

- ▶ 용어 정리
- ▶ 프로그래밍 언어의 종류
- ▶ 프로그램이 만들어지는 과정
- ▶ 컴퓨터의 구조
- ▶ 폰 노이만 구조
- ▶ 프로그램의 동작 원리



Artificial Intelligence Laboratory

# 용어 정리

# 프로그래밍이란?

## ■ 프로그램(program)

- ▶ 컴퓨터를 실행시키기 위해 차례대로 작성된 명령어 모음

## ■ 프로그래밍(programming)

- ▶ 프로그램을 만드는 작업

## ■ 코딩(coding)

- ▶ 프로그램의 명령어를 작성하는 행위

## ■ 프로그래밍 언어(programming language)

- ▶ 프로그램 작성을 위해 인위적으로 만들어낸 언어
- ▶ C, C++, C#, Java, JavaScript, Python 등
- ▶ 프로그래밍 언어는 각자의 개성과 장단점이 있다.

Artificial Intelligence Laboratory

# 프로그래밍 언어의 종류

# 프로그래밍 언어의 종류

## 기계어(machine language)

- 2진 숫자(binary digit, 0과 1)로 나타낸 언어

## 어셈블리어(Assembly Language)

- 기계어를 1:1로 대응하는 단어로 바꿔 놓은 언어. 번역이 간단하다.

Machine code	Assembly code	Description
001 1 000010	LOAD #2	Load the value 2 into the Accumulator
010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
001 1 000101	LOAD #5	Load the value 5 into the Accumulator
010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
111 0 000000	HALT	Stop execution

## I 고급 언어(High-Level Language)

- 사람이 사용하기 편하도록 수식 등을 사용하는 언어.
- 대부분의 프로그램은 고급 언어로 만든다.

```
1      #include <stdio.h>
2
3      int main(void)
4      {
5          int num;
6
7          printf("입력하세요:");
8          scanf_s("%d", &num);
9
10         printf("\n\n입력 받은 값: %d", num);
11         return 0;
12     }
```

Artificial Intelligence Laboratory

# 프로그램이 만들어지는 과정

# C 프로그램의 개발 및 실행 과정(수정)

## 소스 파일(source code) 작성

- .c 확장자를 갖는 파일 작성

## 컴파일(Compile)

- 소스 코드를 기계어로 번역하여 오브젝트 파일 생성
- 컴파일을 수행하는 프로그램을 컴파일러라고 함

## 링크(Link)

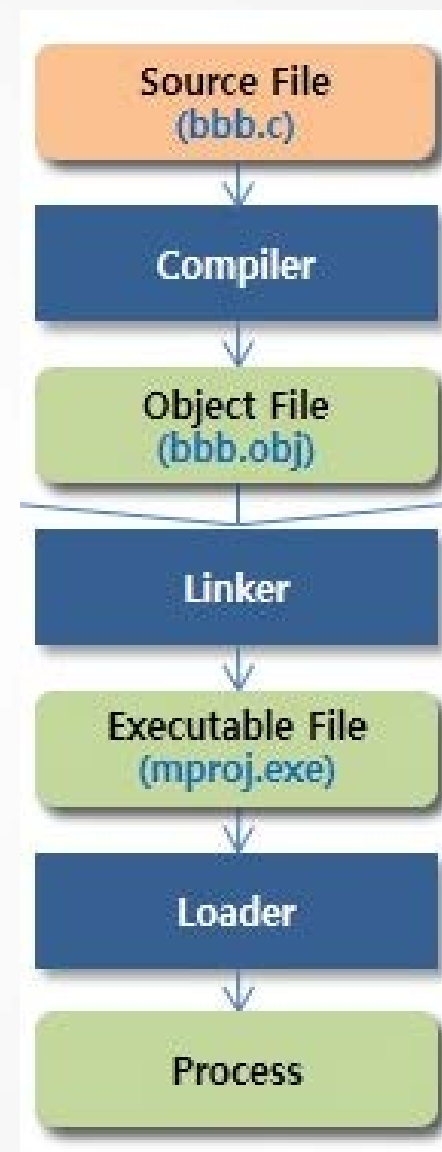
- 오브젝트 파일들을 연결해서 실행파일 생성
- 이런 역할을 하는 프로그램을 링커라고 함

## 로드(Load)

- 실행 파일을 메모리에 적재해서 구동 시키는 단계
- 이런 역할을 하는 프로그램을 로더라고 함

## 프로세스(Process)

- 현재 실행 중인 기계어 프로그램.





Artificial Intelligence Laboratory

# 컴퓨터의 구조

## 컴퓨터의 구조는 사람과 유사하다

### CPU (Central Processing Unit, 씨피유)

- ▶ 두뇌가 모든 지체에 명령을 내리듯, CPU가 모든 구성품에 명령을 내림.

### Memory (주 기억 장치)

- ▶ 기록하지 않으면 그 당시에만 기억하고 금새 잊어 버리듯, 컴퓨터의 Memory도 일시적인 기억 공간이다.

### 보조 기억 장치 (Secondary Storage)

- ▶ 한번 기록해두면 다시 찾아보고 기억해낼 수 있듯, 컴퓨터의 보조 기억 장치는 반영구적인 저장 공간이다.

### 입출력 장치(input and output devices)

- ▶ 눈과 귀로 상황을 인지하고, 손과 입 등으로 생각을 전달하듯, 정보를 받아들이거나 출력하는 역할을 하는 구성요소이다.



# CPU(Central Processing Unit)

■ 컴퓨터의 두뇌에 해당한다.

● i3, i5, i7 등의 용어가 CPU를 가리키는 이름이다.

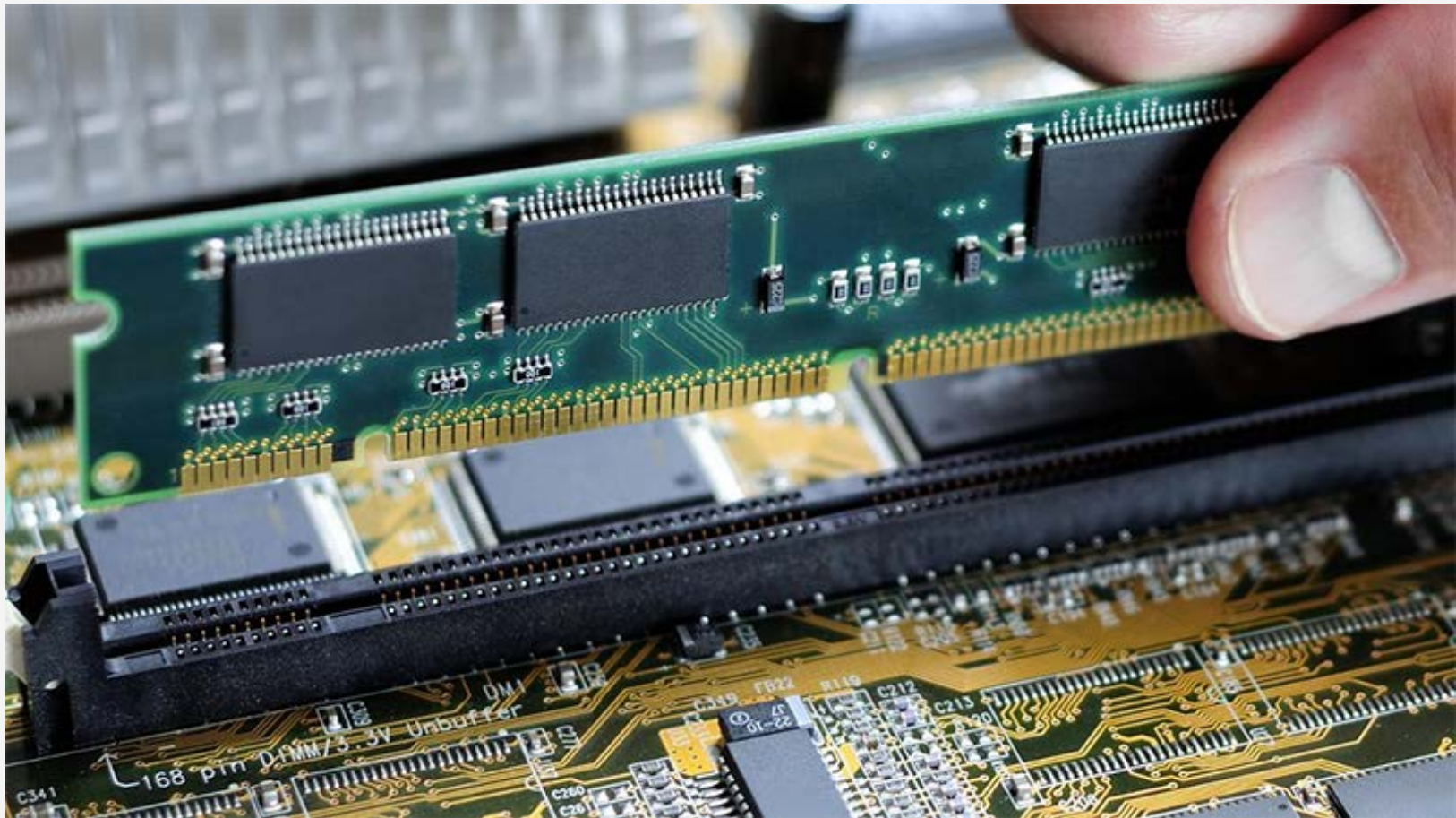




# 메모리(주 기억 장치)

■ 전류가 차단 되면 데이터가 증발한다.

● “우리집은 램이 8기가다.” 는 메모리의 용량이 8기가라는 의미이다.



# 보조 기억 장치(하드 디스크 등)

- 휘발성인 메모리 대신 데이터를 오랫동안 저장할 수 있는 장치.
  - 값이 싸다. 4TB가 몇 만원밖에 안 한다.



# 입출력 장치

## 입력 장치

- 키보드, 마우스, 태블릿 등

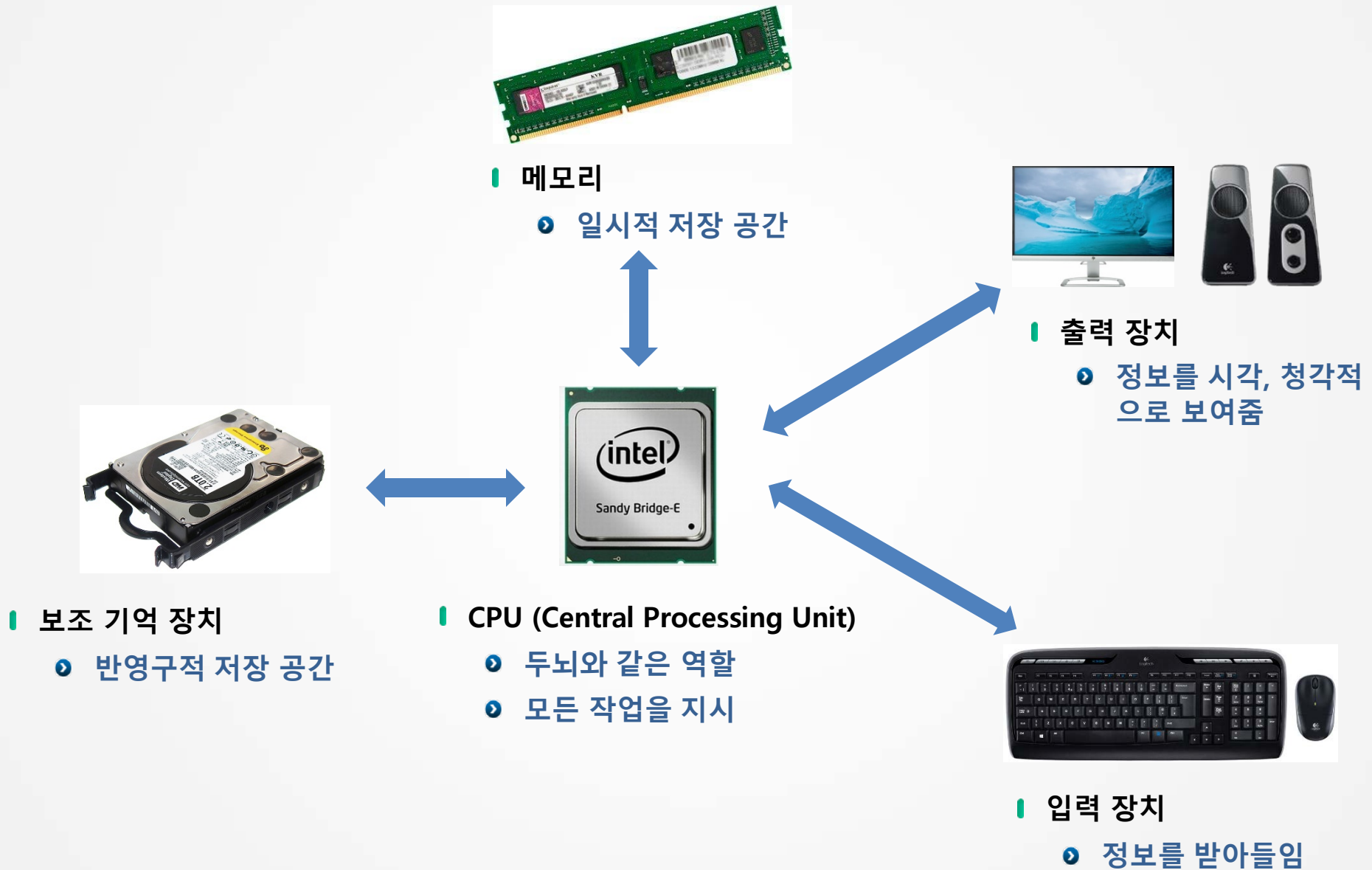
## 출력 장치

- 모니터, 스피커 등





# 현대 컴퓨터의 구조



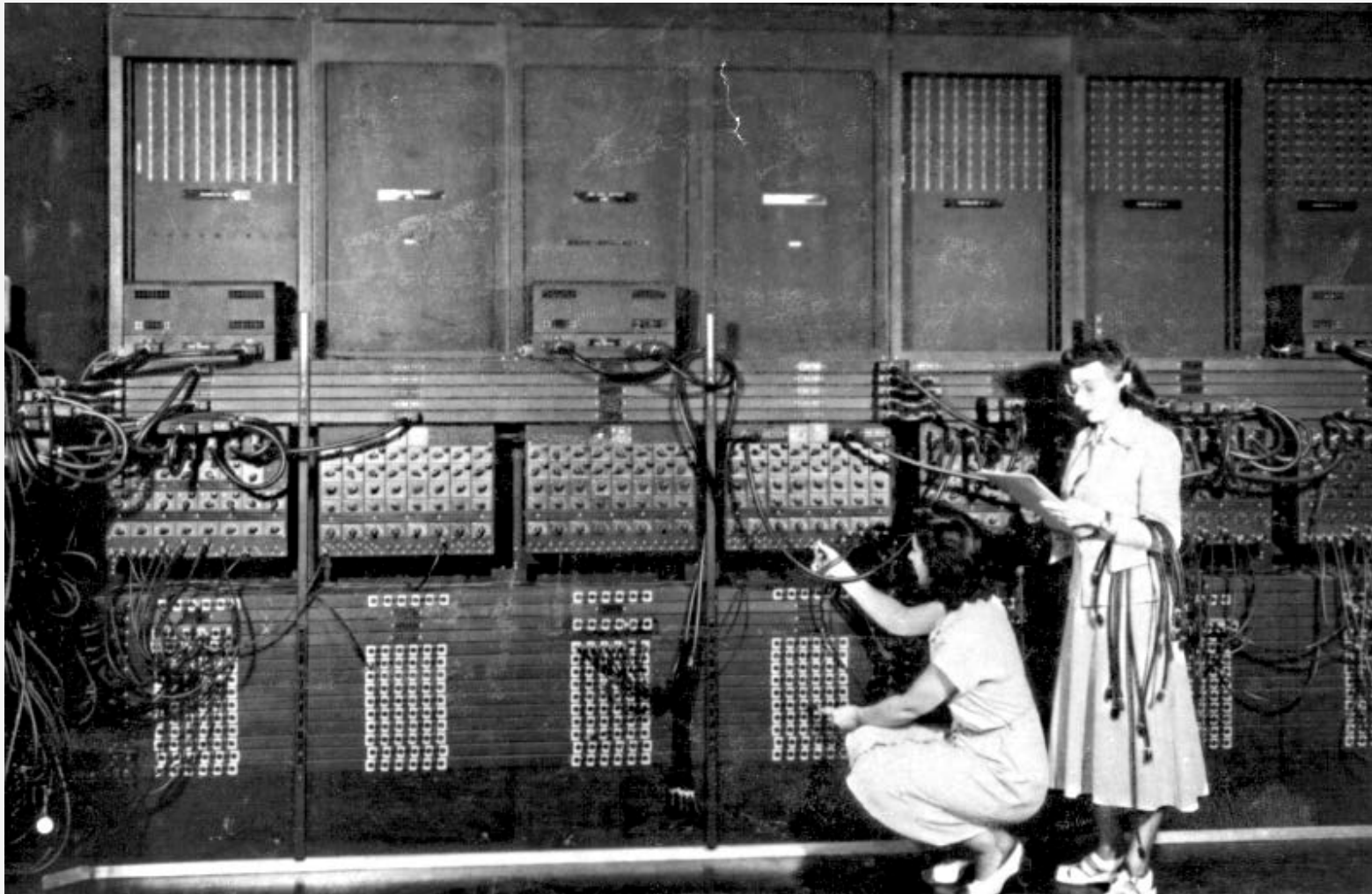
Artificial Intelligence Laboratory

# 폰 노이만 구조



# 과거의 컴퓨터, ENIAC

- 이전의 컴퓨터는 전선을 꽂는 방식이었다.
- 최초의 컴퓨터로 불리는 ENIAC은 이 방식으로 작동되었다.



# 폰 노이만 구조

전기적 정보를 저장할 수 있는 수단을 사용하자!

- 메모리는 전기적 정보를 저장할 수 있는 매체이다.
- 메모리에 저장된 명령을 CPU가 하나씩 가져와서 컴퓨터를 작동시킴.



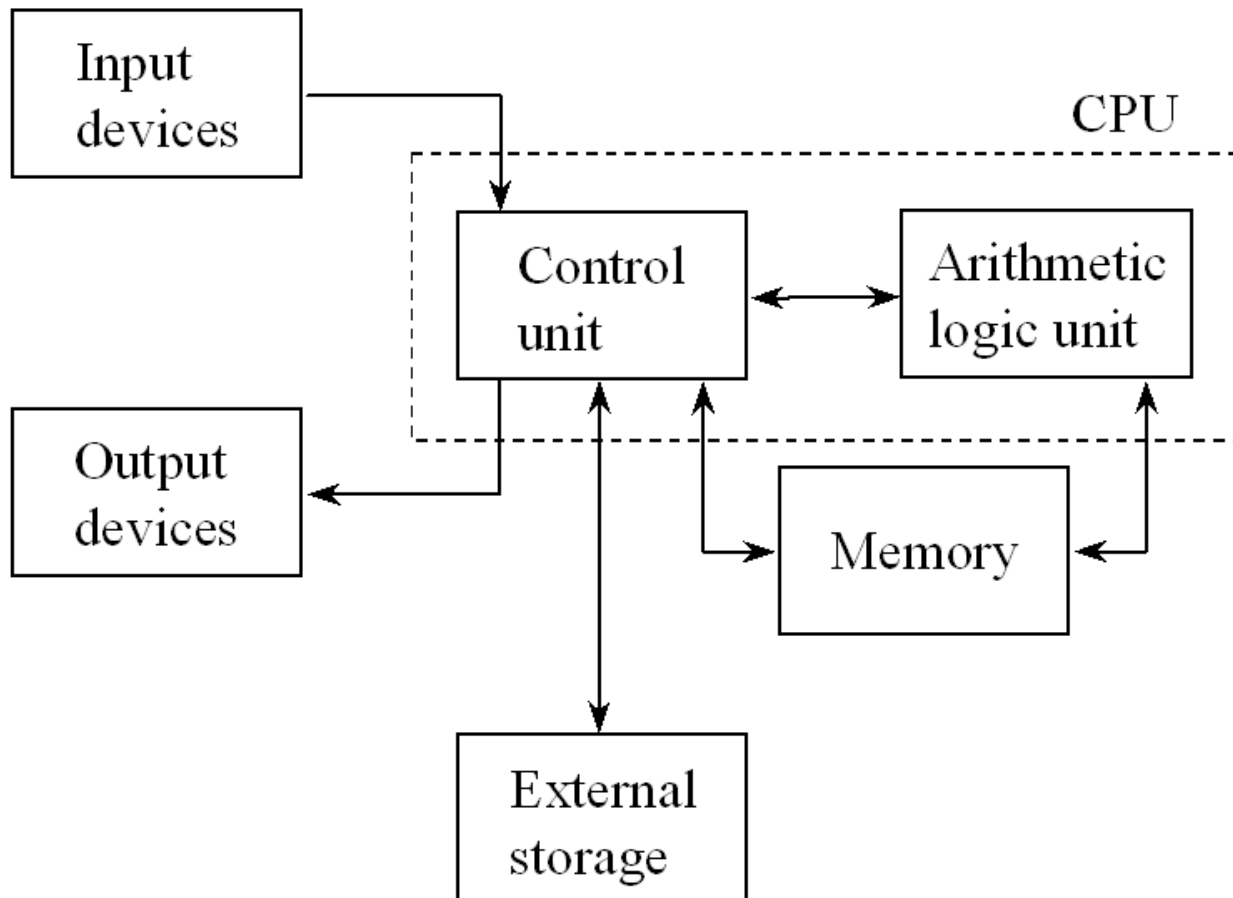
**CPU**



**Memory**

# 폰 노이만 구조

- 현대 컴퓨터는 훨씬 복잡해졌지만 모두 폰 노이만 구조를 따른다.

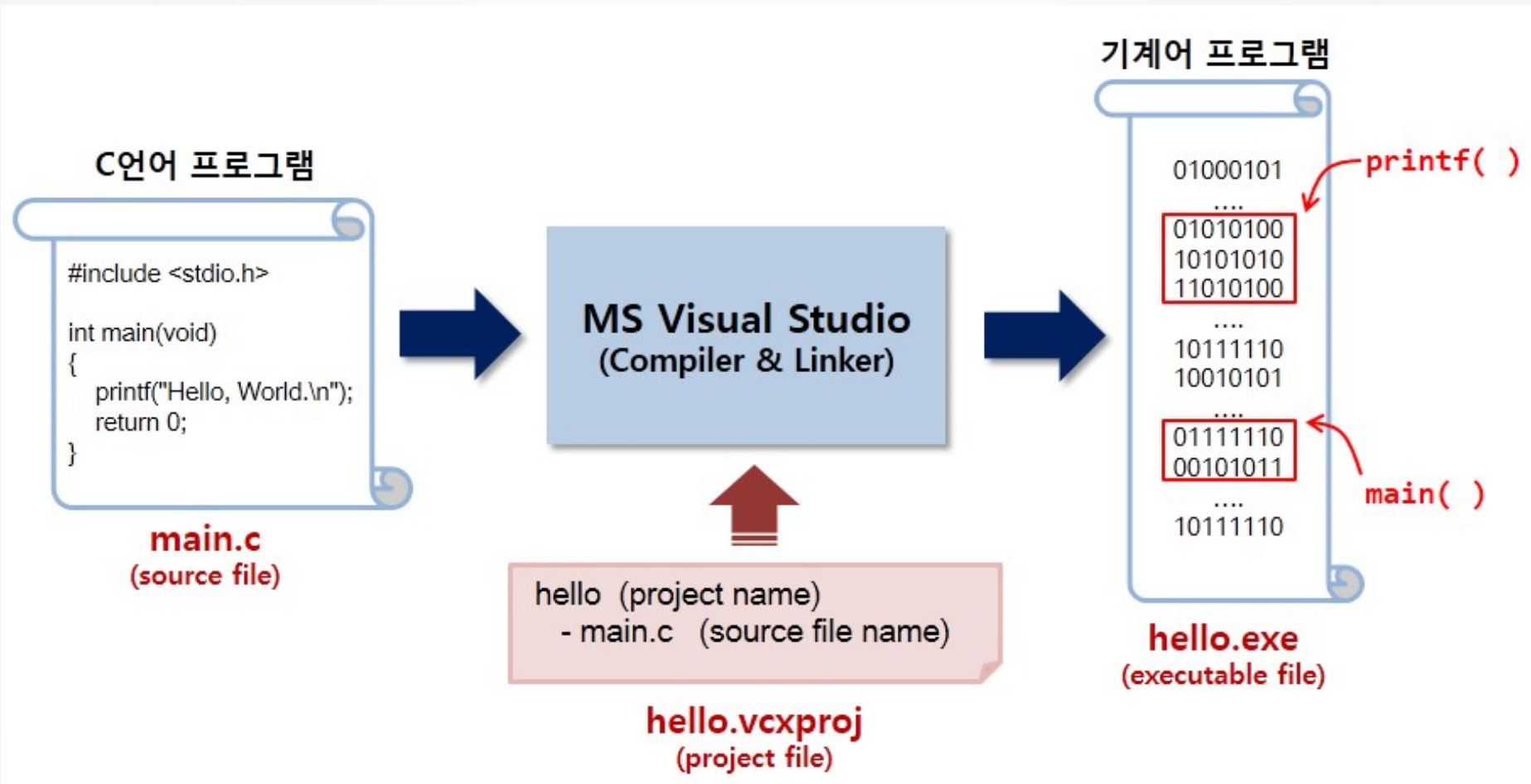


Artificial Intelligence Laboratory

# 프로그램의 동작 원리

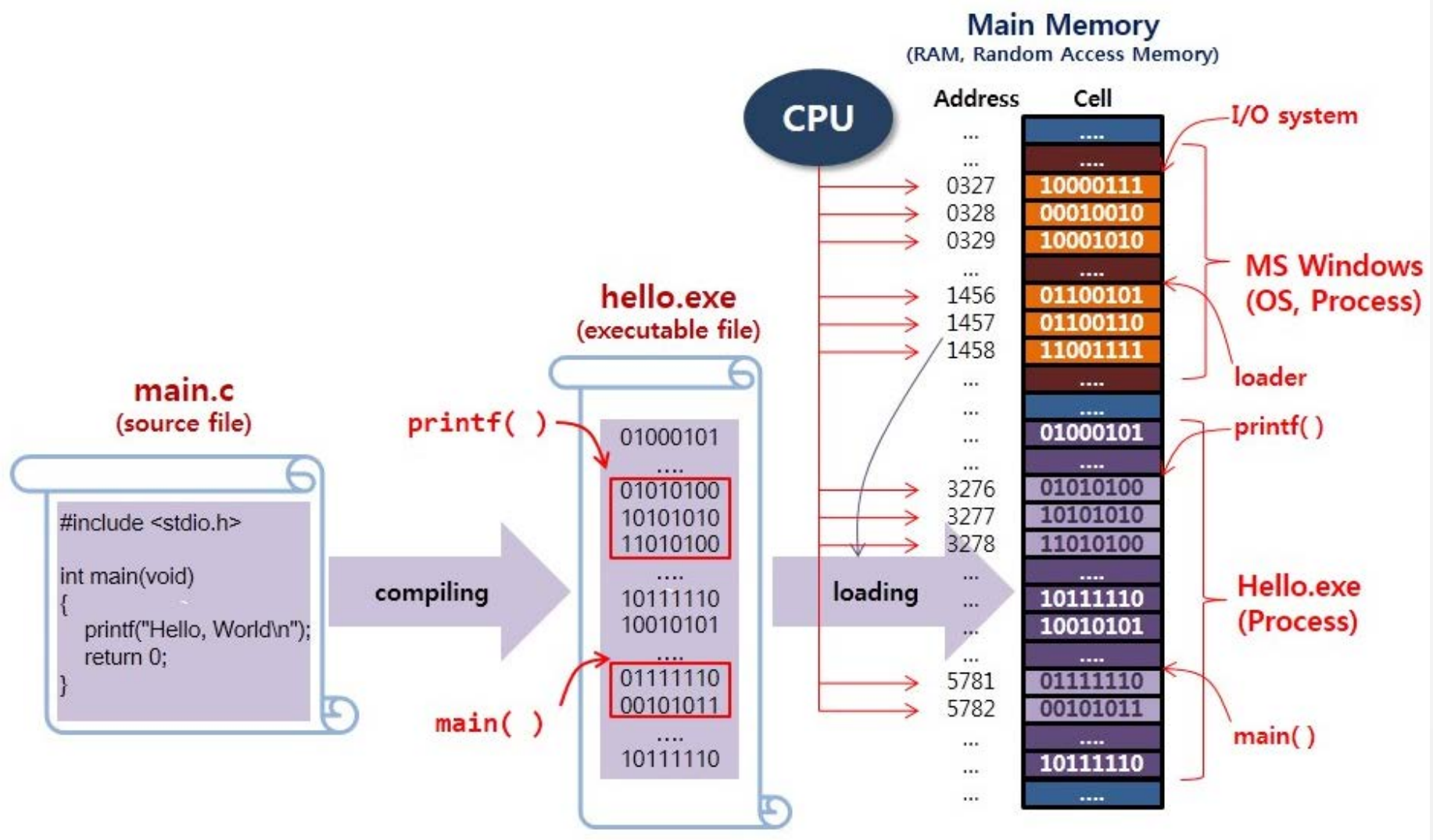
# 프로그램의 동작 원리

1. 컴파일, 링크 과정을 통해 소스 파일로부터 실행 파일을 만든다.



# 프로그램의 동작 원리

## 2. 메모리에 적재 시킨 후, CPU가 이를 수행한다.



- 3. CPU가 프로그램 내의 명령어에 따라 컴퓨터를 동작 시킨다.



# <핵심 요약>

## 프로그래밍 언어에는 세가지가 있다.

- ◆ 기계어
- ◆ 어셈블리어
- ◆ 고급 언어

## 컴퓨터의 구성 요소는 다음과 같다.

- ◆ CPU : 두뇌의 역할.
- ◆ Memory : 일시적인 저장 공간. CPU는 이곳에 저장되어 있는 프로그램을 실행시킴.
- ◆ 보조 기억 장치 : 반영구적 저장 공간.
- ◆ 입출력 장치 : 키보드, 마우스, 모니터, 스피커 등.



# <핵심 요약>

- 컴파일, 링크 과정을 통해 실행 파일(exe)을 만든다.
  - ◆ 우리가 작성한 코드는 소스 코드(source code)라고 한다.
  - ◆ 컴파일(compile)은 소스 코드를 기계어로 번역하는 작업이다.
  - ◆ 링크(link)는 여러 오브젝트 파일을 하나의 실행 파일로 합치는 작업이다.
  - ◆ 실행 파일(executable file)은 온전한 하나의 프로그램이다.
- 현대 컴퓨터는 모두 <폰 노이만 구조>를 따른다.
  - ◆ 프로그램은 메모리에 저장되고, 이를 CPU가 차례로 읽어서 컴퓨터를 작동시킨다.