

리눅스 시스템

3주차



목 차

1 리눅스 텍스트 에디터

2 vi 실습



리눅스 텍스트 에디터

- vi
 - 리눅스의 대표적인 텍스트 에디터
 - 사용자 편의에 맞게 수정가능한 커맨드 기반의 에디터
 - 일반모드와 입력모드가 구분되어 있어 글 수정 및 작성은 입력모드에서만 가능함
- VIM
 - vi가 발전된 버전이라는 의미로 Vi IMproved라고 하며, 현재 리눅스는 실제로 vim을 사용(vi를 실행해도 vim이 실행됨)
- emacs
 - 확장성이 좋은 유닉스 환경 텍스트 에디터
 - 종종 vim과 비교되며 많은 기능을 제공하지만 프로그램 자체가 무겁기 때문에 상대적으로 느림

리눅스 텍스트 에디터

- vi 기본 사용
 - 명령어 'vi'로 실행

| 상태 | 입력 | 내용 |
|-------|-----------------|--------------------|
| 일반모드 | i | 입력모드로 전환 |
| 입력모드 | esc | 일반모드로 전환 |
| 일반모드 | h, j, k, l | 커서 ←, ↓, ↑, → 이동 |
| 일반모드 | : | 커맨드 입력 |
| 명령 모드 | q, q!, wq | 종료, 강제 종료, 저장 후 종료 |
| 명령 모드 | w, w [filename] | 저장, filename으로 저장 |

리눅스 텍스트 에디터

- vi 설정파일
 - 홈 디렉토리(~/)에서 .vimrc 파일 수정(없으면 생성)
 - 설정파일이 있어도 변함이 없으면 vim 설치

| 설정 | 내용 |
|--|-----------------------|
| set nu | 줄 번호 출력 |
| set autoindent | 자동 인덴트 설정 |
| set cindent | C 인덴트 설정 |
| set ts=4 | 탭 스페이스 4로 설정 |
| set shiftwidth=4 | 자동 들여쓰기 스페이스 4로 설정 |
| if has("syntax") syntax on endif | syntax가 있을 때 색깔 하이라이팅 |

일반 모드

- 커서 이동

| 키 | 내용 |
|------------------|--------------------------------|
| h, j, k, l | 커서 ←, ↓, ↑, → 이동 |
| e, E | 다음 word, WORD의 끝으로 이동 |
| w, W | 다음 word, WORD의 처음으로 이동 |
| b, B | 이전 word, WORD의 처음으로 이동 |
| 0, ^ | 공백 포함 줄의 처음, 공백 제외 줄의 처음 |
| \$ | 줄의 마지막 |
| gg | 문서의 처음 |
| (숫자) + Shift + g | 해당 라인으로 이동, 숫자가 없으면 문서의 끝으로 이동 |

- word와 WORD의 차이

```
#include <stdio.h> = 7 word, 2 WORD
```

```
#include <stdio.h> #include <stdio.h>
```

일반 모드

- 커서 이동
 - 참고 Vim 어드벤처 : <https://vim-adventures.com/>
 - 일반 모드에서 익숙치 않은 커서 이동 학습을 위한 게임



일반 모드

- 화면 스크롤

| 키 | 내용 |
|-----------|------------|
| Ctrl + f | 한 화면 아래로 |
| Ctrl + b | 한 화면 위로 |
| Ctrl + d | 반 화면 아래로 |
| Ctrl + u | 반 화면 위로 |
| Ctrl + e | 한 줄 아래로 |
| Ctrl + y | 한 줄 위로 |
| Shift + h | 화면의 맨 윗줄로 |
| Shift + m | 화면의 중간줄로 |
| Shift + l | 화면의 맨 아랫줄로 |

일반 모드

- 입력 명령

| 키 | 내용 |
|---|---------------------------|
| i | 현재 위치에서 입력 모드 |
| I | 현재 줄의 맨 앞에서 입력 모드 |
| a | 현재 위치의 한칸 뒤에서 입력 모드 |
| A | 현재 줄의 맨 뒤에서 입력 모드 |
| o | 현재 줄의 아래에 새로운 줄을 열고 입력 모드 |
| O | 현재 줄의 위에 새로운 줄을 열고 입력 모드 |
| s | 현재 위치의 한 문자를 지우고 입력 모드 |
| S | 현재 줄을 지우고 입력 모드 |

일반 모드

- 편집 명령(none visual block)

| 키 | 내용 |
|----------------|-------------------------------------|
| y | 한줄 복사 |
| yn | 현재 줄부터 +n줄만큼 복사 (ex, n=4이면 총 5줄 복사) |
| p | 복사된 내용 붙이기 |
| dd | 한줄 삭제 |
| dw, dW | 커서 위치부터 한 word, WORD 삭제 |
| Shift + d, d\$ | 커서 위치부터 현재 줄 마지막까지 삭제 |
| Shift + j | 현재 줄 가장 뒤의 개행(wn)을 삭제 |

vim에서 삭제는 더 정확히 말하면 잘라두기이다.
p 명령을 사용하면 가장 마지막에 삭제한 내용이 붙여넣기 된다.

일반 모드

- 되돌리기 (Undo)

| 키 | 내용 |
|---|-----------------|
| u | 직전 명령을 취소하고 되돌림 |

일반 모드

- 블록 지정

| 키 | 내용 |
|-----------|------------------------|
| v | v 입력 후 커서 이동을 통해 블록 지정 |
| Shift + v | 라인 단위 블록 지정 |
| Ctrl + v | 블록 단위 블록 지정 |

일반 모드

- 편집 명령(visual block)

| 키 | 내용 |
|---|-------------|
| y | 지정된 블록 복사 |
| p | 복사된 블록을 붙이기 |
| d | 지정된 블록 삭제 |

명령 모드

- 찾기

| 키 | 내용 |
|-------------------|------------------------|
| :/pattern(정규표현식)/ | pattern 내용 검색 |
| n | 아래로 내려가며 일치하는 다음 단어 검색 |
| N | 위로 올라가며 일치하는 다음 단어 검색 |

명령 모드

• 바꾸기

| 키 | 내용 |
|--|------------------------------------|
| : [범위] s/[old pattern]/[new pattern]/ | 범위 내 old pattern을 new pattern으로 바꿈 |
| : [시작],[마지막] s/[old]/[new]/ | 시작과 마지막 범위 내 old를 new로 바꿈 |

| 범위 표현 | 범위 |
|-------|-------|
| % | 전체 |
| . | 현재위치 |
| \$ | 마지막 |
| 숫자 | 해당 라인 |

명령 모드

- 파일

| 키 | 내용 |
|-------------------|-------------------------|
| :e [filename] | 파일명이 filename인 파일 열기 |
| :q, q!, :wq | 종료, 강제종료, 저장 후 종료 |
| :w, :w [filename] | 저장, filename으로 저장 |
| :<범위>w [filename] | 지정한 범위만 filename으로 저장 |
| :f | 현재 작업중인 파일의 이름과 라인 수 출력 |

명령 모드

- 화면 분할

| 키 | 내용 |
|-------------------------|----------------------------------|
| :sp [file, dir] | 수평으로 화면 분할 |
| :vs [file, dir] | 수직으로 화면 분할 |
| :20vs | 20:80 비율로 수직 화면 분할 |
| Ctrl + ww, wW | 분할된 다음, 이전 화면으로 이동 |
| Ctrl + w + [h, j, k ,l] | 분할된 화면을 커서 이동을 통해 이동 |
| Shift + p | :sp ./ 등으로 디렉토리 네비게이트 상태에서 파일 열기 |

- 기타

| 키 | 내용 |
|----------------------|--|
| vi file1 file2 file3 | 동시에 여러 개 파일 열기, 다음 파일 열기 :n, 이전 파일 열기 :N |
| Ctrl + z | vi 화면을 잠시 내리고 터미널로 돌아감 |
| jobs | Ctrl + z 로 중지된 프로세스 확인 |
| fg | 중지된 프로세스 복귀, jobs가 여러 개일 때 숫자로 선택 (fg 1) |



Q&A

ABSTRACT BACKGROUND