혼자 연구하는 C/C++

김상형 님이 작성한 C/C++ 강좌. 책으로도 출판되었으나 인터넷에 모든 내용이 공개된 상태이다.

★ : 난이도 표시 (많을수록 어려움, 최고 3개)

1부. C 문법

제 4 장 제어문

#움직이기
getch 함수 특성 (확장키 입력시 0xE0 또는 0 반환 후 확장키에 해당하는 아스키코드 반환, int 형 필요)

break
이중 루프 빠져 나오는 법, goto

제 5 장 연산자

증감 연산자
후치 연산자 대상은 변수, 결과는 상수

복잡한 수식
undefined, unspecified

관계 연산자
좌변 상수, 우변 변수 비교 (실수 방지)

산술 연산자
Short Circuit

비트 연산자
전체

쉬프트 연산자
전체

산술 변환
상승 변환, 타입 확장

제 6 장 함수

함수의 원형
함수의 원형 선언 이유, 1패스 방식

모듈
모듈 분할 컴파일 방식 장점

#include
전처리기의 의미

디렉토리 경로 구분시 아닌 / 가 표준

#define
매크로 함수처럼 매크로 상수 정의할 때에도 전체를 괄호로 둘러 쌀 것
행 계속 문자

제 7 장 기억 부류

정적 변수
전역 변수, 정적 변수 구분!
전역 변수는 해당 모듈에서만 쓰일 수 있음. extern 으로도 불러올 수 없음.

정적 함수
언제 쓰이나 알아둘 것

블록 범위
지역 변수는 정확하게는 블록 범위를 가짐
for문의 초기식에 변수 선언하는 구문은 오래된 몇몇 컴파일러에선 지원되지 않음
하지만 표준문서에는 초기식에 변수 선언 가능하다고 표기

선언과 정의
함수나 변수 등의 선언은 중복 가능

제 8 장 표준 함수

절대값 함수
절대값 함수의 조건문 이용

난수의 생성
난수 생성 방법의 예들

time
시간 함수들의 버퍼 공유. 멀티 스레드에서 문제

제 10 장 포인터

★void 형
* 연산자를 직접 쓸 수는 없고 캐스팅 연산자 사용 후 * 연산자를 써야 한다.

할당의 필요성
전역변수일 때에만 스택에 값 저장 가능

할당 및 해제
할당 후 주의해야 할 점 (NULL값 체크 등)

★재할당
malloc, calloc의 차이점
realloc의 사용 예

★★★이중 포인터
이중 포인터 사용 예 - 포인터를 참조 호출로 전달하는 이유

★main함수의 인수

리턴 값의 의미
각 인자의 의미

★★void 이중 포인터
void* 와 void**의 차이점
void**에 저장되어 있는 값을 읽는 방법

제 11 장 배열과 포인터

배열의 내부적 처리
2차원 배열은 1차원 배열의 확장이다(내부적으로는 1차원이다)
→ 함수의 인자가 이중 포인터일 때 2차원 배열은 내부적으로 1차원이므로 넘길 수 없음

[] 연산자
ptr[n] = *(ptr + n) → 교환 법칙 성립
즉, ptr[n] = n[ptr]

★포인터 배열
정의 (뒤에 나올 배열 포인터와 구분)

★포인터 배열의 활용
동적 배열 응용

포인터와 배열
포인터와 배열의 차이
포인터가 배열보다 두 배 정도 빠르다

★★배열 포인터
포인터 배열과의 구분
배열 포인터 선언 방법 (포인터는 자신이 가리킬 대상의 크기를 알아야 한다는 원리로 외울 것)

★★배열 인수
배열 포인터 활용
배열 포인터 캐스팅

★이차 배열 할당
이중 포인터 선언

&ar
ar과 &ar의 차이

문자열 상수
문자열 상수를 컴파일러에서 처리하는 방식 의 사용

문자형 포인터
문자형 포인터와 문자열 배열의 차이

★★문자열 배열
2차원 문자 배열, 1차 포인터 배열의 차이, 3차원 확장

제 12 장 문자열 함수

문자열 복사
strncpy 사용법

문자열 비교
stricmp, strnicmp 사용하는 경우

문자열 검색
strchr, strrchr, strstr, strpbrk, strtok 사용법

문자열 변환
strset, strnset, strlwp, strupr, strrev

문자 관리 함수
나와 있는 함수 모두

메모리 관리 함수
memmove 익혀둘 것

정수와 문자열
정수, 문자열 변환 함수 익혀둘 것

★실수와 문자열
실수, 문자열 변환 함수

sprintf
sprintf의 단점 : 속도가 느리고 실행 파일의 크기가 커진다

★표준 함수 구현
표준 함수들 직접 구현해볼 것

★확장 함수 작성
문자열 처리 함수 작성하면서 필요할 때 다시 한 번 봐도 무방

제 13 장 구조체

정의
배열보다 구조체가 좋은 점

구조체 태그
구조체를 새로운 타입으로 정의하는 방법
구조체를 새로운 타입으로 정의하였을 때 장점

멤버 연산자
멤버의 주소에 접근하는 방법 (중요하진 않음)

초기화
구조체의 초기화는 대입문으로 바뀌므로 구조체의 크기가 커지면 시간이 오래 걸린다

구조체 대입
구조체 안의 배열도 대입 가능
구조체를 함수의 인수로 전달하는 경우에는 포인터로!

깊은 복사
구조체의 포인터 변수에 메모리 할당을 한 경우 깊은 복사 필요

비트 구조체 - 정의
사용법 익힐 것

비트 구조체 - 활용
비트 구조체 장점, 단점

공용체 - 정의
공용체의 변수들이 메모리를 어떻게 공유하는가? (특히 배열의 경우)
초기화하는 경우 첫 번째 멤버 기준

이름없는 공용체
C++컴파일러만 지원
활용 예

2부. C 고급

제 14 장 C 실습

시분할
시분할(Time Sharing) 개념

다른 예제들 한 번씩 분석해볼 것

제 15 장 포인터 고급

포인터와 const
전체

Const 인수
const인수가 꼭 필요한 경우는 포인터를 통한 참조 호출일 때 뿐이며 값 호출일 때는 큰 의미가 없다

volatile
잘 쓰이지 않는다. 개념만 익힐 것.
컴파일러가 보기에 코드가 비효율적이건 어쨌건 개발자가 작성한 코드 그대로 컴파일

함수 포인터 - 정의
함수 포인터 정의 방법 외울 것
배열명처럼 함수이름 자체가 포인터 상수
호출 형식은 두 가지가 있음

함수 포인터 타입
캐스트 연산자 외울 것
typedef 사용 방법

포인터로 함수 호출하기
함수 포인터 사용하는 경우

함수 포인터 리턴
함수 포인터를 리턴하는 함수의 원형 외울 것

가변 인수 함수
가변 인수 함수 쓰는 방법

가변 인수 함수의 조건
가변 인수 함수 사용 시 주의해야 할 점

제 16 장 함수 고급

스택 레지스터가 이미 값을 가지고 있는 경우 스택 이용

★★스택 프레임
다 볼 것 (__cdecl 호출 규약에 대한 설명)

★호출 규약
__cdecl과 __stdcall의 차이 -> 스택 정리의 주체, 가변 인수 지원 여부

호출 규약 불일치
__cdecl로 선언된 함수를 __stdcall로 불러들였을 때 에러 나는 이유

재귀 호출
재귀 호출의 장단점

재귀 호출이 가능한 이유
재귀 호출할 때 함수에서 사용되는 인수나 지역변수들은 호출별로 따로 기억된다. 주의해야할 점

★디렉토리 검색
참고하는 정도로 확인 (원리만 알면 됨)

3부. C++ 문법

제 28 장 연산자 오버로딩

★연산자 함수의 형식
연산자 함수 정의 형식
연산자 정의에 const, &을 쓰는 이유

제 32 장 예외 처리

미처리 예외
set_terminate 함수는 전역 적인 예외 처리

참고

http://soen.kr/

From:
http://obg.co.kr/doku/ - **OBG WiKi**

Permanent link:
http://obg.co.kr/doku/doku.php?id=programming:book:soen_cpp

Last update: **2020/11/29 14:09**