

C Programming - Day 1

JunGu Kang



아주대학교



오리엔테이션



아주대학교



학습 목표

- 리눅스에 익숙해지기
 - GUI 없는 세상에 적응
 - Vim 에디터로 작업하고,
 - GCC로 코드를 컴파일
- C 코드를 쉽게 읽고 쓰기

계획

- 1일차(8월 28일)
 - printf, scanf, 연산자, 변수/상수 선언, 자료형, 분기문, 반복문
- 2일차(8월 31일 → 8월 30일)
 - 함수, 배열
- 3일차(9월 4일)
 - 포인터 1

계획

- 4일차(9월 7일)
 - 포인터 2
- 5일차(9월 11일)
 - 구조체, 공용체, 사용자 정의 자료형, 열거형
- 6일차(9월 14일)
 - 파일 입출력, 문자열 다루기, 헤더파일

과제

- 매 수업마다 과제 출제
- 과제 제출일까지 코드를 슬랙 #submit 채널에 업로드
 - 코드 복사해서 붙여넣지 말고,
 - 코드 파일 모두 압축해서 한 번에 업로드
- 과제도 반드시 리눅스에서 Vim으로 작업

질문

- 질문은 슬랙 #qna 게시판에 작성
 - 반드시 코드 첨부
 - 오류가 발생할 경우 오류 내용 첨부
 - 절대 사진 찍어서 업로드하지 말고,
 - 과제 제출과 동일하게 코드 파일 업로드
- 답변은 누구나 작성 가능
 - 답변 작성 시 댓글로 작성

코딩 스타일

- GNU Style
- K&R Style
- BSD Style

코딩 스타일 - GNU 스타일

```
main()
{
    if(1 == 1)
    {
        printf("1")
    }
}
```

코딩 스타일 - K&R 스타일

```
main() {  
    if(1 == 1) {  
        printf("1");  
    }  
}
```

코딩 스타일 - BSD 스타일

```
main()
{
    if(1 == 1)
    {
        printf("1");
    }
}
```

코딩 스타일 - 탭

- Soft Tab : 탭을 스페이스 2개 또는 4개로 사용
- Hard Tab : 탭을 탭 문자로 사용
 - 절대 하지 말 것
 - 서로 다른 운영체제 / 에디터에서 여는 경우 문제 발생

코딩 스타일

- 과제 / 질문 코드 작성시
 - K&R Style 또는 BSD Style 중 선택
 - Soft Tab으로 하되 스페이스 4개로 사용

Vim 에디터



아주대학교



모드

- 편집 모드
- 명령 모드

단축키

version 1.1
April 1st, 06

Esc

명령 모드

vi / vim 단축키 모음

~ 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄 끝으로 이동	% 일치하는 괄호 찾기	^ 줄의 첫 글자	& :s 반복	* 다음 검색	(문장 시작) 문장 끝	_ 아래줄로 이동	+ 다음 줄
· 매크로 이동	1	2	3	4	5	6	7	8	9	0 줄의 처음	- 이전 줄	= 자동 들여쓰기
Q 실행 모드	W 다음 WORD	E 끝 WORD	R 수정 모드	T 뒤로 검색	Y 줄단위 복사	U 줄단위 실행취소	I 줄 시작에서 삽입	O 행 위에 삽입	P 커서 이전에 붙여넣기	{ 문단 시작	}	문단 끝
q 매크로 기록	w 다음 단어	e 단어 끝	r 한 문자 교체	t 한 문자 검색	y 복사	u 실행취소	i 편집 모드	o 행 아래에 삽입	p 커서 이후에 붙여넣기	[기타]	기타
A 줄 끝에 덧붙이기	S 줄 삭제후 편집모드	D 줄 끝까지 삭제	F 뒤로 검색	G 파일끝으로 이동	H 화면 상단	J 줄 합치기	K 다음알	L 화면 하단	: ex 명령줄	" 레지스터 지정	열 이동	
a 덧붙이기	s 단어 삭제후 편집모드	d 1,3 삭제	f 한 문자 찾기	g 확장 명령	h ←	j ↓	k ↑	l →	: t/T/f/F 명령 반복	' 매크로 이동	\ 사용 안함	
Z 종료	X 백스페이스	C 줄 끝까지 바꾸기	V 줄단위 비주얼모드	B 이전 WORD	N 이전 (찾기)	M 화면 가운데	< 3 내어쓰기	> 3 들여쓰기	? 찾기 (뒤로)			
Z 확장 명령	X 글자 삭제	c 1,3 바꾸기	v 비주얼 모드	b 이전 단어	n 다음 (찾기)	m· 마크 설정	, 역순 검색	. 명령 반복	/· 찾기			

동작

커서를 이동하거나, 연산자가 동작할 범위를 지정합니다.

명령

바로 동작하는 명령, 빨간색은 편집 모드로 변경됩니다.

연산자

이동 관련 문자(숫자나 커서 이동)와 함께 사용해야 하며, 커서의 위치부터 목적지까지 연산합니다.

확장

특별한 키 함수로, 추가적인 키 입력이 필요합니다.

q·

입력후 (숫자를 제외한)으로 끝낼수 있는) 글자를 입력하여야 합니다.

words: 구분자로 공백, 특수기호 모두 사용

WORDS: 구분자로 공백 문자만 사용

words: quux(foo, bar, baz);

WORDS: quux(foo, bar, baz);

주요 명령행 명령 ('ex'):

:w (저장), :q (종료), :q! (저장하지 않고 종료)

:e f (파일 f 열기), :%s/x/y/g (파일 전체에서 'x' 를 'y' 로 교체), :h (vim 도움말), :new (새 파일)

그외 중요한 명령들:

CTRL-R: 재실행 (vim), CTRL-F/-B: 페이지 위로/아래로, CTRL-E/-Y: 줄 스크롤 위로/아래로, CTRL-V: 블록-비주얼 모드 (vim 전용)

비주얼 모드:

커서를 움직여 지정한 범위에 연산자를 적용합니다. (vim 전용)

참고:

(1) 복사/붙여넣기/지우기 명령어를 사용하기 전에 "x"를 입력하여 레지스터(클립보드)를 지정하세요. (x는 a에서 z 또는 * 을 사용할 수 있음) (예: "ay\$ 를 입력하면 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.)

(2) 어떤 명령을 입력하기 전에 횡수를 지정하면, 횡수만큼 반복하게 됩니다.(예: 2p, d2w, 5i, d4j)

(3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다. 예시: dd(현재 라인 지우기), >>(들여쓰기)

(4) ZZ 는 저장후 종료, ZQ는 저장하지 않고 종료.

(5) zt : 커서가 위치한 곳을 제일위로 올리기, zb : 바닥으로, zz : 가운데로

(6) gg : 파일의 처음으로(Vim 전용), gf : 커서가 위치한 곳의 파일 열기(Vim 전용)

vi/vim 에 대한 더 많은 강좌나 팁을 얻으려면 www.viemu.com (ViEmu, MS 비주얼 스튜디오를 위한 vi/vim 에뮬레이션)을 방문하십시오.

단축키

- Esc : 명령 모드로 전환
- a : 덧붙이기(편집 모드로 전환)
- i : 삽입(편집 모드로 전환)
- d : 삭제
- dd : 현재 줄 삭제

명령어

- :w : 저장
- :q : 종료
- 두 명령어를 합칠 수 있음
 - :wq : 저장 후 종료
- 명령에 !를 붙이면 오류가 발생해도 강제로 실행

명령어

- `:set nu` : 줄 번호 보이게 설정
- `:set et` : 탭을 스페이스로 변경

Vim 설정

- `/.vimrc` 파일에서 설정
- Soft Tab 설정
 - `set ts=4`
 - `au Bufenter *.\(c\|cpp\|h\) set et`
 - `.c`, `.cpp`, `.h` 파일인 경우 적용
 - `set et`
 - 모든 파일에 적용
- 줄 번호 보이게 설정
 - `set nu`

C 언어



아주대학교



컴퓨터의 데이터 처리



아주대학교



정수 데이터의 처리 - Unsigned

59

0	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

198

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

정수 데이터의 처리 - Signed

59

0	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

-59

1	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---

정수 데이터의 처리 - ASCII

A	65	0	1	0	0	0	0	0	1
---	----	---	---	---	---	---	---	---	---

자료형



자료형

- 데이터를 어떻게 읽고 쓸 것인가?

자료형

1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2Byte씩 읽으면 {57415}

1Byte씩 읽으면 {224, 71}

자료형

- 정수형
 - char, short, int, long, long long
- 실수형
 - float, double, long double

자료형 - 크기

Type	Size(x84)	Size(x64)
char	1 Byte	1 Byte
short	2Byte	2Byte
int	4Byte	4Byte
long	4Byte	8Byte
long long	8Byte	8Byte
float	4Byte	4Byte
double	8Byte	8Byte
long double	16(12)Byte	16Byte

변수와 상수



아주대학교



입력받고 출력하기



분기문



아주대학교



반복문

