

C프로그래밍

1-1 C언어 개요

01

교과목 소개

교수 소개 (이우신)

1994 ~ 2007

광운대학교 컴퓨터공학과 학사/석사/박사

2008 ~ 2015

<삼성탈레스> 전술이동통신체계 (TMCS, Tactical Mobile Communication System) SW 개발 총괄

2016 ~ 2018

<한화시스템> 군용 무인기, 차세대 전투기, 정찰위성 통신시스템 SW 개발 총괄

2019 ~ 2022

<한화시스템> 국방 인공지능 의사결정지원체계 사업 총괄
("지능형 전장인식 서비스 및 플랫폼/서비스 통합 기술" 등)

2023 ~

광운대학교 컴퓨터정보공학부 부교수

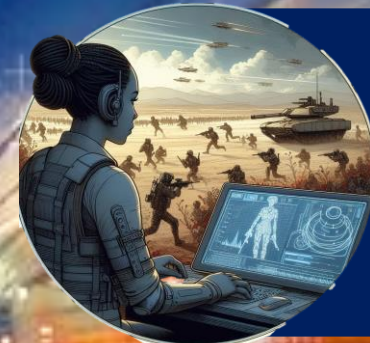
인공지능 의사결정 연구실

- 인간의 의사결정(Decision-Making)을 지원하고, 스스로 주어진 임무를 수행하는 인공지능 에이전트 기술 연구
 - ✓ 관련기술 : 생성형 AI, 자연어 처리, 그래프 신경망, Edge-Cloud AI Computing 등

정보검색 &
질의응답
서비스



전장상황
분석 &
지휘결심
지원



인공지능
비서



인공지능
조종사 &
무인군집로봇
제어



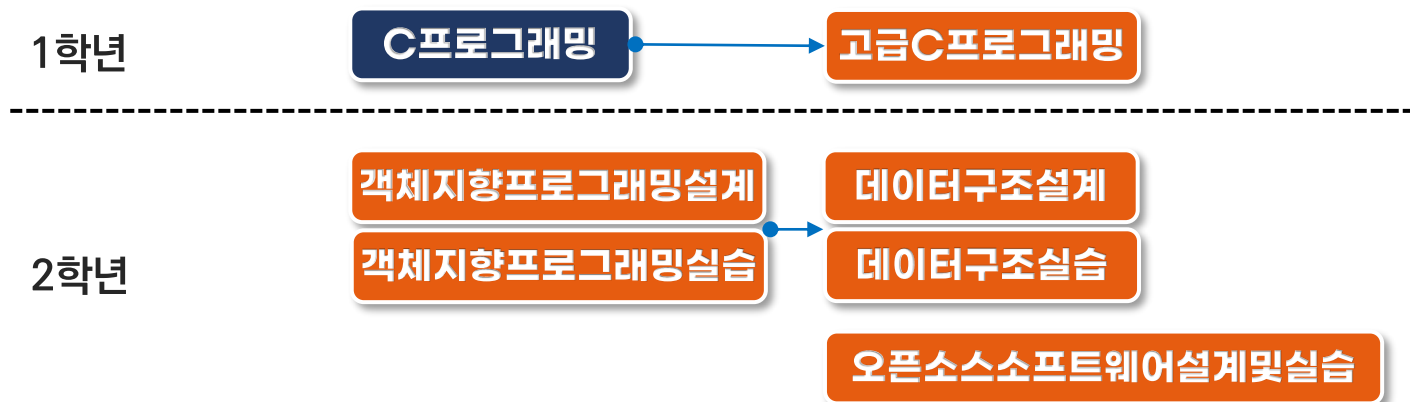
교과목 소개

• 교과목 개요

- ✓ 프로그래밍 언어에 대한 기본적인 이해도 향상 및 공학적 문제를 프로그래밍하는 기본적인 과정을 익힘
- ✓ 변수, 반복문, 조건문, 함수, 포인터 등 C언어의 기본적인 문법을 학습

• 프로그래밍 관련 기본 커리큘럼 (1/2학년 전공과목 기준)

- ✓ 1학년 2개 과목을 통해 C프로그래밍 학습 : 프로그래밍 기초
- ✓ 향후 다양한 프로그래밍 언어와 응용 프로그래밍(HW 포함) 학습 예정



수업과정 소개

- **총 15주 과정으로 구성 (강의계획서 참조)**
 - ✓ 대면 수업 : 12주
 - ✓ 중간/기말 고사 : 2주, 수업 휴강, 토요일 1/2분반 공통시험
 - ✓ 휴강 : 5/5(수)
 - ➔ 보강 : Quiz(오프라인 코딩테스트)로 대체(6/7 토요일)
- **수업진도**
 - ✓ 변수와 연산자 ~ 문자열, 포인터 기초
 - ✓ 2학기 고급C프로그래밍에서 이어서 진행
- **수업방식**
 - ✓ 이론강의와 예제실습 병행
 - ✓ 예제실습 결과물은 수업 종료 전 제출

평가 방안

- 학점부여 : 절대평가 A+/A B+/B F

- 평가방법 비율

출석	중간시험	기말시험	과제물	수업태도	Quiz
10	30	30	0	0	30

- F 부여기준

- ✓ 학교 규정상 $\frac{3}{4}$ 이상 출석하지 않을 경우 F 부여
- ✓ Quiz(코딩테스트) 미참여 시 F 부여
- ✓ 중간고사/기말고사 모두 미참여 시 F 부여
- ✓ 출석 및 코딩테스트, 중간고사/기말고사에 모두 참여하여도 일정기준 성적 미달 시 F 부여될 수 있음

Quiz(오프라인 코딩테스트) 개요

- **목적 : 프로그래밍 실습 및 평가 공정성 확보 (과제 copy 방지)**
- **진행방법 : 오프라인 환경에서 제시되는 문제 프로그래밍 및 제출**
- **문제범위 : 수업시간에 진행한 예제 응용**

02

C언어의 이해

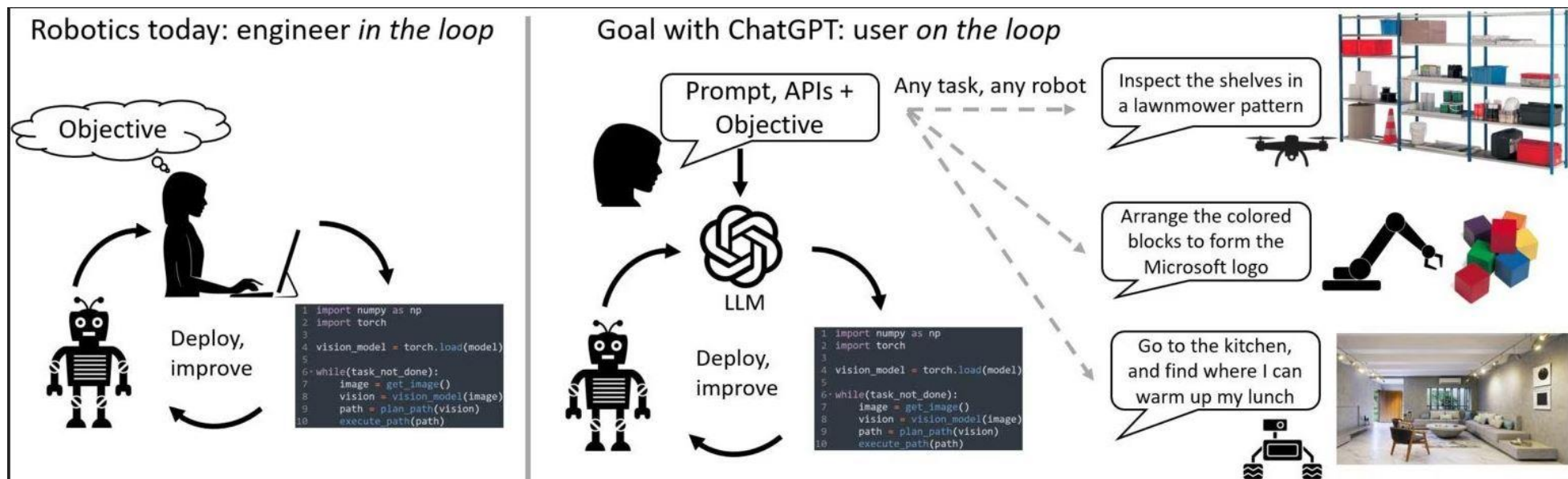
C언어의 이해 (1/6)

- 컴퓨터 프로그래밍 개요

- **컴퓨터에게 특정 결과를 달성할 수 있도록 실행 가능한 컴퓨터 프로그램을 설계, 구축하는 것**
- **프로그래밍 언어란?**
 - ✓ 컴퓨터 프로그램을 만들 때 사용하는 언어 : 인간과 컴파일러 모두 이해 가능
 - ➔ 컴파일러란? 프로그래밍 언어로 작성된 프로그램을 컴퓨터가 이해할 수 있는 기계어로 번역
 - ✓ 자연어 vs. 프로그래밍 언어
 - ✓ 기계어 \leftrightarrow 고급 프로그래밍 언어

자연어 vs. 프로그래밍 언어

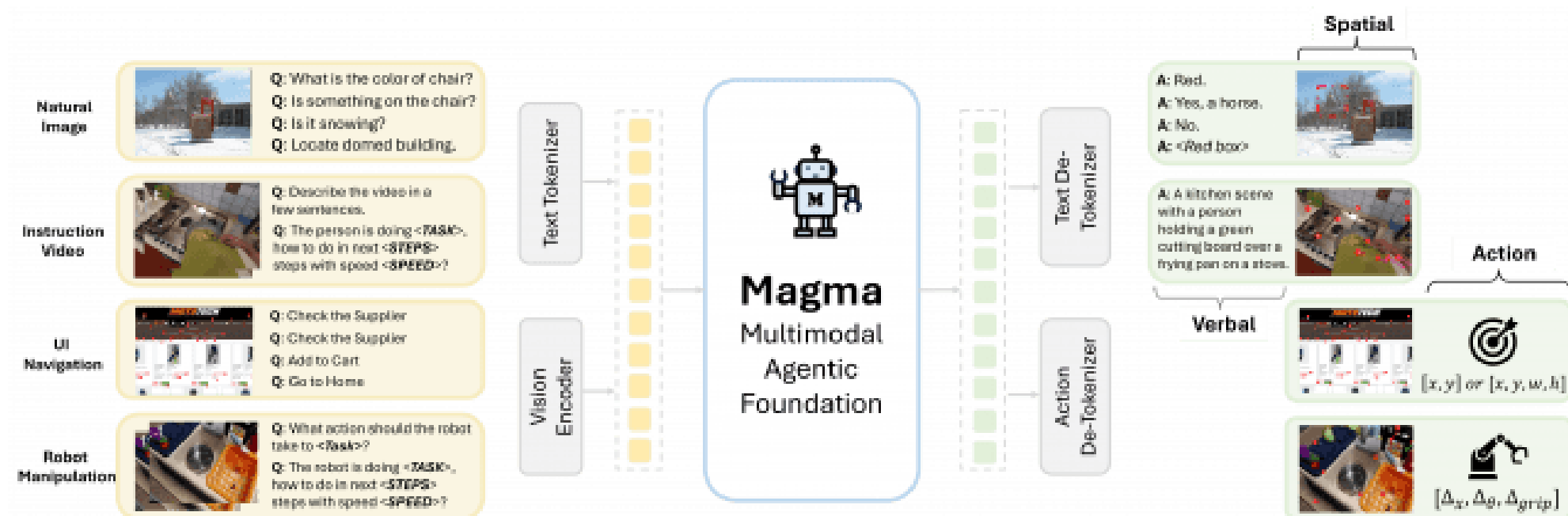
• MS의 ChatGPT를 이용한 로봇 제어 (2023.02)



자연어 vs. 프로그래밍 언어

• MS의 멀티모달 로봇 AI 모델 : Magma AI (2025.02)

- ✓ 단순히 언어와 이미지를 이해하는 수준을 넘어서 실제 환경을 해석하고, 주어진 목표를 스스로 계획해 실행 → Multi-modal AI 에이전트



C언어의 이해 (2/6)

- C언어 개요

- **1971년 UNIX 운영체제 개발을 위해 설계된 범용 고급 프로그래밍 언어**
- **C언어 등장 이전 : 어셈블리(assembly) 등 low-level 언어 사용**
- **C언어가 고급 프로그래밍 언어인가?**
 - ➔ HW(메모리 등)에 대한 직접적인 관리가 가능한 고급 언어
- **C언어를 지금도 사용하나요?**
 - ✓ 컴퓨터 자원이 열악한 환경에서 최적화된 프로그램을 개발하거나, 시스템 프로그래밍을 위해 사용 ex) 임베디드 프로그래밍 등
 - ✓ AI 학습/추론 등으로 인해 대용량 컴퓨팅이 요구되고, 이에 따라 최적화가 중요하기 때문에 더욱 많이 사용 됨

C언어의 이해 (3/6)

- C언어 개요

여기서 잠깐 C 탄생의 비밀

- C가 탄생하기 전인 1960년대 후반에는 프로그래머인 마틴 리처즈(Martin Richards)가 개발한 BCPL이라는 프로그래밍 언어가 있었음
- 벨연구소에서 근무하던 켄 톰프슨(Ken Thompson)은 BCPL을 기반으로 B라는 프로그래밍 언어를 개발했는데 이것이 바로 C의 모태
- 1970년대 초에 벨연구소에서 근무하던 데니스 리치(Dennis Ritchie)가 유닉스 운영체제를 개발하려고 C를 개발
- 데니스는 B를 바탕으로 C에 많은 기능을 추가했는데, 이 때문에 B보다 한 단 계 진보했다는 의미에서 알파벳의 B 다음인 'C'가 이름으로 붙었음
- C는 유닉스 운영체제 개발의 90% 이상을 담당하며 전 세계적으로 가장 많이 사용되는 프로그래밍 언어로 성장

C언어의 이해 (4/6)

- C언어 개요



- **Kenneth Lane Thompson, 1943년생**
- **2006년부터 구글에서 GO 언어 개발에 참여**
- **채용 시 구글의 필수 시험 항목인
C언어 시험을 거절**

C언어의 이해 (5/6)

- C언어 특징

- **이식성이 좋다**

- ✓ 서로 다른 CPU에서 사용 가능. 컴파일 필요. 많은 CPU 계열의 컴파일러 존재.
- ✓ But, 가장 이식성이 좋은가? NO! 예) JAVA

- **구문이 간결하다**

- **효율적인 언어이다**

- ✓ 현재 사용되는 고급 프로그래밍 언어 중에서는 가볍고 빠름

- **HW 접근/관리가 가능하다**

- ✓ 양날의 검 → 목적에 맞게 사용이 필요

C언어의 이해 (6/6)

- C 컴파일러의 종류

- **C 프로그래밍을 하려면 반드시 C 컴파일러가 필요함**

- ✓ 종류 : Turbo C, Turbo C++, Borland C++, MS C++, Visual C++, GCC, G++ 등

- **요즘의 C 컴파일러는 C와 더불어 C++도 동시 지원**

03

C 프로그램 작성 방법

1. 프로그램 코딩 (1/2)

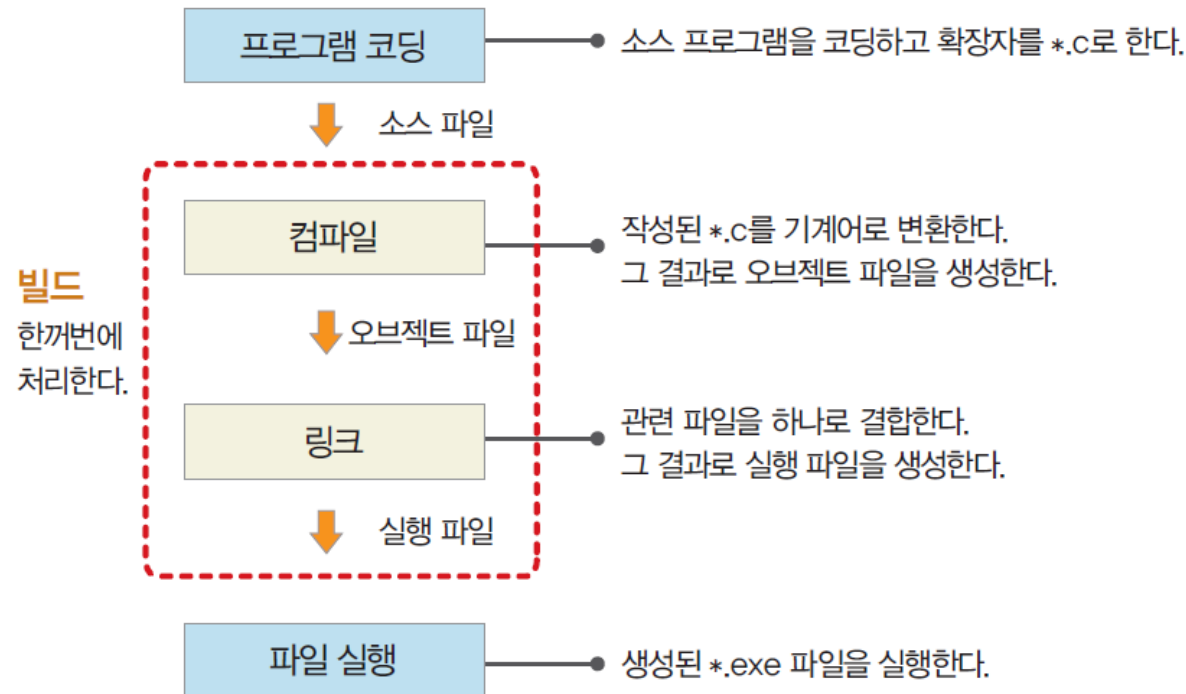


그림 1-4 C의 작성과 실행 순서

- ‘코딩(Coding)’이란 용어는 머릿속에 그려진 결과물을 만들기 위해 C 컴파일러가 알아들을 수 있는 형식으로 문서를 작성한다는 의미
- 사람만 이해하는 코딩

어이 컴퓨터~
100에서 50을 뺀 결과가 뭐지?
한번 모니터에 출력해봐~~

1. 프로그램 코딩 (2/2)

기본 1-1 C 프로그램 맛보기

1-1.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int result;
06     result = 100 - 50;
07     printf(" %d ", result);
08 }
```

- **100에서 50을 뺀 결과를 모니터에 출력하는 전형적인 C 프로그램**

2. 컴파일과 링크 (1/2)

- 컴파일 : 소스 파일을 컴퓨터가 이해할 수 있는 기계어로 바꾸는 과정



그림 1-5 컴파일의 개념

2. 컴파일과 링크 (2/2)

- 링크 : 여러 개의 '오브젝트 파일'을 하나의 '실행 파일'로 묶어주는 과정

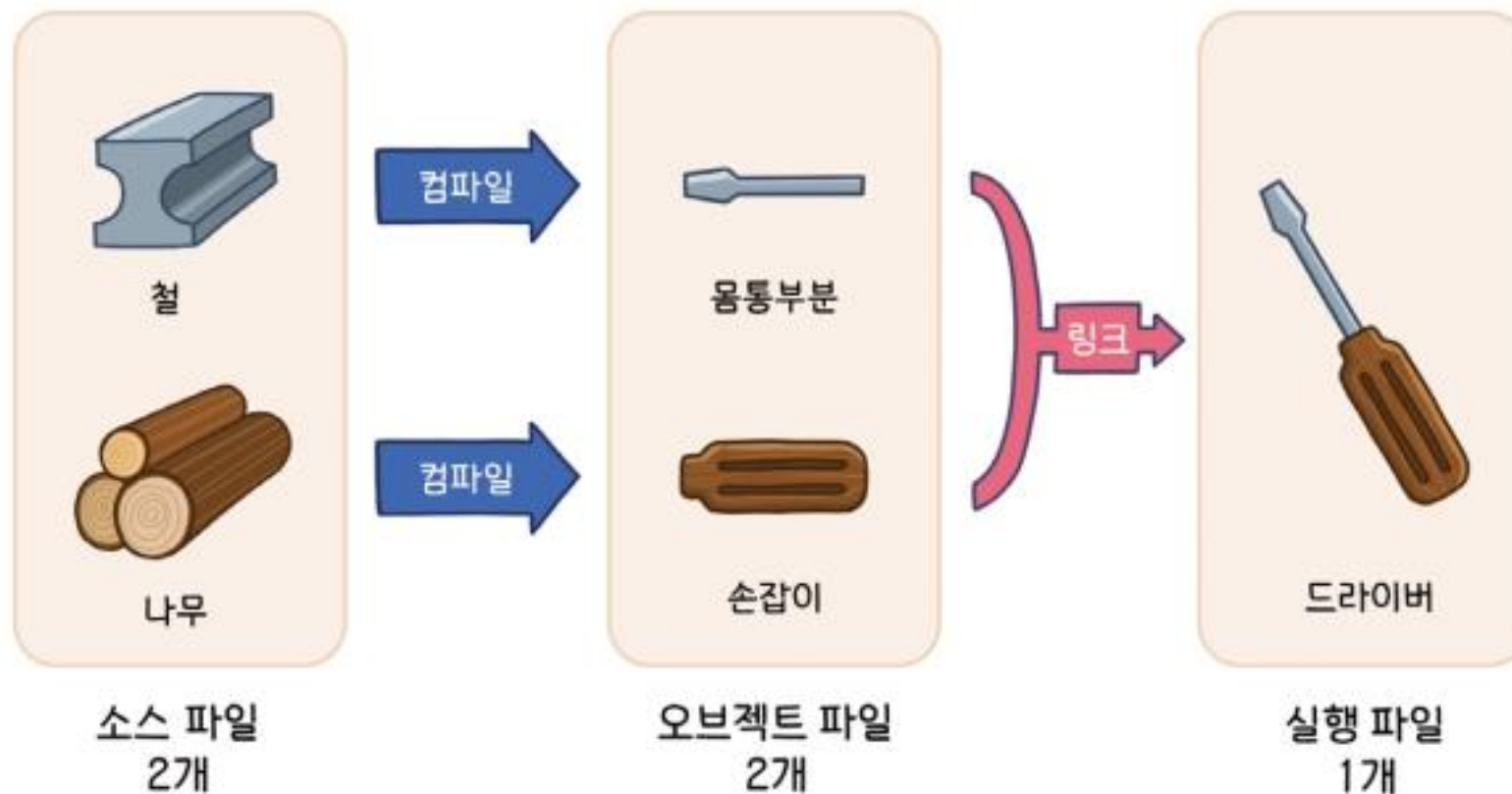
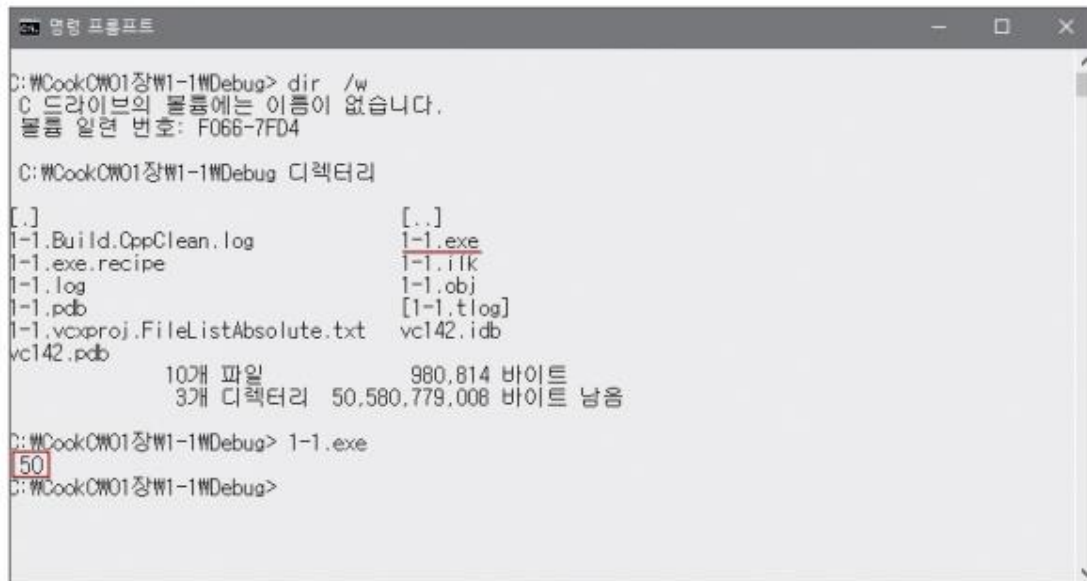


그림 1-6 컴파일과 링크의 개념

3. 프로그램 실행



```
C:\Cook\OW01\장\W1-1\Debug> dir /w
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F066-7FD4

C:\Cook\OW01\장\W1-1\Debug 디렉터리

[.]
1-1.Build.OppClean.log          1-1.exe
1-1.exe.recipe                 1-1.lib
1-1.log                        1-1.obj
1-1.pdb                        [1-1.tlog]
1-1.vcxproj.FileListAbsolute.txt vc142.idb
vc142.pdb
10개 파일                980,814 바이트
3개 디렉터리  50,580,779,008 바이트 남음

C:\Cook\OW01\장\W1-1\Debug> 1-1.exe
50
C:\Cook\OW01\장\W1-1\Debug>
```

그림 1-7 컴파일과 링크가 완료된 1-1.exe 파일

여기서 잠깐 빌드=컴파일+링크

- 컴파일과 링크는 소스 파일이 하나뿐이더라도 반드시 수행해야 함
- 대부분의 컴파일러는 컴파일과 링크 과정을 별도로 처리하지 않음
- 그래서 컴파일과 링크를 합쳐서 컴파일 또는 빌드(build)라고 함
- 필요에 따라 컴파일과 링크를 별도로 수행할 수도 있음

Q & A