



한양대학교
지능통신시스템 연구실

2023 임베디드 AI 연구

강 사

최병찬 (010-4296-3496)

(luwis93@naver.com)



한양대학교
지능통신시스템 연구실

목 차

임베디드 HW 기초

컴퓨터 운영체제 기초

임베디드 HW 기초

임베디드 HW 기초

① 임베디드 시스템 (Embedded System)

- ❖ 특정한 제품이나 솔루션에서 **주어진 특수한 기능을 중점적으로 수행하도록 탑재되는 솔루션이나 시스템**
(※ 출처 : 네이버 지식백과, 두산 백과)
- ❖ 임베디드 시스템은 PC와 달리 범용적인 목적이 아닌 **소수의 특수한 목적 또는 프로그램 수행을 위해서 사용함**
ex : PC - 인터넷 + 동영상 + 게임 + 문서 작업 등을 동시에 같이 수행 가능
ex : 임베디드 시스템 - 유무선 통신 / 영상 처리 / 모터 제어 등 소수 특수한 목적에 집중함
- ❖ 최근 반도체 분야의 기술 혁신으로 인해 **On-Board 기능과 스펙이 비약적인 발전을 이룸**
- ❖ 과거 임베디드 보드에는 간단한 단일 Loop 프로그램만 들어갈 수 있었다면,
최근 임베디드 시스템은 SBC (Single Board Computer) 등장으로 RTOS, Linux, Android 등 OS급 프로그램을 넣어 사용할 수 있는 수준까지 올라옴.



STM32F429I-DISC1
ARM 개발보드



Arduino Uno MCU



Raspberry Pi



Nvidia Jetson Nano

최근 임베디드 SW · HW 설계는 **Open-Source · Open-Hardware 중심의 개발 철학**이 형성되었으며,
사용하기 쉬운 다양한 임베디드 플랫폼이 등장하고 있음.
현재 주요 트렌드는 **오픈 소스 HW 플랫폼을 기반으로**
오픈 소스 SW 이용한 고성능 · 고기능 소프트웨어 (인공지능, 실시간 고용량 센서 처리 등) 개발

임베디드 HW 기초

② MCU (Micro-Controller Unit)

- ❖ Microprocessor, I/O, Memory 등을 하나의 칩 또는 보드로 구성한 제품 (출처 : 네이버 지식 백과)
- ❖ MCU는 다양한 I/O (LED, 모터, 버튼 등) 제어에 특화되어 사용됨
- ❖ 소수의 특수한 목적을 수행하기 위해 만드는 임베디드 시스템을 위해 MCU를 설계함
- ❖ MCU는 PC와 달리 **범용성이 낮기에** 고성능의 하드웨어를 요구하지 않음.
그러나 I/O에 직접 연결되어 제어하기에 반응속도가 비교적 빠름.
- ❖ MCU는 특수한 기능을 고속으로 수행하기 위해 PC 또는 SBC (Single Board Computer)와 같은 범용성 플랫폼과 분리되어 사용됨.
- ❖ 주로 MCU라 부르지만, 아주 가끔씩 'Micom (마이콤 / Micro-Computer)'라 부르는 경우도 있음.



Intel 8051



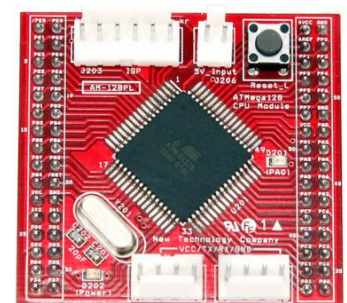
Atmel사 AT89S52



STM32F429I-DISC1
ARM 개발보드



Arduino Uno MCU



NEWTC사 Atmega 128

임베디드 HW 기초

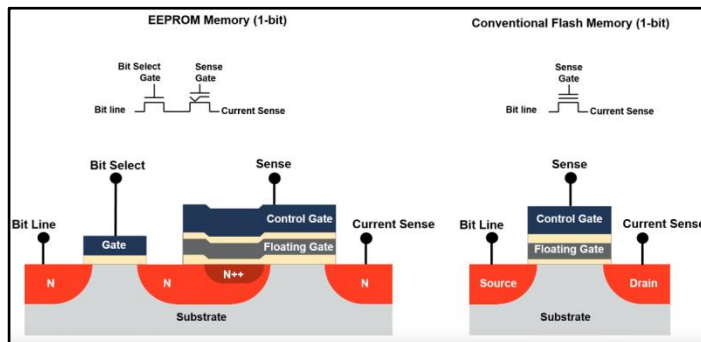
③ EEPROM

- ❖ Electronically Erasable Programmable Read-Only Memory
- ❖ 비휘발성 메모리(Non-Volatile Memory)로 전기적 신호를 이용하여 데이터 저장 · 유지하는 방식
- ❖ 제작한 SW 코드가 저장되는 'Code Memory' 공간으로 사용됨

④ FLASH Memory

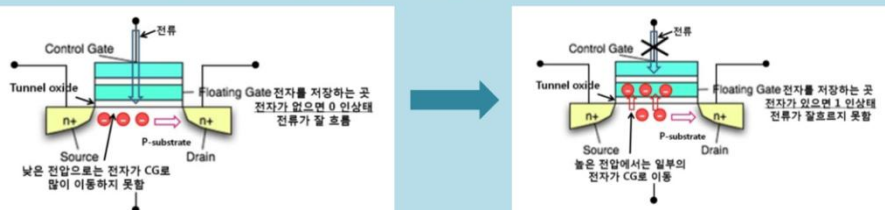
- ❖ EEPROM를 기반으로 발전된 비휘발성 메모리 (Non-Volatile Memory)
- ❖ 전기적 신호를 이용한 고속 Read/Write 목적으로 사용됨
- ❖ 제작한 SW 코드가 저장되는 'Code Memory' 공간으로 사용됨

※ MCU의 EEPROM 또는 FLASH 메모리에 소스 코드를 올리는 행위를 'Flashing'이라고 주로 지칭함



❖ Flash Memory의 원리

- 빠른 속도로 신호를 주고 끊어서 전하가 중간에 빠져서 움직이지 못하게 한다.

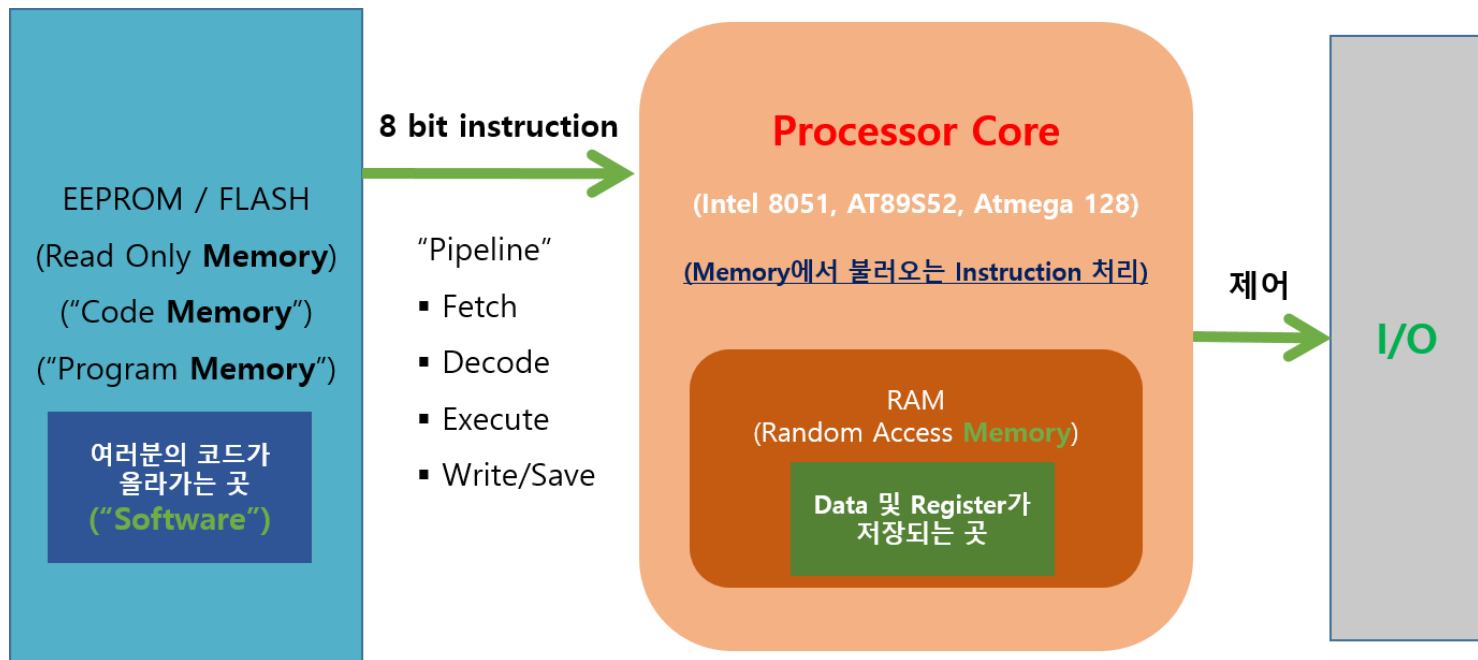


※ 참고자료 : What is EEPROM? <https://www.youtube.com/watch?v=7qa1dsCpMMo>
What is Flash Memory? <https://www.youtube.com/watch?v=ELI3abwYQ90>

임베디드 HW 기초

⑤ RAM (Random Access Memory)

- ❖ 휘발성 메모리 (Volatile Memory)로 프로그램의 일시적 로딩, 데이터의 일시적 저장 등을 위해서 사용함
- ❖ 임베디드 시스템에서는 프로그램의 변수 (전역, 지역, 동적할당) 관리와 Register 설정을 위해서 사용함
- ❖ RAM 크기에 따라 접근할 수 있는 메모리 주소의 크기가 결정됨



컴퓨터 운영체제 기초

컴퓨터 운영체제 기초

① 운영체제 (OS : Operating System)

- ❖ 사용자의 **하드웨어, 시스템 리소스를 제어**하고 **프로그램에 대한 일반적 서비스를 지원**하는 시스템 소프트웨어이다. 하드웨어를 관리할 뿐 아니라 **응용 소프트웨어를 실행하기 위하여 하드웨어 추상화 플랫폼과 공통 시스템 서비스를 제공한다.**
- ❖ 입출력과 메모리 할당과 같은 하드웨어 기능의 경우 **운영체제는 응용 프로그램과 컴퓨터 하드웨어 사이의 중재 역할**을 한다.
- ❖ 운영체제는 실행되는 응용 프로그램들이 **메모리와 CPU, 입출력 장치 등의 자원들을 사용할 수 있도록** 만들어 준다. 더불어, 이들을 추상화하여 **파일 시스템** 등의 서비스를 제공한다.
- ❖ 멀티태스킹 경우, **여러 개의 응용 프로그램들을 스케줄링**하여 마치 그들이 동시에 수행되는 것처럼 보이는 효과를 낸다.

※ 출처 : Wikipedia – 운영체제

(<https://ko.wikipedia.org/wiki/%EC%9A%B4%EC%98%81%EC%B2%B4%EC%A0%9C>)

- ❖ 임베디드 시스템과 달리 OS는 **다양한 프로그램 실행을 위해서 사용함**

ex : PC – 인터넷 + 동영상 + 게임 등을 동시에 같이 수행 가능

ex : 임베디드 – 통신 / 영상 처리 / 모터 제어 등 소수 특수한 목적에 집중

**자원 = DRAM 메모리 사용량 / SSD 저장용량 / CPU 점유율 등
프로그램 실행을 위한 연산 및 저장 소요**

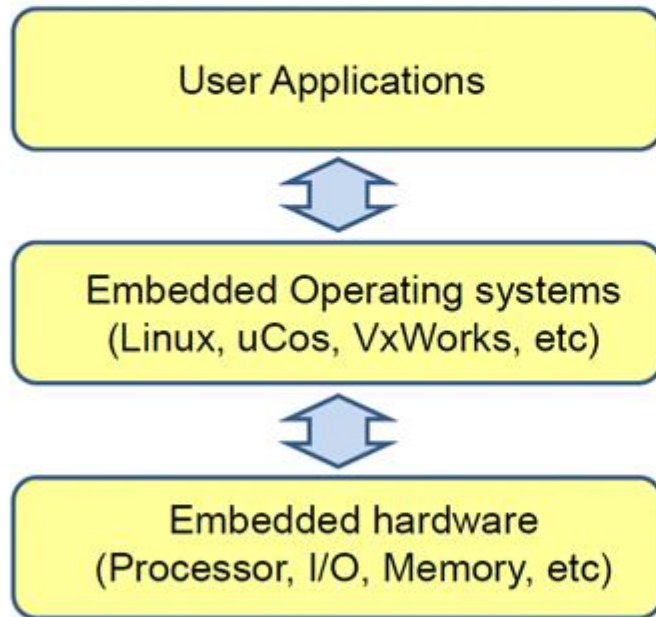
OS = 자원 관리를 통한 자동화된 프로그램 실행 관리자



(※ 출처 : Wikipedia – 운영체제)

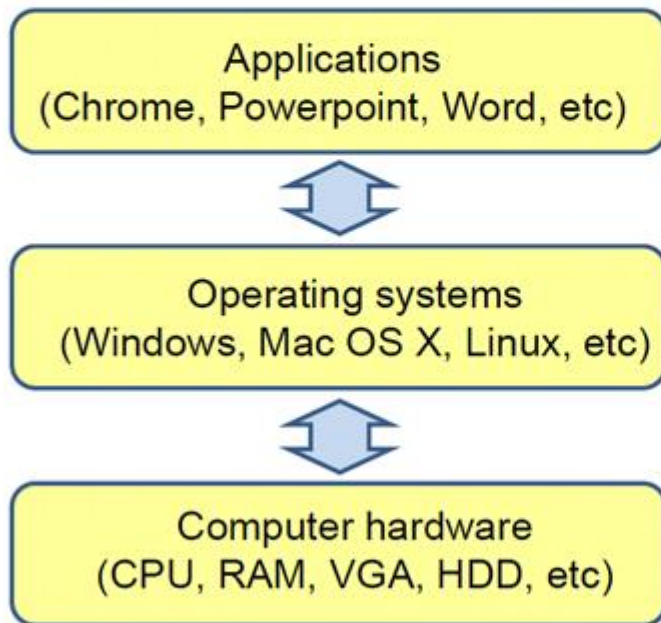
컴퓨터 운영체제 기초

- Embedded systems



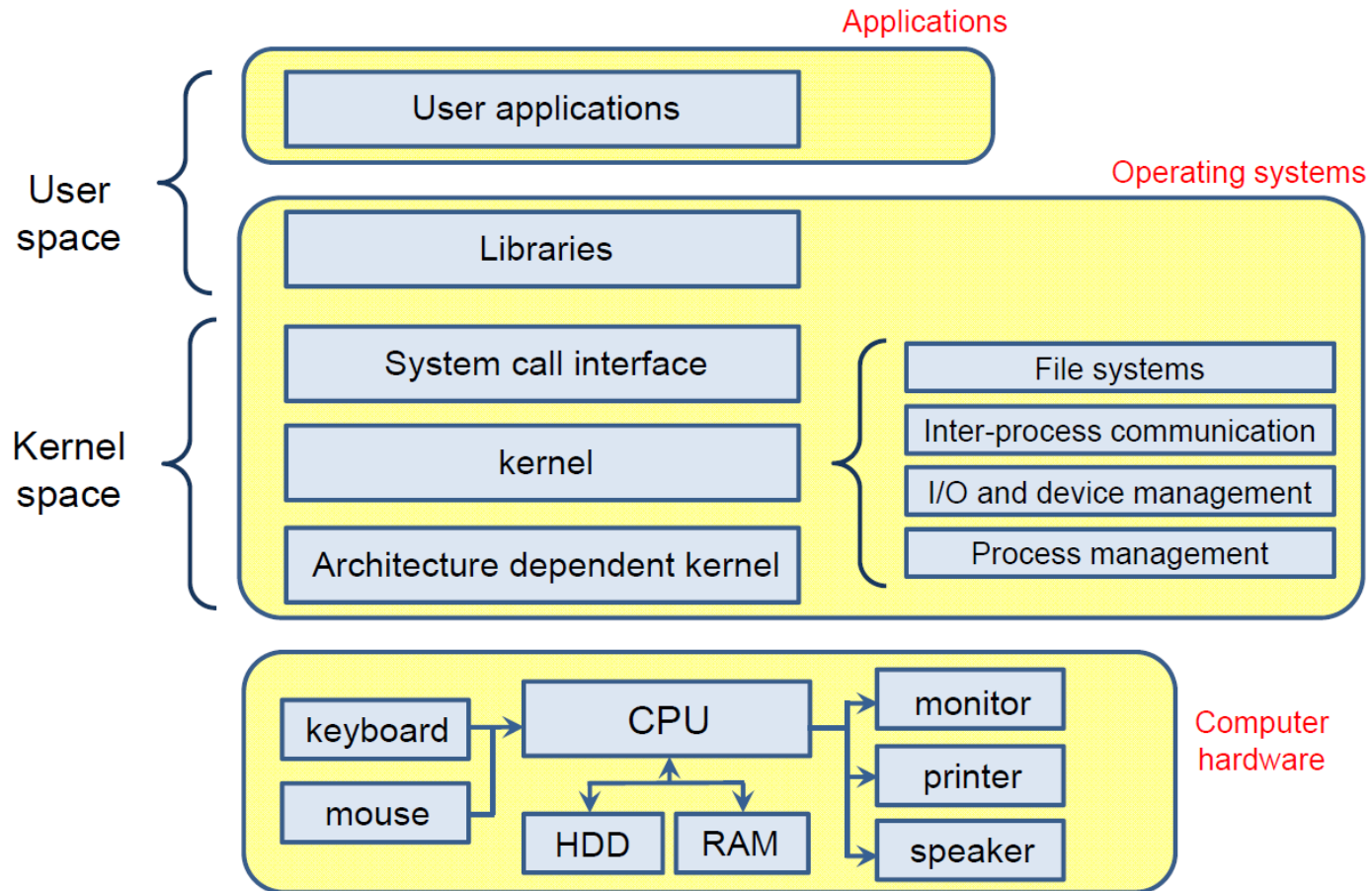
컴퓨터 운영체제 기초

- Computer systems



컴퓨터 운영체제 기초

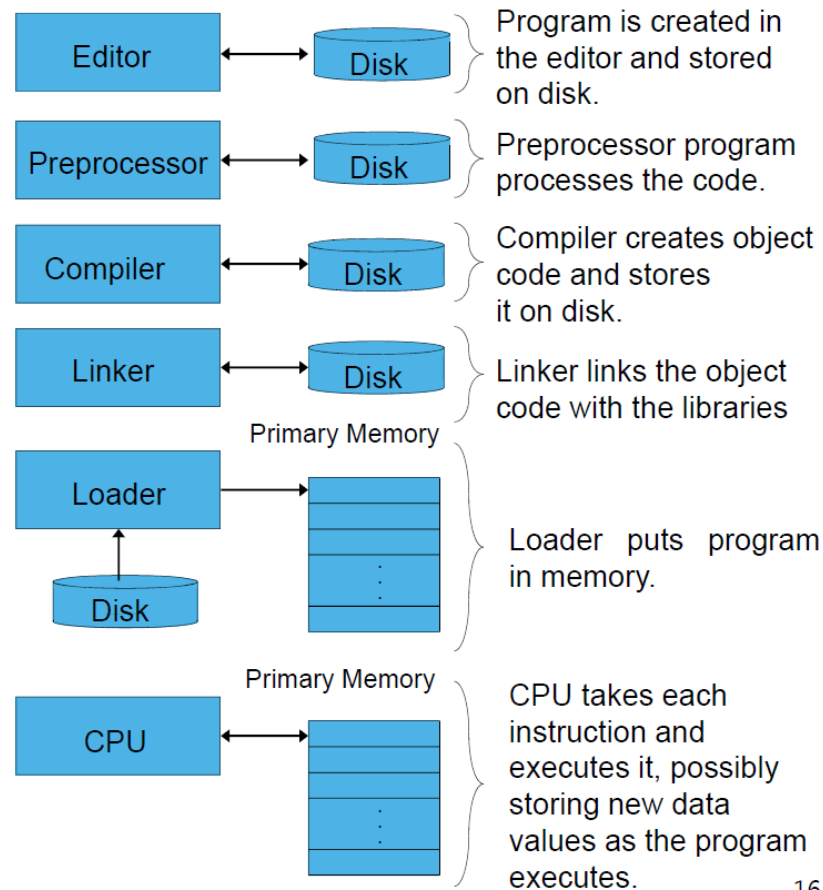
② 운영체제 구조



컴퓨터 운영체제 기초

③ Compile / Link 과정 - 소스코드는 어떻게 실행파일 / 프로그램이 되는가?

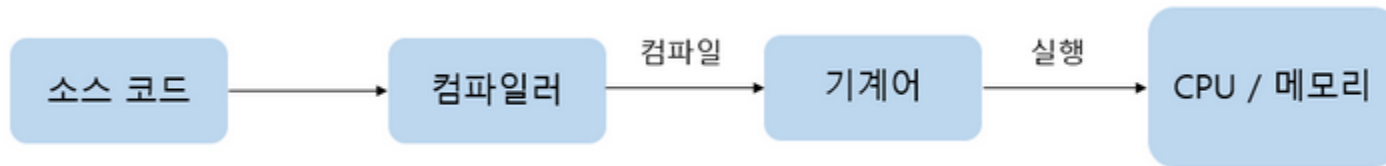
- Phases of C++ (or C) Programs:
 - Edit
 - Preprocess
 - Compile
 - Link
 - Load
 - Execute



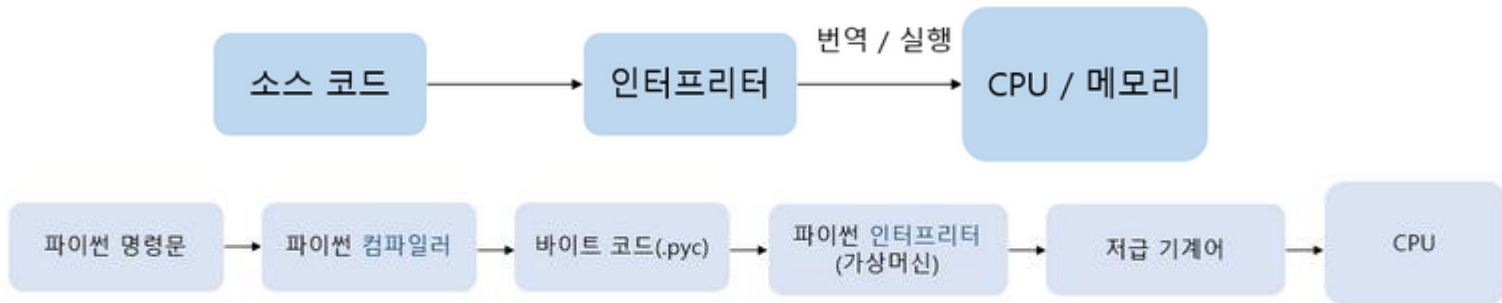
컴퓨터 운영체제 기초

③ Compile / Link 과정 - 소스코드는 어떻게 실행파일 / 프로그램이 되는가?

[컴파일 언어 / Native 언어] (ex : C/C++)



[인터프리터 / Script 언어] (ex : Python)

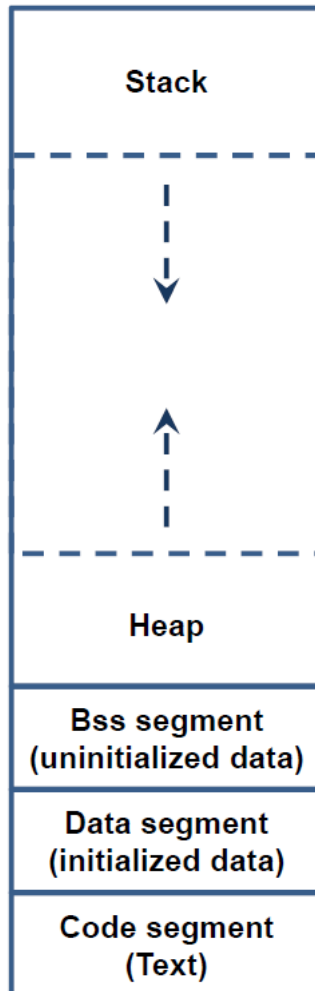


※ 출처 : [Python] 파이썬 동작 방식 (<https://velog.io/@chldppwls12/python-%EB%8F%99%EC%9E%91-%EB%B0%A9%EC%8B%9D>)

- Assembly와 Binary Code를 제외하고 컴퓨터/MCU 하드웨어를 직접적으로 제어할 수 있는 주요 언어는 C언어임
- Python과 같은 인터프리터/스크립트 언어는 사용자 편의를 위한 쉬운 문법과 API를 제공하나, 인터프리터를 통해 C언어로 변환되어 실행됨
- C로 구현하면 복잡한 알고리즘이 Python과 같은 인터프리터/스크립트 언어에서 제공하는 API로 간단하게 구현될 수 있음
- 인터프리터 내부를 보면서 코딩할 필요가 없으나 Python과 같은 인터프리터 언어는 내부적으로는 C언어 Native를 기반으로 작동함

컴퓨터 운영체제 기초

④ Memory Allocation – 프로그램은 RAM에 어떻게 올라가는가?

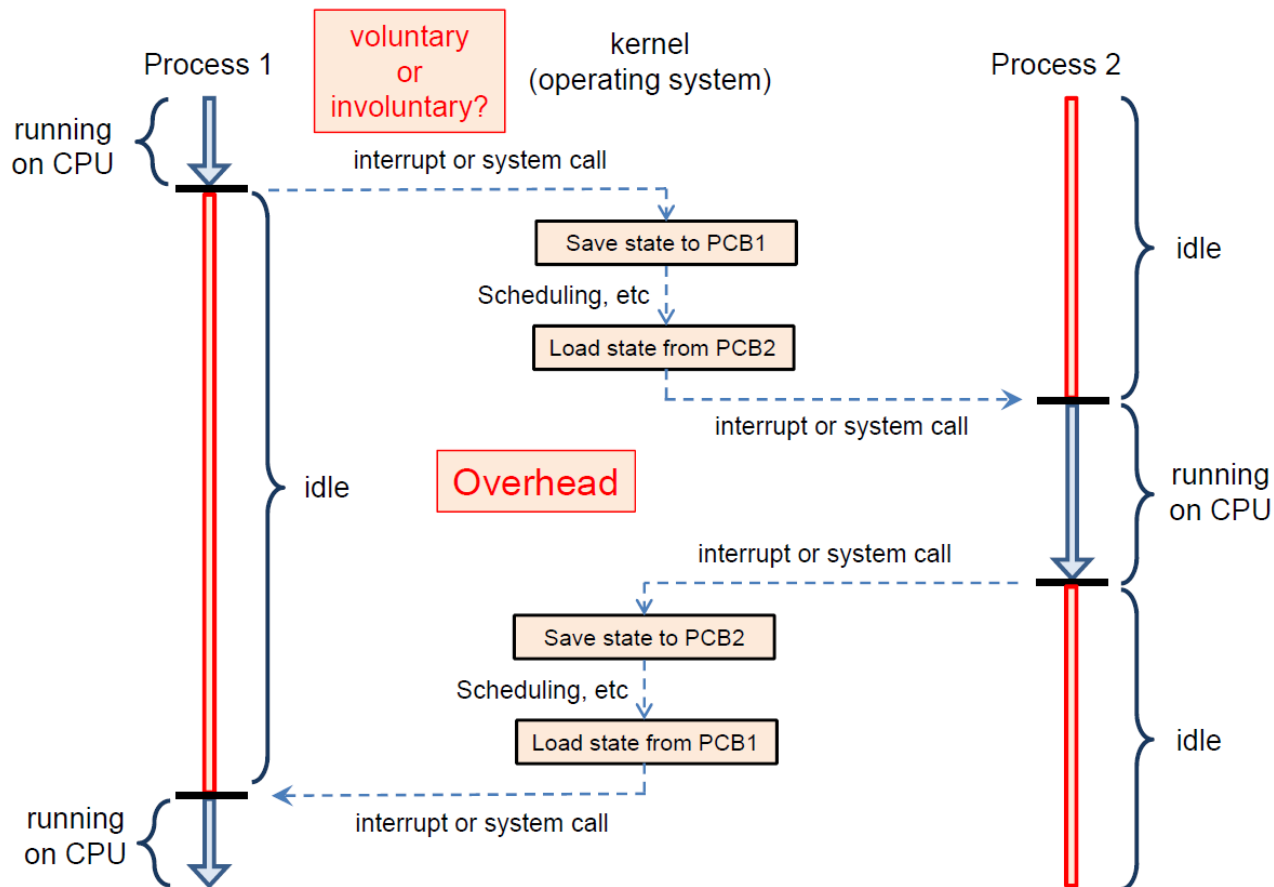


- Memory allocation
 - Static memory allocation (array)
 - Dynamic memory allocation
- Different areas of memory allocated
 - Code: code segment
 - Initialized data: data segment
 - Uninitialized data: bss segment
 - Heap
 - Stack
- Code segment
 - Also called text segment and is read-only
 - Contains an object file with executable instructions
- Data segment
 - Contains global variables and static local variables
 - Variables are initialized but can be changed at run time
 - The size of the segment does not change at run time
- Bss (basic service set) segment
 - Stores uninitialized global and static variables (zero-initialized)
 - The size of the segment does not change at run time

※ 출처 : 한양대학교 대학원 ICSL
CIN9075 Operating
Systems and System
Programming

컴퓨터 운영체제 기초

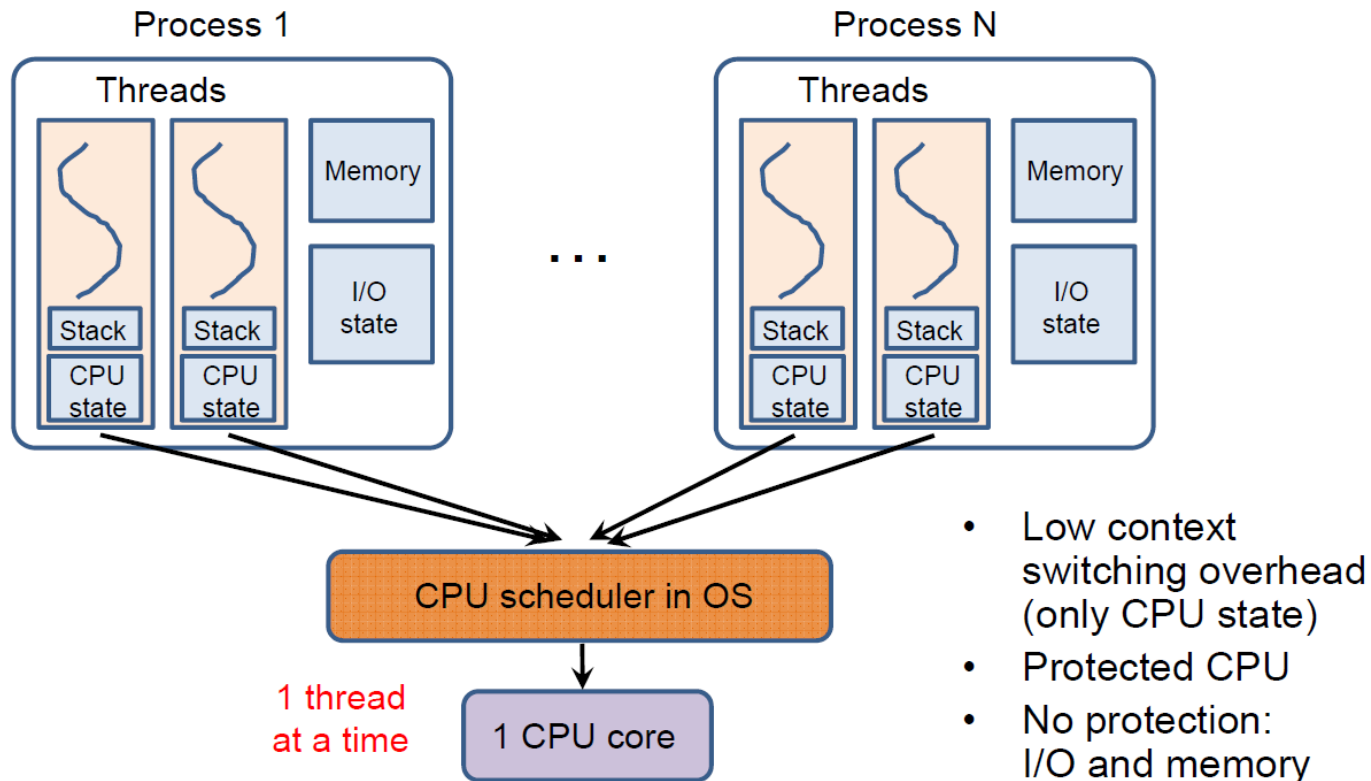
⑤ Context Switching – 수많은 프로그램은 어떻게 CPU에 의해 처리되는가?



컴퓨터 운영체제 기초

⑤ Context Switching – 수많은 프로그램은 어떻게 CPU에 의해 처리되는가?

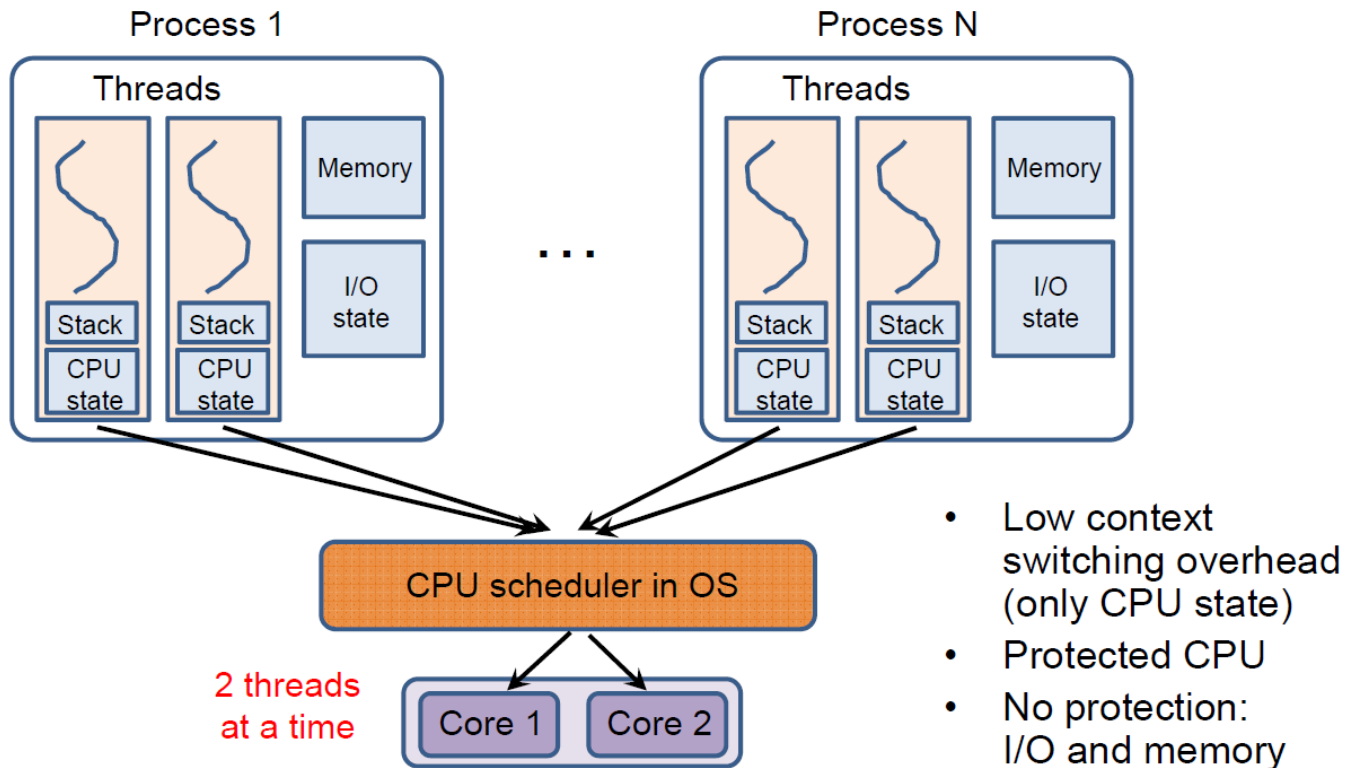
Single Core – Multiple Thread/Process



컴퓨터 운영체제 기초

⑤ Context Switching – 수많은 프로그램은 어떻게 CPU에 의해 처리되는가?

Multiple Core – Multiple Thread/Process



컴퓨터 운영체제 기초

⑤ Context Switching – 수많은 프로그램은 어떻게 CPU에 의해 처리되는가?

Hyper Threading : Multiple (Logical) Core – Multiple Thread/Process

