<자주 사용하는 C 라이브러리>

#include <>

함수를 제작하지 말고, 자기가 필요한 동작이 함수로 만들어져 있는지 검색하는 습관을 지닙시다.

-자주 사용하는 IBM C 라이브러리 헤더파일-

1. stdio.h : Standard Input Output
2. stdlib.h : Standard library
3. math.h : Math
4. ctype.h : C Type
5. string.h : String
6. time.h : Time
7. errno.h : Error number

헤더 파일 안에는

멤버 함수

멤버 상수

멤버 변수

멤버 형식

등이 들어있습니다.

여기서 멤버라는 것은 C 이후로 나온 객체 지향 언어 C++,JAVA에서 쓰이는 클래스라는 정의에서 사용되면서 사용하는 단어로, 2학년 JAVA 시간과 C++ 시간에 자세히 설명을 받길 바랍니다.

간단히 설명하자면 클래스는 특정 객체를 생성하기 위해 변수와 메소드를 정의하는 일종의 틀로,

객체를 정의하기 위한 상태(멤버변수)와 메서드(함수)로 구성됩니다.

클래스와 객체는 서로 다른 존재로,집을 설계한다고 가정할 때

집의 설계도는 클래스를, 실제 집은 객체로 생각하면 편합니다.

<IBM C 라이브러리 헤더파일 목록>

텍스트, 서류이(가) 표시된 사진

자동 생성된 설명

이제부터 설명할 헤더 파일안에는 여러 함수들이 있지만 모두 설명은 하지 못할 것이기에

혼자서 공부하는 자세를 기르길 바랍니다.

1. stdio.h :

Standard Input Output - 핵심 입력과 출력 함수들을 정의한다.

stdio.h의 함수들은 대략 [파일 조작 함수 / 콘솔 입출력 함수]로 볼 수 있습니다.

**-fopen / fclose :**

파일을 읽거나 쓴다. / 파일을 닫는다.

**-printf :**

표준 출력 스트림(stdin, default : 텍스트 터미널==모니터)에 출력한다.

**-scanf :**

표준 입력 스트림(stdout, default : 키보드)으로 입력한다.

**-ferror :**

오류 지시자가 주어진 스트림으로 설정되어 있는지 검사한다.

파일 객체를 쓸 때 오류가 나면 사용한다.

**-perror :**

오류 메시지를 출력한다.

**-getc :**

주어진 스트림으로부터 문자를 읽고 리턴한다.

scanf와 달리 공백 문자도 입력 가능하여 파일에 쓸 때 자주 사용한다.

**-getchar :**

getc(stdin)와 같은 효과를 낸다.

오래전에는 #define getchar() getc(stdin) 라 사용했고 자주 사용되어 함수로 등록됬다.

버퍼를 비우는데 사용하기도 한다.

<예제 코드 01>

#include <stdio.h>

int main() {

printf("getchar 가 왜 쓰일까요?");

int i;

char c;

scanf\_s("%d", &i);

scanf\_s("%c", &c);

printf("입력한 문자 : %c \n", c);

return 0;

}

getchar 가 왜 쓰일까요?

1

입력한 문자 :

이유 :

이전에 호출한 scanf 에 의해 버퍼에 이미 \n 이 남아 있었기 때문이다.

getchar 함수를 호출함으로 통해 버퍼에 남아 있던 \n 을 날려버릴 수 있다.

getc(stdin) 이란 즉 표준 입력 스트림에 남아있는 문자 한 개를 가져온다는 뜻으로,

남아있는 \n 을 가져온다는 것이다.

<예제 코드 02>

#include <stdio.h>

#pragma warning(disable:4996)

int main(){

FILE\* fp = fopen("hello.txt", "w"); // hello.txt 파일을 쓰기 모드(w)로 열기.

// 파일 포인터를 반환

fputs("Hello, world!", fp); // 파일에 문자열 저장

fclose(fp); // 파일 포인터 닫기

///////////////////////////////////////////////////////////////////////////////////////////

fp = fopen("hello.txt", "r"); // hello.txt 파일을 읽기 모드(r)로 열기.

if (fp == NULL){ //fp 파일이 존재하지 않을경우

perror("Error opening file");

// 오류가 발생하면 errno 에 값이 입력되고 perror 함수는 읽어낸 값을 바탕으로 알맞은 에러를 출력한다.

}

else {

fputc('x', fp); //fp에 x를 쓸 경우

if (ferror(fp)) //파일 입력 오류가 있을 경우 ferror 호출

printf("Error Writing to myfile.txt\n");

fclose(fp);

}

return 0;

}

이유:

읽기 전용으로 열은 파일에 fputc 를 이용해 강제적을 파일에 쓰기를 함으로써 오류를

발생시키기 때문이다.