



# PC 실습 @ IT 학부

- UNIX의 주된 편집기 vi -



# 학습 목표

- UNIX에서 사용하는 편집기의 종류를 이해한다.
- UNIX의 대표적 화면 편집기인 vi의 사용 방법을 익힌다.
- vi의 환경 설정 방법을 익힌다.

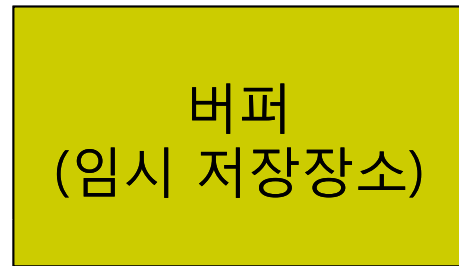
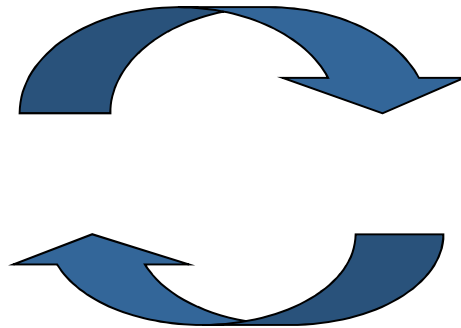
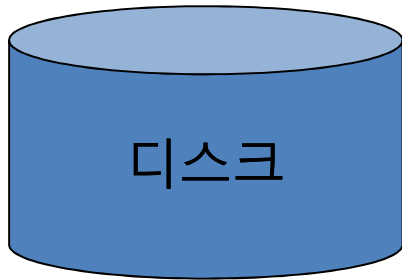
# UNIX의 편집기

- 행 단위 편집기 (라인 편집기)
  - ed
  - ex
  - sed
- 화면 편집기
  - vi
    - 모든 UNIX에 있음
  - emacs
    - 막강한 기능 제공; 추가로 설치해서 사용해야 함

# vi의 사용 방법

① 편집기에서 파일 호출

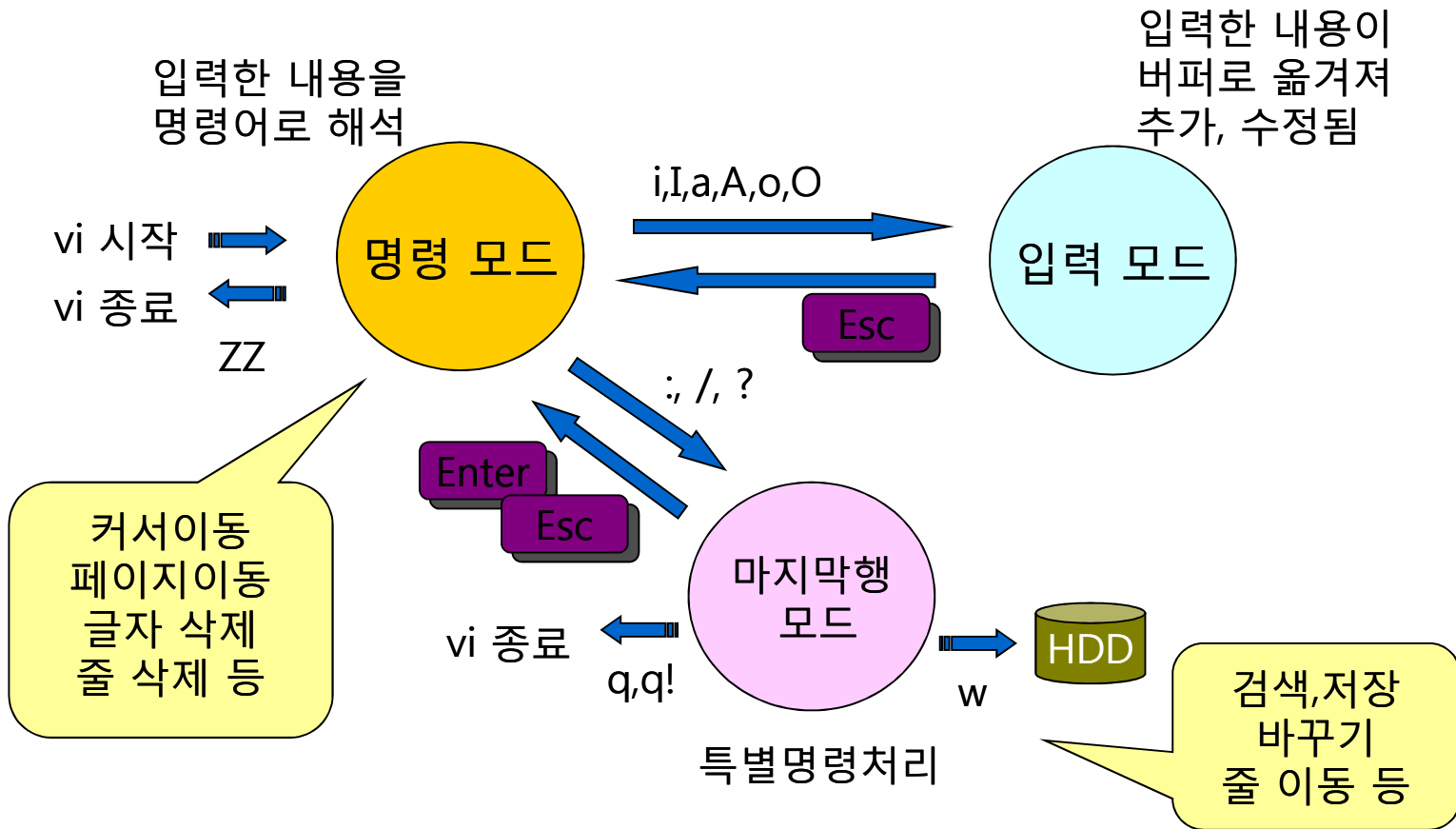
② 편집기에서 파일 수정



③ 파일 저장

- 파일 내용을 잘못 수정한 경우 버퍼(buffer)의 내용 무시하면 됨
- 작업 취소('undo') 기능 제공

# vi의 동작 모드



- vi에서는 대소문자를 별도의 명령으로 구분함

# vi 시작하기

- vi
  - 새로운 파일 시작
  - 파일 저장할 때 이름 지정
- vi 파일\_이름
  - 지정한 이름이 없으면 새로운 파일 생성
  - 지정한 이름이 있으면 기존 파일 열기

```
텔넷 ce.sungshin.ac.kr
```

```
$ vi
```

```
텔넷 ce.sungshin.ac.kr
```

```
$ vi first
```



# 입력 명령

- 입력 명령(명령모드 → 입력모드)

명령키	수행 작업
i	커서 앞에 삽입
a	커서 뒤에 삽입
o	현재 줄 다음에 삽입
I	현재 줄 첫 칸 앞에 텍스트 입력
A	현재 줄 끝에 텍스트 입력
O	현재 줄 앞에 삽입



입력이 끝나면 "입력모드"에서 "명령모드"로 돌아와야 함



# 저장 및 종료 명령

- 저장 명령
  - 저장하거나 종료하려면 "명령모드"로 돌아와야 한다.

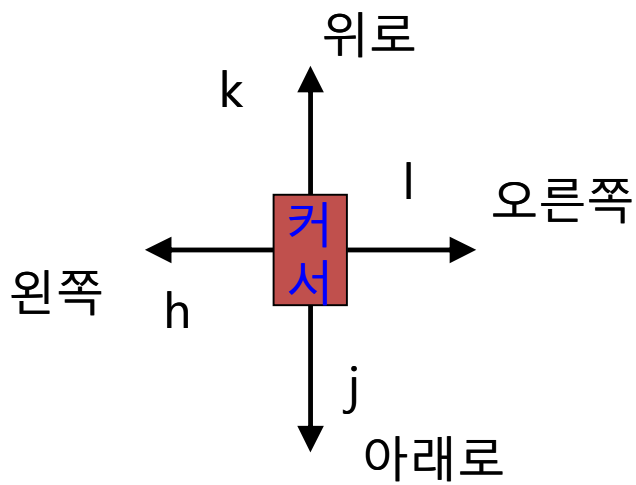
명령키	수행 작업
:w↵	현재의 파일명으로 파일 저장
:w 파일명↵	지정한 파일명으로 파일 저장

- 종료 명령(저장 후 종료 또는 그냥 종료)

명령키	수행 작업
:q↵	작업 내용을 저장하였으면 vi 종료
:q!↵	작업내용을 저장하지 않고 vi 종료
:wq↵	작업 내용을 저장한 후 vi 종료
:wq 파일명↵	작업 내용을 지정한 파일명으로 저장한 후 vi 종료
zz (shift-zz)↵	작업 내용을 저장한 후 vi 종료

# 커서 이동 명령 (1/3)

- 화살표 키 이용
- H, J, K, L 키 이용



이동	명령어
한 행 위	k
한 행 아래	j
한 문자 오른쪽	l
한 문자 왼쪽	h
행의 시작	^ 또는 0
행의 마지막	\$
이전 행의 처음	-
다음 행의 처음	+ 또는 ↵

## 커서 이동 명령 (2/3)

- 현재 화면에서 커서 이동

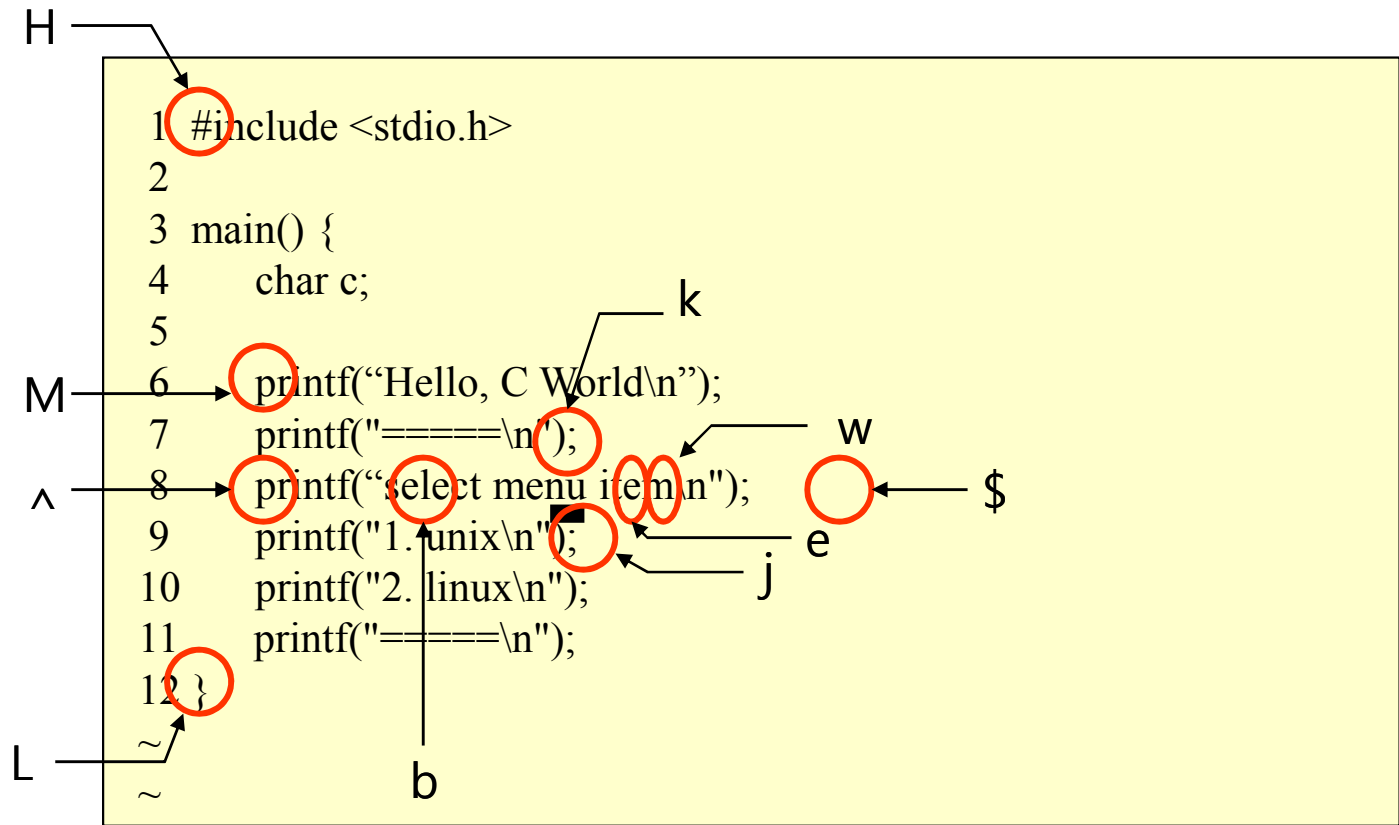
이동	명령키
키 화면 맨 위로	H
키 화면 중간으로	M
키 화면 맨 아래로	L
다음 단어의 첫 문자로	w
이전 단어의 첫 문자로	b
다음 단어의 끝 글자로	e

- 지정한 곳으로 이동

이동	명령키
줄 번호 n 위치로	:n 또는 nG
파일의 끝 줄로 이동	:\$ 또는 G
n줄 만큼 앞으로 이동	n+
n줄 만큼 뒤로	n-
현재 문장의 처음으로	(
다음 문장의 처음으로	)
현재 문단의 처음으로	{
다음 문단의 처음으로	}

# 커서 이동 명령 (3/3)

- 커서 이동 예제



# 화면 이동

- 화면에 나타나지 않은 부분으로 화면 이동

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

telnet ce.sungshin.ac.kr

```
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
한 줄만 위로	^y
한 줄만 아래로	^e
파일내용 다시 출력	^i

# 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		해당 줄의 모든 편집 취소

# 내용 수정

- 명령 모드에서 동작

키	수정 대상	수행 작업
r	문자	현재 커서위치의 한 문자 변경
R	문자열	현재 커서부터 ESC 입력까지 변경
cw	단어	커서 위치부터 현재 단어의 끝까지 내용 변경
cc	줄	커서가 위치한 줄의 내용 변경
s, ns	문자열	현재 커서부터 내용 변경(예: 5s)
C	줄 일부	커서 위치에서 줄 끝까지 내용 변경

# 편집 기능 - 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 행을 잘라내기

- 행 삭제와 잘라내기는 동일한 동작

# 버퍼의 사용

- 버퍼
  - vi는 작업 내용을 버퍼에 저장 → 실행 취소 가능
  - 복사하기, 잘라내기에 사용
- 버퍼 종류
  - Unnamed buffers(이름 없는 버퍼)
  - Named buffers(이름이 있는 버퍼): "a, "b, ... "z
  - Numbered buffers(번호가 있는 버퍼): "1, "2, ..., "9
- 사용 예
  - "a3yy → 현재 행부터 아래로 3줄을 a 버퍼에 저장
  - "ap → a 버퍼의 내용을 붙이기

# 범위 지정 방법

- 편집하는 범위를 지정하는 방법
- 마지막행 모드에서 사용
- “: 범위 편집\_명령” 형태로 사용

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

# 마지막행 모드에서 복사와 잘라내기

- 마지막행 모드에서 사용

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

# 검색 기능

- 마지막행 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

- 마지막행 모드에서 사용

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

# 기타 기능 (1/2)

- 파일 읽어오기 / 여러 파일 편집

명령어	수행
:r 파일명	지정한 파일을 현재 커서 위치에 삽입
:e 파일명	현재 파일 대신 지정한 파일을 읽음
:n	vi 시작시 여러 파일을 지정하였을 경우 다음 파일로 이동

- vi에서 셸 명령 실행

명령어	수행 작업
!:명령	vi를 중단하고 지정한 명령 수행 (vi로 돌아올 때: ↵)
:sh	vi를 잠시 빠져나가서 셸을 수행 (vi로 돌아올 때: exit)

## 기타 기능 (2/2)

- 알아두면 유용한 명령키들

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

# vi 환경 설정

- vi의 환경을 설정하는 특수명령과 변수들

명령어	수행 작업
:set nu	파일 내용의 각 줄에 줄 번호 표시 (보이기만 할 뿐 저장은 되지 않는다.)
:set nonu	줄 번호 취소
:set list	눈에 보이지 않는 특수문자 표시(tab: ^I, eol: \$ 등)
:set nolist	특수문자 보기 기능 취소
:set showmode	현재 모드 표시
:set noshowmode	현재 모드 표시 기능 취소
:set	set으로 설정한 모든 vi 변수 출력
:set all	모든 vi 변수와 현재 값 출력