

프로그램 구성요소 살펴보기

제주대학교 컴퓨터공학과

변영철 교수

(ycb@jejunu.ac.kr)

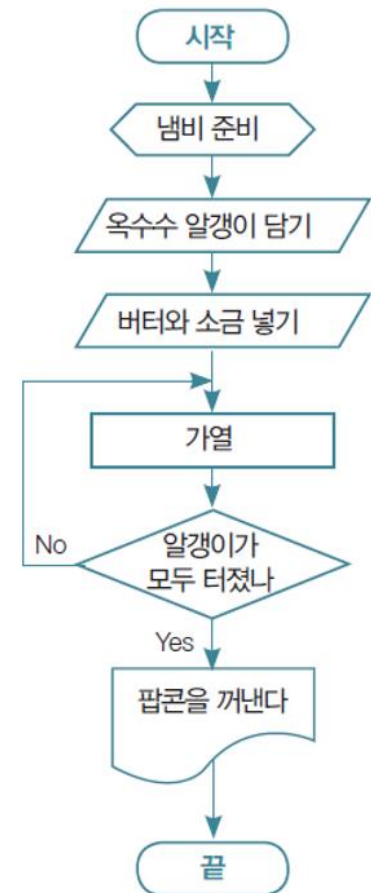
프로그램이란?

- 무엇인가를 하는 순서를 기술한 것
- 방송 순서 = TV 프로그램
- 요리 순서, '라면 끓이기', '팝콘 만들기'

글로 표현한 프로그램

1. 냄비를 준비한다.
2. 옥수수 알갱이를 냄비에 담는다.
3. 버터와 소금을 넣는다.
4. 가열한다.
5. 알갱이가 모두 터질 때까지 4번을 반복한다.
6. 팝콘을 꺼낸다.

순서도로 표현한 프로그램



컴퓨터 프로그램

- 컴퓨터에서 무언가를 하는 것을 기술하는 것
- 컴퓨터 프로그래밍 언어
 - 컴퓨터에서 프로그램을 기술하는 언어
 - C 언어도 프로그래밍 언어 중 하나

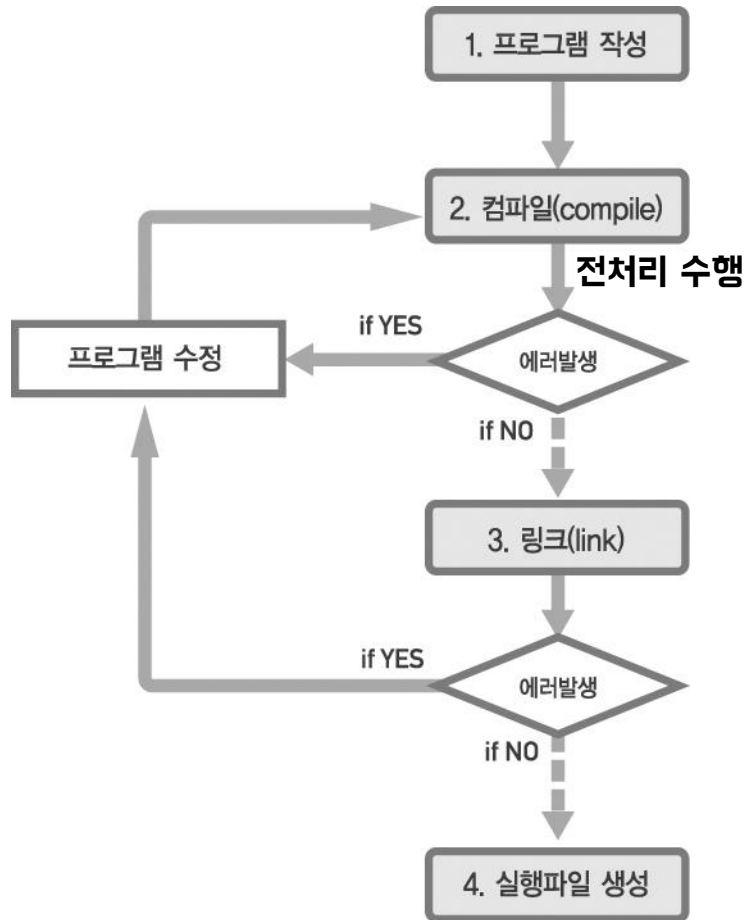
프로그램 작성부터 실행까지

1. C언어로 컴퓨터 프로그램 작성
2. 컴파일 & 링크 (빌드 | 솔루션 빌드)
3. 실행 (디버그 | 디버깅 하지 않고 시작)

```
#include <stdio.h>
```

```
void main() {  
    printf("Hello,World!\n");  
    getchar();  
}
```

프로그램 작성부터 실행까지

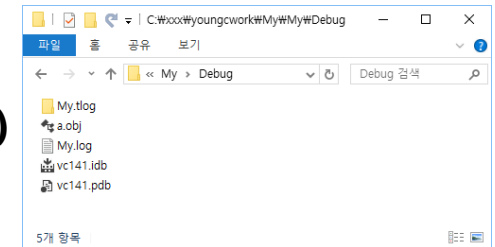


프로그램 소스 파일(a.cpp)

“사람이 이해할 수 있는 것”

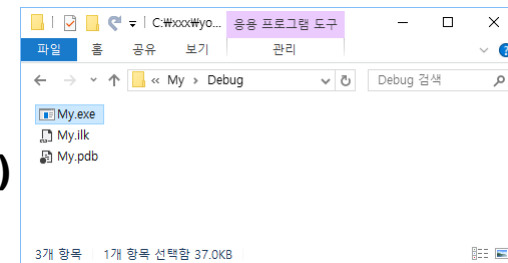
“컴퓨터는 실행 불가”

목적 파일(a.obj)



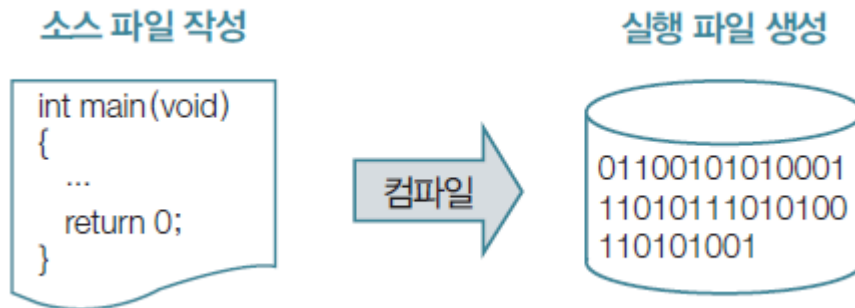
실행 파일(My.exe)

“컴퓨터가 실행 가능”



컴파일과 링크

- 컴파일(compile)
 - 우리가 작성한 프로그램 → 컴퓨터에서 실행할 수 있는 기계어로 변환 (번역)



메인 함수 main

- 프로그램을 실행하면 운영체제가 맨 처음 호출하는 함수
- 진입점(entry point)라고 불림

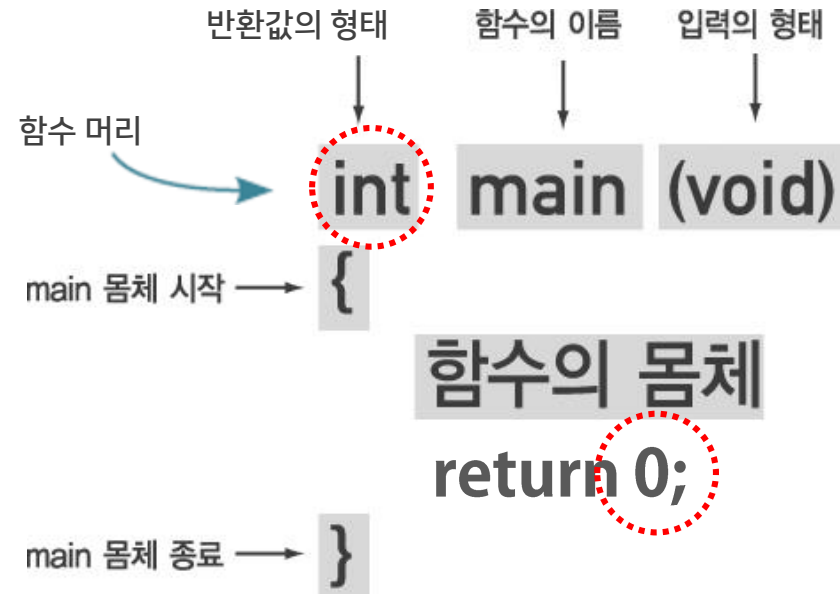
```
#include <stdio.h>

void main()
{
    printf("Hello, World!\n");
    getchar();
}
```

메인 함수 main

- 입력값이 없고 반환 값도 없는 main 함수
- 입력값은 없고 반환 값이 정수(int)인 main 함수

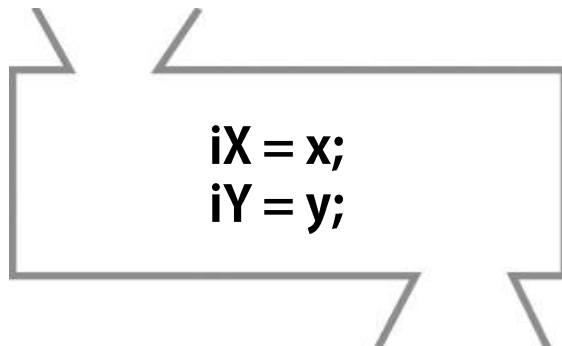
"void" = 입력이 없다는 의미



함수 Add와 Assign

- 코드를 간단히 표현한 것 = 코드 추상화
- C 언어의 기본 단위
- 함수호출, 입력 파라미터, 리턴값 전달

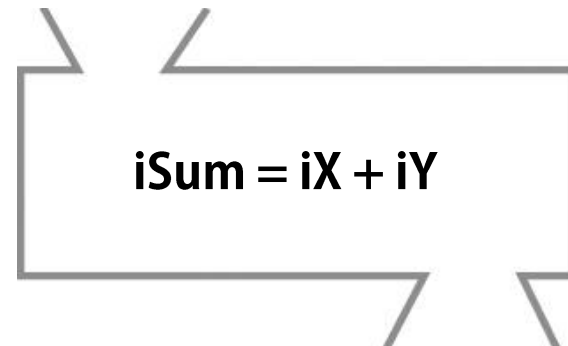
입력 파라미터 x, y



리턴값 없음

<Assign 함수>

없음



리턴값 = iSum

<Add 함수>

함수 구조

- 함수 머리(header)
 - 함수 원형(function prototype)이라고도 함
 - 함수의 이름, 입력 파라미터
- 함수 몸체
 - 함수에서 실행할 일들을 중괄호 안에 작성
 - 몸체의 마지막에는 `return 0;` 넣어 프로그램 종료와 함께 값을 반환(리턴)함.

프로그램 작성 문법

- 주석문(comment)
 - 소스 코드를 이해하는데 도움을 주는 메모
 - '/'와 '/' 사이의 모든 내용
 - '/' 이후 해당 줄의 끝까지
 - 컴파일러는 주석을 없는 것으로 간주
- 세미콜론(;) → 문장 끝 표시
- 한 줄에 한 문장씩 작성
- 일정한 간격으로 들여쓰기

```
/* 한 줄 짜리 주석 */
```

```
/*  
여러 줄에  
걸친 주석  
*/
```

```
// 주석 하나.  
// 주석 둘.  
// 주석 셋.
```

헤더 파일

- `stdio.h`
- 함수에 대한 정보를 가지고 있음.
- 가급적 프로그램 맨 위에 작성



The diagram shows a C program structure with two distinct regions highlighted by speech bubbles. The first region, labeled '첫번째 영역' (First Area), contains the preprocessor directive `#include <stdio.h>`. The second region, labeled '두번째 영역' (Second Area), contains the `main` function definition. The code is as follows:

```
#include <stdio.h>

int main(void)
{
    printf("Hello, World! \n");
    return 0;
}
```

printf 함수

- printf 함수
 - 이미 작성되어 있는 함수로서 호출하여 쉽게 사용할 수 있음.
 - 이미 표준화 해서 만들어 놓은 함수들의 집합으로서 표준 라이브러리라고 함.
 - 라이브러리에 있는 함수들의 입력 파라미터와 리턴값 정보를 선언해 놓은 것이 **헤더 파일**

printf 함수

- printf = print with format, 어떤 형태로 모니터에 출력

예제 2-2 문자열을 화면에 출력하는 프로그램

```
1. #include <stdio.h>
2.
3. int main(void)
4. {
5.     printf("Be happy");    // 문자열 "Be happy" 출력
6.     printf("My friend");   // 문자열 "My friend" 출력
7.
8.     return 0;
9. }
```

실행
결과

Be happyMy friend

printf 함수

```
/* printf1.c */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello Everybody \n");  
    printf("%d \n", 1234);  
    printf("%d %d \n", 10, 20);  
    return 0;  
}
```

printf 함수

```
/* printf2.c */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("My age : %d \n", 20);  
    printf("%d is my point \n", 100);  
    printf("Good \nmorning \neverybody\n");  
  
    return 0;  
}
```


printf 함수

표 2-1 printf 함수의 사용법

출력 데이터	사용 예	출력 결과
문자열	<code>printf("Be happy!");</code>	화면에 "Be happy!" 출력
제어 문자	<code>printf("Be happy!\n");</code>	"Be happy!" 출력 후 줄 바꿈
정수	<code>printf("%d", 10);</code>	화면에 정수 "10" 출력
실수	<code>printf("%lf", 3.5);</code>	화면에 실수 "3.500000" 출력
수식	<code>printf("%d", 10 + 20);</code>	10과 20을 더한 결과인 "30" 출력

printf 함수

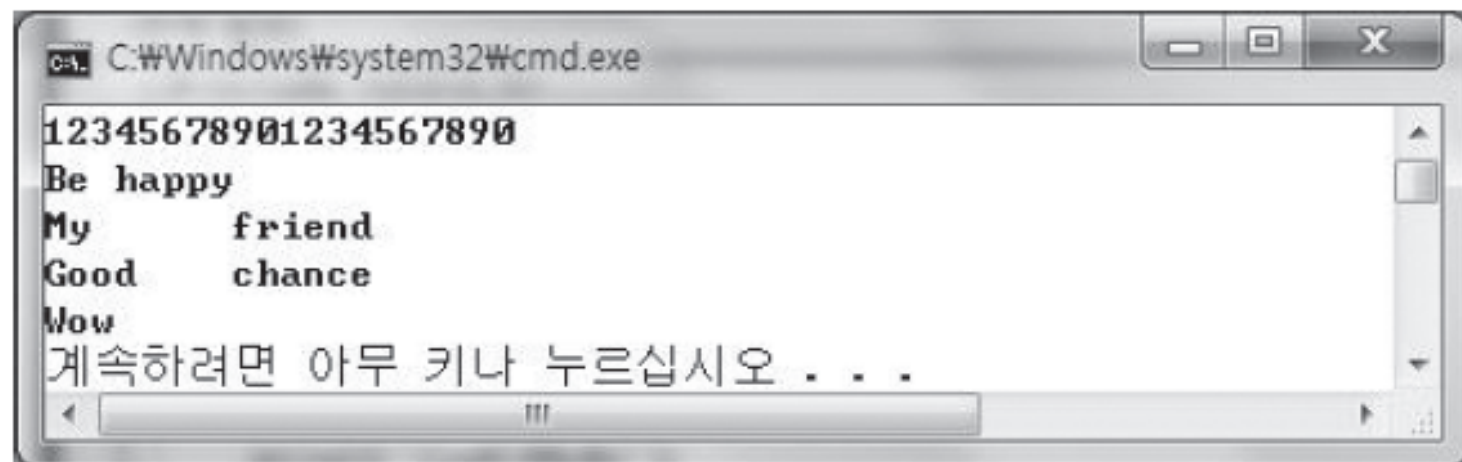
표 2-2 제어 문자의 종류

제어 문자	의미	기능
\n	개행(new line)	줄을 바꾼다.
\t	탭(tab)	출력 위치를 다음 탭(tab) 위치로 옮긴다.
\r	캐리지 리턴(carriage return)	출력 위치를 줄의 맨 앞으로 옮긴다.
\b	백스페이스(back space)	출력 위치를 한 칸 왼쪽으로 옮긴다.
\a	알럿(alert)	경보 벨(bell) 소리를 낸다.

printf 함수

예제 2-3 제어 문자를 사용한 출력

```
1. #include <stdio.h>
2.
3. int main(void)
4. {
5.     printf("12345678901234567890\n"); // 화면에 칸 번호 출력
6.     printf("Be happy\n");             // "Be happy"를 출력하고 줄을 바꿈
7.     printf("My\tfriend\n");           // "My"를 출력하고 탭 위치로 이동 후에 "friend" 출력
8.     printf("Goot\bd\tchance\n");     // t를 d로 바꾸고 탭 위치로 이동 후에 "chance" 출력
9.     printf("Cow\rW\la\n");           // C를 W로 바꾸고 벨 소리를 낸다.
10.
11.     return 0;
12. }
```



```
C:\Windows\system32\cmd.exe  
12345678901234567890  
Be happy  
My      friend  
Good    chance  
Wow  
계속하려면 아무 키나 누르십시오 . . .
```

printf 함수

예제 2-4 정수와 실수의 출력

```
1. #include <stdio.h>
2.
3. int main(void)
4. {
5.     printf("%d\n", 10);        // %d 위치에 10 출력
6.     printf("%lf\n", 3.4);      // %lf 위치에 3.4를 소수점 이하 6자리까지 출력
7.     printf("%.1lf\n", 3.45);   // 소수점 이하 첫째 자리까지 출력
8.     printf("%.10lf\n", 3.4);   // 소수점 이하 10자리까지 출력
9.
10.    printf("%d과 %d의 합은 %d입니다.\n", 10, 20, 10+20);
11.    printf("%.1lf - %.1lf = %.1lf\n", 3.4, 1.2, 3.4-1.2);
12.
13.    return 0;
14. }
```

실행
결과

10

3.4000000

3.5

3.400000000000

10과 20의 합은 30입니다.

$3.4 - 1.2 = 2.2$

공부한 내용

- 프로그램이란
- 컴파일과 링크
- 메인 함수에 대하여
- printf 함수로 화면 출력 방법