으로 검색
    - n : 검색할 다음 문자열을 검색 방향과 같은 방향으로 검색
    - N : 검색할 다음 문자열을 검색 방향의 반대 방향으로 검색

```
kdh@cs: ~/linux
1 Open Source linux
2 linux text editor
3 vi editor
~
~
~
~/linux
```





# 리눅스 문서편집기

- 바꾸기
  - 기존의 문자열을 다른 문자열로 바꾸기

---

|                             |                           |
|-----------------------------|---------------------------|
| <code>:s/old/new</code>     | 현재 줄의 처음 old를 new로 교체     |
| <code>:s/old/new/g</code>   | 현재 줄의 모든 old를 new로 교체     |
| <code>:%s/old/new/g</code>  | 문서 전체에서 old를 new로 교체      |
| <code>:%s/old/new/gc</code> | 문서 전체에서 old를 new로 확인하며 교체 |

---



# Linux

## 리눅스 문서편집기

- 바꾸기 예
  - 현재 행의 linux 를 LINUX로 바꾸려면

```
kdh@cs: ~/linux
1 Open Source linux
2 linux text editor
3 vi editor
~
~
~
1,2 모두
```

:s/linux/LINUX

```
kdh@cs: ~/linux
1 Open Source LINUX
2 linux text editor
3 vi editor
~
~
~
: NUX 1,1 모두
```

- 모든 행의 linux 를 LINUX로 바꾸려면

```
kdh@cs: ~/linux
1 Open Source linux
2 linux text editor
3 vi editor
~
~
~
1,2 모두
```

:%s/linux/LINUX/g

```
kdh@cs: ~/linux
1 Open Source LINUX
2 LINUX text editor
3 vi editor
~
~
~
: INUX2,1 모두
```



Linux

# 리눅스 문서편집기

- 파일 읽어오기
  - :r 파일 : 지정한 파일을 읽어 현재 커서 위치에 삽입
  - :e 파일 : 지정한 파일로 작업 전환 ( 기존파일은 저장되어 있어야함 )
  - :n : vi 로 여러 파일을 열었을 경우 다음 파일로 작업 이동

# 리눅스 문서편집기



- 셸 명령 실행

- `:! 셸명령` : vi 에디터 작업을 잠시 중단하고 셸 명령을 수행한다. 엔터를 누르면 vi 에디터로 돌아온다.
- `:sh` : 셸 명령을 수행하기 위해 vi 에디터를 잠시 빠져나온다. exit 명령을 입력하면 vi 에디터로 돌아온다.
- `r![명령어]` : vi 화면에 명령어 결과를 출력한다.

```
kdh@cs: ~/linux
1 Open Source LINUX
2 INUX text editor
3 vi editor
~
~
~
: INUX2,1 모두
```

`:! ls`

```
kdh@cs: ~/linux
[마지막으로 고친 뒤 저장 안함]
ex exam.link
ex exam.softlink
계속하려면 엔터 혹은 명령을 입력하십시오
```

```
kdh@cs: ~/linux
1 Open Source LINUX
2 INUX text editor
3 vi editor
~
~
~
: INUX2,1 모두
```

`:sh`

```
kdh@cs: ~/linux
ex exam.softlink
계속하려면 엔터 혹은 명령을 입력하십시오
[마지막으로 고친 뒤 저장 안함]
kdh@cs:~/linux$
```

# 리눅스 문서편집기



Linux

- 다중 창 사용

- Ctrl +w, n 창 가로 분할
- Ctrl +w ,c 분할 닫기 ( 저장된 파일에 한해서 )
- Ctrl +w, w 창 간 전환
- Ctrl +w, s 현재 파일로 수평 분할
- Ctrl +w, v 현재 파일로 수직 분할

