# Fundamental Concepts

**Minsoo Ryu**

**Operating Systems and Distributed Computing Lab.**
**Hanyang University**

**msryu@hanyang.ac.kr**

# Key Concepts of OS

1. Abstraction
2. Protection
3. Illusion
4. Coordination and Optimization

# Abstraction 추상화

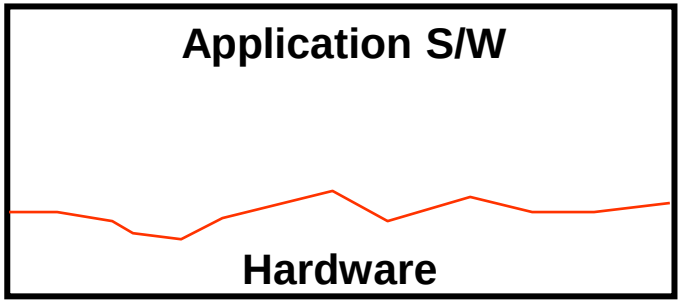
## ❒ In general

- **Abstraction means picking out only key features to reduce complexity** 중요한 것만 간추려 제공하는 것

## ❒ In the context of OS 하드웨어의 복잡성을 없애고 프로그래머가 하드웨어를 손쉽게 제어할수 있도록 인터페이스를 제공하는 것

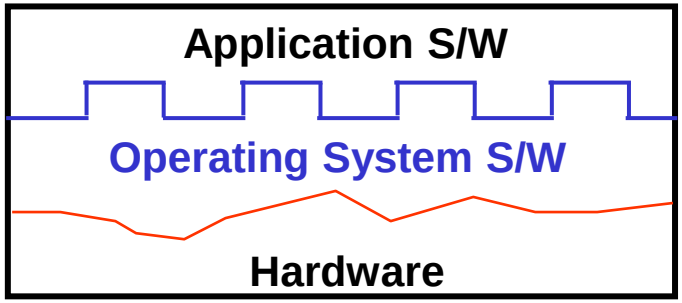
- **OS hides hardware complexity and provides a clean, uniform, and standard interface to programmers**
  - e.g.) POSIX APIs (Application Programming Interfaces) 예) API

How can we display “Hello World” on the screen?




Without OS

We can simply use printf( ) that is provided by OS

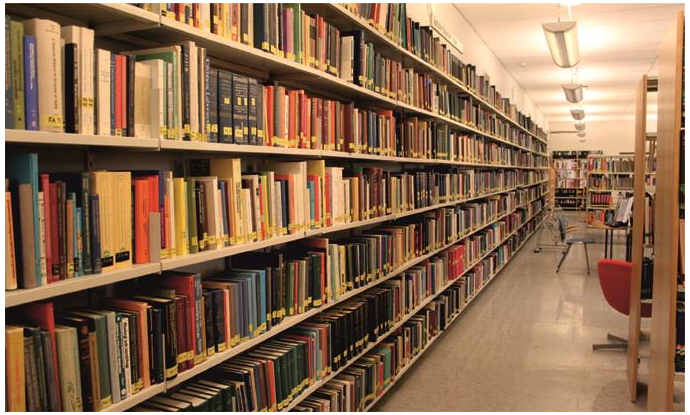



With OS

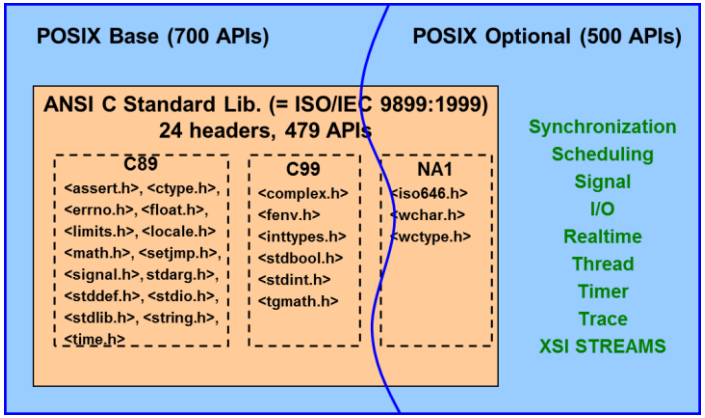
# OS can be viewed as a Big Library

함수들을 제공하는 거대한 library

- An organized collection of reusable program code
  - Bootstrapping, device drivers, scheduling, GUI …
  - POSIX Standard API implementations

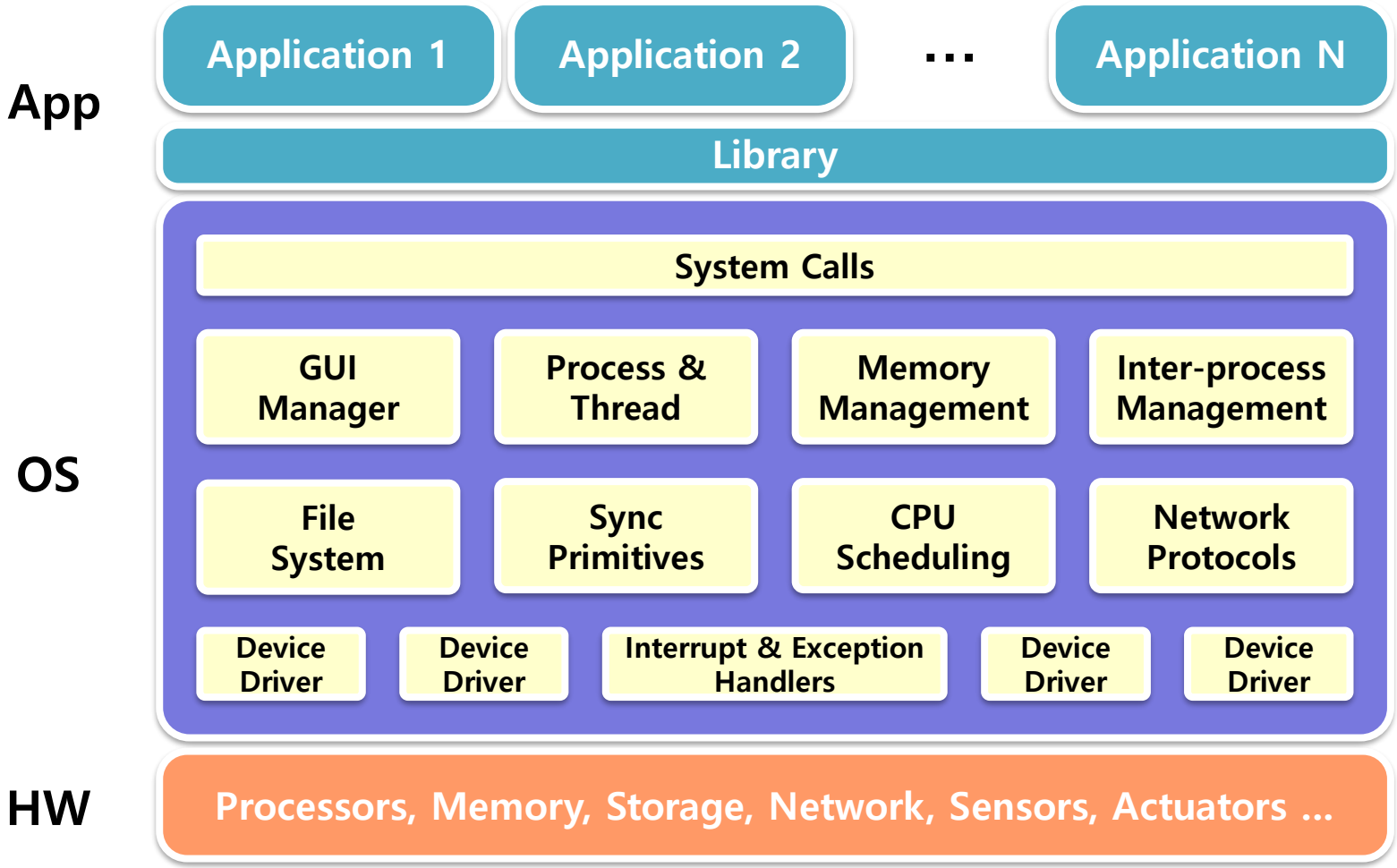


University Library



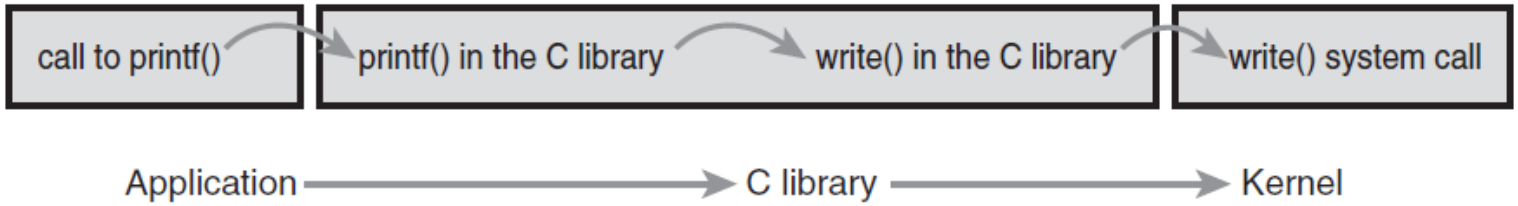

POSIX Standard 1003.1

# Typical System Structure



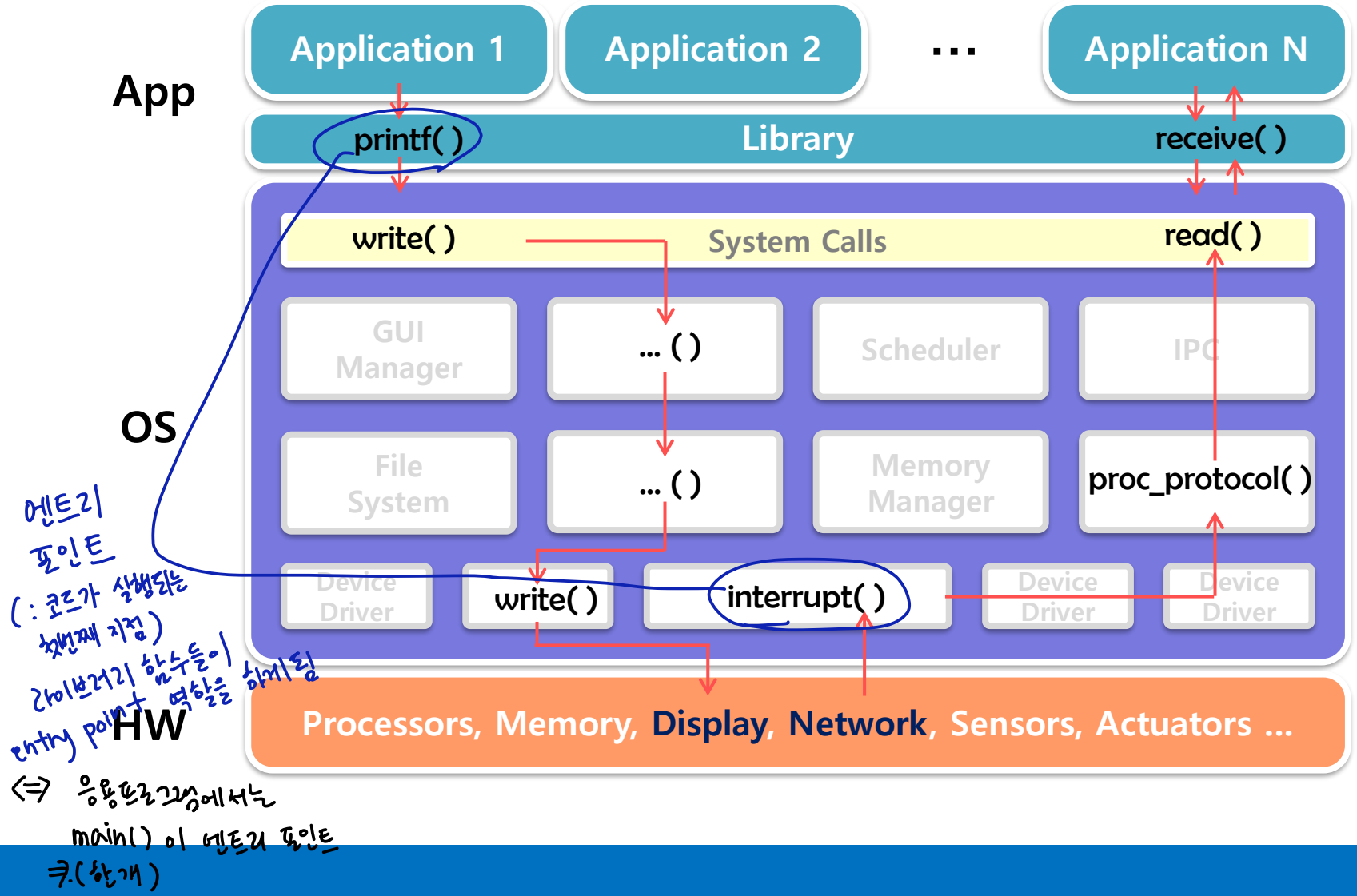
# System Calls vs. Library Calls

- System calls provide the interface between application SW and the operating system




| System calls | Library calls |
|---|---|
| open | fopen |
| close | fclose |
| read | fread, getchar, scanf, fscanf, getc, fgetc, gets, fgets |
| write | fwrite, putchar, printf, fprintf putc, fputc, puts, fputs |
| lseek | fseek |

# Execution Flow

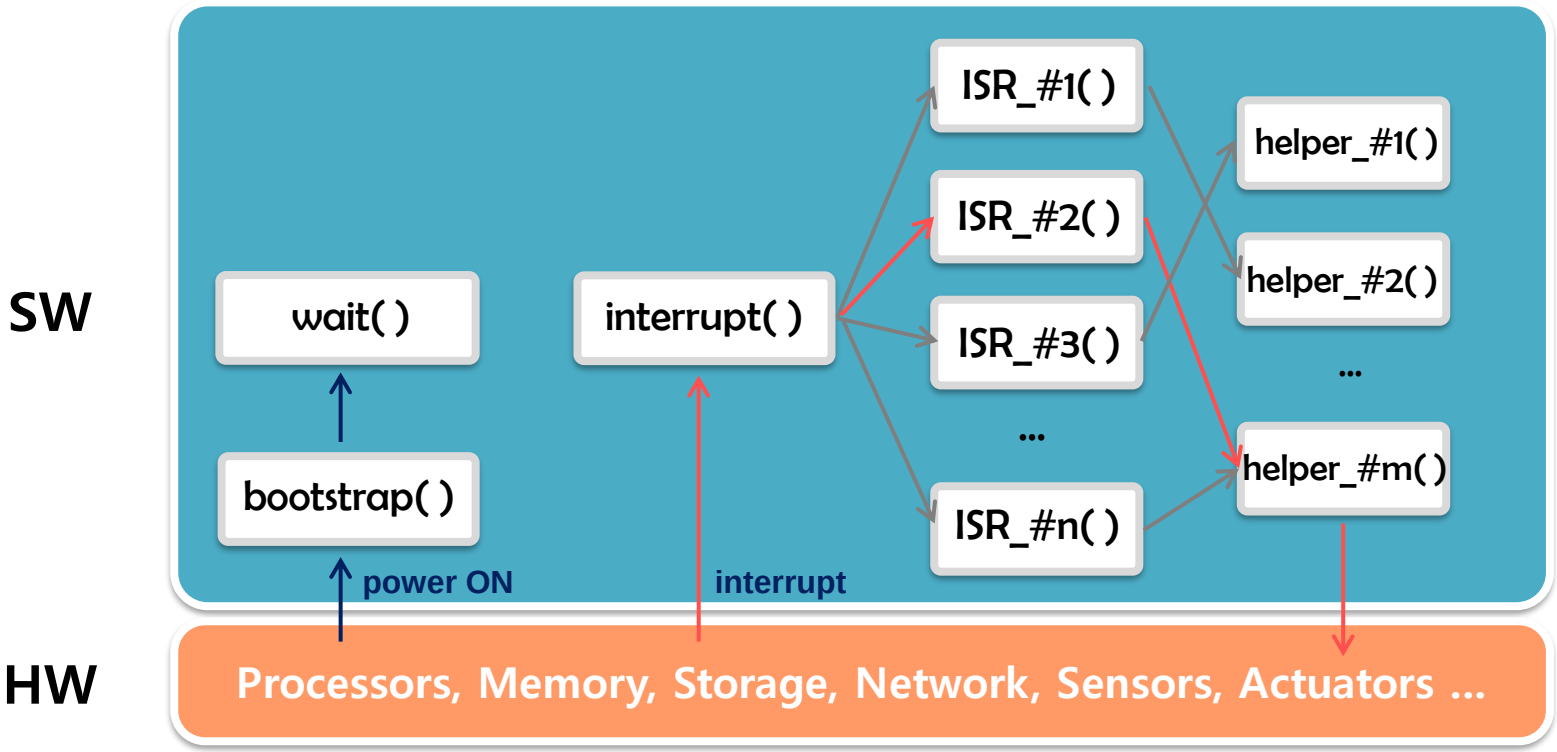


# Execution Flow without OS

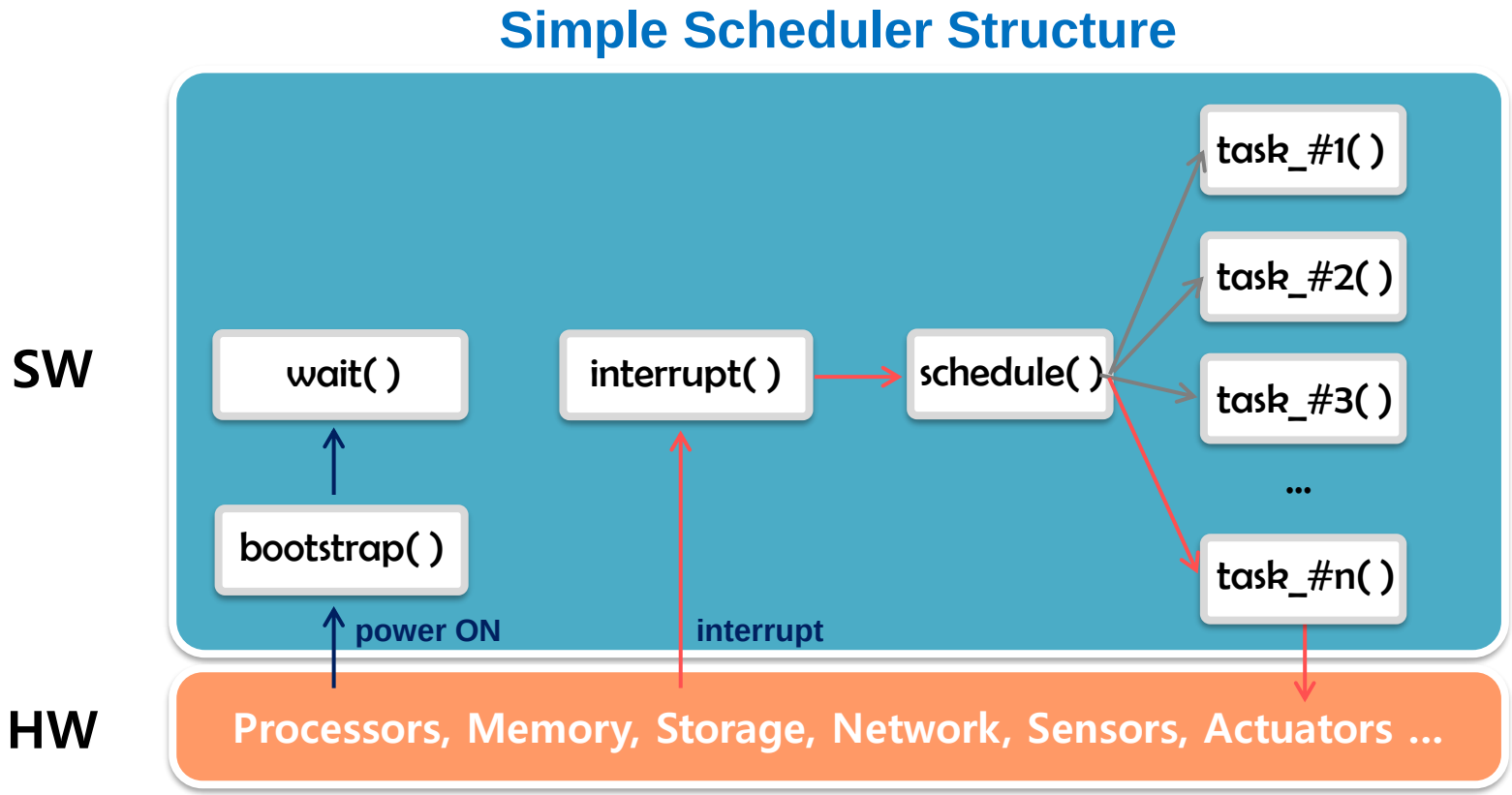
ex) 임베디드 시스템

interrupt driven system 구조
event "

## Event Handler Structure



# Execution Flow without OS



# Protection

시스템을 보호

- OS protects the system from many types of application faults

응용프로그램으로부터 하드웨어를 포함한 운영체제를 보호함
↓
(신뢰하기 어려움)

- Example: separation of each program's memory space

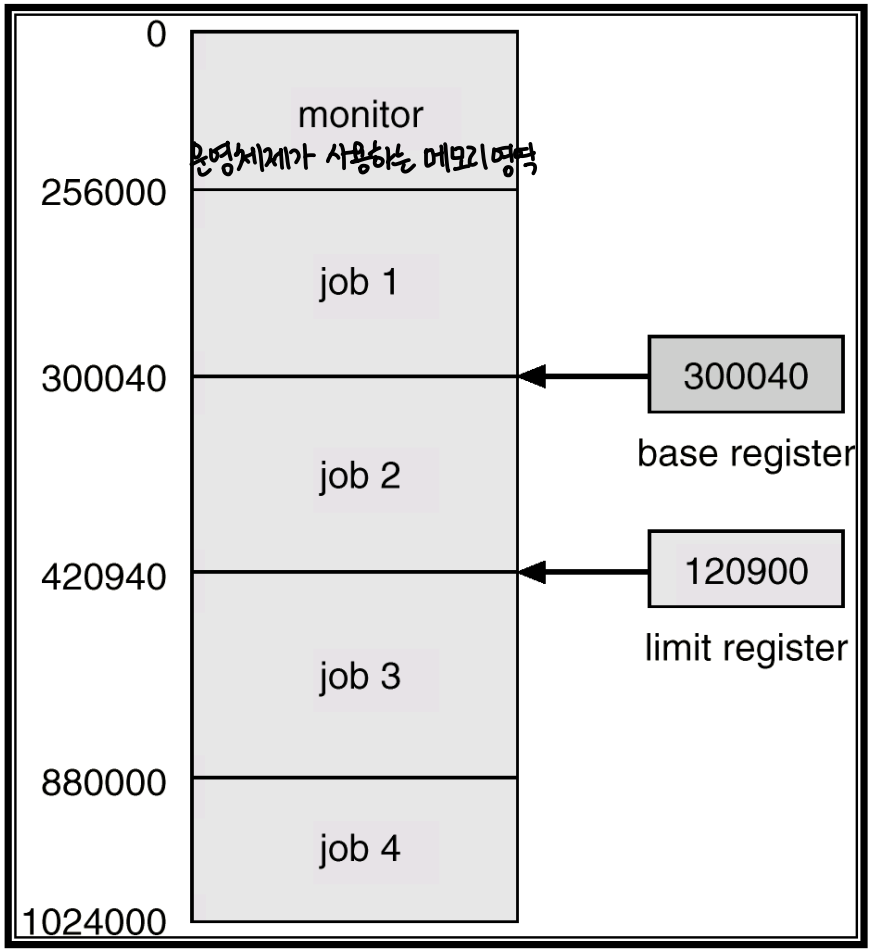
특정 프로그램이 사용하는 메모리의 시작주소

  - Base register – holds the smallest legal memory address
  - Limit register – contains the size of the range

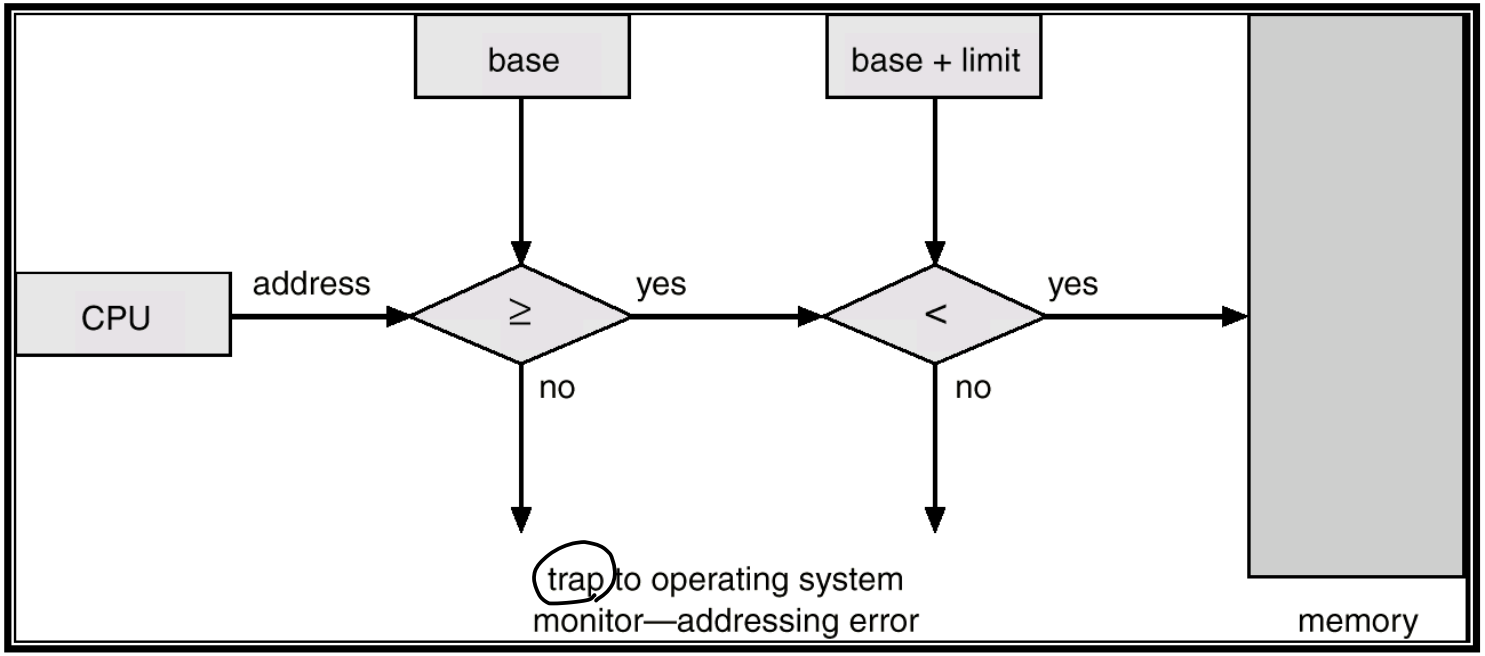
프로그램이 사용하는 메모리의 최대크기

  - Determine the range of legal addresses
  - Memory outside the defined range is protected

# Use of A Base and Limit Register



# Hardware Address Protection




운영 체제의 exception handler 호출

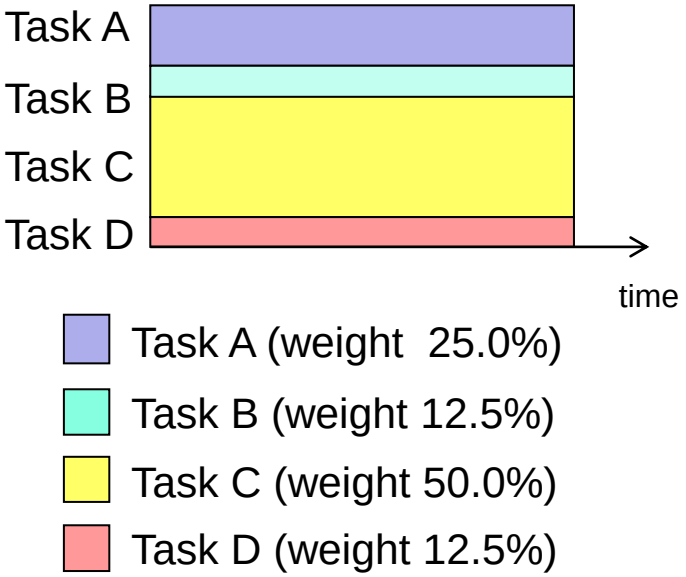
↓

해당 프로그램을 kill

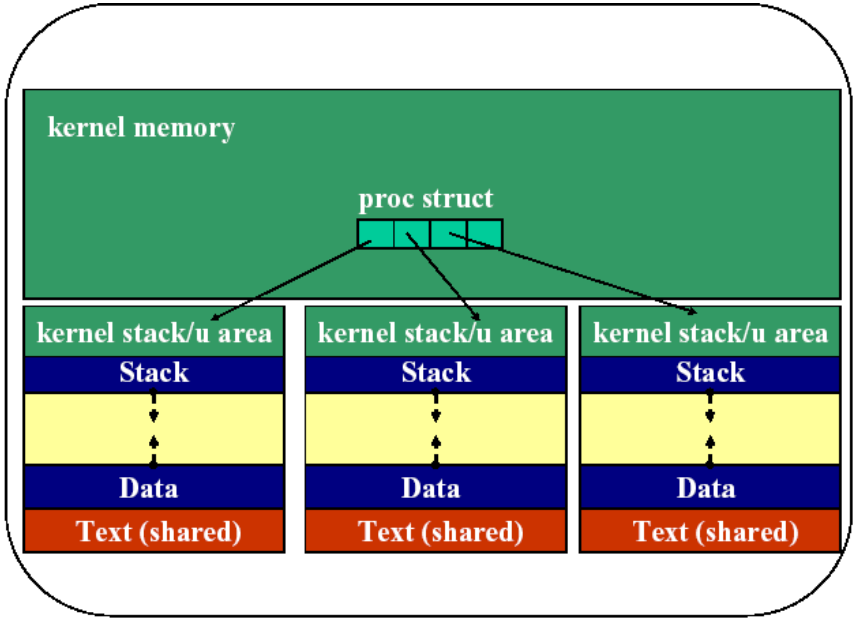
# Illusion

시스템이 가진 물리적 제약을
사용자가 느끼지 못하도록
운영체제가 속임수를 쓰는것

- **Make hardware limitations go away**
  - **As if there are infinite number of processors (time sharing)**
  - **As if there is extremely large memory (virtual memory)**




**Time Shared Multitasking**




**Virtual Memory**

# Coordination and Optimization

- **Make many things work together efficiently**

- **Concurrency**
  - **Multitasking (synchronization and scheduling)**
  - **Multi-user support**
  - **Multithreading**

- **Efficiency**
  - **Overlapped I/O and Processing**
    - I/O devices and the CPU can execute concurrently
  - **Effective storage management**
    - Caching, paging, and swapping
  - **Fast interrupt handling**

# Multitasking and Synchronization


thank you!