

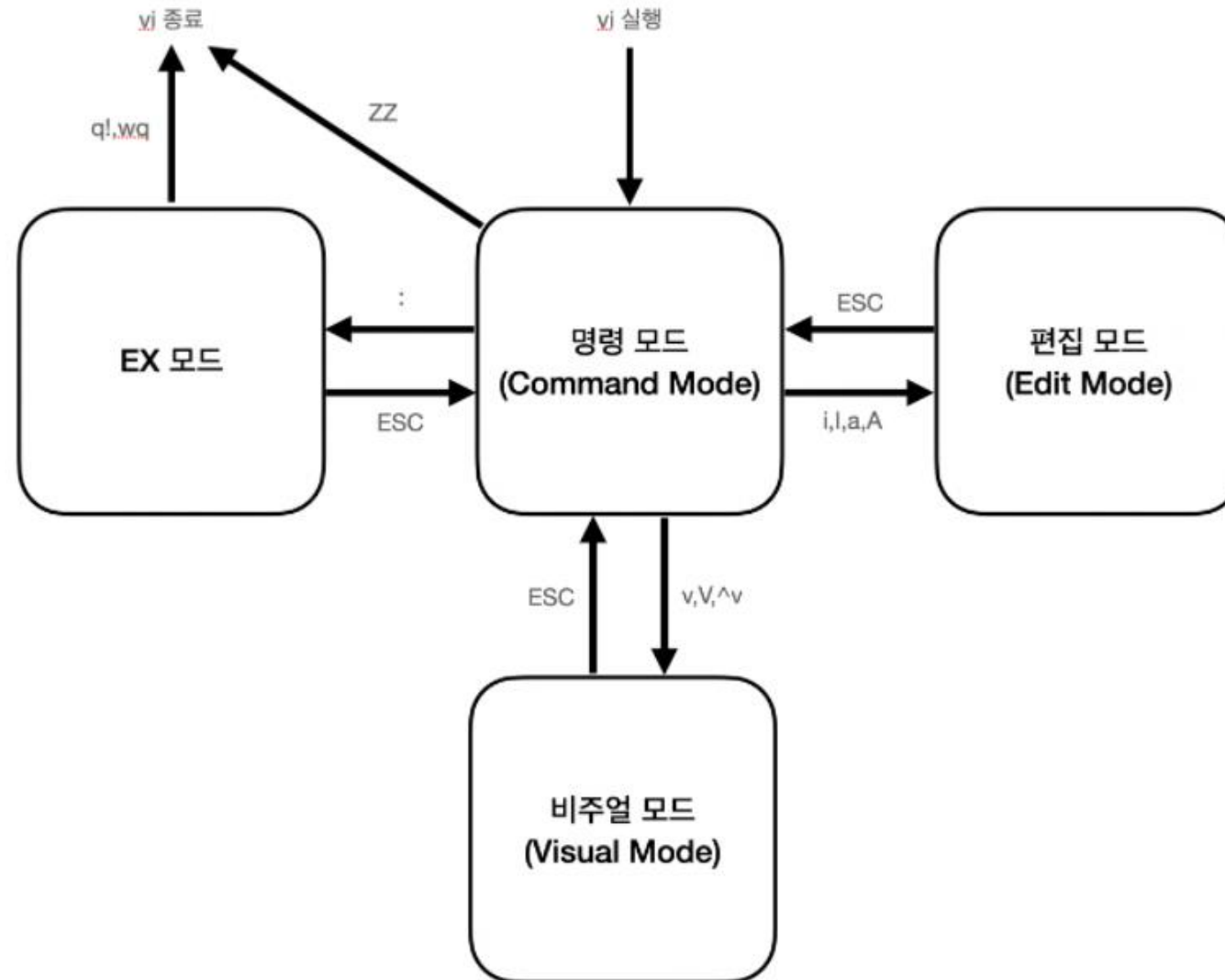
시스템 프로그래밍

Linux system programming

Vim & C 프로그래밍 환경



vi 상태 및 상태 전환



명령 모드

- 현재 커서 위치에서 편집 모드로 전환 **i**
- 현재 커서 위치 줄의 첫문자 위치에서 편집 모드로 전환 **I**
- 현재 커서의 바로 뒤에서 편집 모드로 전환 **a**
- 현재 커서 위치 줄의 마지막 위치에서 편집 모드로 전환 **A**
- 왼쪽으로 커서 위치 이동 **h**, ←
- 오른쪽으로 커서 위치 이동 **l**, →
- 위쪽으로 커서 위치 이동 **k**, ↑
- 아래쪽으로 커서 위치 이동 **j**, ↓
- 다음 단어 단위로 커서 이동 **w**
- 이전 단어 단위로 커서 이동 **b**
- 커서 위치 라인의 첫 문자로 이동 **^**, **0**
- 커서 위치 라인의 끝 문자로 이동 **\$**
- 마지막 라인으로 이동 **G**
- 100번째(1부터 시작) 라인으로 이동 **100G**
- 편집된 내용 되돌리기 **u**

- 커서 위치 한글자 삭제 **x**
- 커서 위치를 포함해 세 글자 삭제 **3x**
- 커서 앞의 글자 삭제 **X**
- 커서 위치의 단어 단위로 삭제 **dw**
- 커서 위치 줄 삭제 **dd**
- 커서 위치의 줄을 포함해 세 줄 삭제 **3dd**
- 커서 앞의 글자 삭제 **X**
- 커서 위치에서 줄 끝 문자까지 삭제 **D**
- 커서 위치의 줄을 버퍼에 복사 **yy**
- 커서 위치의 줄을 포함해 세 줄을 버퍼에 복사 **3yy**
- 버퍼에 복사된 문자열을 커서 위치에 붙여넣기 **p**
- 버퍼에 복사된 문자열을 커서의 앞 위치에 붙여넣기 **P**
- 한 글자만 수정 **r**
- 커서 다음 위치에 빈줄 추가 **o**
- 커서 이전 위치에 빈줄 추가 **O**
- 커서 위치의 라인과 다음 라인을 한 라인으로 붙임 **J**



EX 모드

- 변경된 내용 저장 **w**
- 변경된 내용 저장하고 종료 **wq**
- 변경된 내용 무시하고 종료 **q!**
- 종료 **q**
- 문자열 검색(예: **boy**) **/boy**
 - 다음 검색 문자열 **n**
 - 이전 검색 문자열 **N**
- 문자열 변경(예: **boy**를 **guy**로 변경) **%s/boy/guy/ig**
- 줄 번호 표시 **set nu**
- 줄 번호 감추기 **set nonu**

비주얼 모드

- 글자 단위로 선택(명령모드에서 전환시) **v**
- 라인 단위로 선택(명령모드에서 전환시) **V**
- 블록 단위로 선택(명령모드에서 전환시) **^v**
- 선택된 텍스트를 버퍼에 복사 **y**
- 선택된 텍스트를 삭제 **d, x**



gcc 컴파일러

- gcc(GNU cc) 컴파일러 상업용 C 컴파일러(cc)
\$ gcc [-옵션] 파일 \$ cc [-옵션] 파일
- 컴파일
\$ gcc long.c
\$ a.out // 실행 파일
- -c 옵션
\$ gcc -c long.c
- -o 옵션
\$ gcc -o long long.o
혹은
\$ gcc -o long long.c
\$ long // 실행 파일



단일 모듈 프로그램 : long.c

```
#include <stdio.h>
#define MAXLINE 100
void copy(char from[], char to[]);
char line[MAXLINE]; // 입력 줄
char longest[MAXLINE]; // 가장 긴 줄
/*입력 줄 가운데 가장 긴 줄 프린트 */
main()
{
    int len;
    int max;
    max = 0;
    while (gets(line) != NULL) {
        len = strlen(line);
        if (len > max) {
            max = len;
            copy(line, longest);
        }
    }
}
```

```
if (max > 0) // 입력 줄이 있었다면
    printf("%s", longest);

return 0;
}
/* copy: from을 to에 복사; to가 충분히 크다고 가정*/
void copy(char from[], char to[])
{
    int i;
    i = 0;
    while ((to[i] = from[i]) != '\0')
        ++i;
}
```



다중 모듈 프로그램

- 단일 모듈 프로그램
 - 코드의 재사용(reuse)이 어렵고,
 - 여러 사람이 참여하는 프로그래밍이 어렵다
 - 예를 들어 다른 프로그램에서 copy 함수를 재사용하기 힘들다
- 다중 모듈 프로그램
 - 여러 개의 .c 파일들로 이루어진 프로그램
 - 일반적으로 복잡하며 대단위 프로그램인 경우에 적합



다중 모듈 프로그램: 예

- main 프로그램과 copy 함수를 분리하여 별도 파일로 작성
 - main.c
 - copy.c
 - copy.h // 함수의 프로토타입을 포함하는 헤더 파일
- 컴파일

```
$ gcc -c main.c
$ gcc -c copy.c
$ gcc -o main main.o copy.o
혹은
$ gcc -o main main.c copy.c
```



main.c

```
#include <stdio.h>
#include "copy.h"
char line[MAXLINE]; // 입력 줄
char longest[MAXLINE]; // 가장 긴 줄
/*입력 줄 가운데 가장 긴 줄 프린트 */
main()
{
    int len;
    int max;
    max = 0;
    while (gets(line) != NULL) {
        len = strlen(line);
        if (len > max) {
            max = len;
            copy(line, longest);
        }
    }
}
```

```
if (max > 0) // 입력 줄이 있었다면
    printf("%s", longest);

return 0;
}
```



copy.c

```
#include <stdio.h>
#include "copy.h"
/* copy: from을 to에 복사; to가 충분히 크다고 가정*/
void copy(char from[], char to[])
{
    int i;
    i = 0;
    while ((to[i] = from[i]) != '\0')
        ++i;
}
```

copy.h

```
#define MAXLINE 100
void copy(char from[], char to[]);
```



make 시스템

- make 시스템
 - 대규모 프로그램의 경우에는 헤더, 소스 파일, 목적 파일, 실행 파일의 모든 관계를 기억하고 체계적으로 관리하는 것이 필요
 - make 시스템을 이용하여 효과적으로 작업
- Makefile
 - 실행 파일을 만들기 위해 필요한 파일들과 만드는 방법을 기술
 - make 시스템은 파일의 상호 의존 관계를 파악하여 실행 파일을 쉽게 다시 만듦.
- \$ make [-f 메이크파일]
 - 옵션이 없으면 Makefile 혹은 makefile을 사용



메이크파일의 구성

- Makefile의 구성 형식

대상리스트: 의존리스트
명령리스트

- 예: Makefile

```
main:main.o copy.o
```

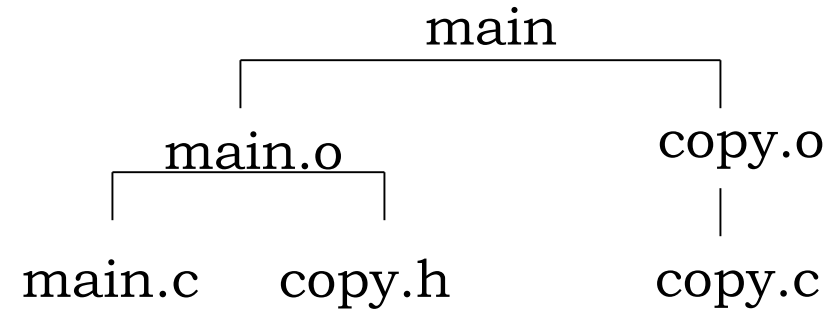
```
    gcc -o main main.o copy.o
```

```
main.o: main.c copy.h
```

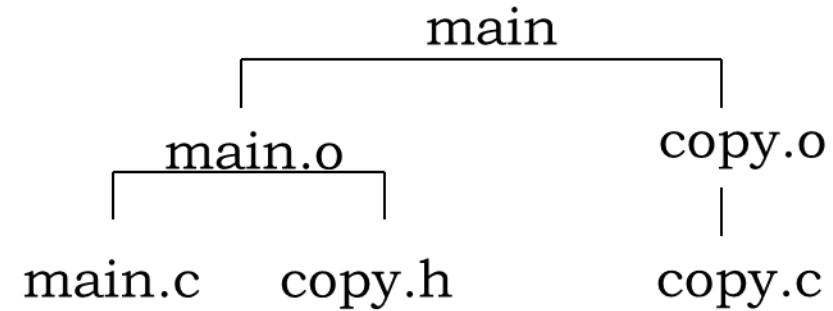
```
    gcc -c main.c
```

```
copy.o: copy.c
```

```
    gcc -c copy.c
```



- make 실행
\$ make 혹은 \$ make main
gcc -c main.c
gcc -c copy.c
gcc -o main main.o copy.o
- copy.c 파일이 변경된 후
\$ make
gcc -c copy.c
gcc -o main main.o copy.o



Reference

- ✓ 리눅스 프로그래밍, 창병모, 생능출판
- ✓ <https://www.44bits.io/ko/post/wsl2-install-and-basic-usage>

