

4장. vi 사용법 익히기

화면 이동

❖ 이전 혹은 다음 화면 보기

```
#include <stdio.h>
main()
{
    printf("Hello 1\n");
    printf("Hello 2\n");

    printf("Hello 3\n");
    printf("Hello 4\n");
    printf("Hello 5\n");
    printf("Hello 6\n");
    printf("Hello 7\n");

    printf ("Hello 8\n");
}
```

이 동	명령키
반 화면 위로	^u
반 화면 아래로	^d
한 화면 위로	^b
한 화면 아래로	^f

내용 삭제 및 취소

❖ 명령 모드에서 동작

명령어	삭제 대상	수행 작업
x, #x	문자	커서 위치의 문자 삭제(예:3x)
dw, #dw	단어	커서 위치의 단어 삭제
dd, #dd	줄	커서 위치의 줄 삭제
D(shift+d)	줄의 일부	커서 위치부터 줄 끝까지 삭제
u		방금 수행한 명령 취소
U	줄	해당 줄의 모든 편집 취소
:e!	파일	마지막 저장 이후 모든 편집 취소

- # 위치에 숫자를 넣으면 해당 숫자만큼 작업 반복

편집하기

- ❖ 복사, 잘라내기, 붙이기
- ❖ 명령 모드에서 동작

명령어	수행 작업
yy, #yy	현재 행을 버퍼로 복사 (예:4yy)
p	현재 행 다음에 버퍼 내용 삽입
P	현재 행 위쪽에 버퍼 내용을 삽입
dd, #dd	현재 행 잘라내기 (= 행 삭제)

- 행 삭제와 잘라내기는 동일한 동작임!

범위지정 방법

- ❖ 편집하는 범위를 지정하는 방법
- ❖ **마지막행 모드**에서 사용
- ❖ “: **범위 편집명령**” 형태로 사용

위치 표현 특수 문자
.: 현재 줄
\$: 마지막 줄

범위	의 미
: 1,\$	첫 줄에서 마지막 줄까지(파일내의 모든 줄)
: %	첫 줄에서 마지막 줄까지(파일내의 모든 줄)
: 1,.	첫 줄에서 현재 줄까지
: .,\$	현재 줄에서 마지막 줄까지
: .,-2	현재 줄에서 앞쪽으로 2번째 줄
: 10,20	10번째 줄에서 20번째 줄까지

마지막 행 모드 편집 명령

❖ 마지막 행 모드에서 사용

명령어	수행 작업
: #y	#으로 지정한 행을 복사(:10y -> 10행을 복사)
: <범위>y	범위로 지정한 행을 복사(예, :10,20y -> 10행~20행까지 복사)
: #d	#으로 지정한 행을 삭제(:10d -> 10행을 삭제)
: <범위>d	범위로 지정한 행을 삭제(예, :10,20d -> 10행~20행을 삭제)
: pu	현재 행 다음에 버퍼내용 붙이기
: #pu	#으로 지정한 행 다음에 버퍼내용 붙이기(예, :5pu)

검색 기능

❖ 마지막 행 모드에서 사용

명령어	수행 작업
/문자열	현재 위치부터 파일 앞쪽으로 문자열 탐색
?문자열	현재 위치부터 파일 뒤쪽으로 문자열 탐색
n	다음 문자열 탐색
N	역방향으로 문자열 탐색

바꾸기 기능

❖ 마지막 행 모드에서 사용

명령어	수행 작업
:s/문자열1/문자열2/	커서가 위치한 줄에서만 문자열1을 문자열2로 바꿈
:<범위>s/문자열1/문자열2/	<범위>안의 모든 줄에 대해서 각 줄의 첫 번째 문자열1을 찾아 문자열2로 바꿈
:<범위>s/문자열1/문자열2/g	<범위>안의 모든 줄에 대해서 모든 문자열1을 문자열2로 바꿈
:<범위>s/문자열1/문자열2/gc	<범위>안의 모든 줄에 대해서 각 문자열1을 문자열2로 치환할 때 수정할지 안 할지를 묻는다

기타 기능

❖ 파일 읽어오기 / 여러 파일 편집

명령어	수 행
:r 파일명	지정한 파일을 현재 커서 위치에 삽입
:e 파일명	현재 파일 대신 지정한 파일을 읽음
:n	vi 시작 시 여러 파일을 지정하였을 경우 다음 파일로 이동

❖ vi에서 쉘 명령 실행

명령어	수행 작업
:!명령	vi를 중단하고 지정한 명령 수행 (vi로 돌아올 때 :.)
:sh	vi를 잠시빠져나가서 쉘을 수행 (vi로 돌아올 때 : exit)

기타 기능

❖ 알아두면 유용한 명령키들

명령어	수 행
:f 파일명	파일 이름을 지정한 이름으로 변경
:w %.old	현재 파일을 .old 이름으로 저장해 둘 때
^g	기본적인 파일정보 출력(파일명, 라인수 등)
J	현재 줄과 다음 줄 연결
.	바로 이전에 수행한 명령 재 실행
~	현재 커서 위치의 한 문자를 소문자 혹은 대문자로 전환

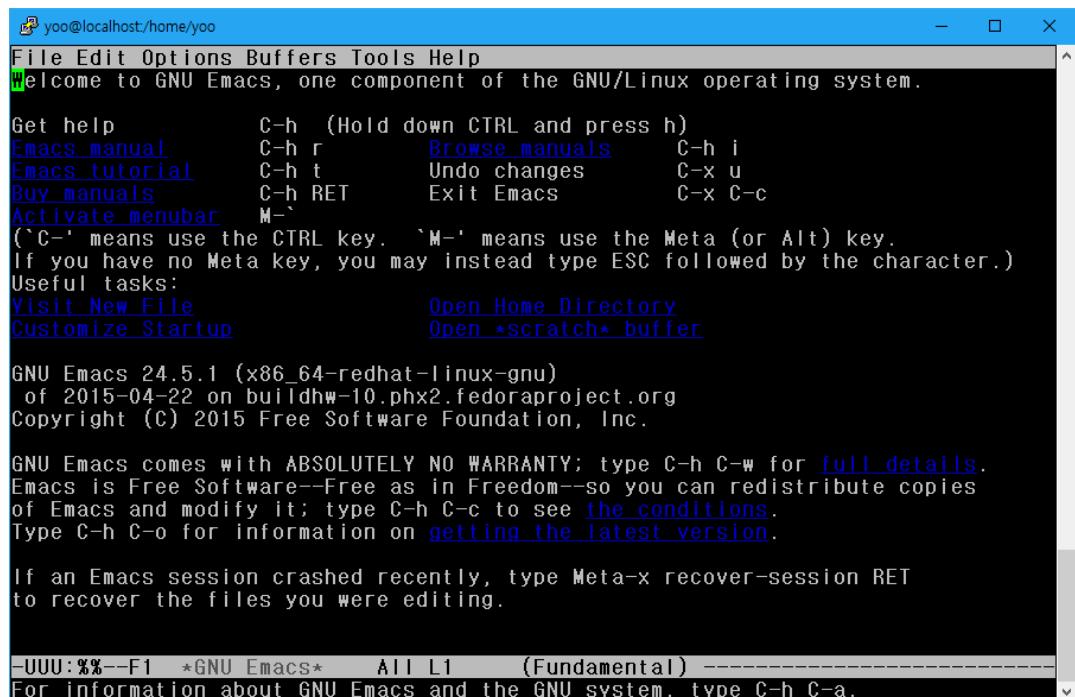
03. vi 환경 설정

❖ vi 의 환경을 설정하는 특수 명령과 변수들

명령어	수행 작업
:set nu	파일 내용의 각 줄에 줄 번호 표시
:set nonu	줄 번호 취소
:set list	눈에 보이지 않는 특수문자표시(tab:^I, eol:\$ 등)
:set nolist	특수문자보기 기능 취소
:set showmode	현재 모드 표시
:set noshowmode	현재 모드 표시기능 취소
:set	set으로 설정한 모든 vi변수 출력
:set all	모든 vi 변수와 현재 값 출력

Emacs

- ❖ 이맥스(Emacs)는 사용자가 많은 부분을 설정할 수 있는 고성능 문서 편집기이다.
- ❖ 이맥스는 단순한 편집기를 넘어서는 텍스트 처리를 위한 포괄적인 통합환경, 또는 응용 프로그램 실행 환경이라고 할 수 있다.
- ❖ 이맥스의 개발은 유닉스 환경을 주된 대상으로 하고 있지만, 마이크로 소프트 윈도우즈 등 다양한 환경에서도 이용할 수 있다.



Emacs 사용해보기

- ❖ “emacs” 실행 후에 q입력하면 입력 모드
 - ❖ “emacs 파일이름” 실행 시에 파일이 없으면 새파일을 생성하고 기존 파일이 존재하면 파일 열기
 - ❖ 파일 저장 : ^x+^w
 - ❖ 종료 : ^x+^c
-
- ❖ 윈도우 두개로 분할 : ^x+2
 - ❖ 윈도우 다시 한 개로 : ^x+1
 - ❖ 다른 윈도우로 건너가지 : ^x+o

4장 요약 [1/2]

❖ vi 동작 모드

- 입력 모드 : 입력한 키는 내용으로 저장
- 명령 모드 : 입력한 키는 커서 이동같은 명령으로 해석되어 실행
- 마지막 행 모드 : 화면의 하단으로 커서가 이동하여 특별한 명령 실행

❖ vi 명령

- 입력 : a, i, o, A, I, O
- 커서 이동 : h, j, k, l, \$, ^, #G, G, :#, :\$
- 화면 이동 : ^f, ^b, ^u, ^d
- 저장 및 종료 : ZZ, :w, :w 파일명, :q, :q!, :wq
- 내용 삭제 : x, dw, dd, D
- 이전동작 취소 : u, U, :e!
- 복사 및 붙이기 : yy, p, P, :y, :<범위>y, :pu, :#pu

4장 요약 [2/2]

❖ vi 환경 설정

명령어	수행 작업
:set nu	파일 내용의 각 줄에 줄 번호 표시
:set nonu	줄 번호 취소
:set list	눈에 보이지 않는 특수문자표시(tab:^I, eol:\$ 등)
:set nolist	특수문자보기 기능 취소
:set showmode	현재 모드 표시
:set nowrapmode	현재 모드 표시기능 취소
:set	set으로 설정한 모든 vi변수 출력
:set all	모든 vi 변수와 현재 값 출력