
컴퓨터 프로그래밍 개론

프로그래밍 개요

Contents

I. C언어 개요

A. 역사

II. 프로그래밍 작성 및 실행 과정

A. HelloWorld

B. main()함수

C. 중괄호

D. 들여쓰기

E. 주석

F. printf()함수

G. 세미콜론

H. 전처리기

1. 라이브러리

프로그래밍 개요

▶ 프로그램(Program)이란

- ▶ 일반적으로는 어떤 작업이 나열된 계획표 또는 순서표를 의미
 - ▷ 워크샷의 프로그램, 주간 TV 방영 프로그램
- ▶ 컴퓨터에서 프로그램은 컴퓨터가 해야할 일이 순서대로 작성된 파일
 - ▷ 어떤 목적을 이룰 수 있도록 프로그래밍 언어로 진행 순서를 기술한 것
 - ▷ 작업할 내용을 컴퓨터가 이해할 수 있는 기계어로 만들어져 있음

```
011010010110011000100000011110010110111101110101001001110
111001001100101001000000111001001100101011000010110010001
10100101101110011001100100000011101000110100001101001011
100110010110000100000011110010110111101110101001001110111
001001100101001000000111011001100101011100100111100100100
000011000110110110001100101011101100110010101110010001011
100110100101100110001000000111100101101111011101010010011
101110010011001010010000001110010011001010110000101100100
011010010110111001100111001000000111010001101000011010010
111001100101100001000000111100101101111011101010010011101
110010011001010010000001110110011001010111001001111001001
000000110001101101100011001010111011001100101011100100010
111001101001011001100010000001111001011011110111010100100
111011100100110010100100000011100100110010101100001011001
000110100101101110011001110010000001110100011010000110100
101110011001011000010000001111001011011110111010100100111
011100100110010100100000011101100110010101110010011110010
010000001100011011011000110010101110110011001010111001000
101110
```

프로그래밍 개요

▶ 기계어(machine language)

- ▶ 0,1의 이진수로 구성된 언어

- ▶ 컴퓨터는 0과 1이라는 기계어를 인식 가능

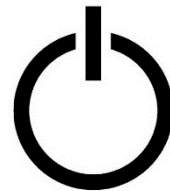
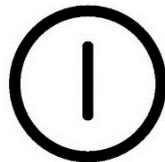
 - ▶ 컴퓨터의 CPU는 본질적으로 기계어만 처리 가능

- ▶ 자연어와 기계어 비교

 - ▶ 자연어 : 숫자 3을 컴퓨터로 읽어 들이시오.

 - ▶ 기계어 : 10011111 00000011

 - ▶ 사람이 모든 명령을 기계어로 작성할 수 없으므로 프로그래밍 언어로 작성 후 기계어로 해석



프로그래밍 개요

▶ 컴퓨터에서 정보 처리 방식

- ▶ 컴퓨터에서 이미지나 사운드 등의 모든 데이터들은 0과 1로 변경되어 처리됨
- ▶ 0과 1은 하드웨어에서 전기신호로 바뀜
 - ▶ 하드웨어의 전자 스위치들은 0을 OFF, 1을 ON상태로 간주



프로그래밍 개요

▶ 컴퓨터에서 정보 처리 방식

▶ 초기에는 인간이 10진수를 사용하기 때문에 10진수로 처리

▷ 컴퓨터 시스템에서 10진수를 사용하면 0~5V의 범위를 10개로 나누어서 사용해야 함

▷ 잡음에 민감해짐(내부 전자파, 정전기 등)

▶ 그러므로 하드웨어에서 2개의 전압 레벨을 구분하는 회로 설계가 효율적임

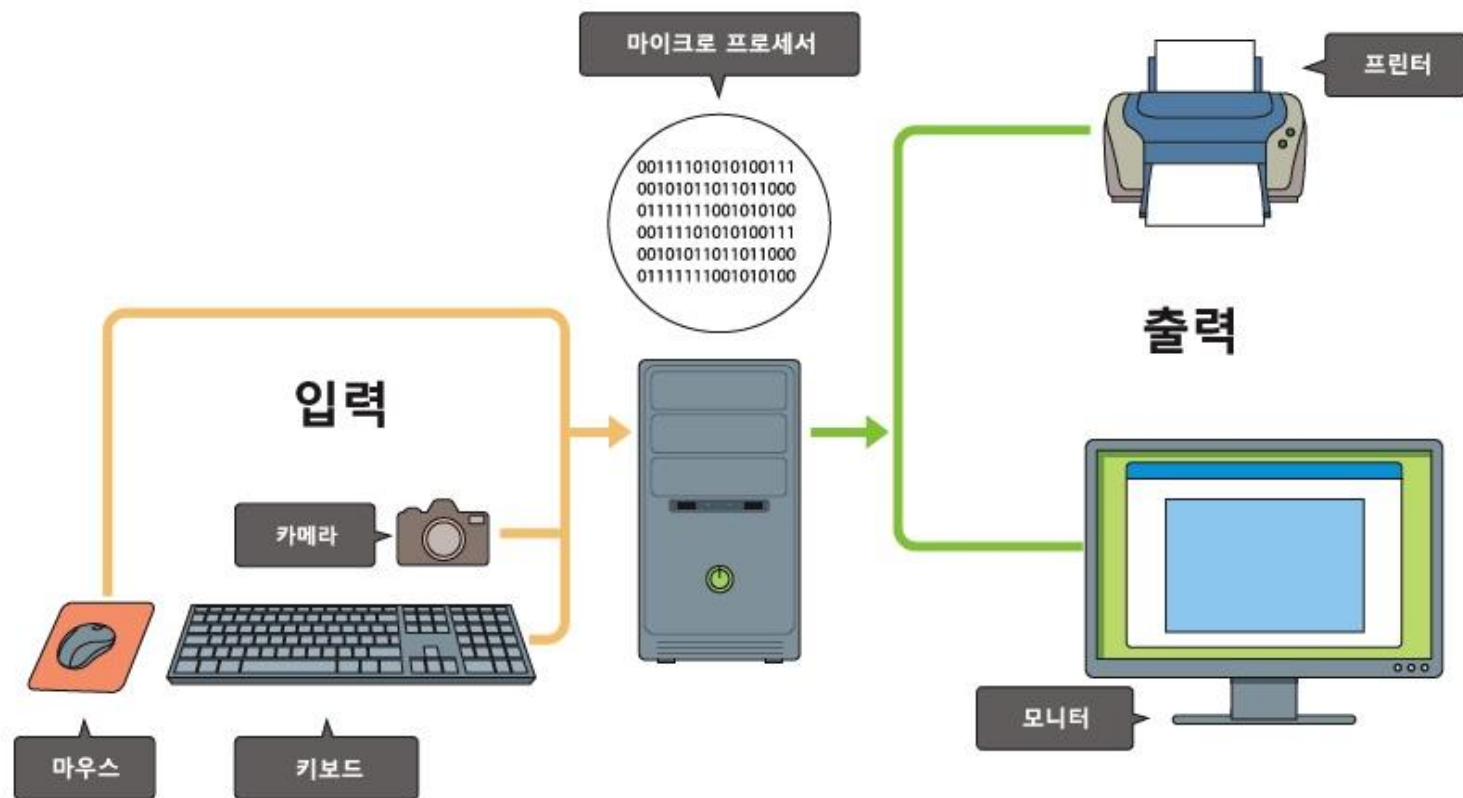
▷ 전자 공학 관점에서 전압이 0V와 0.5V를 구별하는 것보다 0V와 5V를 구별하는 것이 쉬움

전압	값
0.0	0
0.5	1
1.0	2
...	...

프로그래밍 개요

▶ 컴퓨터에서 정보 처리 방식

- ▶ 컴퓨터 내부의 모든 데이터는 1과 0의 신호로 저장, 전송, 처리
- ▶ 데이터가 이진수로 변환되어 처리되지 않으면 컴퓨터는 이해할 수 없음



프로그래밍 개요

▶ 비트(bit)와 바이트(byte)

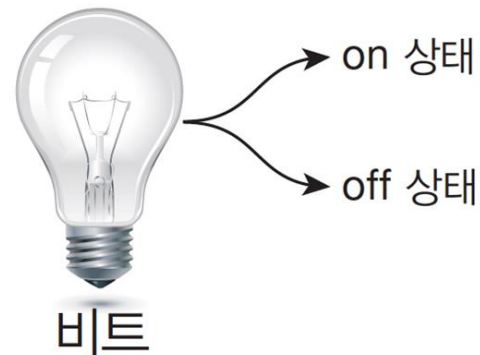
▶ 비트(Bit : binary digit)

- ▶ 컴퓨팅 세계의 표현법
- ▶ 2진수(0,1)를 저장할 수 있는 최소 메모리 공간
- ▶ 논리적으로는 참(true) 또는 거짓(false)을 나타냄
 - ▶ 데이터가 1인 경우에 전기신호를 보내고 0인 경우에는 차단

▶ 바이트(byte)

- ▶ 8개의 비트로 구성된 저장 단위
- ▶ 1byte = 8bit
- ▶ 실질적 의미는 ASCII 문자 하나의 크기를 뜻함

Bit = Binary Digit



1비트	2비트	3비트	...	n비트
$2^1=2$ 개	$2^2=4$ 개	$2^3=8$ 개	...	2^n 개



프로그래밍 개요

▶ 프로그래밍(Programming)이란

- ▶ 프로그램을 만드는 행위
- ▶ 단순히 코드를 작성하는 코딩(Coding)과는 다르게 시스템을 전반적으로 이해하고 설계하는 과정을 포함

▶ 프로그래밍 언어

- ▶ 계산을 위한 표현에 집중되어 있는 인공언어
 - ▷ 인간이 사용하는 한국어, 일본어, 영어 등은 자연어
- ▶ 자연어와 프로그래밍 언어의 차이를 이해하는 것이 프로그램을 이해하는데 매우 중요
 - ▷ 의미가 명확해야 함
 - ▷ 명확한 순서로 명령해야 함
 - ▷ <https://www.youtube.com/watch?v=I5cq54MFQCo>
- ▶ 용도에 따라 많은 종류가 있음
 - ▷ C언어 , C++언어, C#언어, Java언어, Python언어, Kotlin언어, Swift언어, Objective-C

프로그래밍 개요

▶ 컴퓨터 프로그램 종류

▶ 프로그래밍 언어로 프로그래밍한 작업 결과

▷ C프로그램, C++프로그램, C#프로그램, Java프로그램 ...

▶ 프로그래머

▶ 프로그래밍을 하는 사람

▷ C 프로그래머, C++ 프로그래머, C# 프로그래머, Java 프로그래머 ...

C언어 개요

▶ C언어란

- ▶ 1972년 미국 AT&T社の 벨(Bell)연구소에서 유닉스 운영체제(Unix Operating System) 개발
 - ▷ 켄 톰슨(Kenneth Lane Thompson), 데니스 리치(Dennis Ritchie)가 개발
- ▶ 이전의 B언어를 발전시킨 언어
- ▶ HW에 관계 없이 사용하기 위해 개발
 - ▷ 초기의 프로그래밍 언어는 하드웨어가 변경되면 새로 개발해야 함
- ▶ 컴퓨터 분야에서는 초기에 다양한 언어가 등장했지만 C언어는 지금까지 유지되면 가장 많이 사용되는 프로그래밍 언어



켄 톰슨

데니스 리치

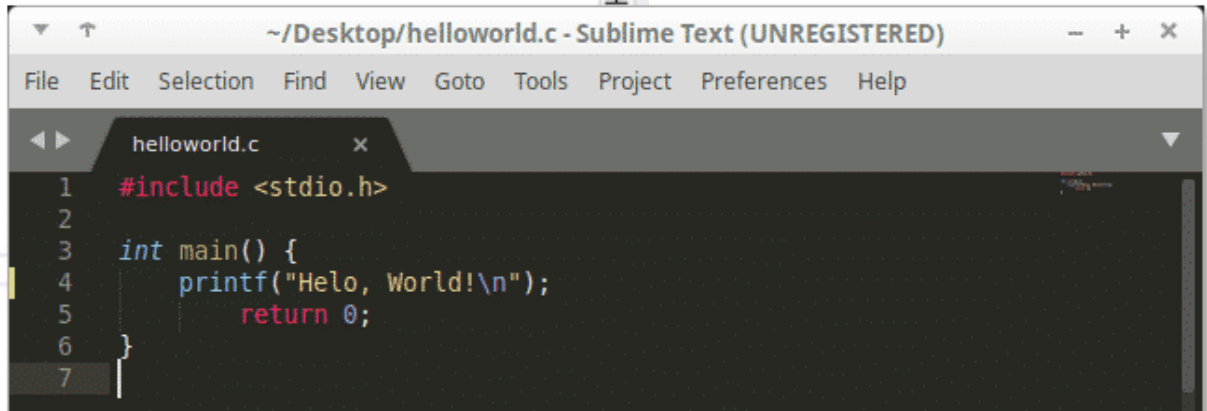
C언어 개요

▶ C언어의 특징

- ▶ 소스파일에 작성된 코드의 순서대로 실행되는 절차지향 언어
- ▶ 메모리에 직접 접근할 수 있으므로 하드웨어 제어 가능
 - ▷ 시스템의 밀접한 부분까지 접근할 수 있어서 HW를 제어하거나 접근하는 프로그래밍인 시스템 프로그래밍에 사용
- ▶ C언어로 프로그래밍을 작성하면 다른 운영체제에서 거의 그대로 사용 가능
 - ▷ 약간의 수정이 필요한 경우도 있으나 대체로 이식성이 좋음
 - ▷ 윈도우에서 프로그래밍한 코드를 유닉스에서 실행 가능



```
hello.c  [icon] X
hello (전역 범위)
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, world!\n");
6
7     return 0;
8 }
9
```



```
~/Desktop/helloworld.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
helloworld.c X
1 #include <stdio.h>
2
3 int main() {
4     printf("Helo, World!\n");
5     return 0;
6 }
7
```

C언어 개요

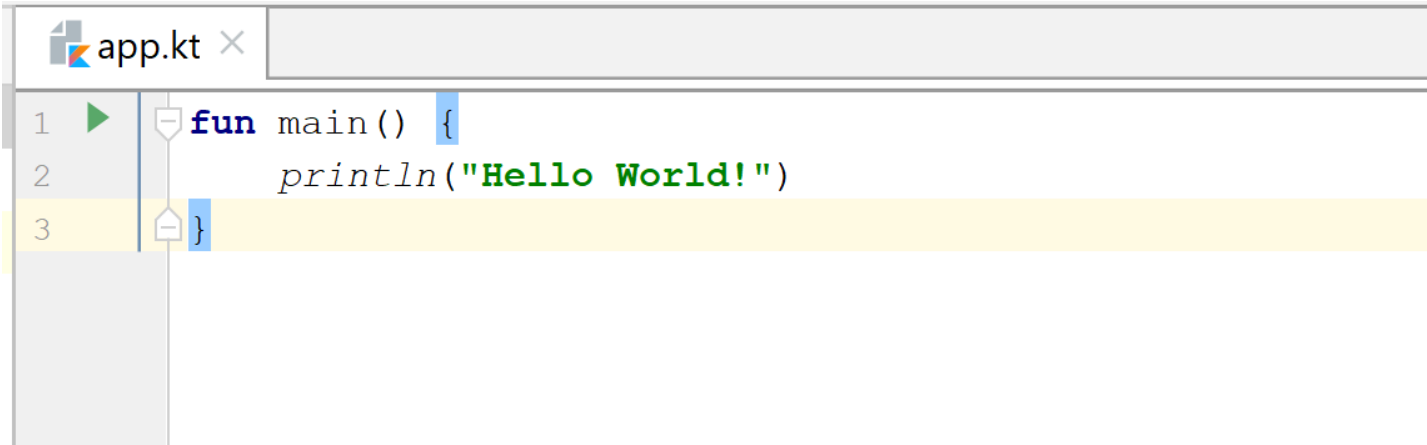
▶ C언어의 특징

- ▶ 모든 언어의 기본이기 때문에 C언어를 학습한 후 다른 언어를 배우기가 수월해짐



A screenshot of a Java IDE window titled 'HelloWorld.java'. The code is as follows:

```
1  
2 public class HelloWorld {  
3     public static void main(String arg[]){  
4         System.out.println("Hello, World!!");  
5     }  
6 }  
7
```



A screenshot of a Kotlin IDE window titled 'app.kt'. The code is as follows:

```
1 ▶ fun main() {  
2     println("Hello World!")  
3 }
```

프로그래밍 개발 과정

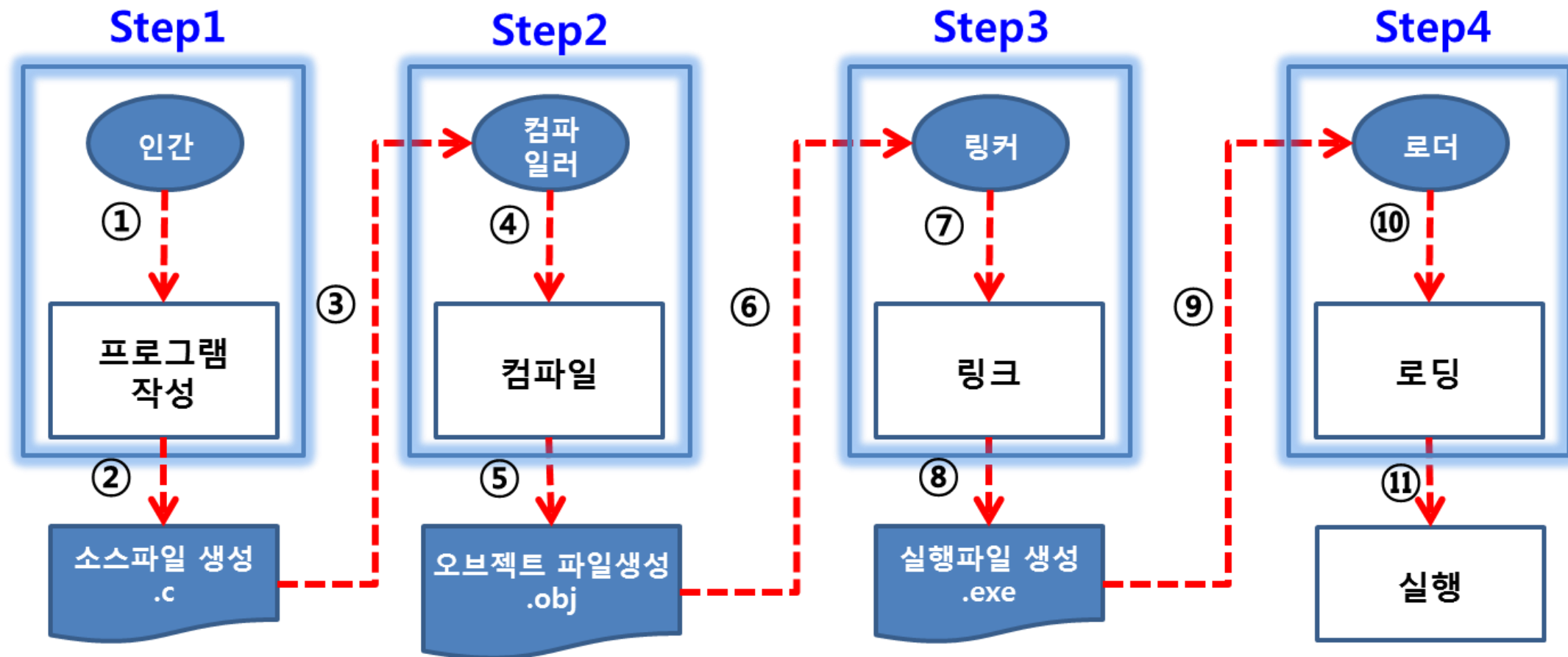
▶ 소스파일(Source File) 작성

- ▶ 다양한 종류의 프로그래밍 언어로 코드를 작성할 수 있으며 그 결과물로 소스파일이 생성됨
- ▶ 소스 파일은 사람이 내용을 이해하거나 수정이 가능하지만 소스코드 상태로는 컴퓨터에서 바로 실행 불가
 - ▷ 컴퓨터는 0과 1로 된 신호만 이해하기 때문에 소스 파일의 빌드(Build) 과정이 필요
 - ▷ 빌드 : 텍스트로 된 소스코드가 실행 가능한 프로그램 파일로 변하는 과정
- ▶ 비주얼스튜디오에서 아래 코드를 작성
 - ▷ 코드에 오타가 포함되어 있으면 컴파일 과정에서 에러 발생
 - ▷ 가독성을 높이기 위해 들여쓰기(Tab/4칸 들여쓰기) 사용

```
#include<stdio.h>
int main(void)
{
    printf("HelloWorld \n");
    return 0;
}
```

프로그래밍 개발 과정

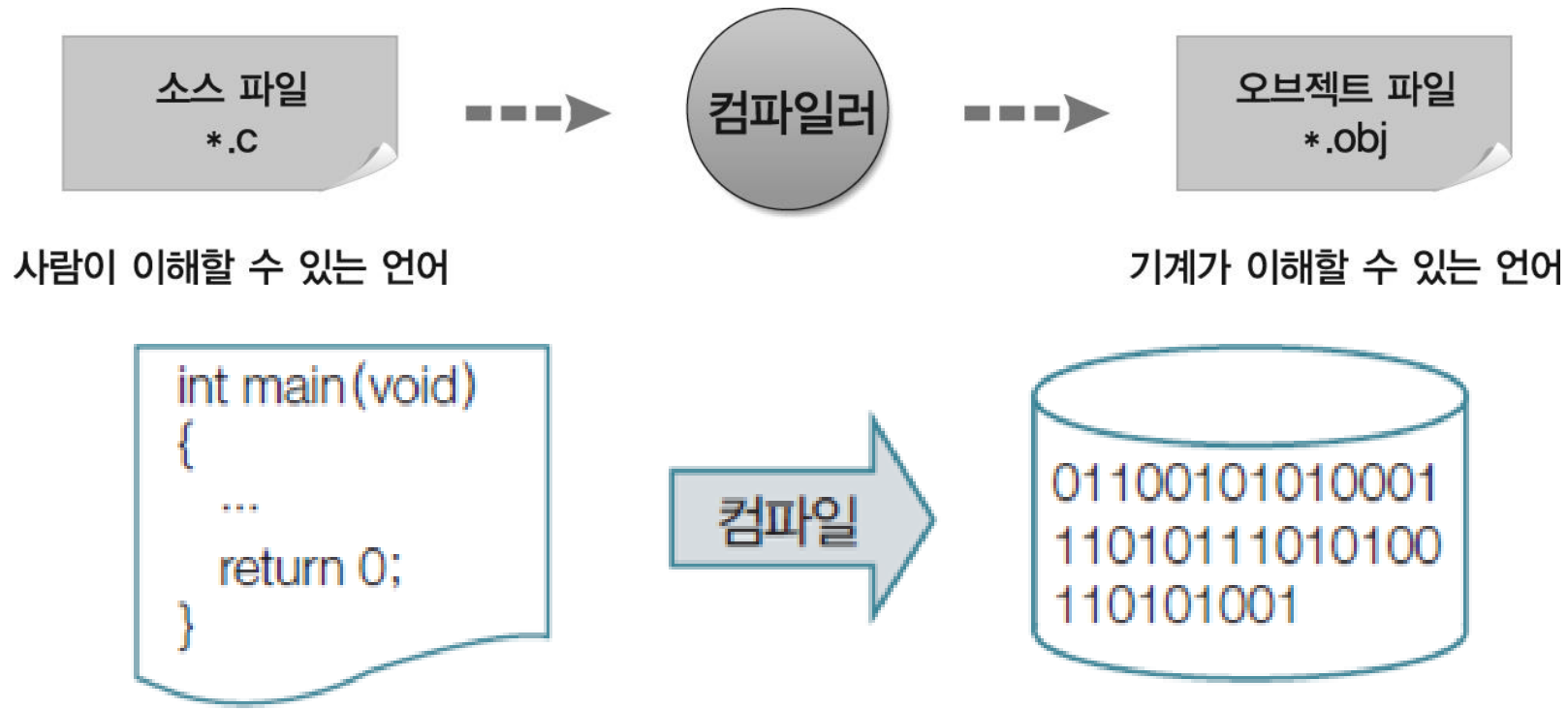
▶ 프로그래밍 4단계



프로그래밍 개발 과정

▶ 컴파일(Compile)

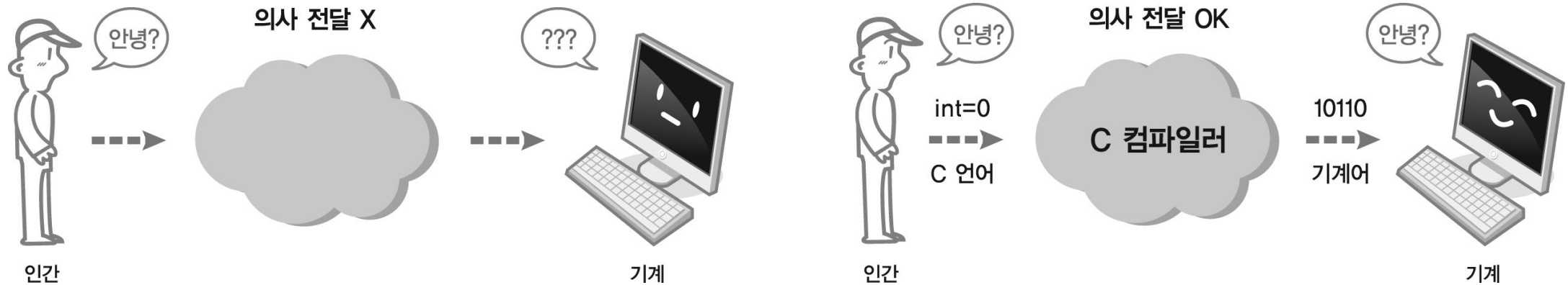
- ▶ C 언어로 된 소스파일을 컴파일러를 통하여 기계어로 바꾸는 과정
- ▶ 컴파일 과정이 정상적으로 수행되면 .obj 파일이 생성됨
 - ▶ 기계어로 번역된 오브젝트 파일(.obj)은 링커(linker)에 의해 링크가 되기 전까지는 실행할 수 없음



프로그래밍 개발 과정

▶ 컴파일러(Compiler)

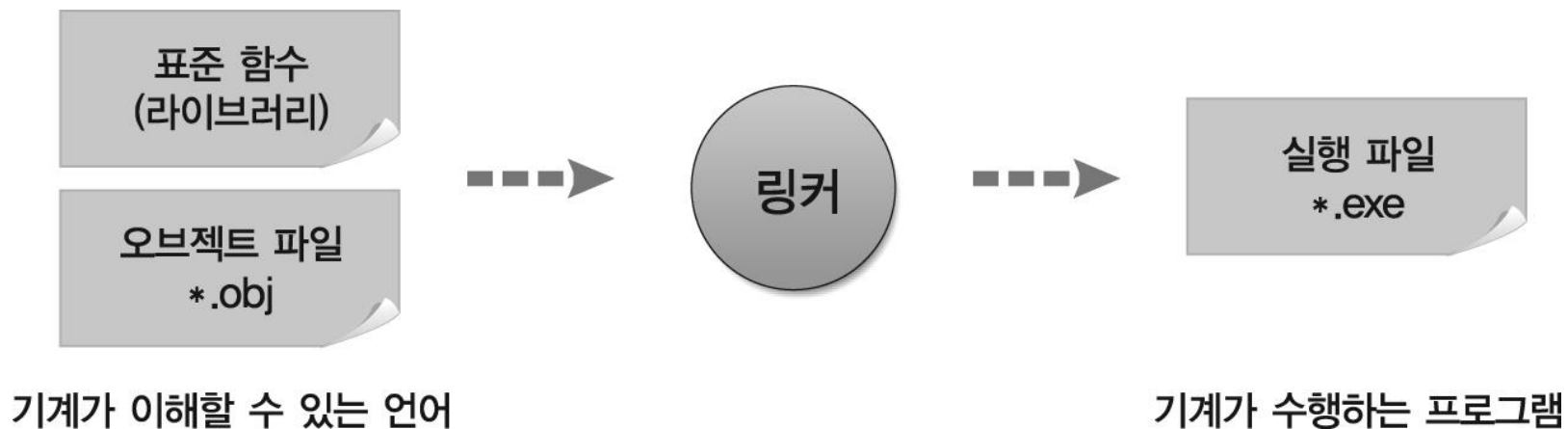
- ▶ 인간과 컴퓨터 사이에서 통역관의 역할
- ▶ 인간이 만든 프로그램을 기계가 이해 하도록 기계어로 변환하는 변환기
 - ▷ 기계어 : 기계가 이해하는 2진 숫자(0과 1)로 작성된 언어
- ▶ 동일한 코드도 번역 방법에 따라 실행속도가 달라지므로 컴파일러는 프로그래밍 더 빨리 동작할 수 있도록 최적화하여 번역을 실행



프로그래밍 개발 과정

▶ 링크 (link)

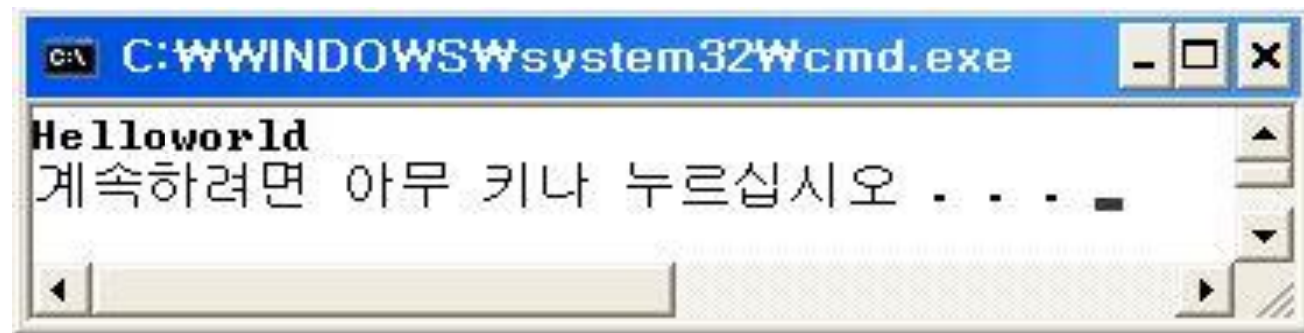
- ▶ 링커(linker)가 오브젝트 파일과 프로그래밍에 사용된 라이브러리를 연결하여 실행파일을 생성
 - ▶ 정상적인 실행을 위하여 OS에서 인식할 수 있는 실행파일 형태로 변경하는 과정
- ▶ 링크과정이 정상적으로 진행되면 .exe파일이 생성됨



프로그래밍 개발 과정

▶ 로딩(loading)

- ▶ 로더(loader)가 실행파일을 주기억 장치(RAM)에 로딩(적재)하여 CPU에서 실행될 수 있도록 하는 과정
 - ▶ 컴퓨터에서 모든 프로그램은 CPU에서 실행되기 위하여 RAM에 로딩되는 과정이 필요함

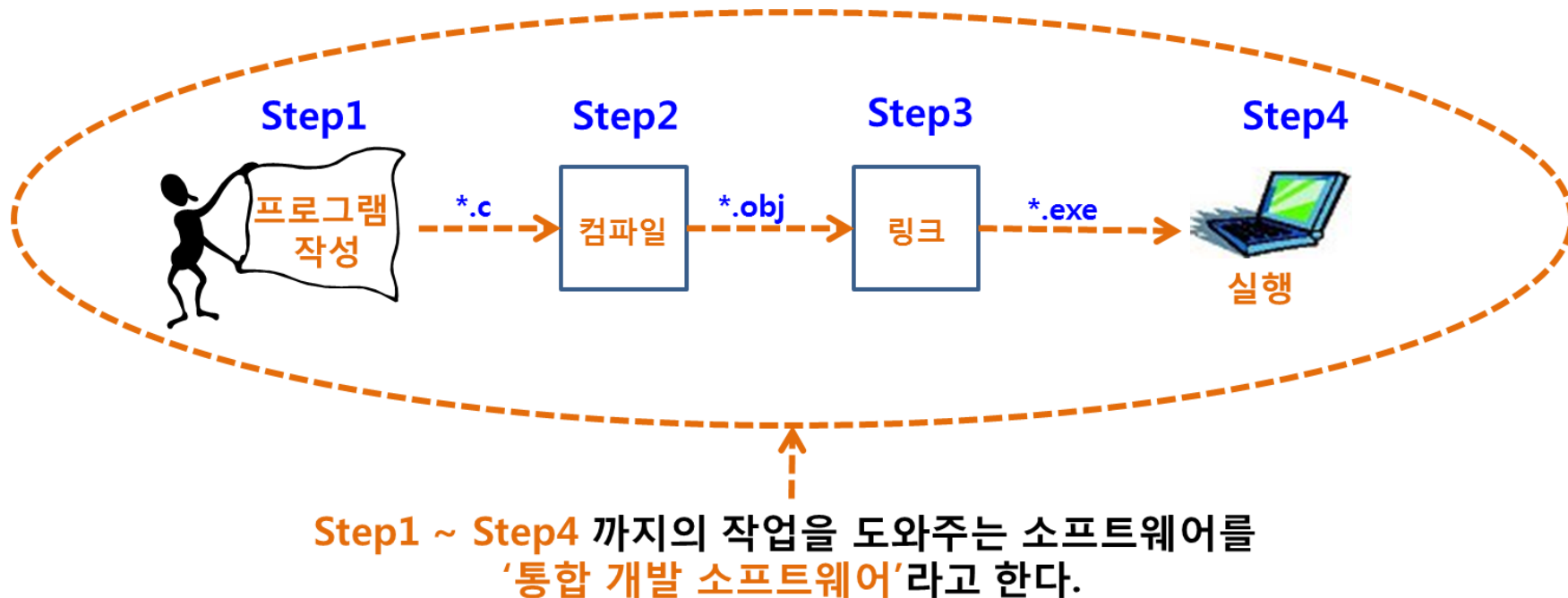


프로그래밍 개발 과정

▶ 통합 개발 소프트웨어(또는 통합 개발 환경)

▶ IDE(Integrated Development Environment)

▶ 비주얼 스튜디오와 같이 프로그램 작성부터 컴파일, 링크, 실행까지 가능하게하는 프로그램을 지칭



프로그래밍 개발 과정

▶ CPU(Central Processing Unit)

- ▶ 컴퓨터 내에서 다양하고 많은 양의 계산식을 아주 빠른 속도로 계산
 - ▷ 일반적으로 사용하는 범용 컴퓨터에는 반드시 CPU가 탑재되어 있음
- ▶ 대부분의 CPU는 2~3GHz(기가헤르츠)정도의 속도를 가짐
 - ▷ G(giga)는 10억을 나타내는 접두사(接頭辭)
 - ▷ Hz는 1초에 몇 번 동작을 할 수 있는지를 나타내는 단위
 - ▷ 2~3GHz의 CPU는 1초에 약 20~30억 번의 계산을 실행
- ▶ CPU 내부에는 계산된 결과를 임시로 저장하는 공간이 있지만 너무 작아서 모든 계산 결과를 저장할 수 없음



프로그래밍 개발 과정

▶ 저장장치의 종류

▶ 디스크(Disk)

- ▶ 가격이 저렴(대용량), 속도가 느림
- ▶ 비휘발성(non-volatile)
 - ▶ 전원이 꺼져도 데이터 유지되므로 저장의 목적으로 사용



▶ 램(RAM:Read Only Memory)

- ▶ 속도가 빠름, 가격이 비쌈(소용량)
- ▶ 휘발성(volatile)
 - ▶ 전원이 꺼지면 데이터 삭제되기 때문에 실행 중에 처리할 데이터를 임시저장하여 사용
- ▶ 프로그래밍에서는 메모리에 변수라는 이름으로 공간을 만들어서 사용
 - ▶ 자료형(DataType)과 이름이 필요
 - 저장할 자료의 최대 크기와 어떤 형식의 데이터를 입력할지 미리 시스템에게 알려줌
 - 이는 프로그램에서 필요한 공간을 예약하고 효율적으로 처리하기 위함

프로그래밍 개발 과정

▶ 메모리(Memory)

▶ 2~16GB 정도의 메모리가 일반적

▷ 1바이트는 8bit와 같으며 8자리의 2진수로 표현(0~255까지의 범위를 표현)

▶ 메모리가 많을수록 많은 계산결과를 기억할 수 있음

▷ 계산 결과를 기억하는 것은 같은 계산식을 반복하지 않아도 된다는 의미

▷ 한번 계산된 결과를 메모리에 저장한 후 재활용

▷ 컴퓨터의 속도 향상과 관련이 있음

▶ 휘발성

▷ 전원이 꺼지면 저장된 데이터는 사라짐

DDR



DDR2



DDR3



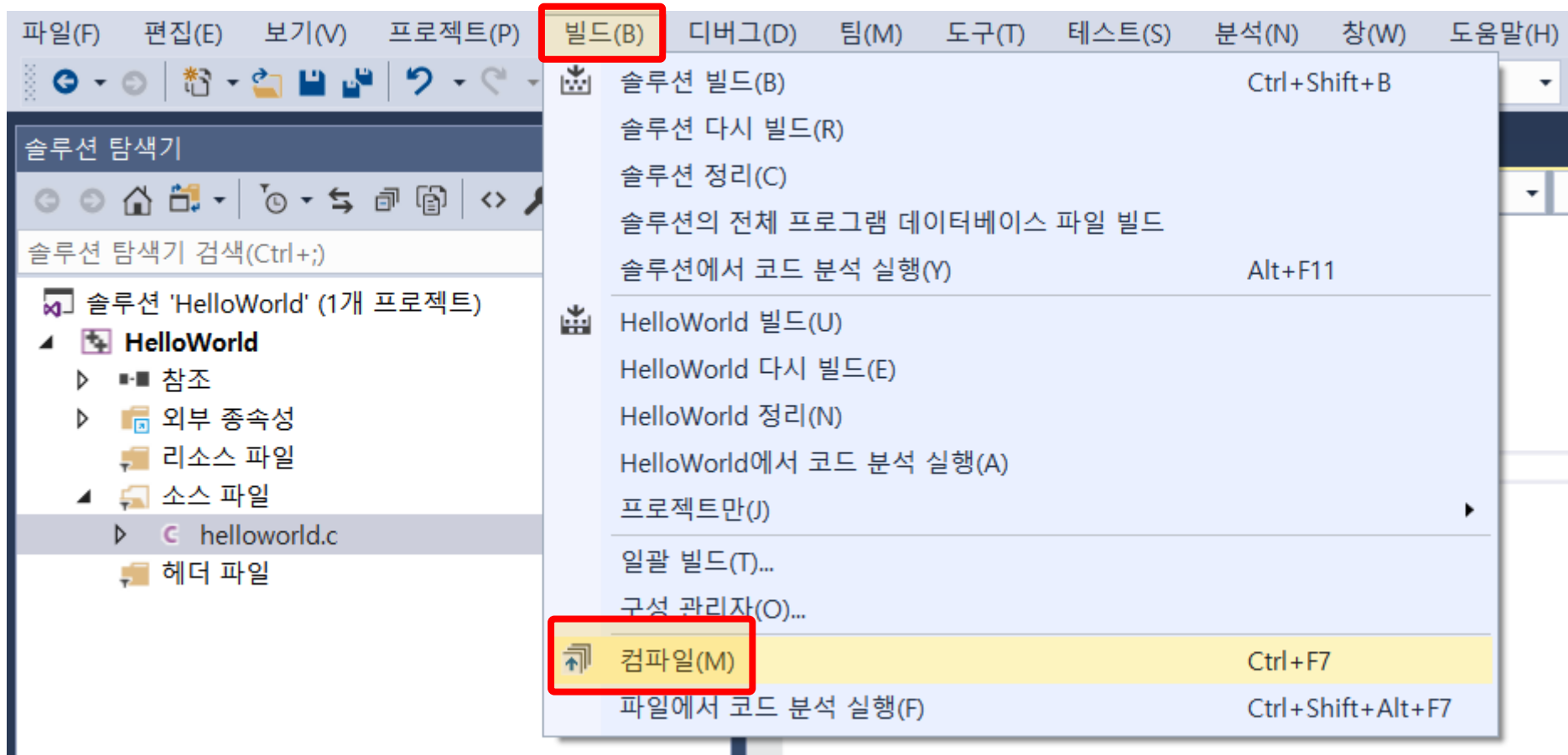
DDR4



프로그래밍 개발 과정

▶ 컴파일 실습

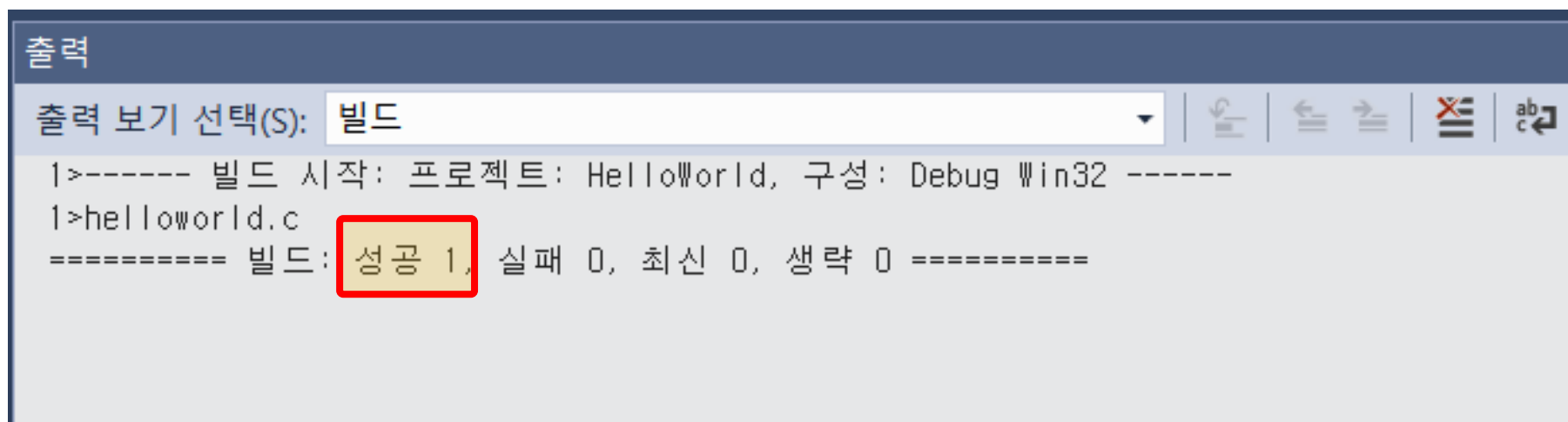
▶ 오브젝트 파일 (helloworld.obj)을 생성



프로그래밍 개발 과정

▶ 컴파일 성공

- ▶ 비주얼 스튜디오의 출력창에 아래와 같이 성공 메시지를 확인



The screenshot shows the '출력' (Output) window in Visual Studio. The '출력 보기 선택(S):' (Output view selection) dropdown is set to '빌드' (Build). The output text shows a successful build for the project 'HelloWorld' in 'Debug Win32' configuration. The message '=====
1>----- 빌드 시작: 프로젝트: HelloWorld, 구성: Debug Win32 -----
1>helloworld.c
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====' is displayed. The word '성공' (Success) is highlighted with a red box.

```
출력  
출력 보기 선택(S): 빌드  
1>----- 빌드 시작: 프로젝트: HelloWorld, 구성: Debug Win32 -----  
1>helloworld.c  
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

프로그래밍 개발 과정

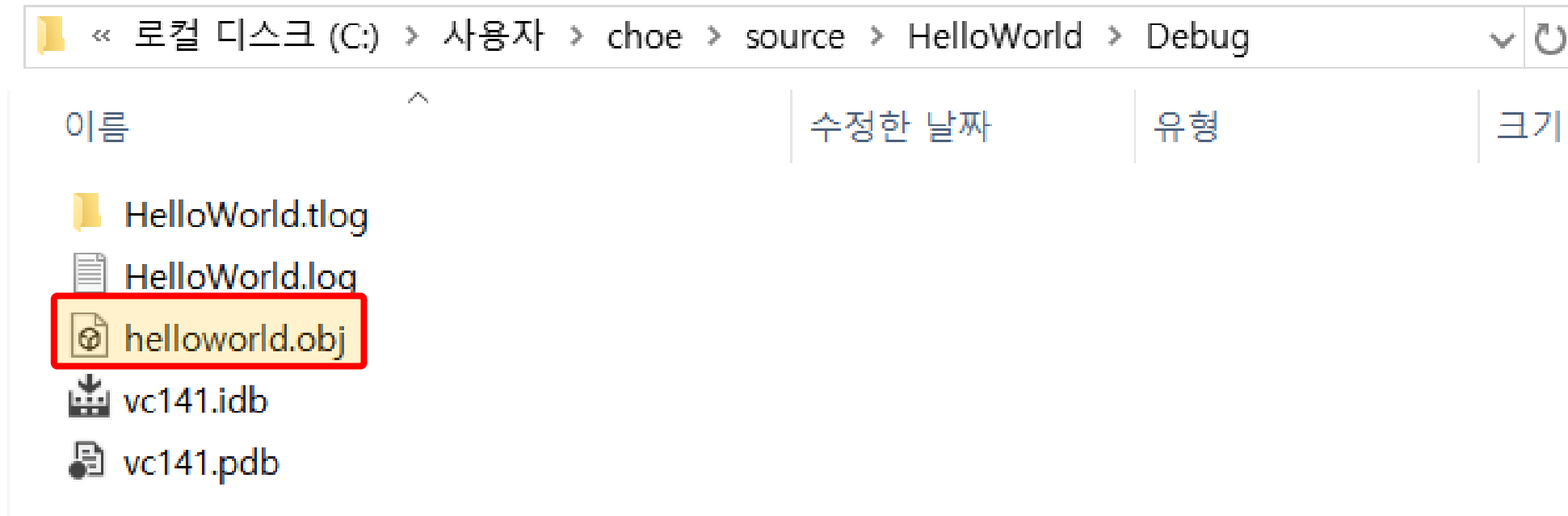
▶ 컴파일 된 오브젝트 파일 확인

▶ 컴파일을 성공하면 오브젝트 파일(.obj)이 생성됨

▷ 아래 그림의 위치 또는 C:\Users\choe\source\repos\HelloWorld\Debug 에 오브젝트 파일이 생성

▷ “choe”는 컴퓨터에 설정된 사용자 이름이므로 PC마다 다름

▷ 오브젝트 파일은 기계어로 구성

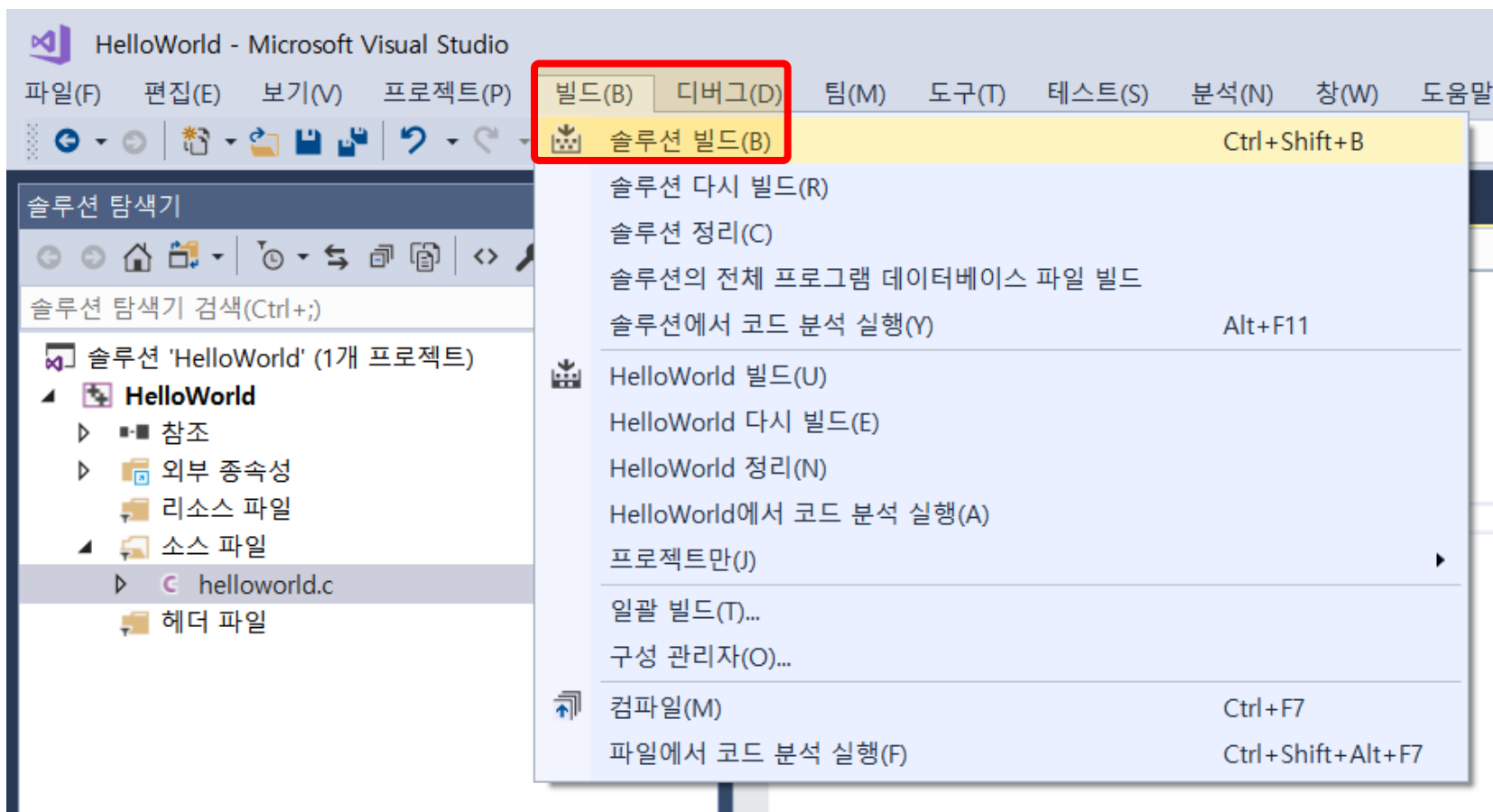


프로그래밍 개발 과정

▶ 프로그램 링크

▶ 실행 파일 (HelloWorld.exe) 생성하는 과정

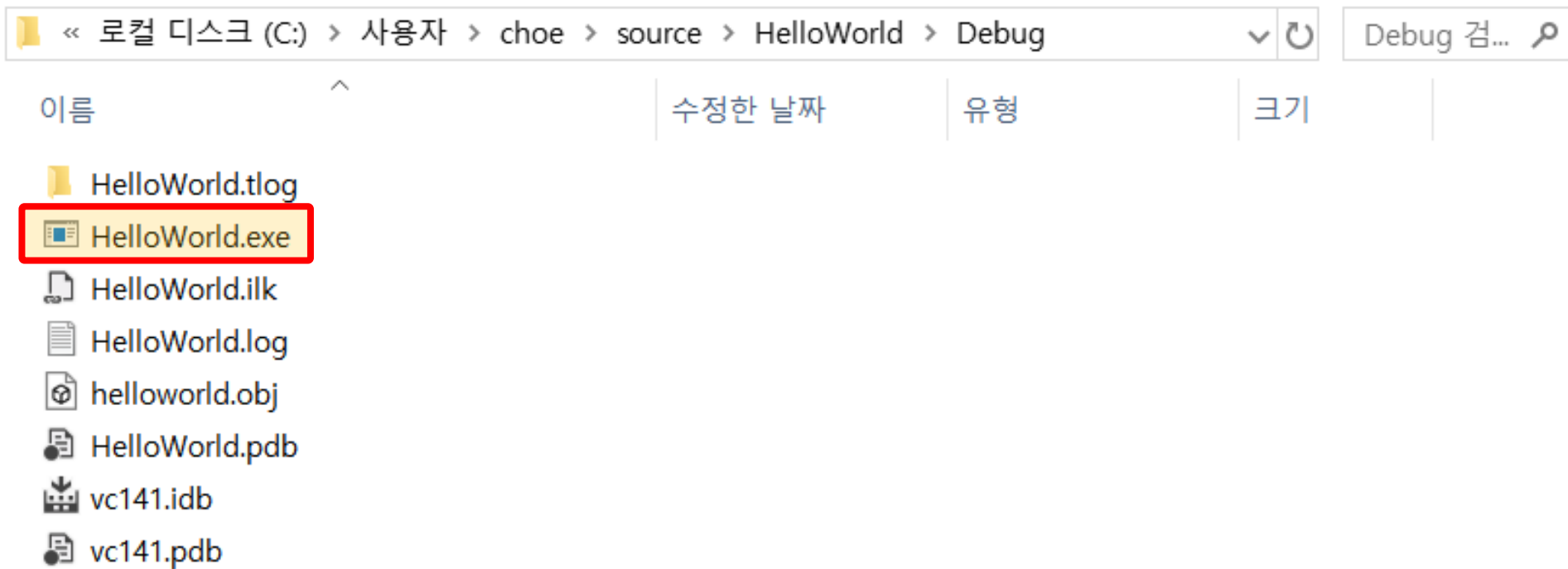
▶ 앞서 컴파일을 미리 했기때문에 링크만 수행되며 만일에 소스코드가 수정되었거나 컴파일되지 않았다면 컴파일과 링크가 동시에 수행



프로그래밍 개발 과정

▶ 링크된 실행파일 확인 후 실행하기

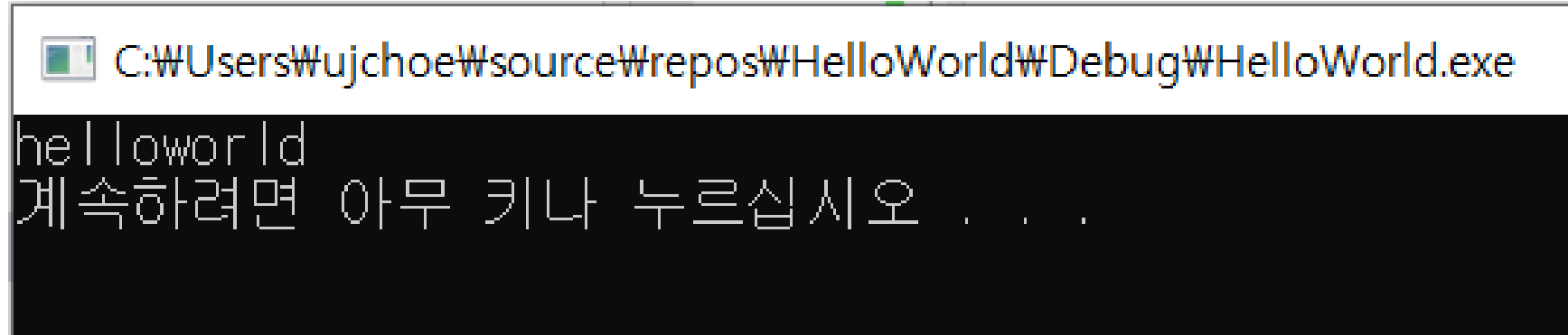
▶ 링크 과정 후에 생성된 실행 파일 (HelloWorld.exe)을 실행



프로그래밍 개발 과정

▶ 실행 확인

- ▶ 콘솔 창이 나타나며 helloworld가 출력됨



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\Wujchoe\source\repos\HelloWorld\Debug\HelloWorld.exe. The command prompt displays the output 'helloworld' on the first line and '계속하려면 아무 키나 누르십시오 . . .' (Press any key to continue) on the second line.

```
C:\Users\Wujchoe\source\repos\HelloWorld\Debug\HelloWorld.exe  
helloworld  
계속하려면 아무 키나 누르십시오 . . .
```

HelloWorld 출력

▶ 소스코드 수정

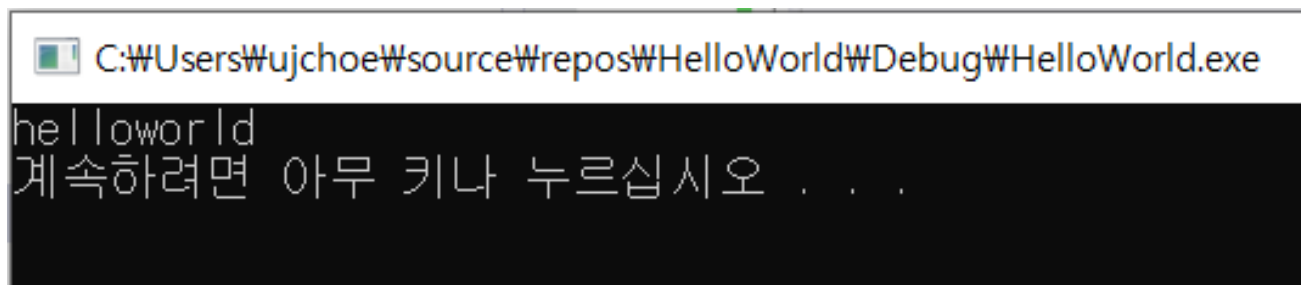
▶ 실행화면이 금방 사라질 경우 아래와 같이 소스코드 수정 후 다시 빌드

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

int main(void) {
    printf("helloworld \n");
    system("pause");
    return 0;
}
```

▶ 실행파일을 더블 클릭하여 다시 실행

- 📁 HelloWorld.tlog
- 📄 HelloWorld.exe
- 📄 HelloWorld.ilc
- 📄 HelloWorld.log



```
C:\Users\Wujchoe\source\repos\HelloWorld\Debug\HelloWorld.exe
helloworld
계속하려면 아무 키나 누르십시오 . . .
```

Q & A