



인하공업전문대학
INHA TECHNICAL COLLEGE

C 프로그래밍(1주차)

인하공업전문대학 컴퓨터 정보과
김한결 강사

강의 진행 안내

❖ 이론/실습 병행 진행

- 실습 강의실 : 4-403

❖ 교재

- 서명 : C언어 일취월장 2판
- 저자 : 신윤환
- 출판사: 한빛아카데미(주)
- 기타 참고 자료 활용(C언어 for Beginner 4판)

❖ 평가

❖ 출석 : 20% , 과제물 : 20 % , 중간 : 30%, 기말 : 30% (합 100%)

❖ 연락처

▪ 010-2314-4882, sonnonet@sonnonet.com



강의 진행 안내 - 2

제4조(출석 인정) ① 강의담당교원은 다음 각 호의 1에 해당하는 학생을 출석 인정한다.

1. 천재지변 등 불가항력의 사유로 인하여 출석하지 못한 경우 <개정 2016.11.16., 2019.11.25.>
2. 공적 의무 또는 공권력의 행사로 인하여 출석하지 못한 경우
3. 학과장의 허가를 받아 대학을 대표하는 행사참여로 출석하지 못한 경우
4. 조기취업인 경우 <신설 2016.11.16.>
5. 경조사로 인하여 출석하지 못한 경우 <개정 2016.11.16., 2019.11.25.>

구분	대상	인정일수 (휴일포함)
결혼	본인	5
	자녀, 형제자매, 부모의 형제자매	1
사망 (본인 및 배우자 기준)	배우자, 본인 및 배우자의 부모	5
	조부모, 형제자매 및 그의 배우자, 자녀 및 자녀의 배우자	3
	부모의 형제자매	1

※ 경조사 출결 인정일수는 경조사 발생일 이후부터 산정하여 인정한다.

6. 법정전염병 또는 질병이나 사고로 입원하여 출석하지 못한 경우 <신설 2019.11.25.>

구분	증빙서류	인정일수 (휴일포함)
법정전염병	병원에서 발행한 진단서	법정전염병 격리기간
질병 및 사고 입원	병원에서 발행한 입퇴원확인서 및 진단서	2주 이내 (학기별 1회)

7. 그 밖에 총장이 필요하다고 허가하는 경우

② 위 4조의1항에 의거하여 출석의 인정을 원하는 학생은 사전 또는 사유 발생일로부터 7일 이내에 해당 증빙서류를 첨부한 출석인정원을 제출해야 한다. (단, 6호의 경우, 법정 전염병 격리해지일 또는 퇴원일로부터 7일 이내에 제출해야 한다.) <개정 2019.11.25.>

③ 제1항4호의 세부사항은 별도의 규정을 정하여 시행한다. <신설 2016.11.16.>

강의 진행 안내 - 3

1. 수업대체일

가. 2024-1학기 휴업일에 대한 수업대체일 우선 지정 운영

일자	내용	보강구분	일수
04.10.(수)	국회의원 선거일	보강진행	1
04.26.(금)	개교기념일(대체휴일)	보강진행	1
05.06.(월)	어린이날대체공휴일	보강진행	1
05.15.(수)	부처님오신날	보강진행	1
06.06.(목)	현충일	보강진행	1
계			5

강의 진행 안내 - 4

C 언어 기초	1장	C 언어의 개요 : C 언어 소개, C 프로그램 개발 과정
	2장	C 프로그램 작성과 실행 : 실습 환경 구축, C 프로그램의 구성요소, 오류 처리
	3장	표준 입출력 함수 : 표준 출력 함수, 표준 입력 함수, 문자 코드(아스키코드)
	4장	변수와 자료형 : 변수, 자료형, 상수, 자료형 변환
	5장	연산자 : 산술/대입/증감/관계/논리/조건/비트/주소 참조 연산자
	6장	조건문 : 제어문, if 문, switch 문
	7장	반복문 : for 문, while 문, do~while 문, 분기문
	8장	함수 : 라이브러리 함수, 사용자 정의 함수, 순환 함수
C 언어 심화	9장	함수와 변수 : 지역 변수, 전역 변수, 동적 변수, 정적 변수
	10장	배열 : 배열 선언과 초기화, 배열 활용, 다차원 배열
	11장	포인터 기초 : 포인터 사용법, 포인터 연산, 포인터 배열
	12장	포인터 활용 : 배열/함수/이중/void형 포인터, 포인터를 활용한 정렬과 탐색
	13장	문자와 문자열 : 문자/문자열 처리, 문자/문자열 함수, 배열로 문자열 다루기
	14장	구조체 : 구조체 개념과 활용, 공용체, 열거형
	15장	스트림과 파일 입출력 : 스트림 개념, 파일 입출력
	16장	동적 메모리와 연결 리스트 : 동적 메모리 할당과 반납, 연결 리스트 구조
	17장	선행처리기와 다중 소스 파일 : 선행처리기, 매크로, 조건 컴파일, 다중 소스 파일
부록	워크북	중간/기말고사 대비 워크북 6회차

강의 진행 안내 - 5

주	해당 장	주제
1	01장, 02장	C 언어 개요와 개발 환경 구축 및 프로그램 작성과 실행 방법
2	03장	프로그램 기본 문법 작성법
3	04장	변수의 개념과 자료형의 종류 및 자료형 변환
4	05장	수식의 개념과 연산자의 종류 및 우선순위
5	06장	제어문의 종류와 사용 방법
6	07장	반복문의 종류와 사용 방법
7	08장	라이브러리 함수와 사용자 정의 함수
8	시험	중간고사
9	09장	지역 변수와 전역 변수의 경계 및 동적 변수와 정적 변수 사용 방법
10	10장	배열의 선언과 초기화 및 배열의 활용 방법
11	11장	포인터 개념과 포인터 사용 방법
12	12장	포인터 활용 방법
13	13장	문자와 문자열 구분 방법 및 문자열 입출력 함수
14	14장, 15장	구조체와 스트림 및 파일 입출력 방법
15	16장, 17장	동적 메모리와 연결 리스트 및 선행처리기와 다중 소스 파일
16	시험	기말고사

강의 평가 안내

❖ 지식

- 구조적 프로그래밍 언어 기본 문법
- 구조적 프로그래밍 언어 문법 오류 -> 디버깅
- 제어구조, 하향식 설계 개념

❖ 기술

- 컴파일러 사용 능력
- IDE(Integrated Development Environment) 도구 활용 능력
- 구조적 프로그래밍 언어 디버깅 기법, 코드 검토 기술

- 동영상 참고 교육 (1시간)
- 연구실안전교육시스템 <http://edu.labs.go.kr>
- 소화기 사용방법
- 자동세제동기 사용방법 (AED)
- 심폐소생술 방법

Chapter 01

C 언어의 개요

01 사용자와 프로그램

02 C 언어를 배우는 이유

03 C 프로그램의 개발 과정

학습목표

- 프로그램과 프로그래밍 언어의 개념을 이해합니다.
- 프로그래밍 언어의 종류를 알아봅니다.
- C 언어의 유래와 특징을 알아봅니다.
- C 프로그램의 개발 과정을 알아봅니다.

1. 프로그램이란?

■ 프로그램

- 컴퓨터를 실행하기 위해 차례대로 작성한 명령어의 모음
- 컴퓨터에서 어떤 문제를 해결하기 위해
처리하는 방법과 순서를 기술한 명령문의 집합체
- 사람이 직접 하기 힘든 작업을 효율적으로 수행

1. 프로그램이란?

■ 프로그래밍

- 프로그래밍 언어를 이용하여 프로그램을 만드는 과정
- 컴퓨터에게 작업을 처리하도록 명령하는 작업지시서
- 프로그래밍 = 컴퓨터 하드웨어 + 소프트웨어 + 프로그래머

➤ 프로그래밍 필요 요건



3. 프로그램 언어의 종류

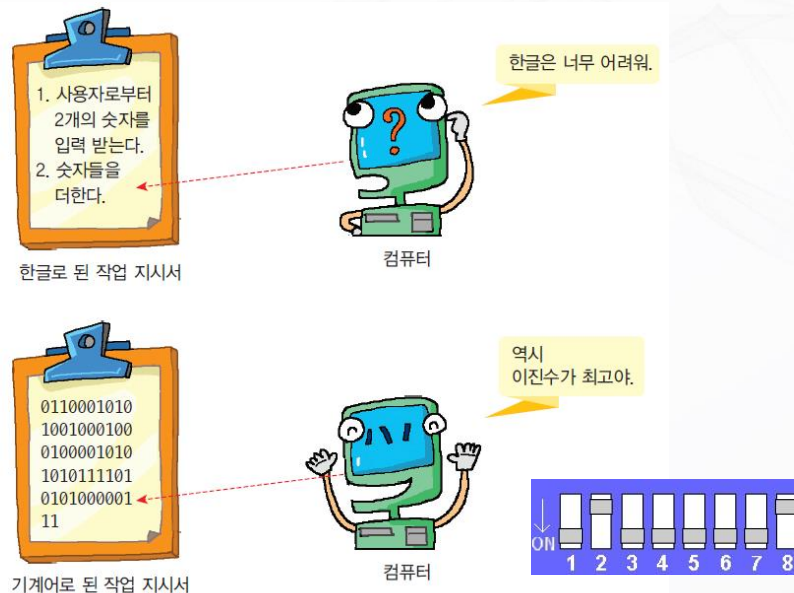
■ 프로그래밍 언어 구분 기준 : 사람과 컴퓨터가 이해하는 수준

- 저급 언어 – 컴퓨터가 쉽게 해석할 수 있는 명령어로 구성된 언어
 - 기계어 : 이진수 0과 1로 명령어를 표현한 언어이며 컴퓨터 하드웨어를 설계할 때 주로 사용합니다.
 - 어셈블리어 : 기계어가 사용하기 어렵고 오류가 자주 발생하는 등의 불편함을 개선하기 위해 개발한 프로그래밍 언어입니다.
- 컴퓨터가 알아듣는 언어는 한가지이다. 즉 0과 1로 구성되어 있는 “001101110001010...”과 같은 기계어이다.
- 컴퓨터는 모든 것을 0과 1로 표현하고 0과 1에 의하여 내부 스위치 회로들이 ON/OFF 상태로 변경되면서 작업을 한다.

3. 프로그램 언어의 종류

■ 프로그래밍 언어 구분 기준 : 사람과 컴퓨터가 이해하는 수준

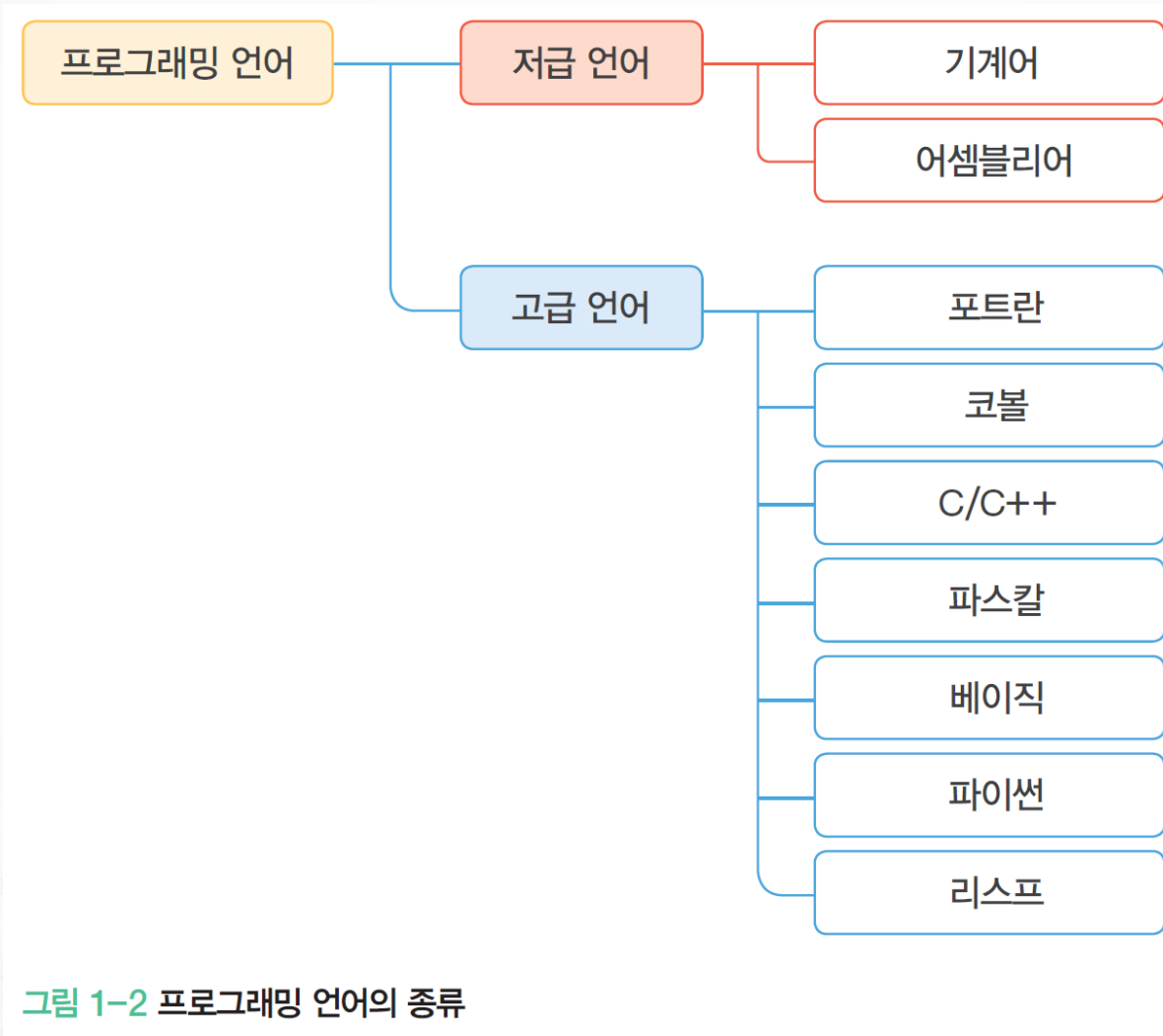
- 저급 언어 - 컴퓨터가 쉽게 해석할 수 있는 명령어로 구성된 언어
 - 기계어 : 이진수 0과 1로 명령어를 표현한 언어이며 컴퓨터 하드웨어를 설계할 때 주로 사용합니다.
 - 어셈블리어 : 기계어가 사용하기 어렵고 오류가 자주 발생하는 등의 불편함을 개선하기 위해 개발한 프로그래밍 언어입니다.
- 컴퓨터는 모든 것을 0과 1로 표현하고 0과 1에 의하여 내부 스위치 회로들이 ON/OFF 상태로 변경되면서 작업을 한다.



3. 프로그램 언어의 종류

- 고급 언어 - 사람이 쉽게 해석할 수 있는 명령어로 구성된 언어
 - 포트란 : 과학용 수치 계산에 적합한 프로그래밍 언어입니다.
 - 코볼 : 사무용 작업을 수행하기 위해 개발한 프로그래밍 언어로, 프로그램을 개발할 때 구어체 문장으로 명령어를 기술합니다.
 - C 언어 : 명령어를 차례대로 수행하는 순차적 프로그래밍 언어입니다.
 - 파스칼 : 교육용 언어로 개발한 프로그래밍 언어입니다.
 - 베이직 : 기억하기 쉬운 회화형으로 이루어진 PC용 프로그램 개발 언어입니다.
 - 파이썬 : 문법이 간결하고 표현 구조가 인간의 사고 체계와 닮아 초보자도 쉽게 배울 수 있는 언어입니다. 웹뿐만 아니라 데이터 분석, 머신 러닝, 그래픽 등 여러 분야에서 활용할 수 있어 생산성이 높습니다.
 - 리스프 : 함수에 의해서만 기능을 수행하는 프로그래밍 언어입니다.

3. 프로그램 언어의 종류



3. 프로그램 언어의 종류

```
1  #include<stdio.h>
2
3  int main() {
4      int a = 1;
5      int b = 2;
6      int c = a + b;
7
8      printf("%d", c);
9  }
```

C언어



```
1  push    ebp
2  mov     ebp,esp
3  sub     esp,0E4h
4  push    ebx
5  push    esi
6  push    edi
7  lea     edi,[ebp+FFFFFF1Ch]
8  mov     ecx,39h
9  mov     eax,0CCCCCCCCh
10 rep stos dword ptr es:[edi]
11 mov     ecx,9AC003h
12 call    009A1316
13 mov     dword ptr [ebp-8],1
14 mov     dword ptr [ebp-14h],2
15 mov     eax,dword ptr [ebp-8]
16 add     eax,dword ptr [ebp-14h]
17 mov     dword ptr [ebp-20h],eax
18 mov     eax,dword ptr [ebp-20h]
19 push    eax
20 push    9A7D08h
21 call    009A10CD
22 add     esp,8
23 xor     eax,eax
24 pop     edi
25 pop     esi
26 pop     ebx
27 add     esp,0E4h
28 cmp     ebp,esp
29 call    009A123F
```

어셈블리어

1. C 언어 유래

■ 탄생 배경

- 1970년대 초 AT&T 벨(Bell) 연구소에서 UNIX 개발 목적으로 고안
- 1972년부터 데니스 리치와 켄 톰슨이 B 언어를 개선하여 C 언어를 개발
- B 언어보다 한 단계 발전하였다는 의미로 C 언어라 명명

■ 표준 제정

- 프로그래머가 가장 많이 사용하는 언어
- 미국표준협회(ANSI)에서 C 언어 표준 제정
- 다양한 컴파일러 개발과 함께 일반 PC에서도 C 언어를 사용하게 된 계기

■ C 언어의 장점

- 범용 프로그래밍 언어의 기능 수행
- 기계 지향적인 저급 언어와 문제 지향적인 고급 언어의 장점만 반영
- 다양한 자료형과 연산자로 간결한 표현 가능
- 프로그램을 함수의 집합으로 구성할 수 있어 모듈화 용이
- 이식성이 좋음
- 모든 함수가 순환 구조 허용
- 자료의 주소를 자유롭게 조작할 수 있는 포인터 제공
- 자체 입출력 기능 없이 라이브러리 함수로 입출력 명령 수행
- 하드웨어와 관련된 장치를 제어하기 쉬움
- 구문이 간결하여 배우기 쉬움

1. 요구 사항 분석

■ 요구사항 (needs) : 프로그램 개발 시 첫 번째 수행 단계

- 프로그래머는 개발하고자 하는 프로그램의 요구사항 분석
- 요구 사항을 명확히 분석하지 못할 경우 엉뚱한 결과 초래



그림 1-3 프로그램 개발 첫 단계인 요구 사항 분석

2. 알고리즘 설계

■ 알고리즘 (Algorithm)

- 프로그램 개발을 위한 일련의 과정을 차례대로 정리해 놓은 명령 순서
- 일정한 순서에 따라 반복되며 자연어, 의사 코드, 순서도 등으로 표현

학사 일정 관리 알고리즘

시작 :

1. 월은 1부터 12까지의 숫자만 허용한다.
 - 1.1 월의 구분은 <알고리즘_1>을 적용한다.
 - 1.2 월별 스케줄은 <명세서_1>을 적용한다.
 - 1.3 유효 범위를 초과하는 숫자에 대해서는 에러 메시지를 출력한다.
 - 1.4 허용되지 않는 값을 입력했을 경우에는 월을 다시 입력하도록 한다.
2. 요일은 일요일부터 순서대로 적용한다.
 - 2.1 요일의 구분은 <알고리즘_2>를 적용한다.
 - 2.2 요일별 스케줄은 <명세서_2>를 적용한다.
 - 2.3 유효 범위를 초과하는 숫자에 대해서는 에러 메시지를 출력한다.
 - 2.4 허용되지 않는 값을 입력했을 경우에는 요일을 다시 입력하도록 한다.
3. 수업 유무는 2항의 요건에 따라 각각의 스케줄을 적용한다.
 - 3.1 스케줄 적용은 <알고리즘_3>을 적용한다.
 - 3.2 시간대별 스케줄은 <명세서_3>과 <명세서_4>를 적용한다.
 - 3.3 허용되지 않는 값을 입력했을 경우에는 시간을 다시 입력하도록 한다.

끝 :

그림 1-4 학사 일정 관리 프로그램의 알고리즘

2. 알고리즘 설계

■ 의사코드 (슈도,수도 코드 Pseudo-code)

- 프로그래밍 언어를 코드로 작성 전, 무엇을 어떻게 동작 하도록 작성할 지에 대하여 사람이 이해할 수 있는 언어로 작성하는 것
- 의사코드를 작성하기 전 문제를 이해하고, 논리적으로 해석 이후 컴퓨팅 사고로 전환하여 의사코드를 작성하고 개발 언어로 코드를 작성

```
// 1. 단순 명세
1. 두 가지 숫자를 준비한다.
2. 더하기를 수행한다.
3. 결과값을 도출한다.

// 2. 두 수를 더하는 함수 만들기
입력 : 두 가지 정수(num1, num2)
처리 : num1 + num2
출력 : 두 가지 정수를 더한 값(result)
```

```
// 1. 단순 명세
int num1 = 5;
int num2 = 10;
int result = num1 + num2;

// 2. 두 수를 더하는 함수 만들기
int add(int num1, int num2) {
    result = num1 + num2;
    return result;
}
```

3. 명세서 작성

■ 명세서 (Statement)

- 알고리즘에 필요한 내용을 특정 기준이나 목록 등으로 구분한 표의 형태
- 프로그램에 적용할 기본 규칙이나 기준을 제시하는 역할
- 학사 일정 관리 프로그램의 알고리즘은 다음 명세서로 나타낼 수 있음

요일별 스케줄

요일	코드	1학기 수업	코드	하계 학기	코드	2학기 수업	코드	동계 학기
일	211	1학기-없음	231	하계-없음	221	2학기-없음	241	동계-없음
월	212	1학기-월	232	하계-월	222	2학기-월	242	동계-월
화	213	1학기-화	233	하계-화	223	2학기-화	243	동계-화
수	214	1학기-수	234	하계-수	224	2학기-수	244	동계-수
목	215	1학기-목	235	하계-목	225	2학기-목	245	동계-목
금	216	1학기-금	236	하계-없음	226	2학기-금	246	동계-없음
토	217	1학기-없음	237	하계-없음	227	2학기-없음	247	동계-없음

그림 1-5 학사 일정 관리 프로그램의 명세서

4. 순서도 작성

■ 순서도 (Flowchart)

- 서술형으로 기술된 알고리즘 명령을 그림으로 표현
- 순서도를 그릴 때 사용하는 도형은 [범례]에서 제시하는 도형만 사용

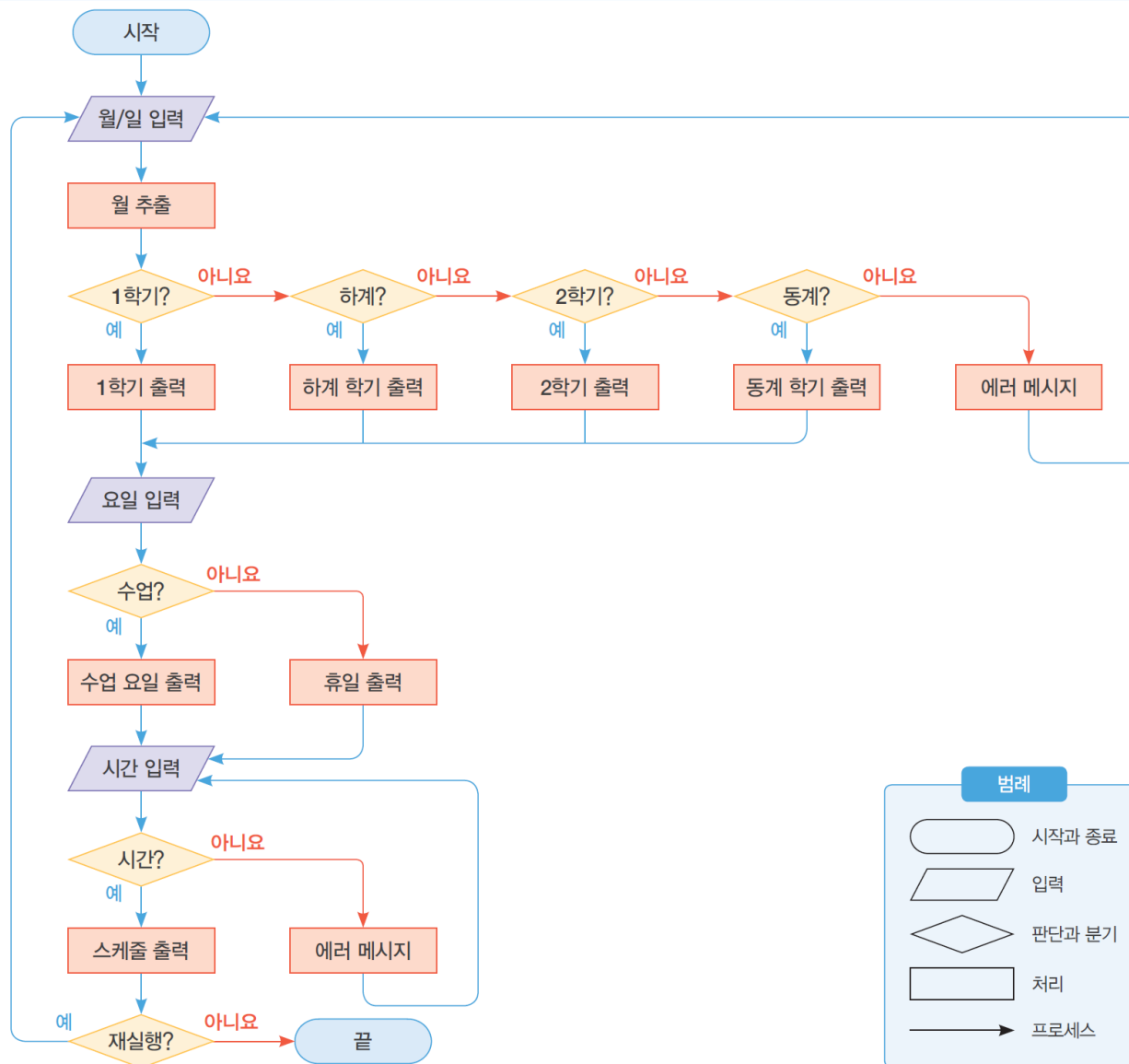


그림 1-6 학사 일정 관리 프로그램의 순서도

5. 소스 코드 작성

■ 소스 코드 (Source code)

- 프로그래밍 언어로 알고리즘을 표현한 코드
- 알고리즘과 순서도를 기준으로 컴파일러를 통해 작성

```

test.c x
Project1 (전역 범위) main(void)
1  /* c 언어 소스 코드
2     작성 날짜 : 2022년 12월 25일
3     작성자 : 아무개
4     연락처 : 핸드폰과 이메일 주소 기입 */
5
6     # include <stdio.h>
7     int main(void)
8     {
9         printf("학사 일정 관리 프로그램");
10        printf("이 프로그램은 다음과 같은 순서로 실행됩니다.\n");
11        printf("1. 요일 입력 : \n");
12        printf("2. 코드 입력 : \n");
13        printf("3. 학기 입력 : \n");
14        printf("4. 수업 여부 입력 : \n");
15        printf("5. 계절 입력 : \n");
16        printf("6. 입력한 내용에 대한 학사 일정은 다음과 같습니다.\n");
17        return 0;
18    }
  
```

그림 1-7 학사 일정 관리 프로그램의 소스 코드

6. 컴파일과 링크

■ 컴파일 (Compile)

- 인간이 이해하는 명령어를 컴퓨터가 이해 할 수 있도록 번역하는 과정
- 컴파일은 비주얼 스튜디오와 같은 컴파일러에서 소스 코드를 대상으로 수행

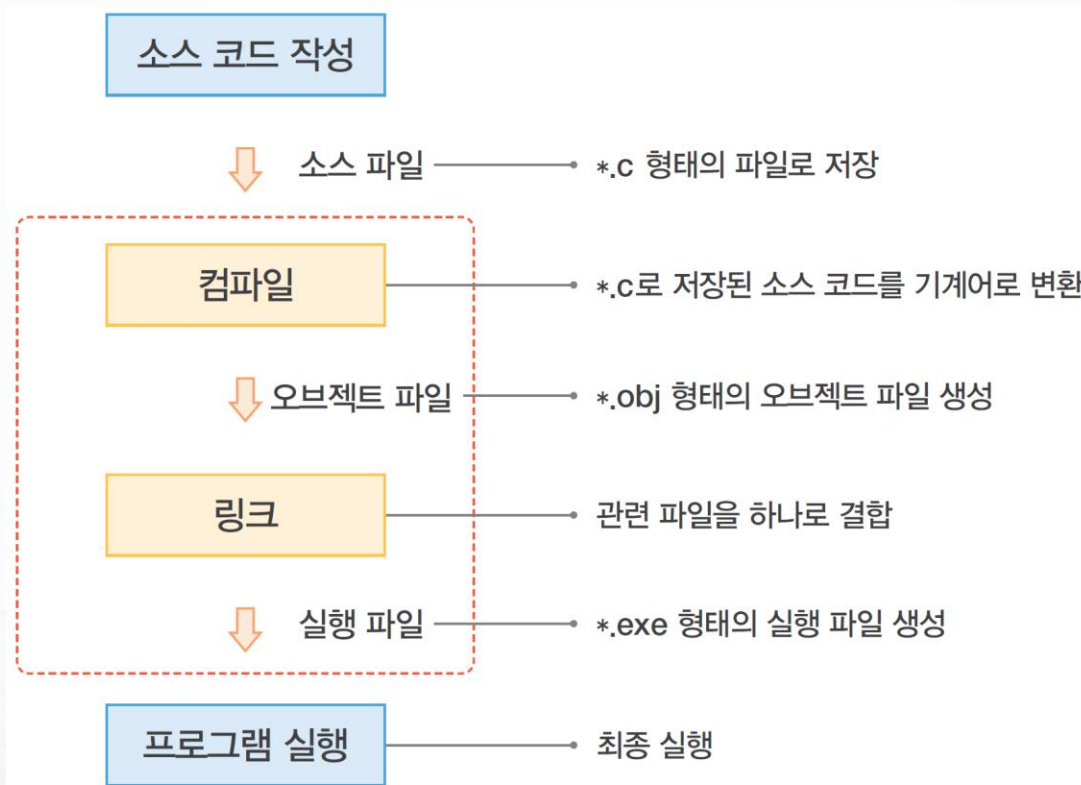
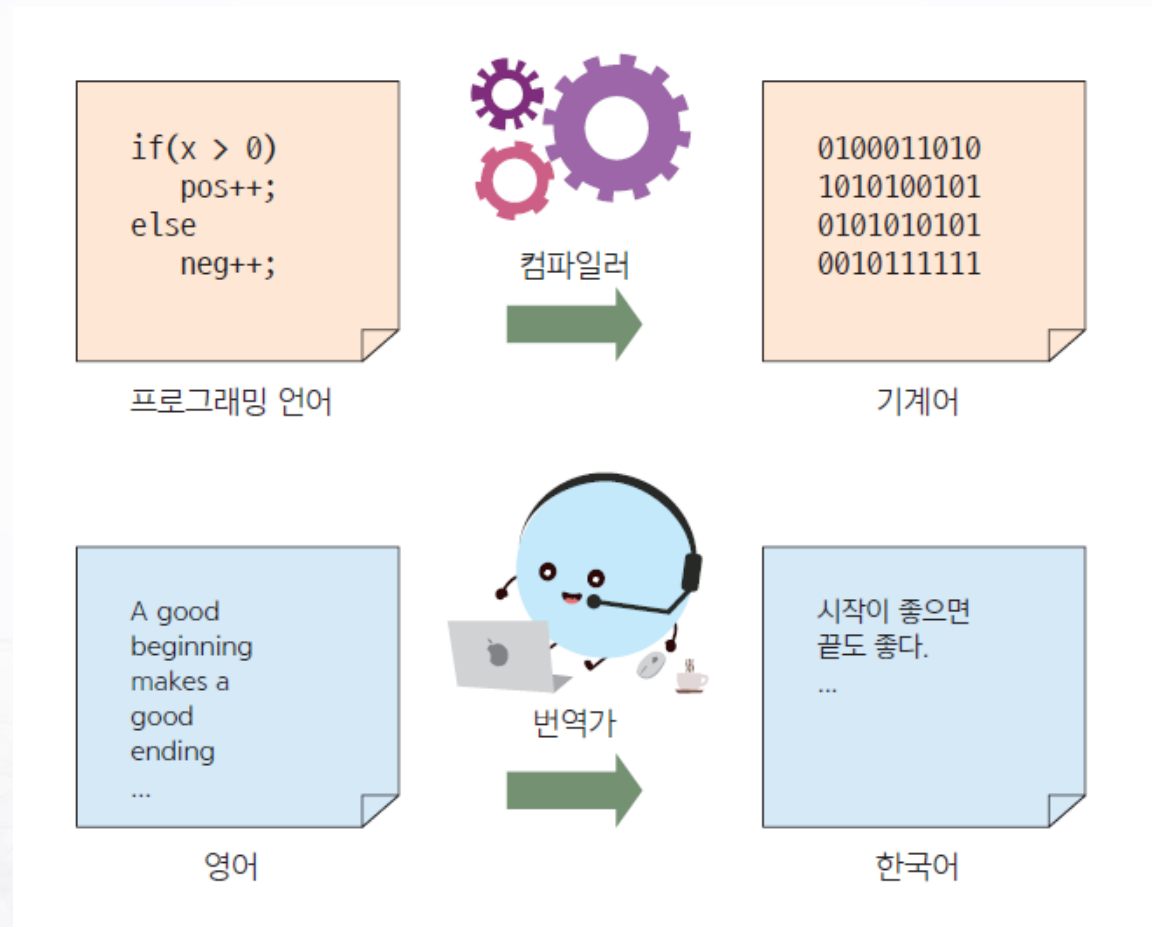


그림 1-8 C 프로그램 작성과 실행 순서

6. 컴파일과 링크

■ 컴파일러 (Compiler)

- 컴파일러(compiler)는 인간과 컴퓨터 사이의 통역이라 할 수 있다



6. 컴파일과 링크

■ 오브젝트(object) 파일

- 기계어 혹은 그에 준하는 이진 코드로 이루어진 파일
- 오브젝트 파일의 필수 요소는 컴퓨터의 CPU가 직접 실행하는 코드인 기계어

■ 링크 (Link)

- 여러 개의 오브젝트 파일을 묶어 커널과 연결
- 실행 파일이나 라이브러리를 만들 때 사용

■ 커널 (Kernel)

- 컴퓨터 운영체제의 가장 중요한 핵심요소
- 컴퓨터 시스템을 통제
- 운영체제의 다른 모든 부분에 여러 가지 기본 서비스를 제공

7. 프로그램 실행과 디버깅

■ 실행 파일 (*.exe)

- 소스 코드를 컴파일과 링크 과정을 거쳐 실행 파일 생성
- 실행 파일은 완벽한 기계어로 된 파일

■ 디버깅 (Debuging)

- 소스 코드 문법 오류 또는 논리 오류가 있는지를 발견하여 오류 사항을 해결하는 과정

7. 프로그램 실행과 디버깅

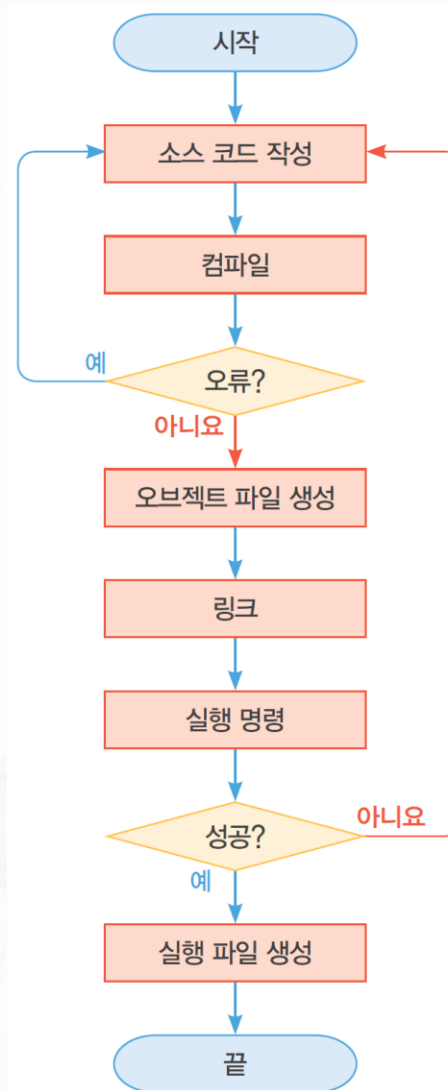
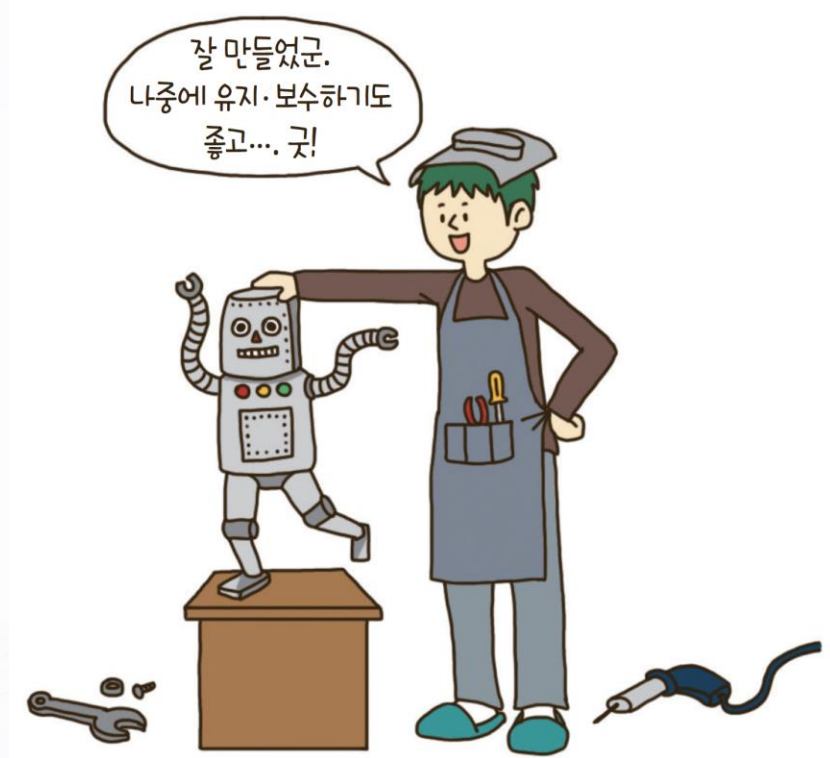


그림 1-9 프로그램 실행과 디버깅 프로세스

8. 프로그램 유지·보수

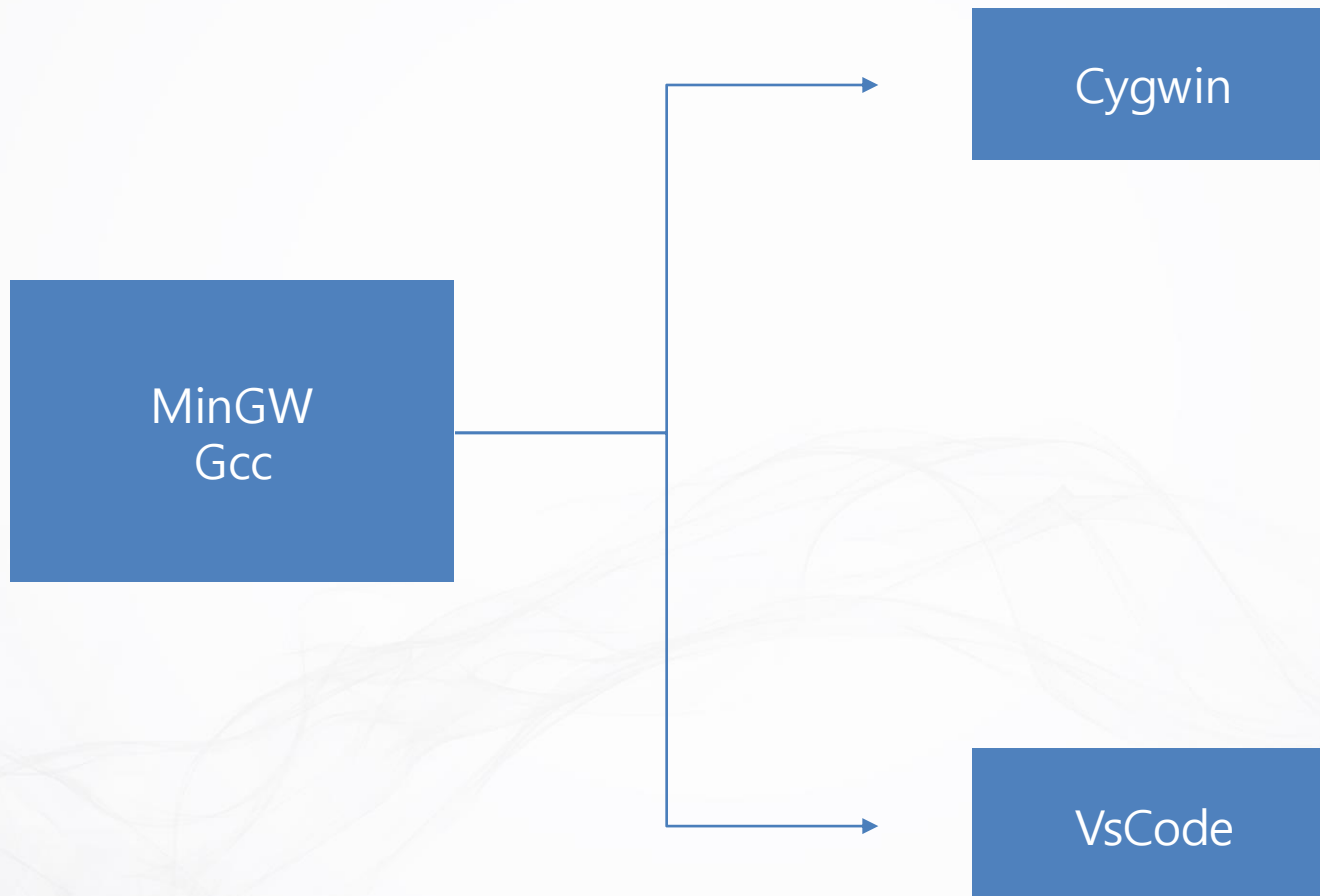
■ 유지·보수가 필요한 이유

- 시대의 흐름과 환경의 변화
- 요구 사항의 변동
- 소프트웨어 호환성
- 기타 문제 발생



C 언어 개발을 위한 IDE 설치

- Visual Studio Code
- Cygwin



<https://sourceforge.net/projects/mingw-w64/files/mingw-w64/>

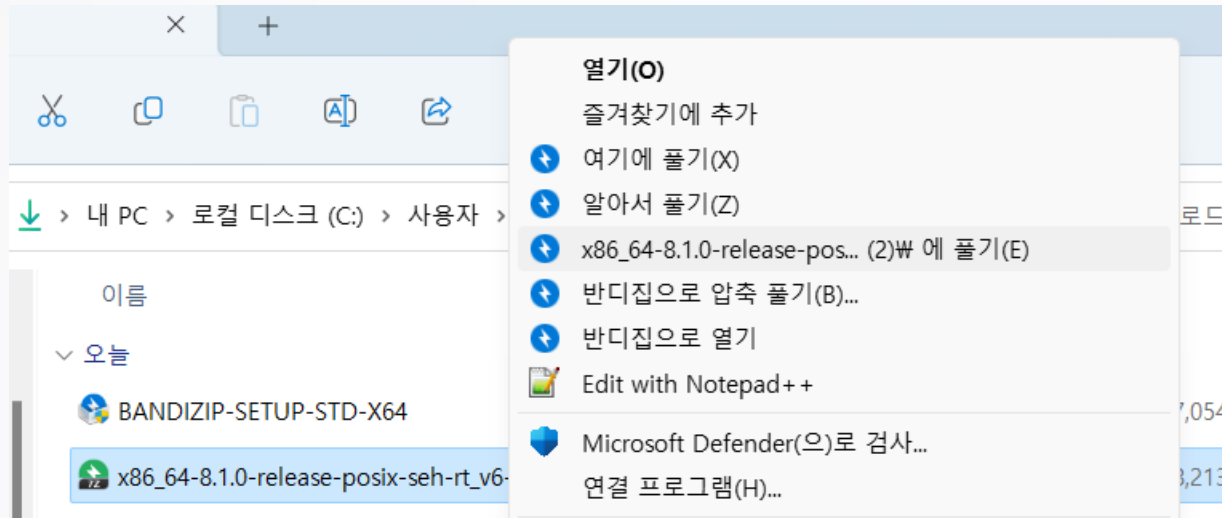
MinGW-W64 GCC-8.1.0

- x86_64-posix-sjlj
- x86_64-posix-seh
- x86_64-win32-sjlj
- x86_64-win32-seh
- i686-posix-sjlj
- i686-posix-dwarf
- i686-win32-sjlj
- i686-win32-dwarf

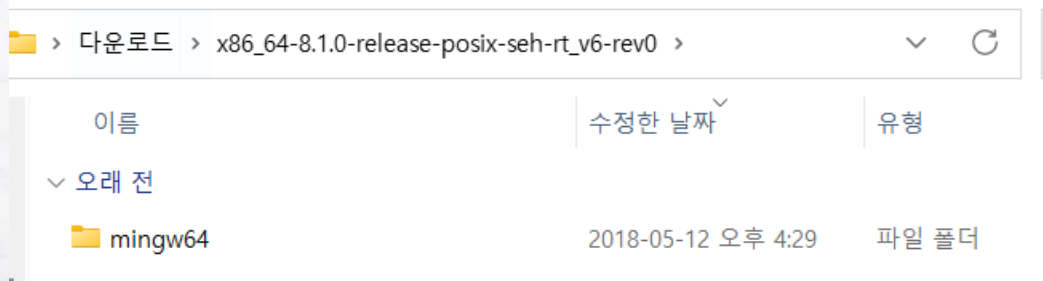
64bit

32bit

MinGW 설치 - 1



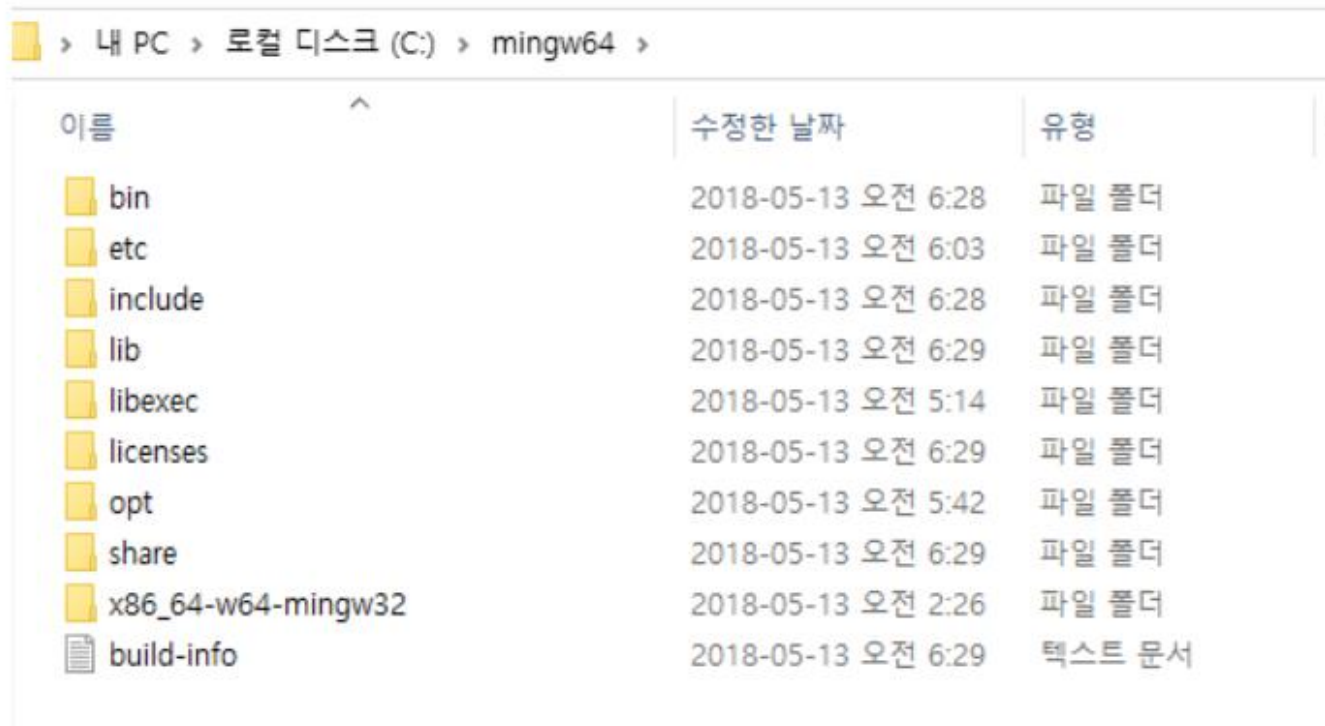
- 압축 프로그램 -> 압축 풀기



MinGW 설치 - 2

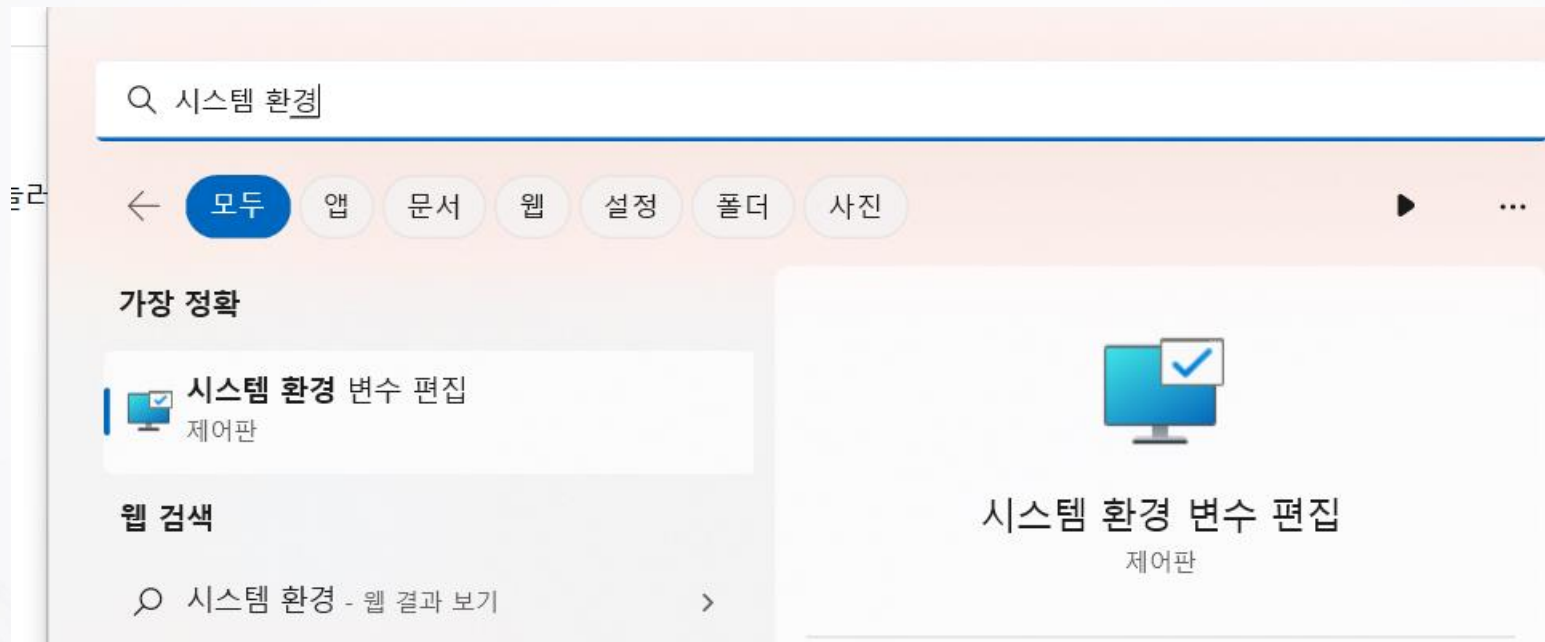
- 복사 -> C:\mingw64

압축을 풀어 C:\mingw64에 복사해줍니다.

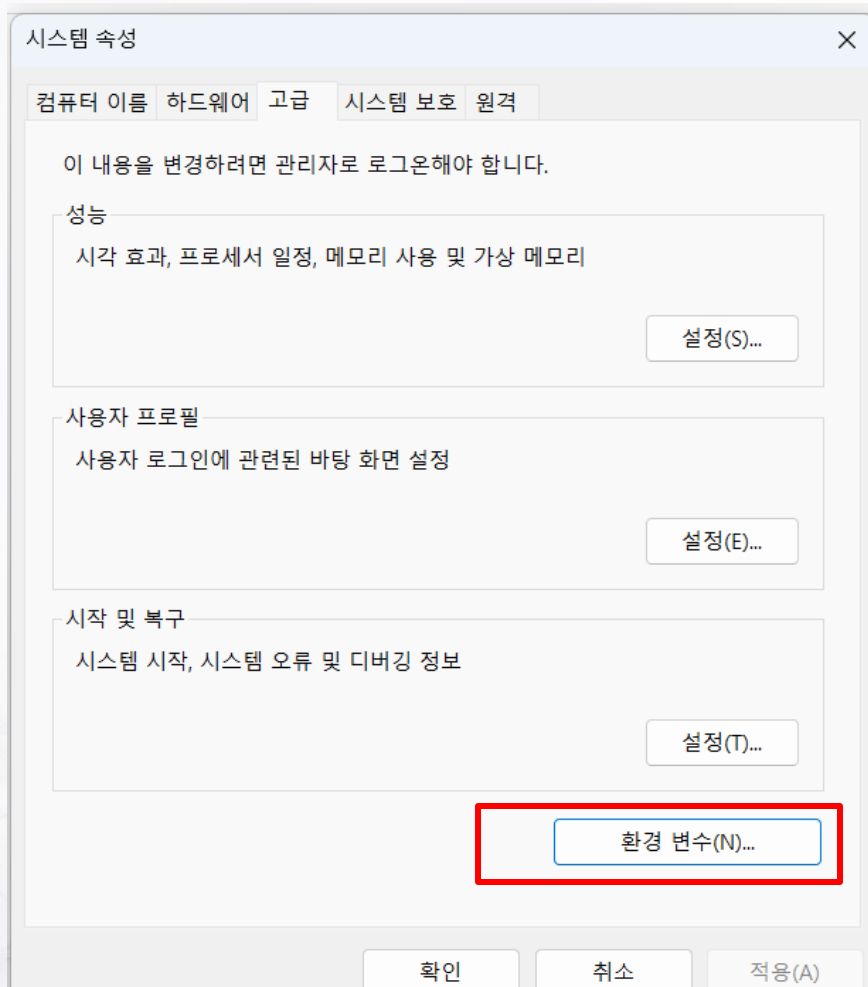


이름	수정한 날짜	유형
bin	2018-05-13 오전 6:28	파일 폴더
etc	2018-05-13 오전 6:03	파일 폴더
include	2018-05-13 오전 6:28	파일 폴더
lib	2018-05-13 오전 6:29	파일 폴더
libexec	2018-05-13 오전 5:14	파일 폴더
licenses	2018-05-13 오전 6:29	파일 폴더
opt	2018-05-13 오전 5:42	파일 폴더
share	2018-05-13 오전 6:29	파일 폴더
x86_64-w64-mingw32	2018-05-13 오전 2:26	파일 폴더
build-info	2018-05-13 오전 6:29	텍스트 문서

- 작업표시줄 검색 창 에서 시스템 환경 변수



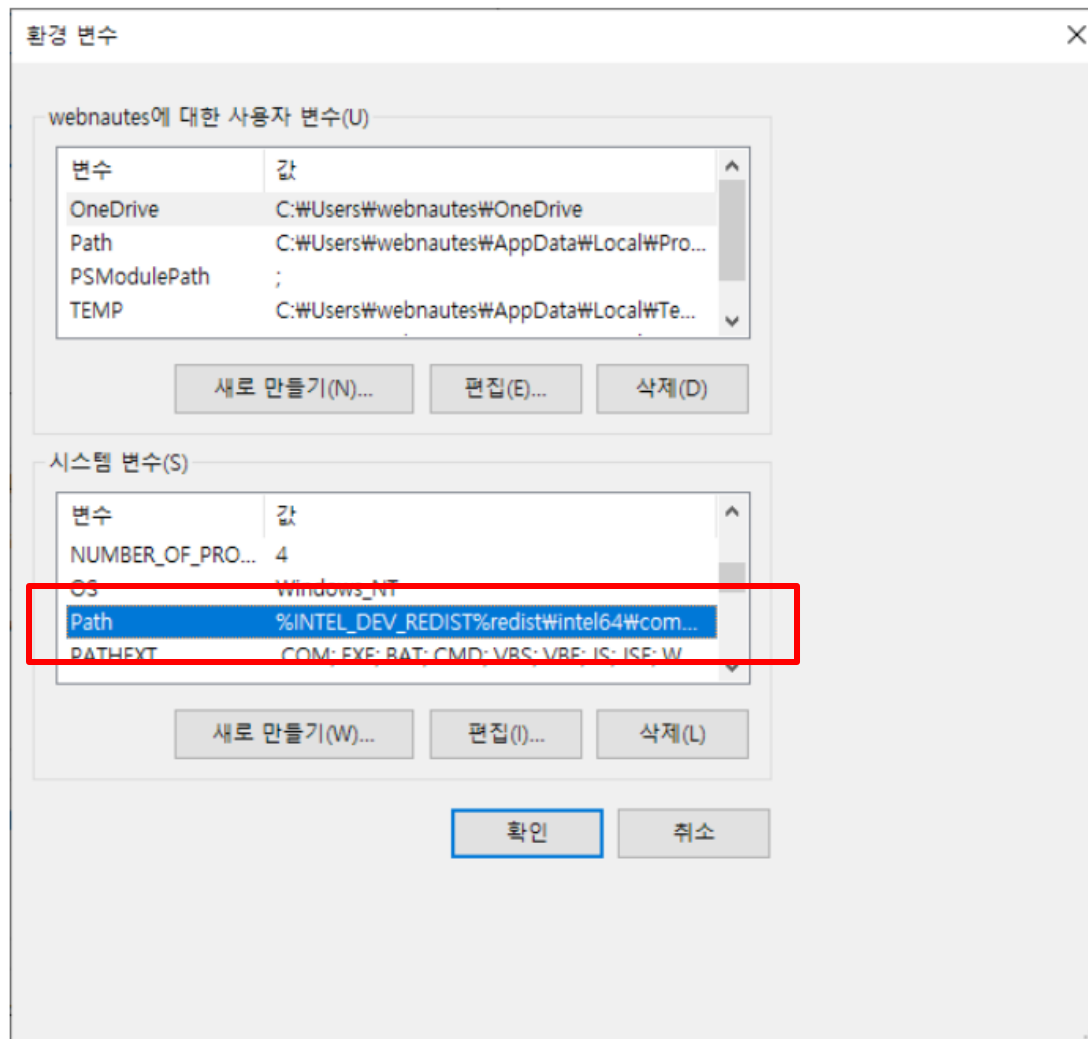
- 환경변수



MinGW 설치 - 5

- Path -> 편집

시스템 변수에서 Path를 선택하고 편집 버튼을 클릭합니다.



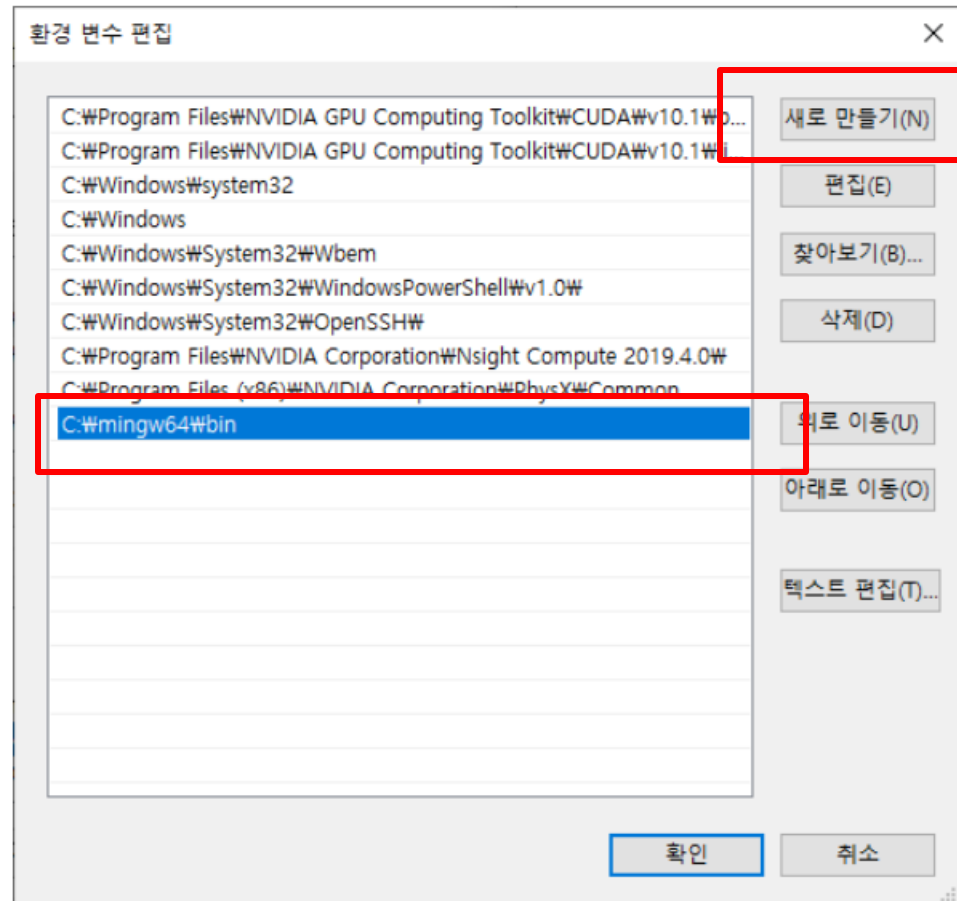
MinGW 설치 - 6

- 새로 만들기

-> C:\mingw64\bin

새로 만들기 버튼을 클릭한 후.. 맨 아래에 다음 경로를 입력합니다.

C:\mingw64\bin



MinGW 설치 - 6

- 윈도우키 + R -> cmd
- gcc -v

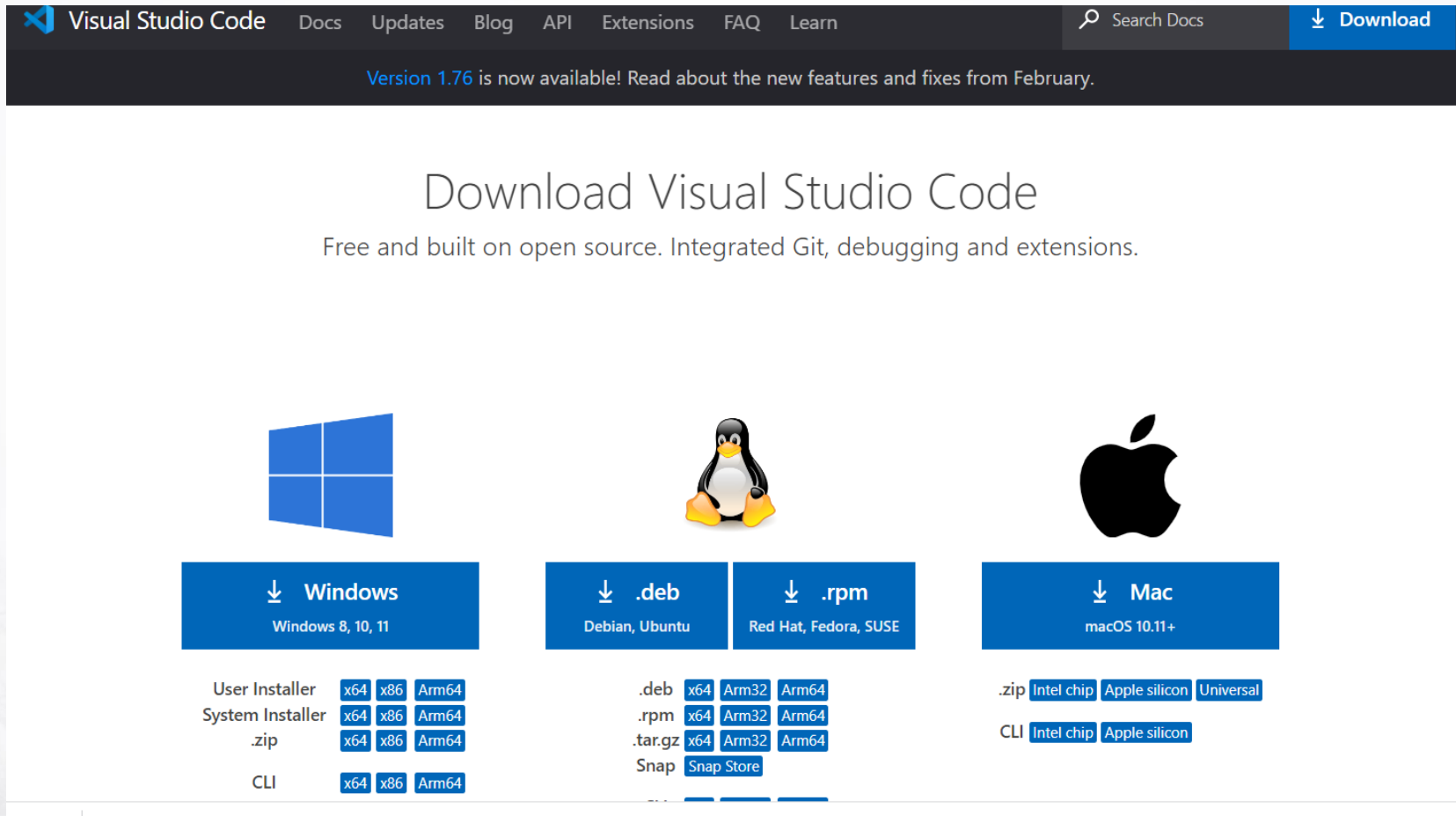
```
C:\Windows\System32\cmd
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=C:/mingw64/bin/./libexec/gcc/x86_64-w64-mingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../.././src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --prefix=/mingw64 --with-sysroot=/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64 --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=nocore --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpfr=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-isl=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-pkgversion='x86_64-posix-seh-rev0, Built by MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident -I/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw810/prerequisites/x86_64-zlib-static/include -I/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw810/prerequisites/x86_64-zlib-static/include -I/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CPPFLAGS='-I/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw810/prerequisites/x86_64-zlib-static/include -I/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/include' LDFLAGS='-pipe -fno-ident -L/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/lib -L/c:/mingw810/prerequisites/x86_64-zlib-static/lib -L/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/lib '
Thread model: posix
gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)

C:\Users\user>
```

Visual Code 설치

- <https://code.visualstudio.com/download>



The screenshot shows the Visual Studio Code download page. At the top, there's a navigation bar with links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A search bar and a 'Download' button are also present. Below the navigation bar, a message states 'Version 1.76 is now available! Read about the new features and fixes from February.' The main heading is 'Download Visual Studio Code', followed by the tagline 'Free and built on open source. Integrated Git, debugging and extensions.' The page is divided into three main sections for Windows, Linux, and Mac. Each section has a download button and a list of available installers or packages with their respective architectures.


Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Search Docs Download

Version 1.76 is now available! Read about the new features and fixes from February.


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows
Windows 8, 10, 11


User Installer	x64	x86	Arm64
System Installer	x64	x86	Arm64
.zip	x64	x86	Arm64
CLI	x64	x86	Arm64



↓ .deb
Debian, Ubuntu

↓ .rpm
Red Hat, Fedora, SUSE

.deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		

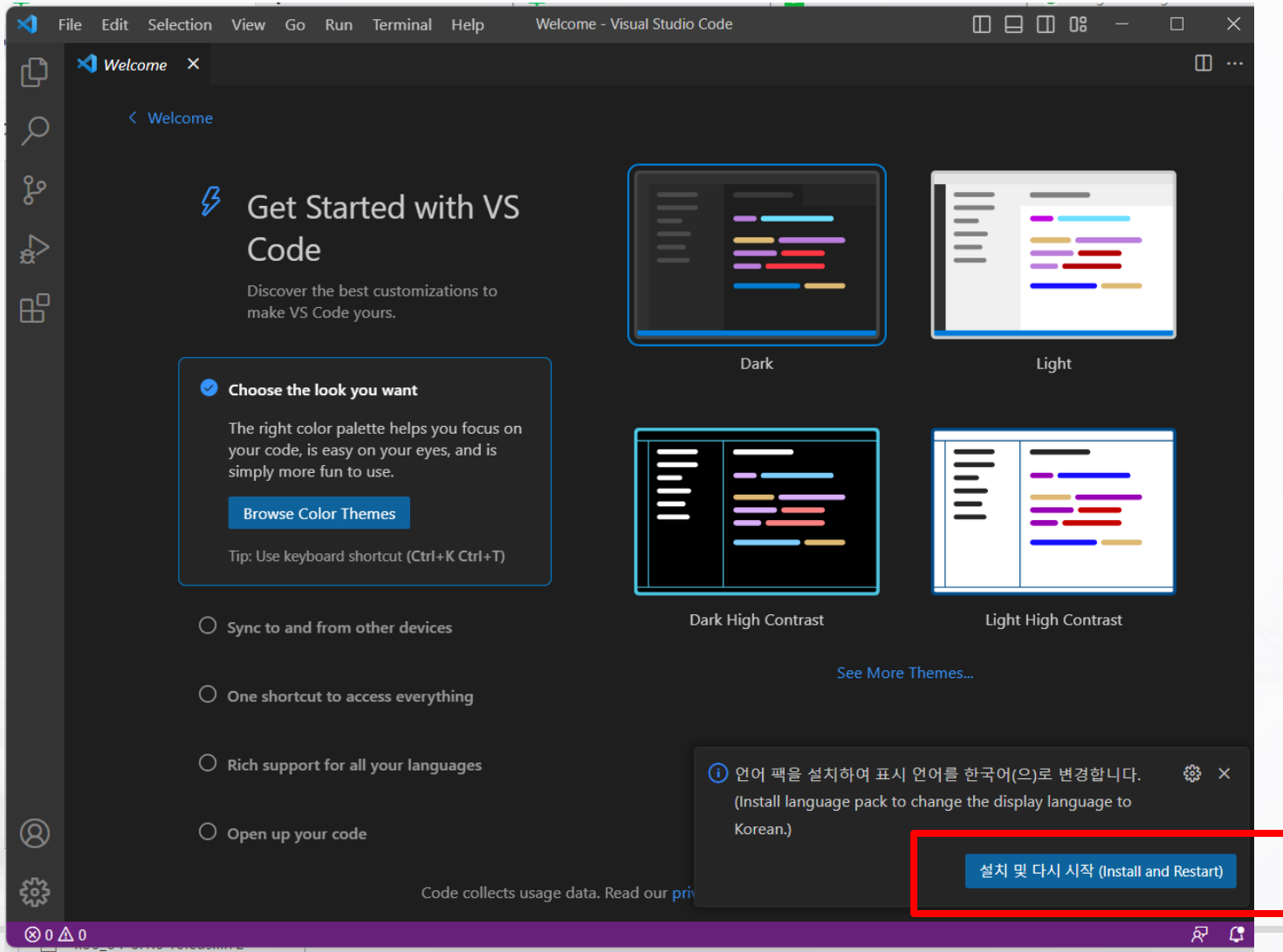


↓ Mac
macOS 10.11+

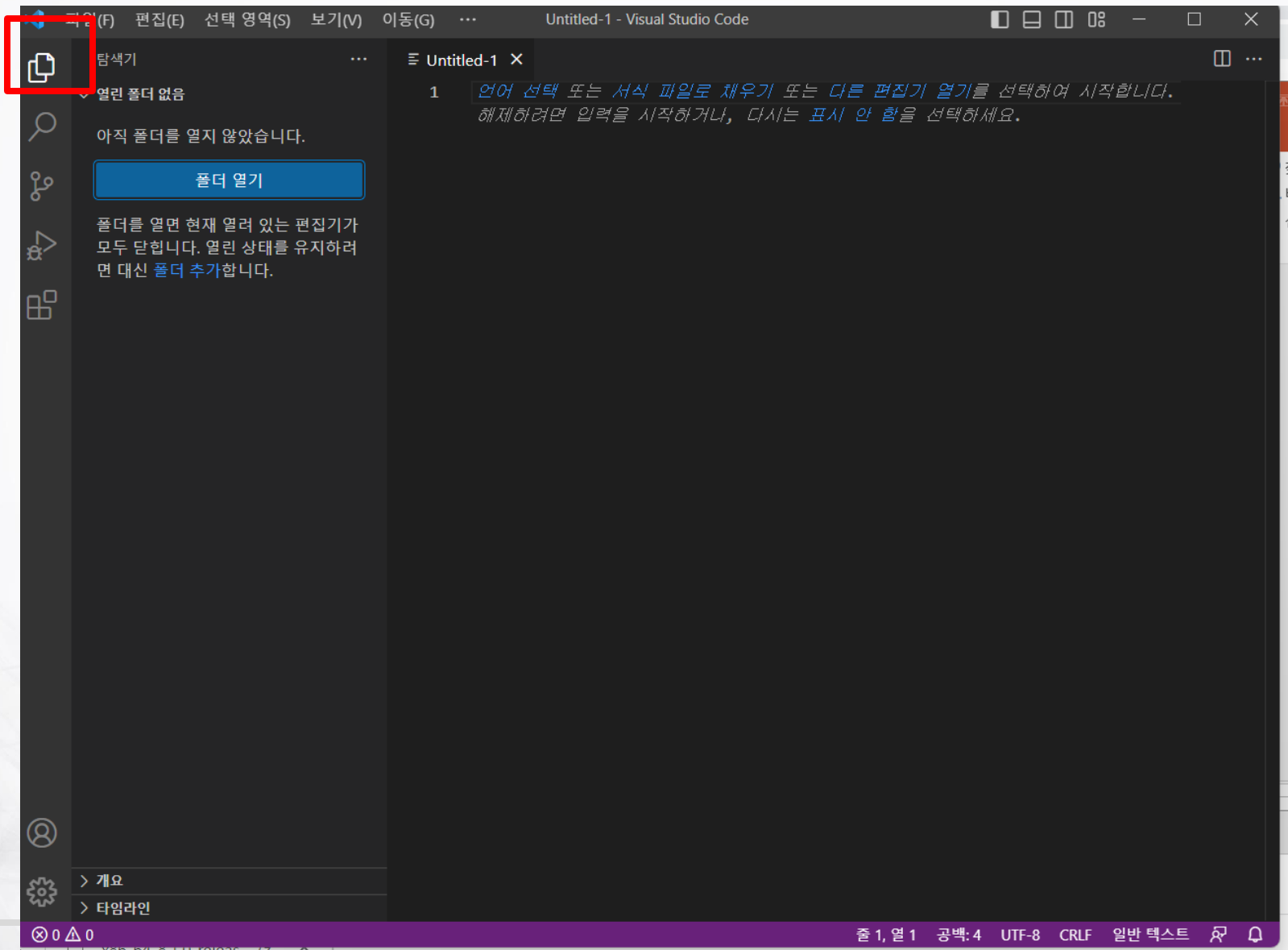
.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	

Visual Code 설치 - 1

• 한국어 설치

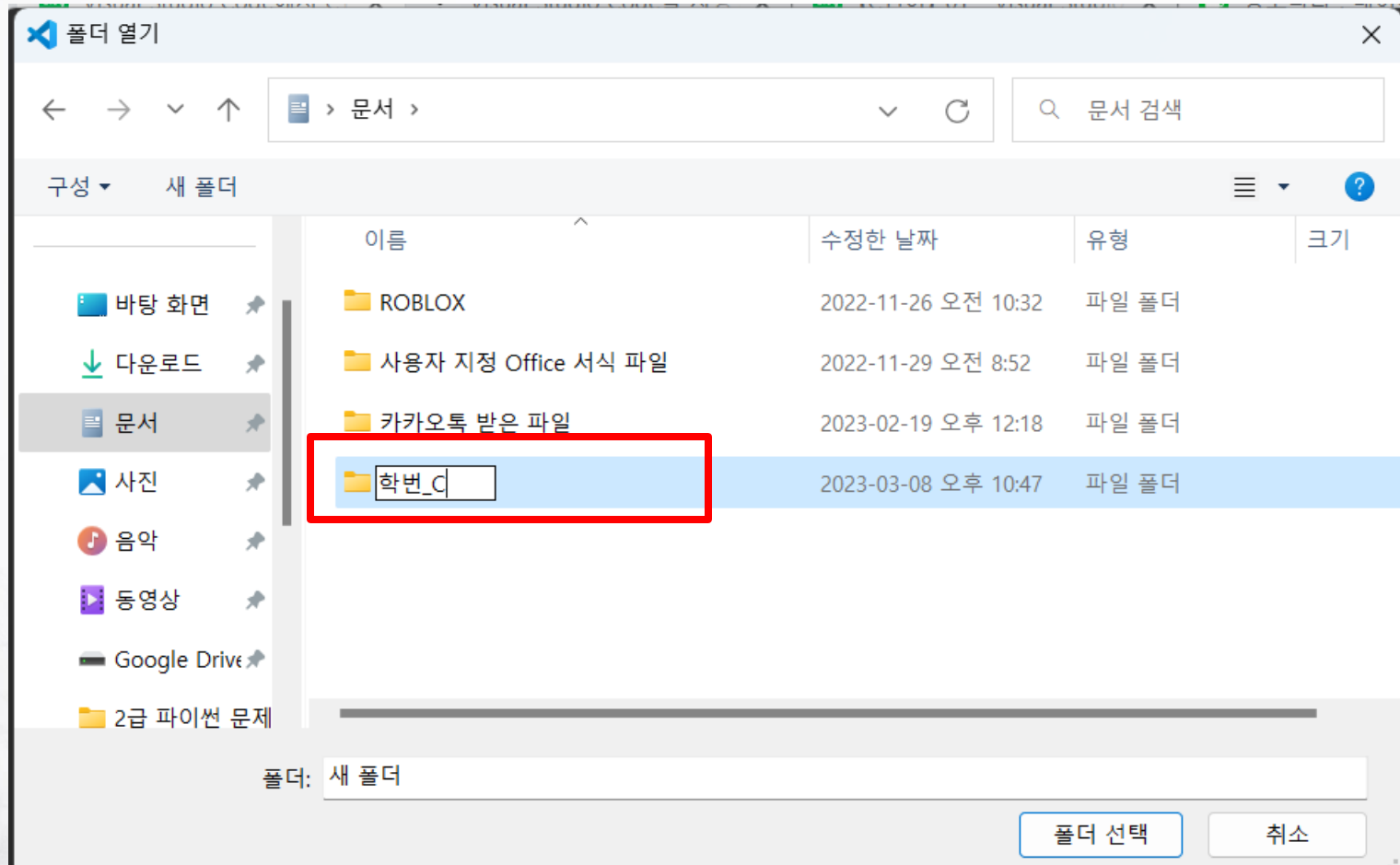


Visual Code 설치 - 2



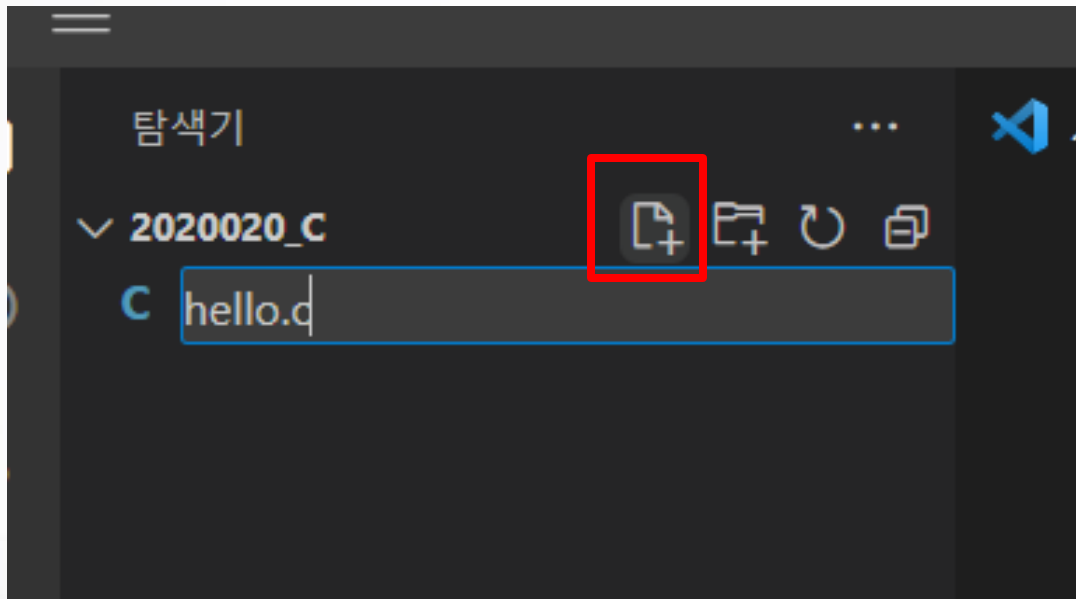
Visual Code 설치 - 3

- 폴더 생성



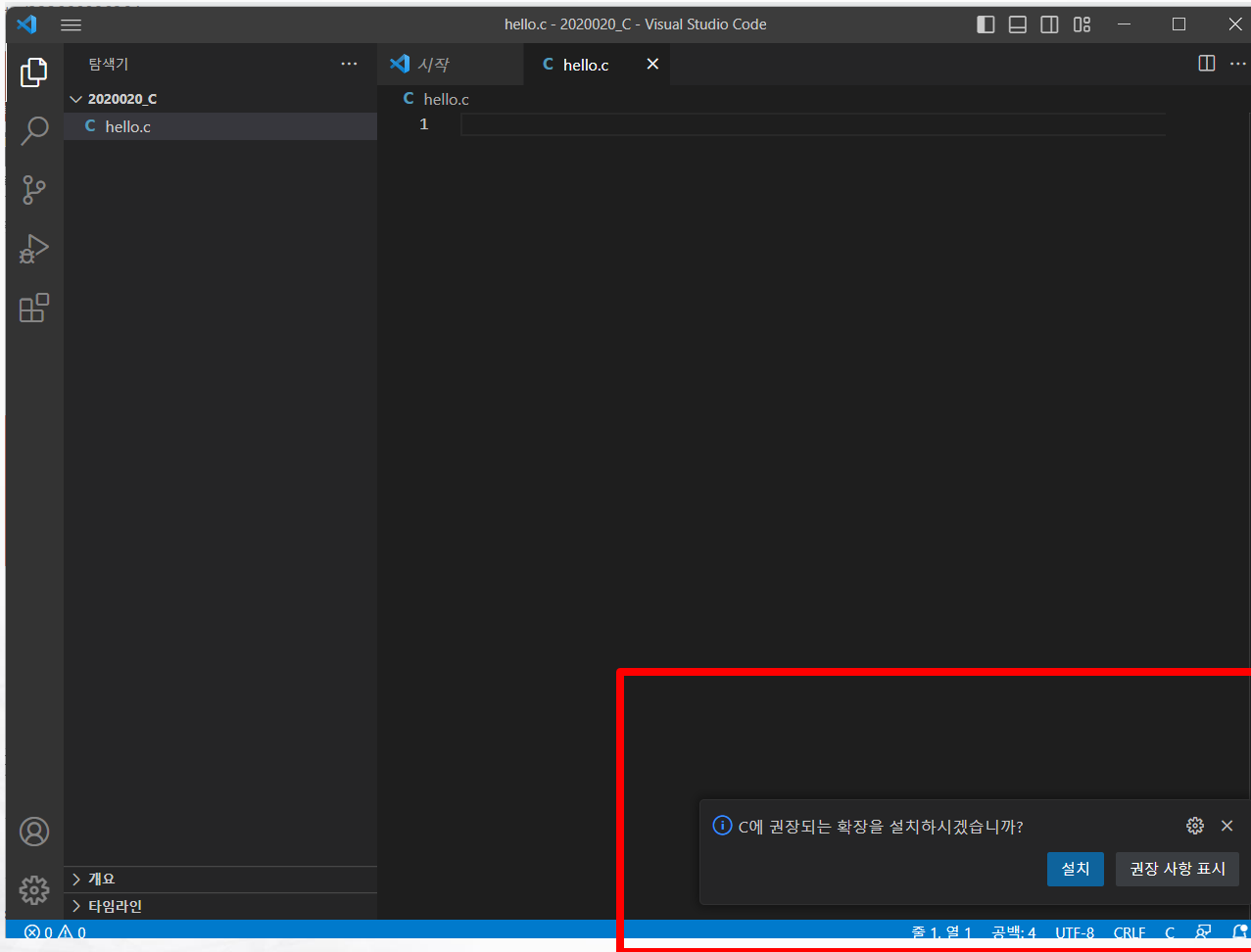
Visual Code 설치 - 4

- 폴더 생성



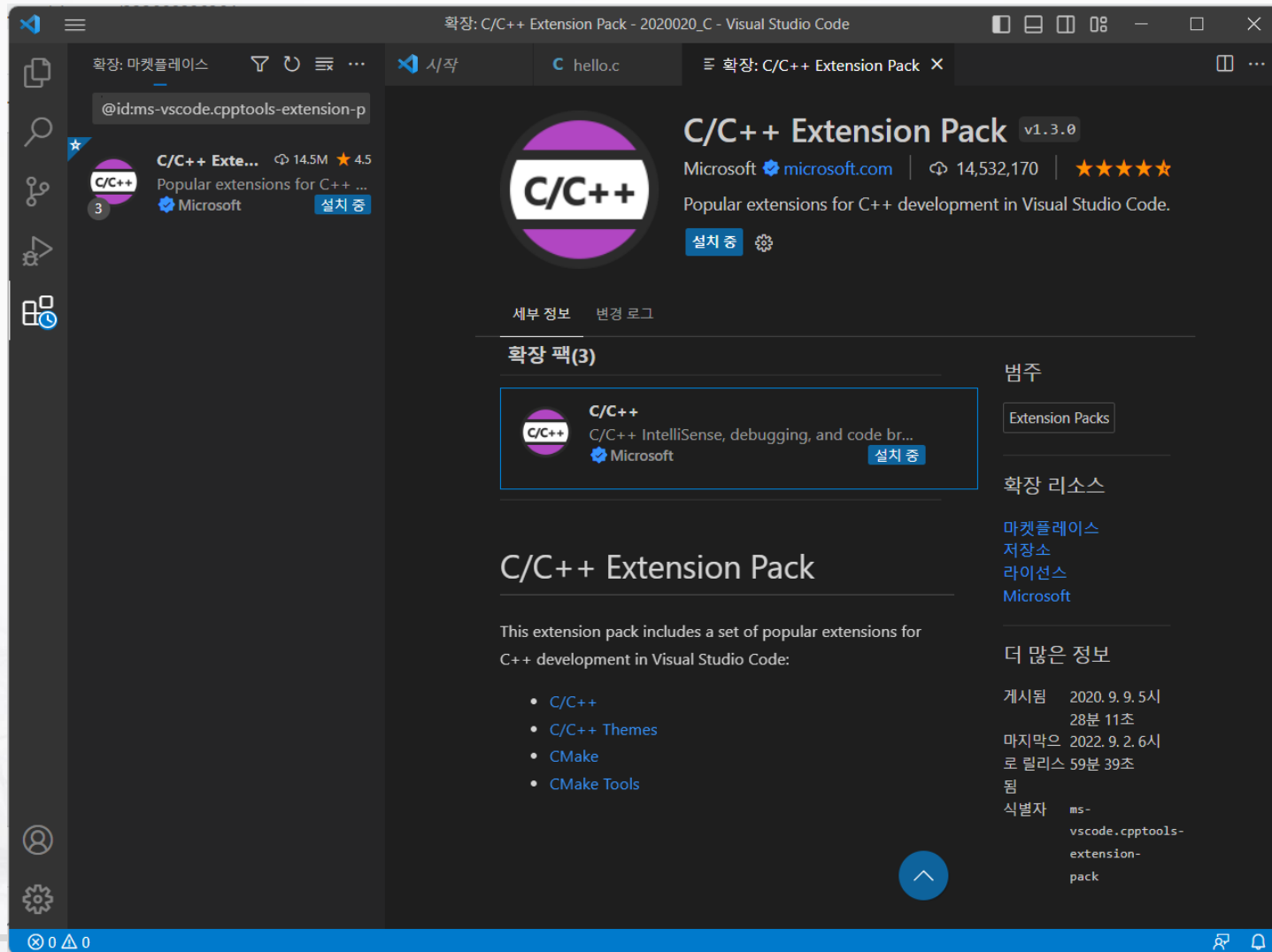
Visual Code 설치 - 5

- C에 권장되는 항목 설치



Visual Code 설치 - 6

• C 확장팩 설치



- 코드 테스트

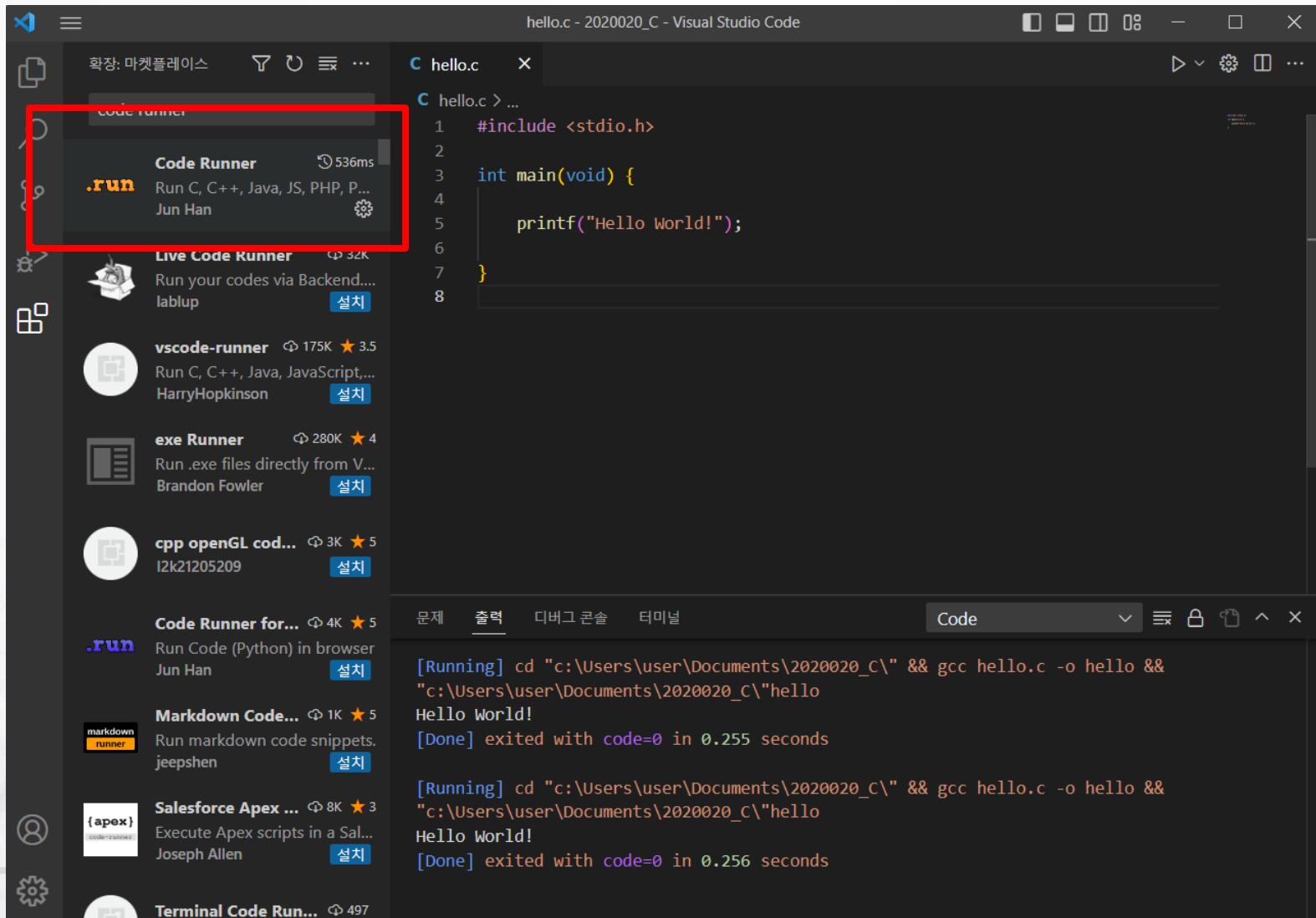
C hello.c X

C hello.c > ...

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      printf("Hello World!");
6
7  }
8
```

Visual Code 설치 - 8

- 옵션 -> Code Runner

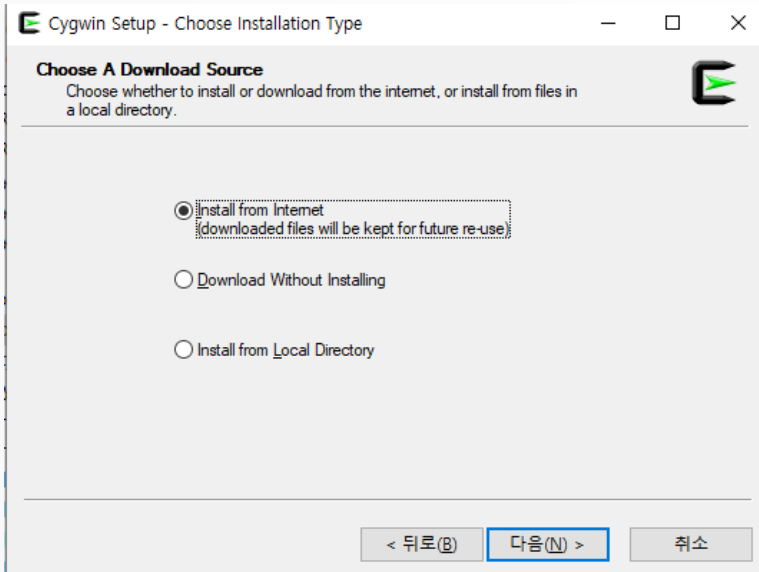


- 구글에서 Cygwin 검색 후 사이트 이동

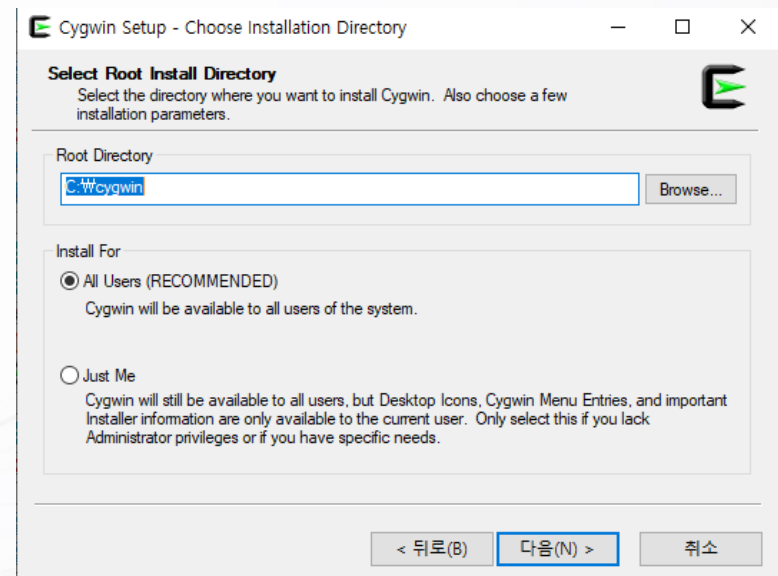
<https://cygwin.com/install.html>



Cygwin install

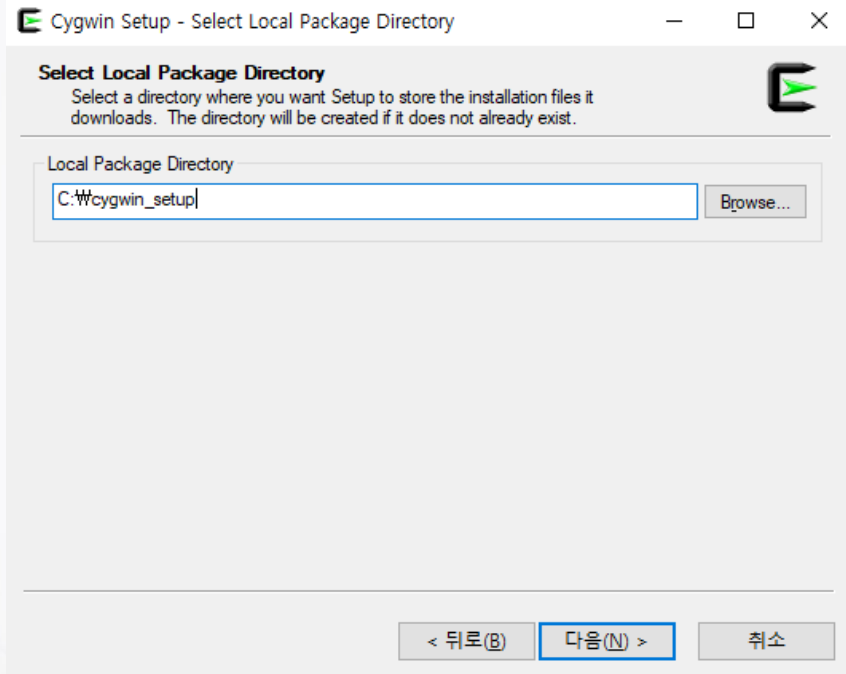


1. Install from internet

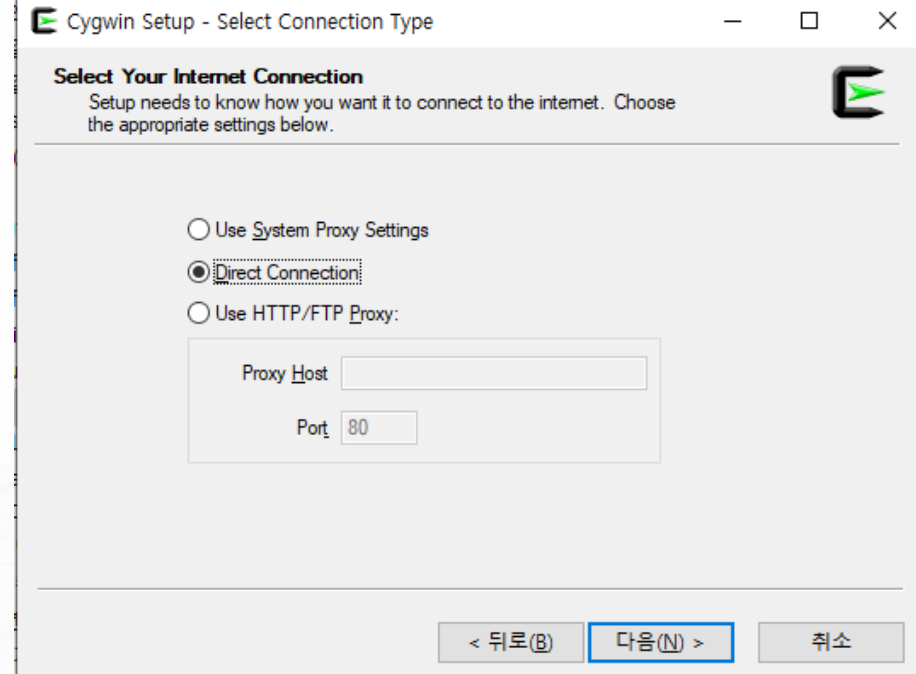


2. C:\wcygwin

Cygwin install (Const.)

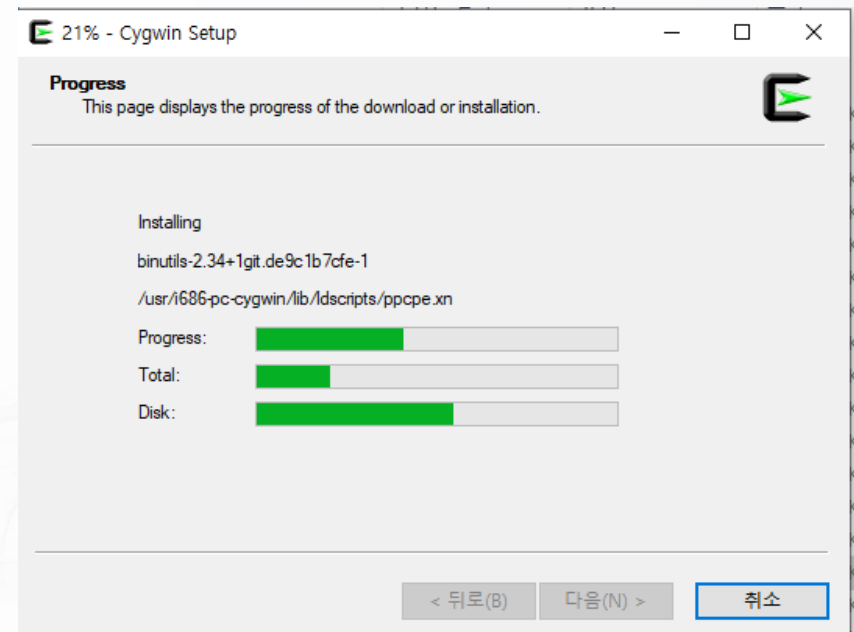
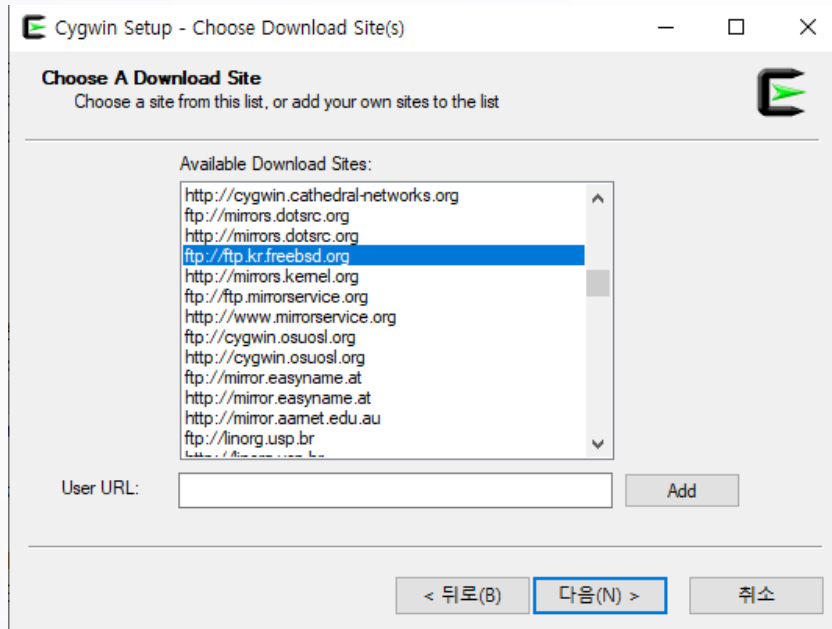


3. Setup package directory



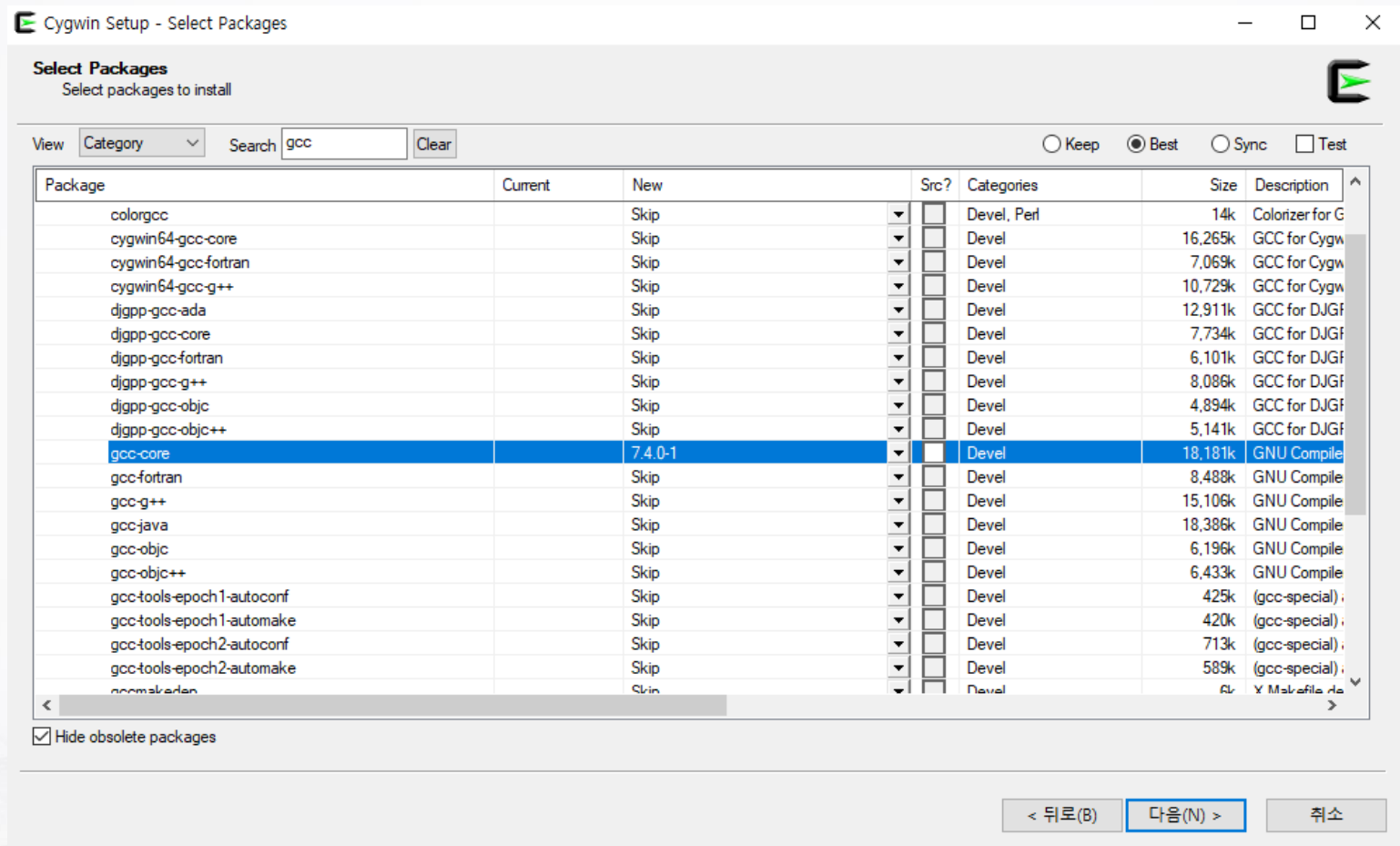
4. Direct Connection

Cygwin install (Const.)



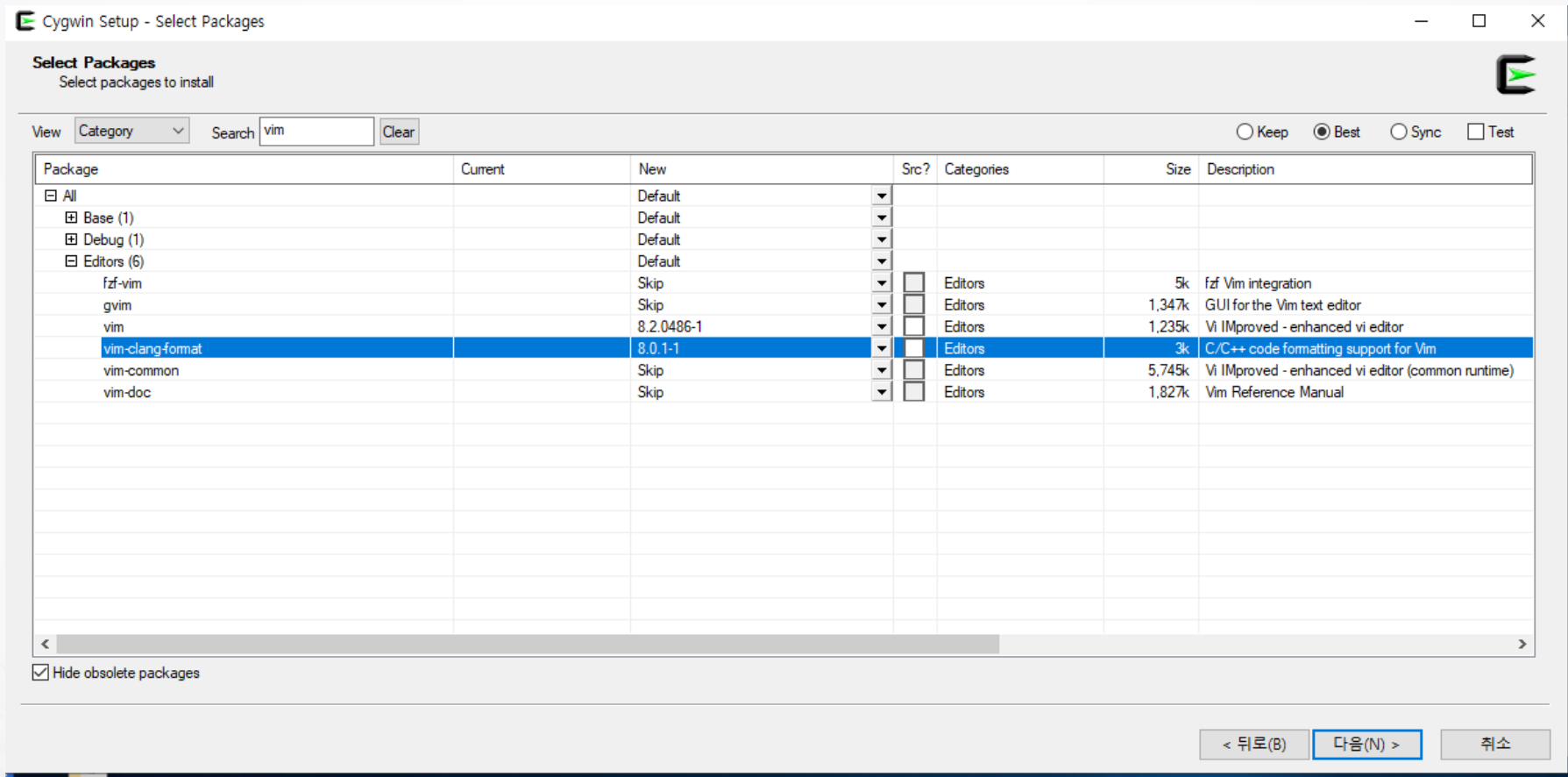
5. <ftp://ftp.kr.freebsd.org>

Cygwin install (Const.)



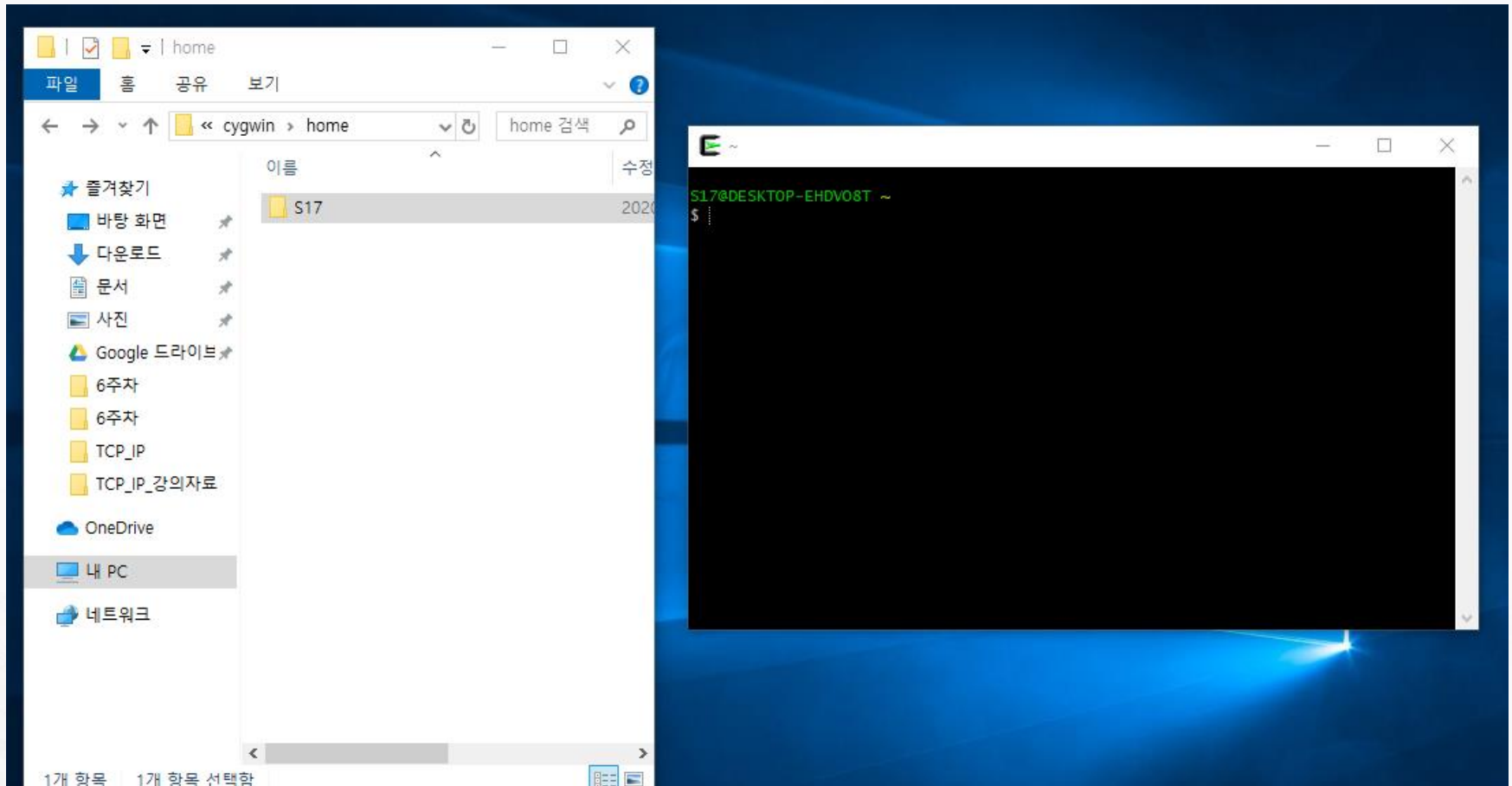
6. Category -> gcc -> gcc-core -> 7.4.0-1

Cygwin install (Const.)



7. Category -> vim -> vim , vim-clang-format

Cygwin view



- gcc c파일이름 -o 실행파일이름
- **Ex) gcc hello.c -o helloTest**

수고 하셨습니다.

