Visual C++

About..

Chapter 1. 운영체제와 시스템 프로그래밍

김 원 일

운영체제의 정의

운영체제(Operating System)

- 모든 컴퓨터 시스템에 필수적인 요소
- 컴퓨터의 하드웨어를 관리하는 프로그램
- 사용자를 위한 응용 프로그램이 실행 가능하도록 기반을 제공
- 사용자와 하드웨어 사이의 인터페이스를 제공

컴퓨터 시스템의 구성 요소

- Hardware, OS, Application
- Hardware
  - 컴퓨터 시스템을 구성하는 물리적인 구성 요소
- OS
  - 컴퓨터 시스템을 구성하는 모든 자원을 관리하는 관리 프로그램
- Application
  - 컴퓨터 시스템의 모든 자원을 요청하여 사용자가 원하는 목적을 이루도록 도와주는 프로그램

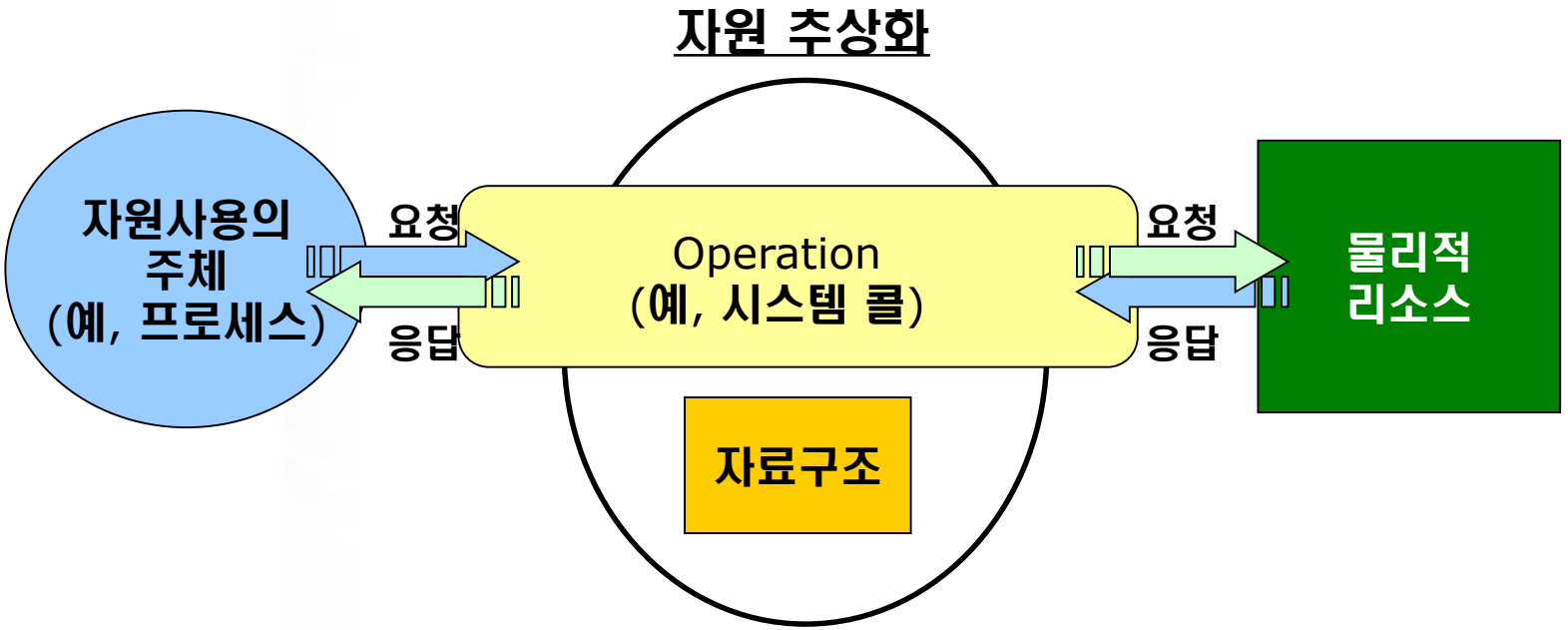
운영체제의 목적

- 시스템에 존재하는 자원의 관리
  - 컴퓨터 시스템을 구성하고 있는 모든 자원을 관리
  - 물리적인 자원 뿐만 아니라 논리적인 자원도 포함
    - CPU
    - Memory
    - Space on a disk
    - Process
    - Scheduling
    - Display
- 자원 관리
  - "An OS creates resource abstractions"
  - "An OS manages resource sharing"

Resource Abstraction – 1

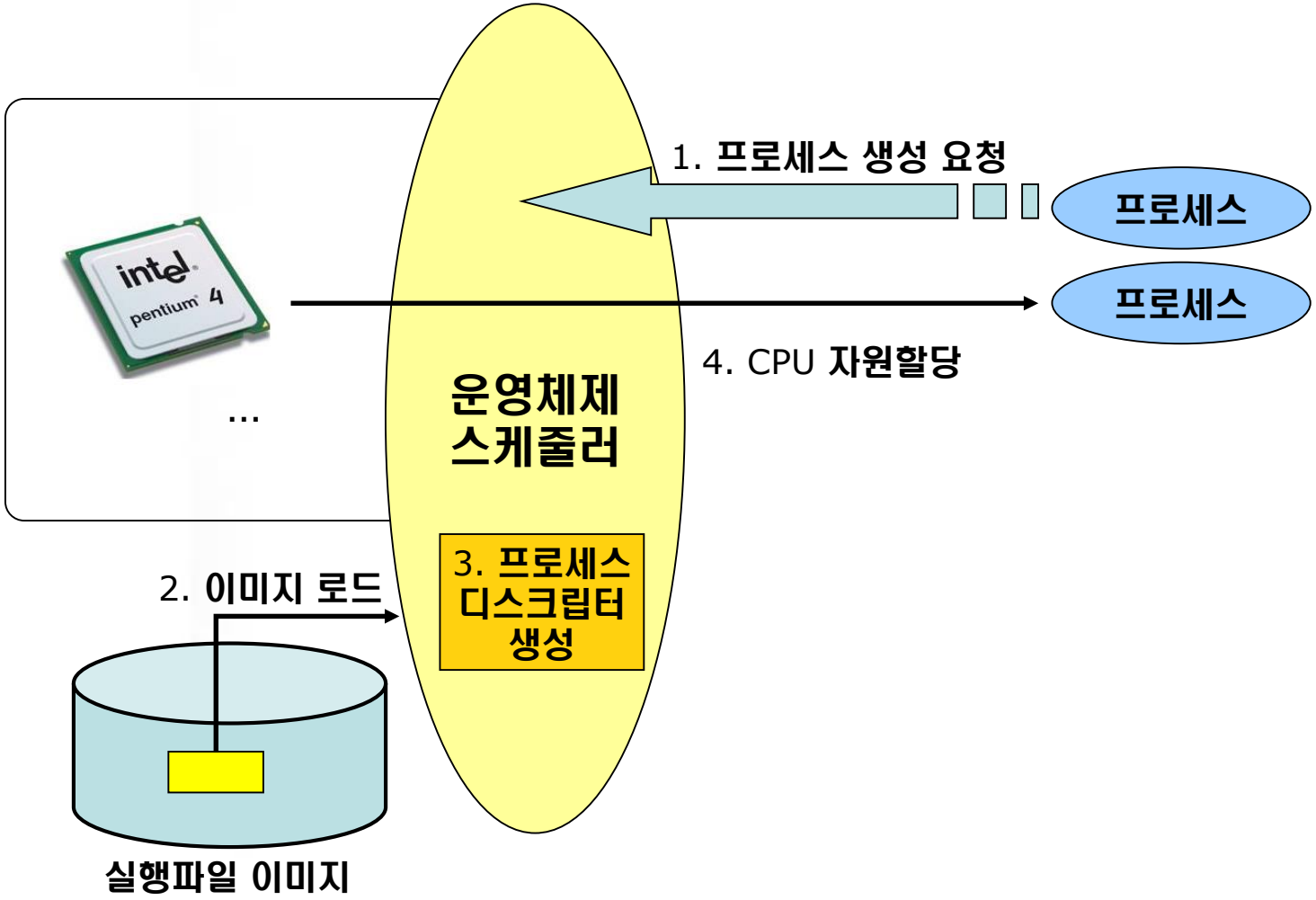
자원 추상화의 구성 요소

- 물리적 리소스에 대한 논리적 자료구조
- 잘 정의된 오퍼레이션의 집합



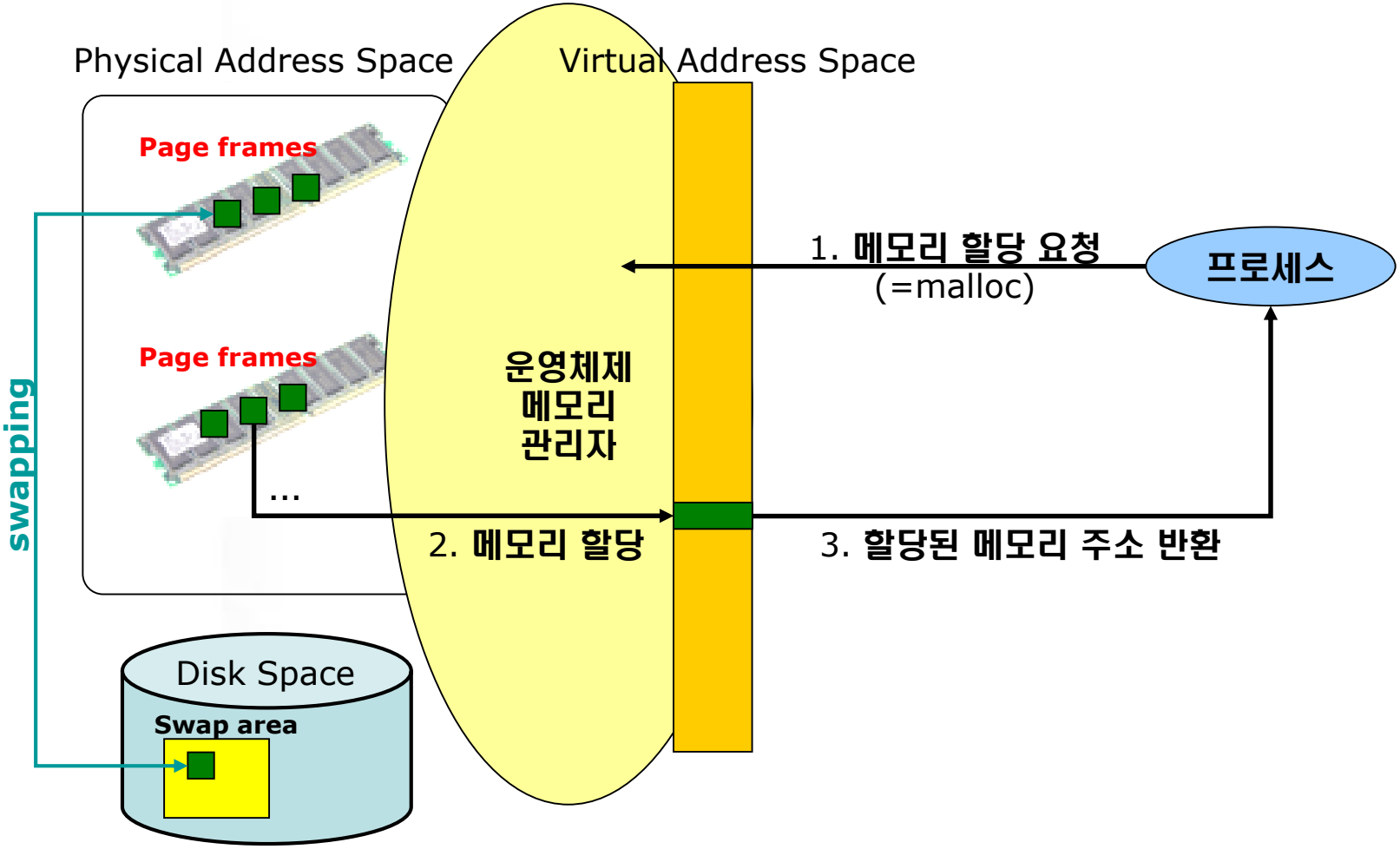
Resource Abstraction – 2

CPU Abstraction



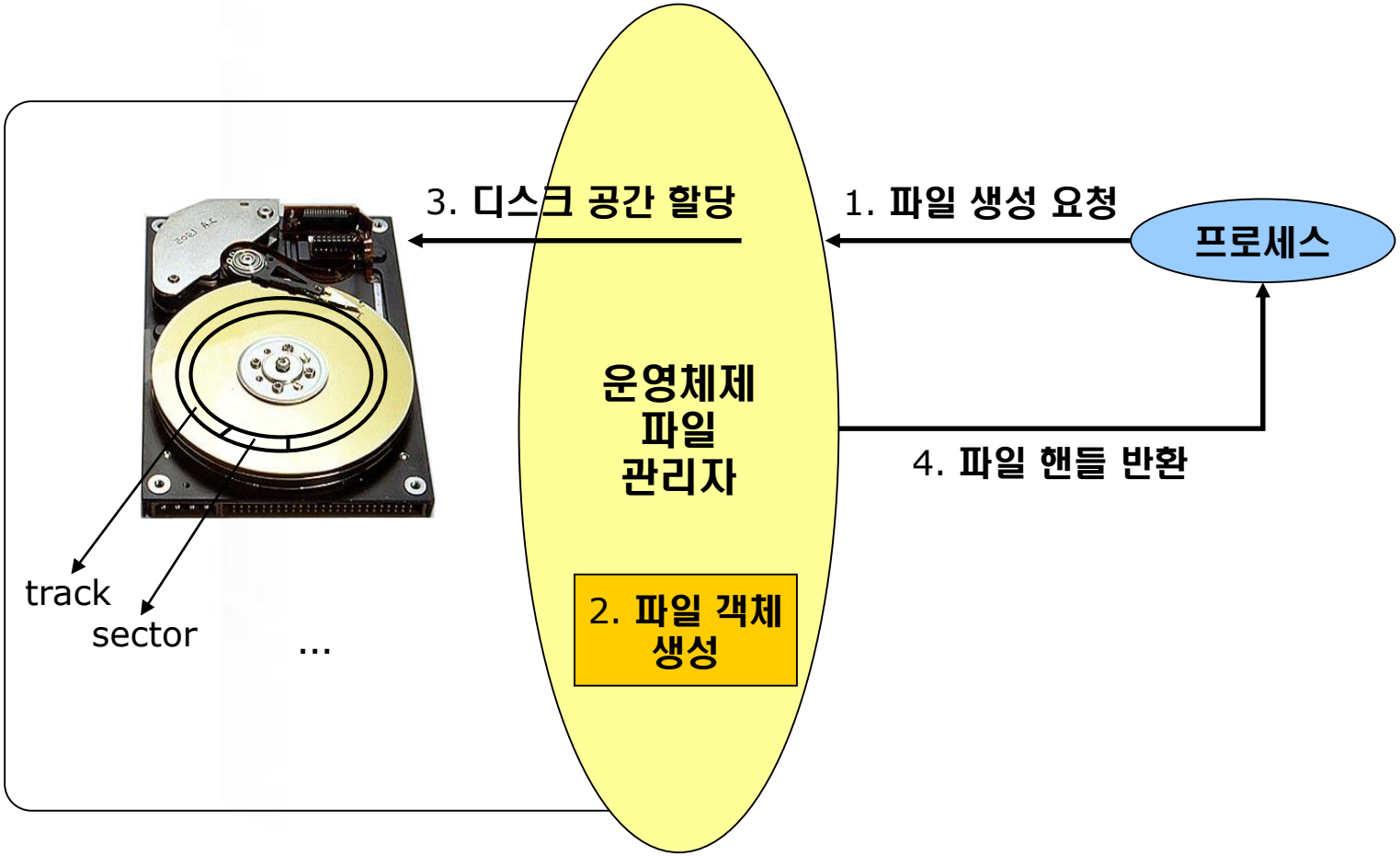
Resource Abstraction – 3

Memory Abstraction



Resource Abstraction – 4

Disk Abstraction



Resource Sharing – 1

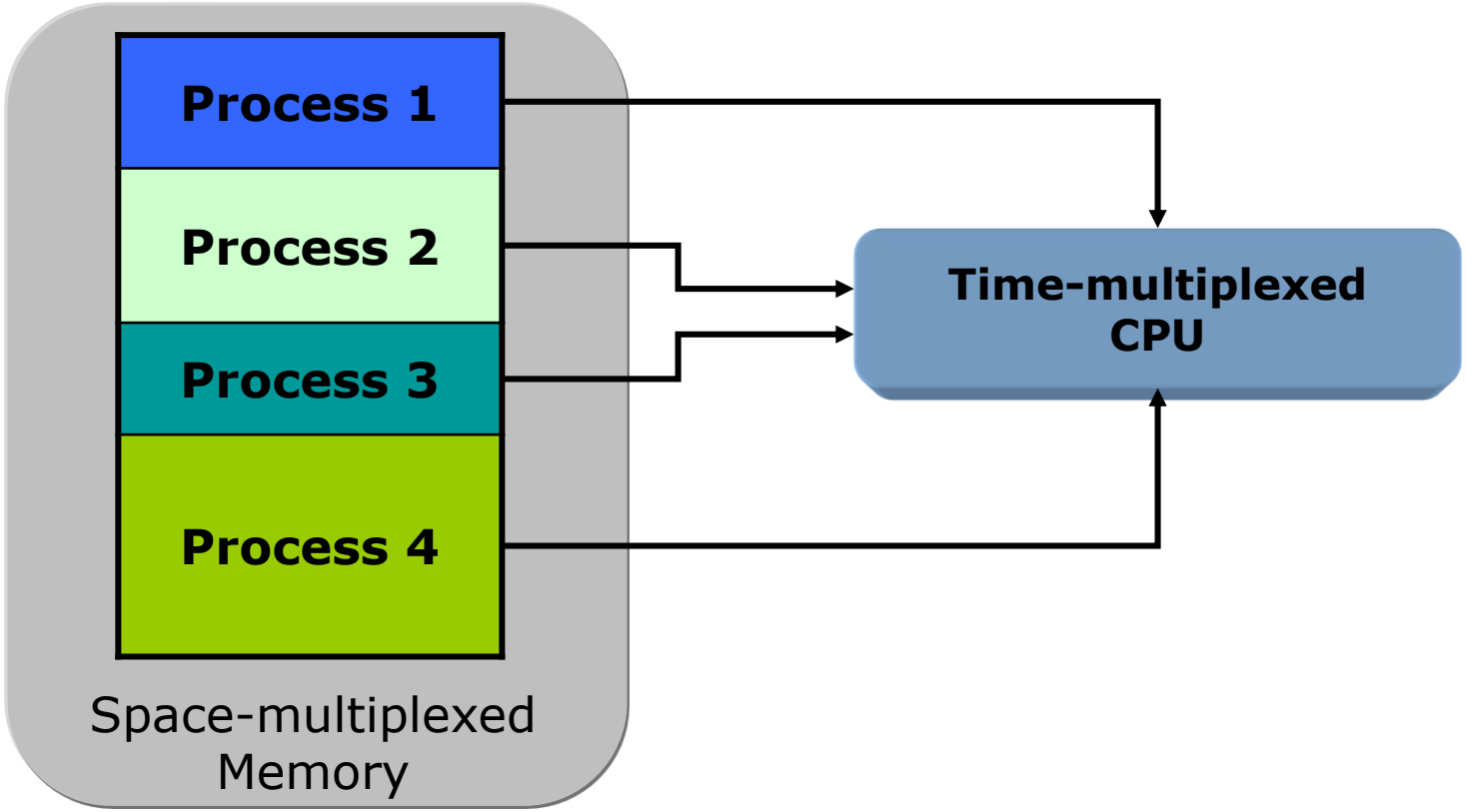
- 자원 공유의 기법
  - Space- Vs time-multiplexed sharing
  - 시간과 공간으로 다중화된 자원 공유가 기본적임
- 공유 자원 제어
  - 공유 자원을 제어하려면 반드시 자원이 분리되어 존재해야 함
- 운영체제의 자원 분리 기법
  - 자원을 어떻게 분리하여 제공할 것인지를 결정
  - 분리된 자원의 사용과 할당에 대한 정책 등을 결정
- 자원 사용에의 동시성 문제가 발생
  - 동기화 문제 해결 방법들을 제시

Resource Sharing strategy – 1

- Multiprogramming
  - CPU의 자원을 공유하기 위한 프로세스들의 공유 기법
  - 프로세스는 입출력 시에 블록 여부가 결정
  - 프로세스는 다른 프로세스의 입출력을 대기해야 함
  - 프로세스가 블록 되면 다른 프로세스가 수행 가능
  - Multiprogramming OS는 CPU 공유가 가능
    - 이러한 공유는 전체적으로 자동화되어 실행
  - 모든 프로세스의 실행 시간이 줄어듦
    - 수행을 위한 순서 결정을 위한 결정 시간이 필요
    - 각 자원의 실행 시간을 비교 설정할 수 있어야 함

Resource Sharing strategy – 2

- Multiprogramming **방법**
  - **메모리 공간의 다중화를 통해 다수의 프로세스 실행**
  - CPU **시간의 다중화를 통해 다수의 프로세스가 수행**



Resource-Sharing strategy – 3

- Batch Processing
  - Multiprogramming **기법을 이용**
  - Job(**작업**)
    - **운영체제의 각종 명령어를 파일로 생성**
    - **해당 파일의 수행은 실행 시간을 위해 대기 상태**
  - **할당된 작업들은 운영체제가 한번에 실행되도록 구성**
  - **운영체제는 나열된 명령어의 리스트를 차례로 수행**
  - **사용자의 개입 없이도 모든 작업이 수행**
  - **운영체제는 자원의 사용 효율을 최적화 할 수 있음**
  - Batch processing**는 현재도 선택적으로 사용되고 있음**

Resource-Sharing strategy – 4

- Timesharing
  - Multiprogramming **기법을 이용**
  - **상용 작용하는 컴퓨팅 모델을 제공**
    - **다수의 콘솔을 사용하고 있는 것처럼 보임**
    - **다수의 프로세스가 동시에 실행되고 있는 것처럼 보임**
  - Batch**와는 다른 형태의 스케줄링과 메모리 할당 기법을 사용**
  - Tends to propagate processes
  - **자원의 분리를 위해 많은 주의를 기울이고 있음**
    - Security & protection
  - **응답 시간의 최적화를 제공**

운영체제 구조와 역사

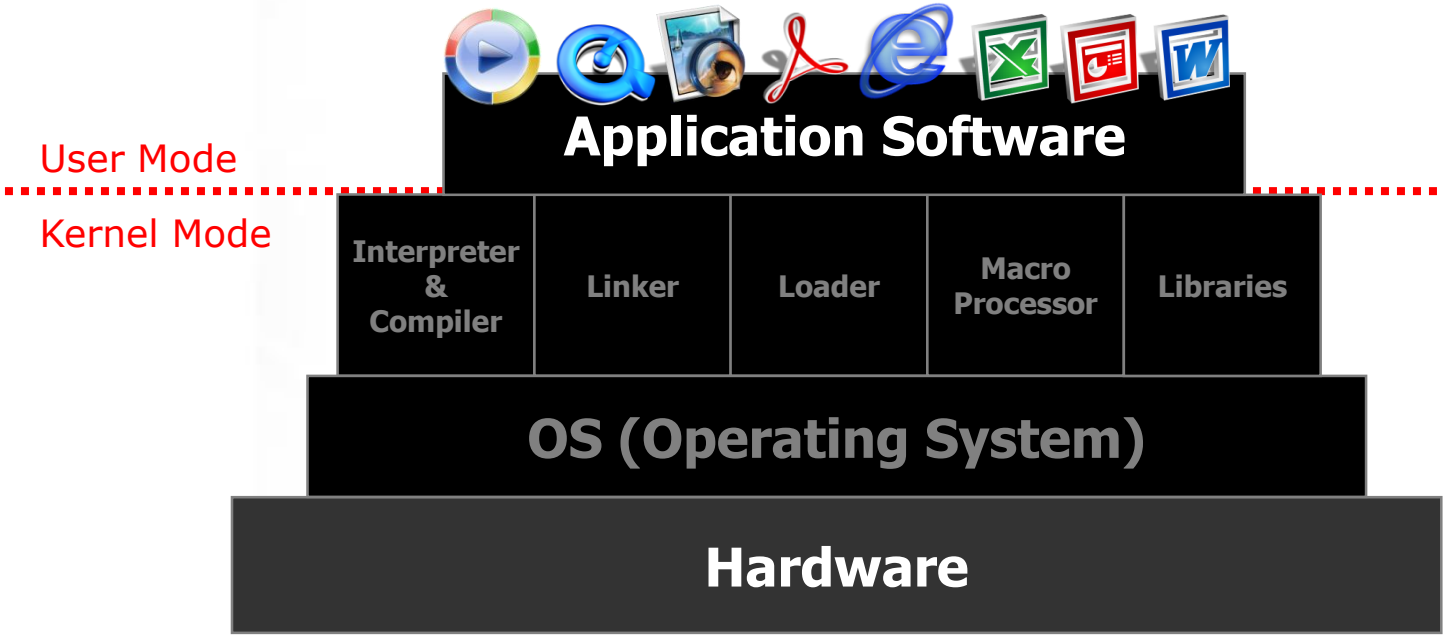
[About..

]

운영체제의 구조

컴퓨터 소프트웨어의 구성

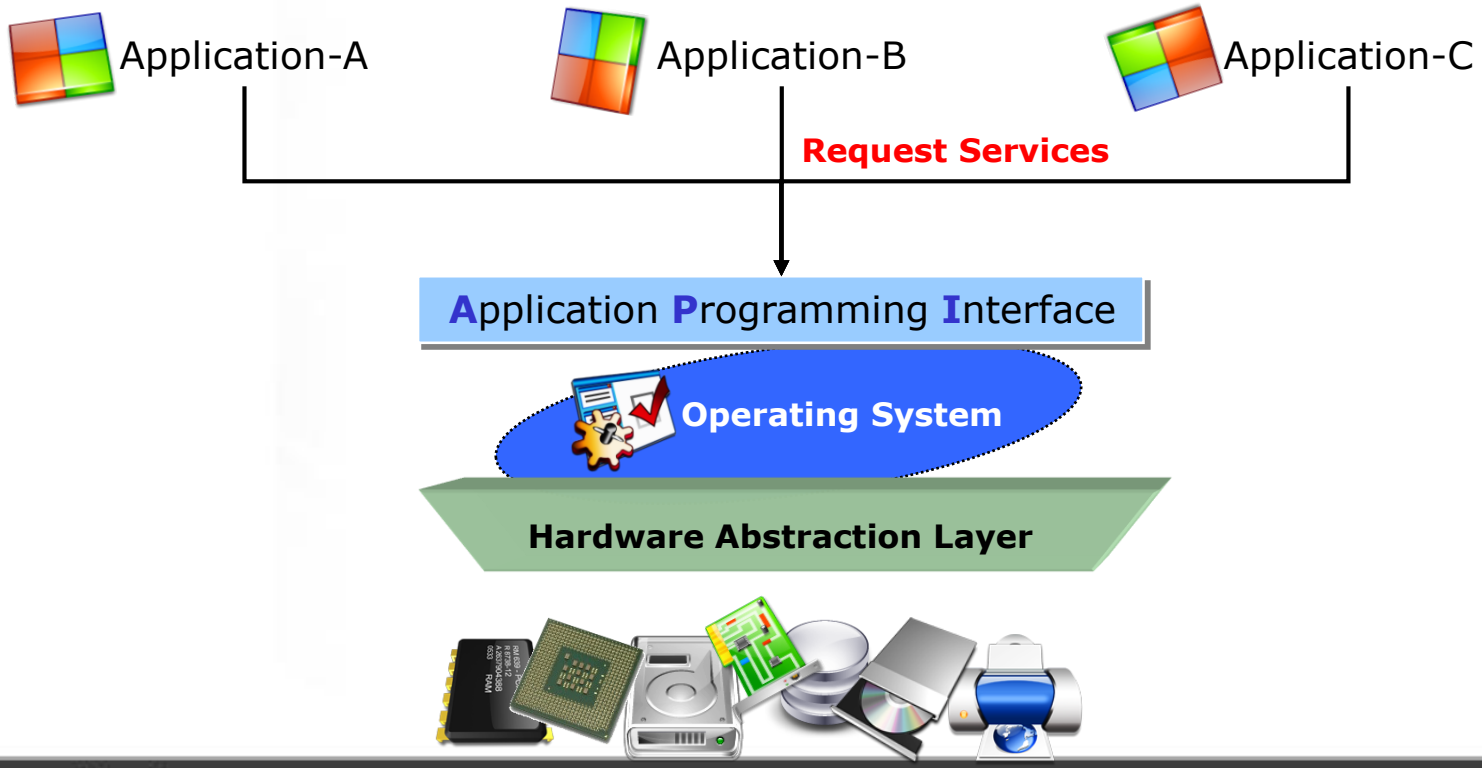
- 응용 소프트웨어 (프로그램)
  - 사용자와 직접적인 상호작용을 수행하는 프로그램
- 시스템 소프트웨어 (프로그램)
  - 응용 프로그램이 컴퓨터 자원을 사용할 수 있도록 지원
  - 컴퓨터 자원의 효과적인 활용을 위해 자원 관리를 수행



운영체제에서 API

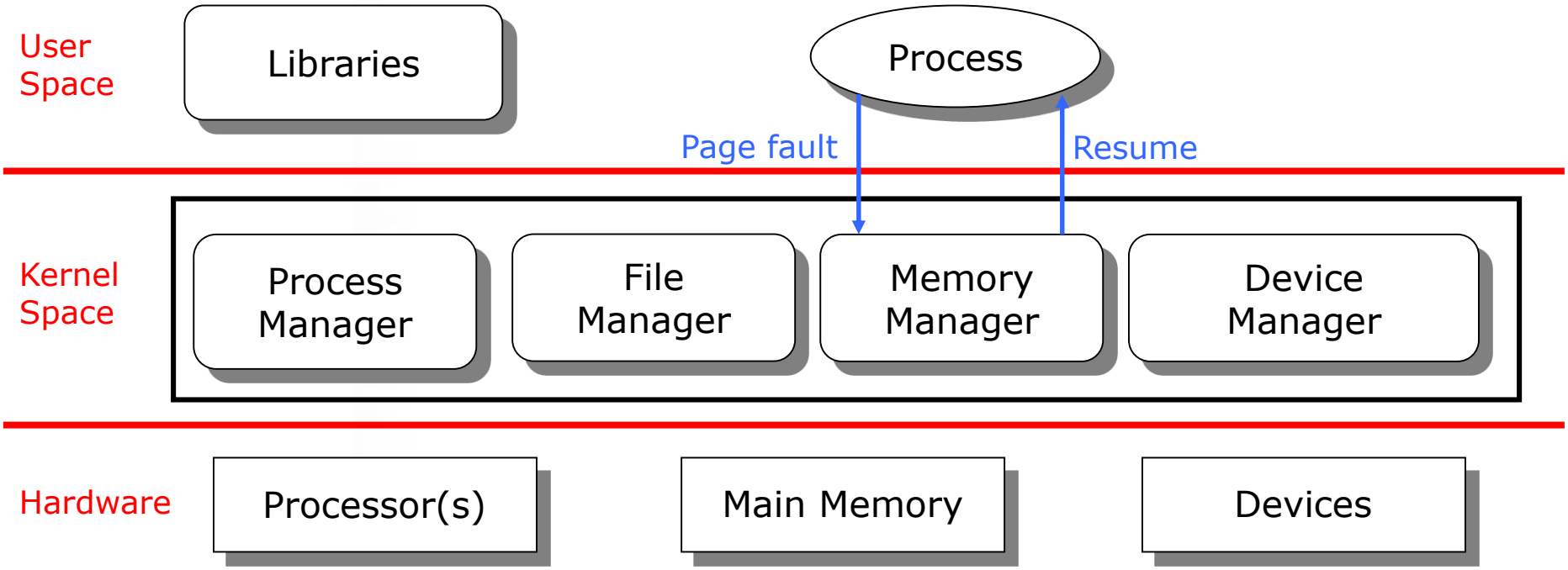
OS and API

- API와 운영체제와의 관계
  - 애플리케이션이 운영체제에 접근하는 기능을 제공
  - API는 하드웨어에 대한 추상화된 접근을 제공
  - 직접적인 접근을 허용하지 않음 (Real Vs Protected Mode)



Monolithic Kernel Organization

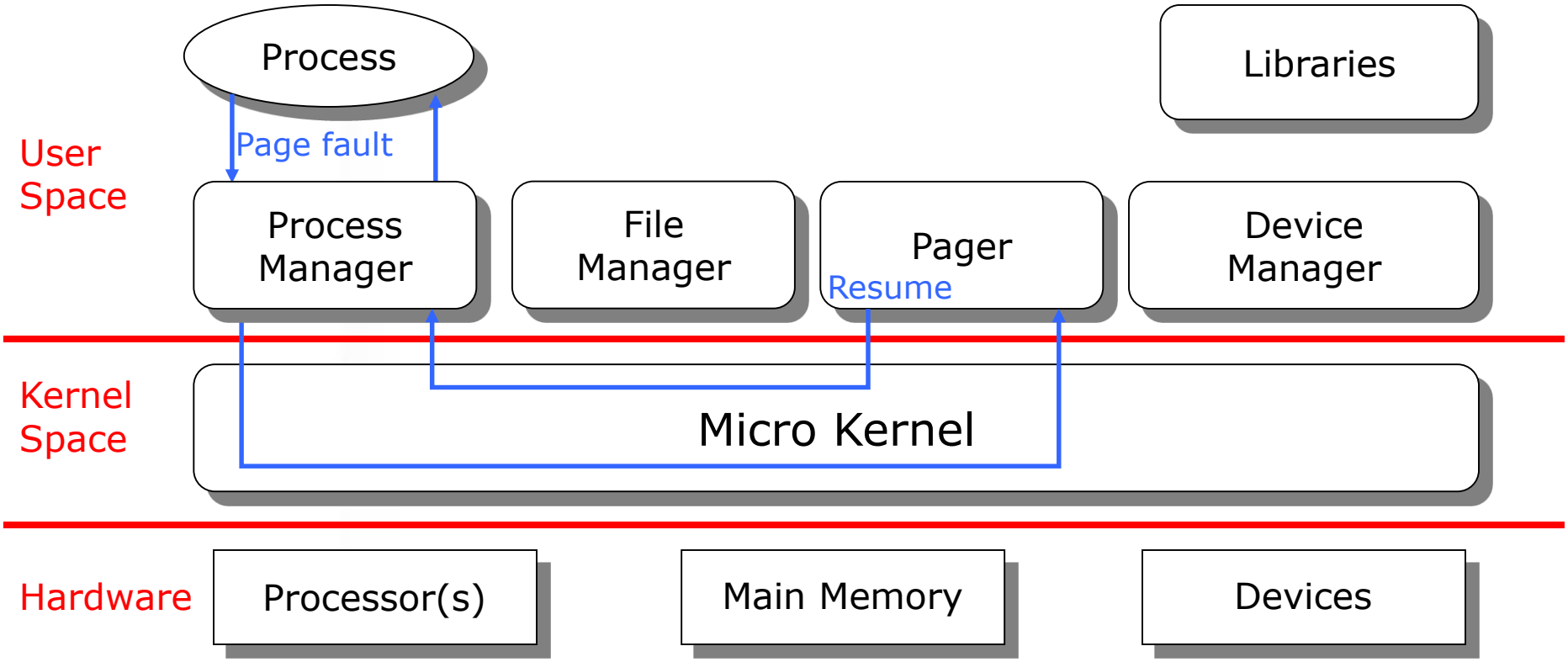
- 커널 구조가 단순하고 성능이 빠름
- 커널의 유연성과 확장성이 떨어짐
  - 예) Page fault 발생시 커널영역과 사용자영역 사이의 스위칭이 적게 일어나 빠르지만, Paging 전략을 교체하기 위해서는 커널 전체를 새로 빌드해야함



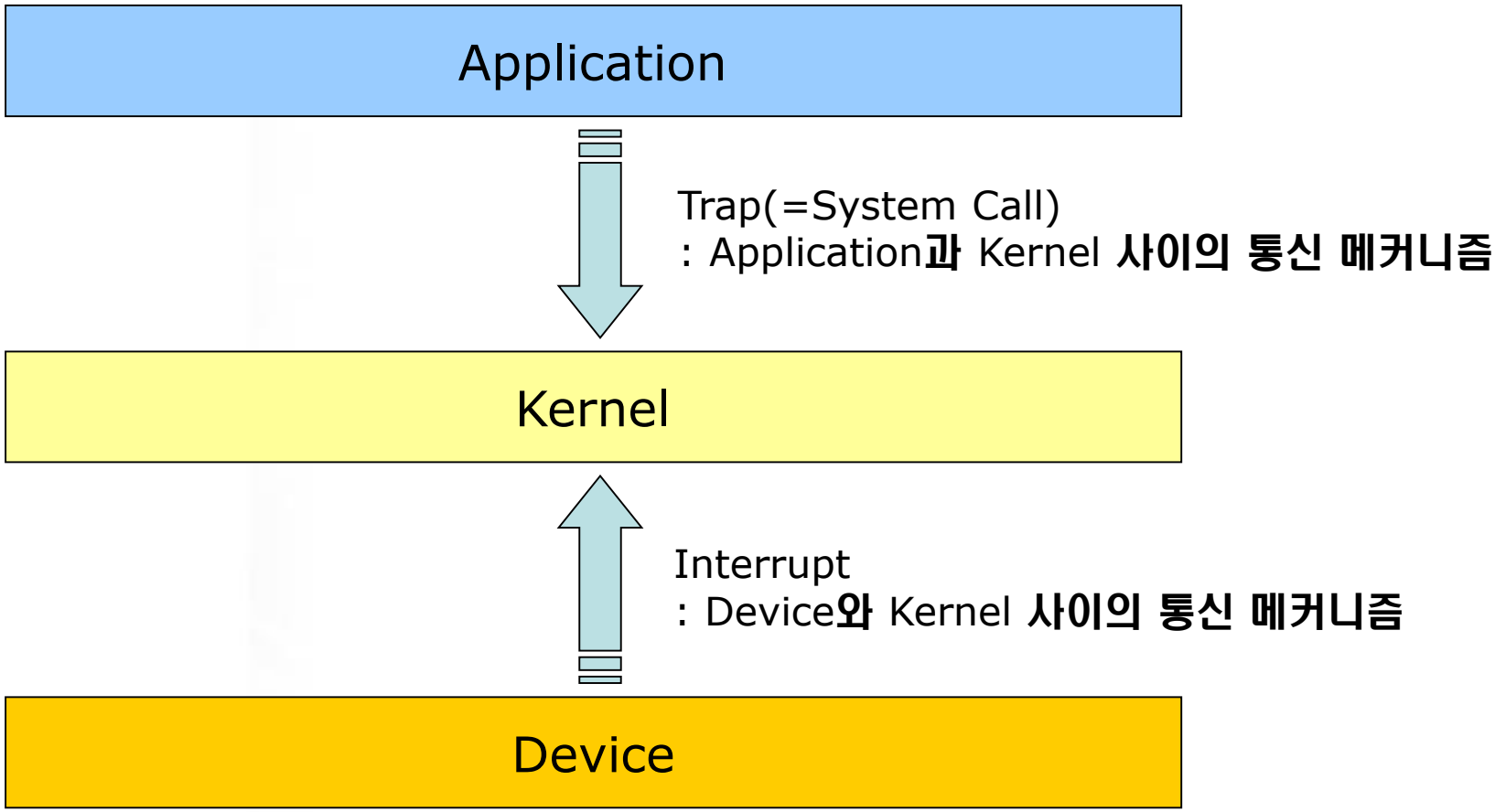
Micro kernel Organization

커널의 유연성과 확장성 증대

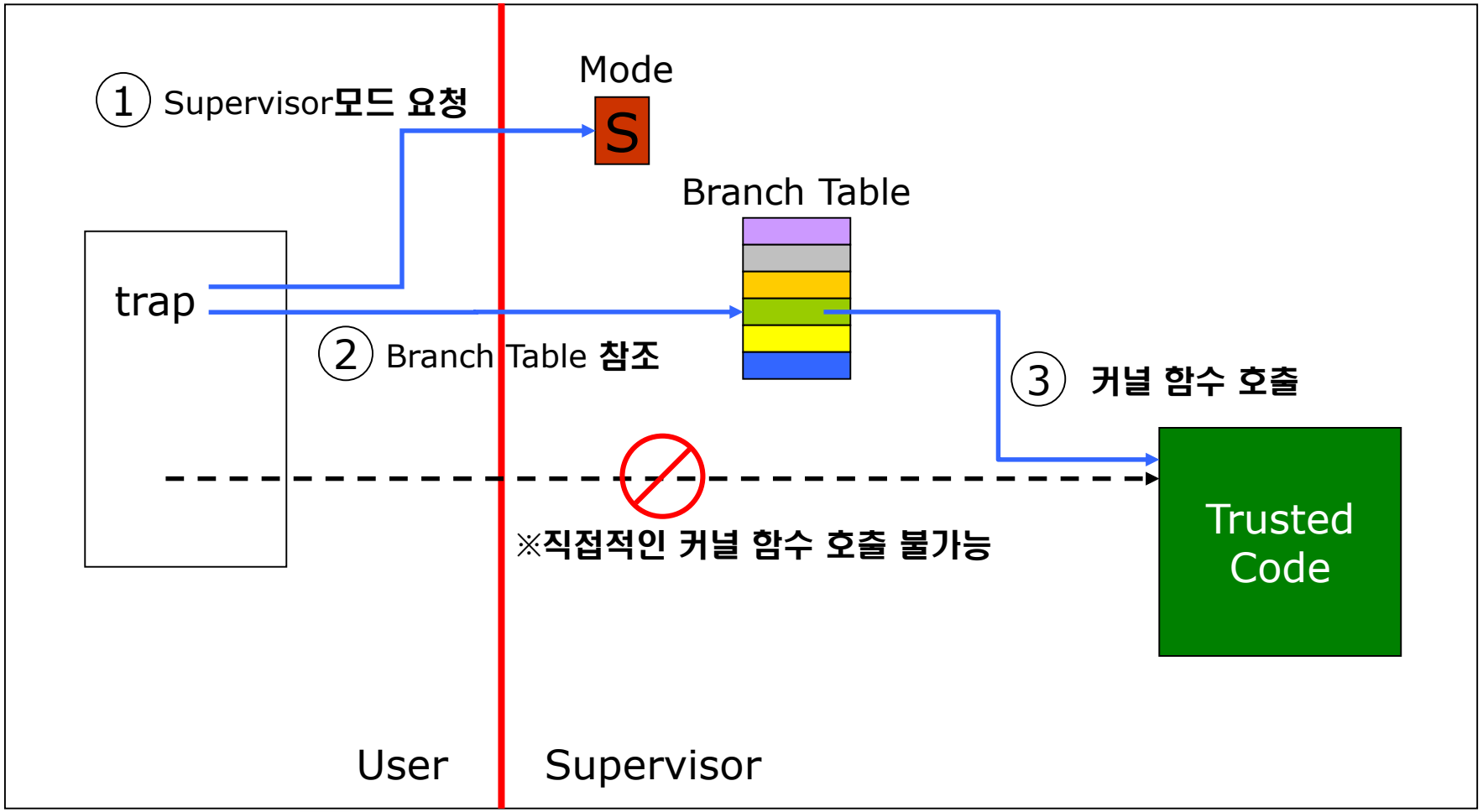
- 시스템의 성능이 떨어짐
  - 예) Page fault 발생시 커널영역과 사용자영역 사이에서 여러 번의 스위칭이 일어나 성능저하를 초래하지만 Paging 전략을 사용자 영역에서 쉽게 교체할 수 있음



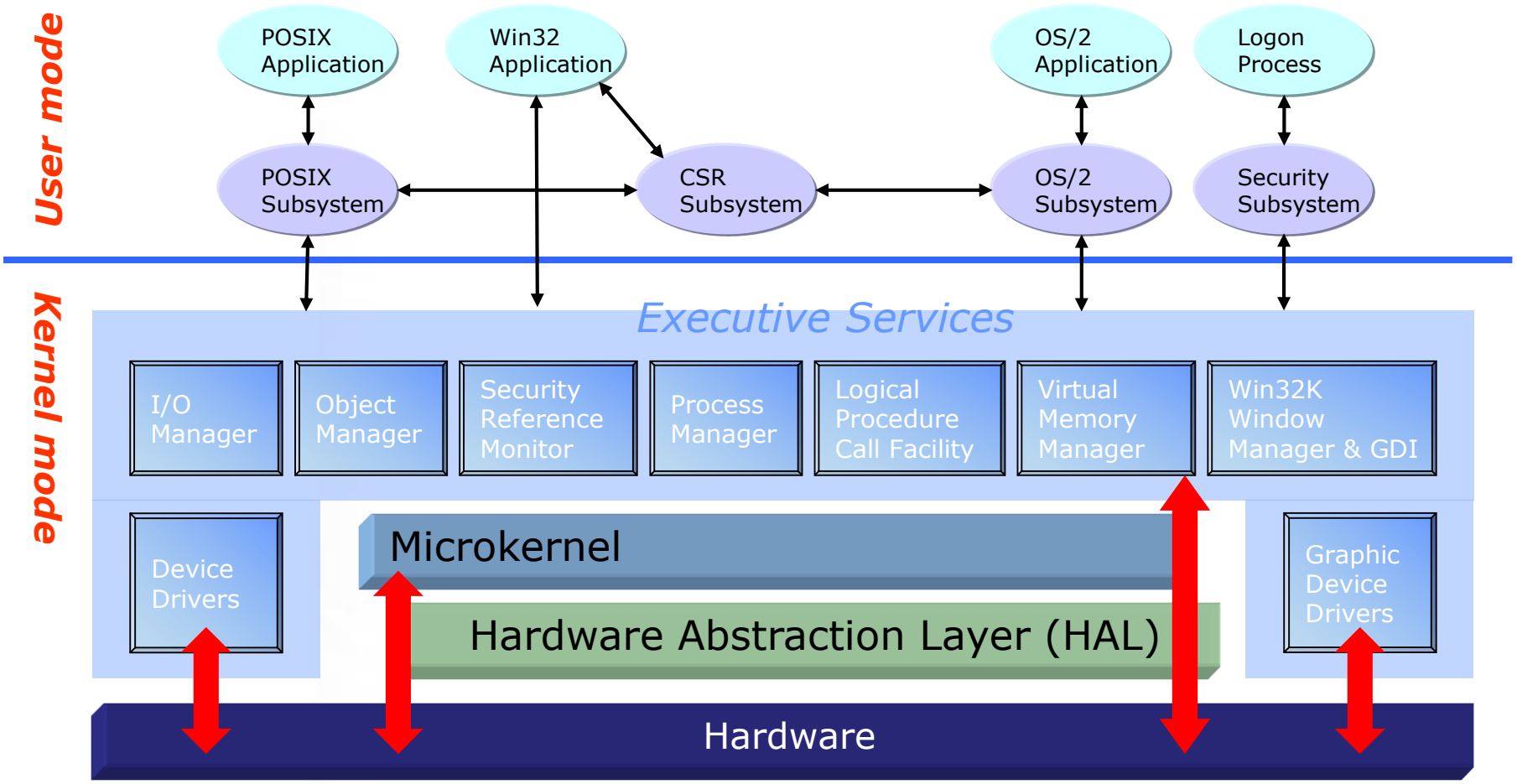
Interrupt vs. Trap



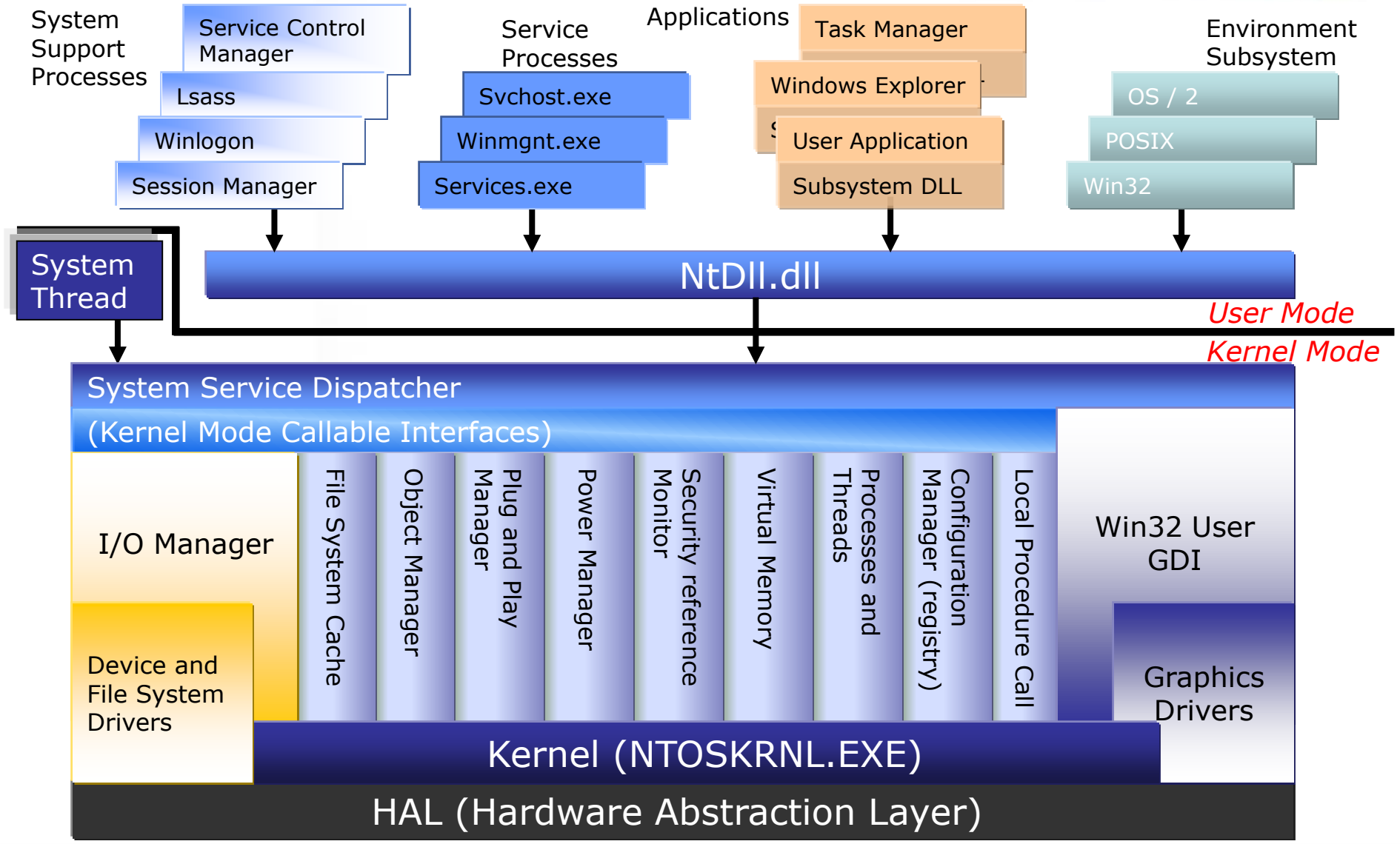
The Trap Instruction Operation



Windows OS Architecture – 1



Windows OS Architecture – 2



What is Win32 API – 1

- Windows OS
  - Microsoft사에서 85년 11월에 개발/발표한 GUI 기반 운영체제
  - 사용자를 위한 강력한 그래픽 환경, 쉬운 인터페이스
  - 현 윈도우 체계는 95년 8월에 발표된 윈도우 95 이후
- Win32 API
  - Win32: Windows가 지원하는 CPU버전
    - Win16: 16bit CPU를 지원하기 위해 사용했던 API 버전
  - API (Application Programming Interface)
    - 운영체제가 응용프로그램을 위해 제공하는 운영체제 함수의 집합
- Framework?
  - Microsoft가 제시하는 차세대 API
  - 내부적으로는 동일하게 Win32 또는 Win64를 이용

What is Win32 API – 2

- Win32 API
  - API에 대한 이해
    - API함수는 Windows OS가 제공하는 일종의 함수
    - 운영체제가 다양한 서비스를 제공하기 위해 공개하는 함수
    - 즉, API의 이해는 운영체제 자원 사용에 대한 전반적인 이해
  - Windows 프로그램 개발의 기반
    - 다른 방법, 도구를 사용하더라도 API 기반이기에 완전한 분리는 불가
    - 고급 기능은 API의 이해유무가 OS 이해와 접근에 큰 비중을 차지
  - 높은 프로그래밍의 자유도 제공
    - 다양한 클래스 라이브러리의 제공
    - Visual Tool 사용시 도구 자체의 기능을 뛰어넘는 설계가 가능