

메모리와 프로세스 구조

안태진(taejin@codecure.smuc.ac.kr)

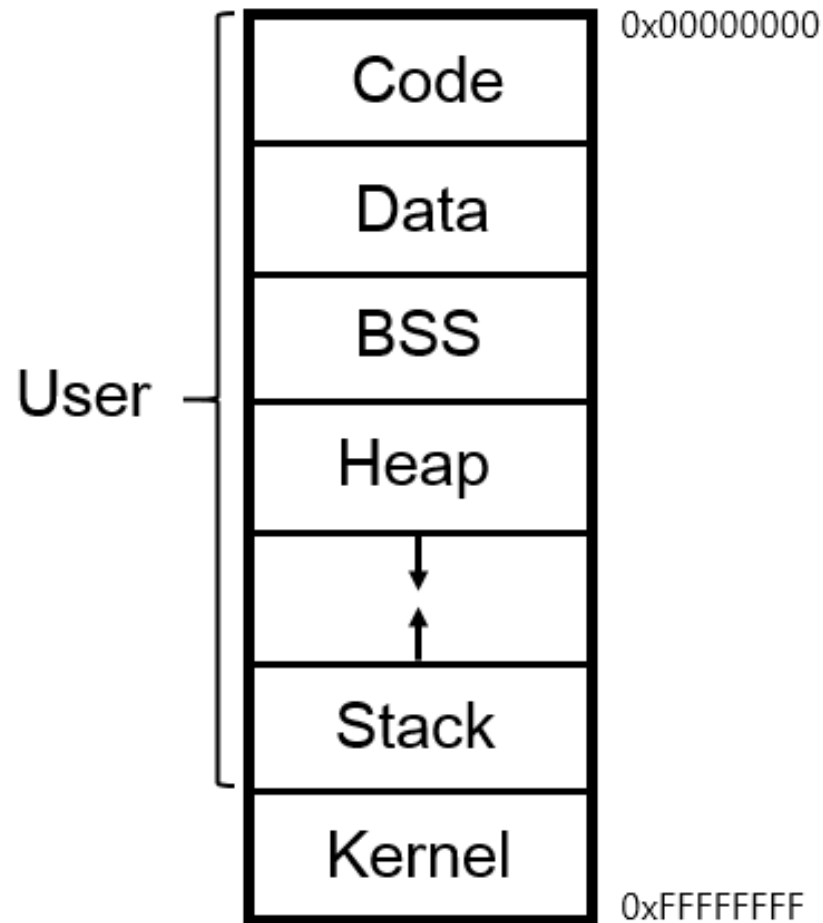
상명대학교 보안동아리 CodeCure

목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

메모리 구조

- 메모리 영역



메모리 구조

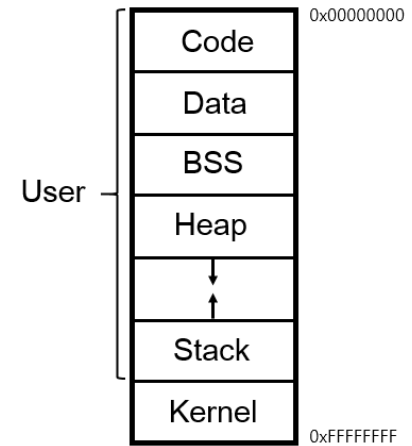
- 메모리 영역

- 커널 영역(Kernel Space)

- 운영체제가 구동되는 영역
- 시스템 운영에 필요한 메모리
 - 시스템 운영?
 - 메모리나 저장장치 내에서 주소공간 관리

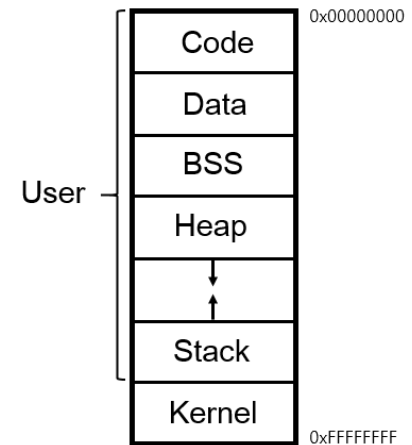
- 유저모드에서 접근불가능

- 유저모드?
 - 유저 어플리케이션 코드만 실행 가능한 모드
 - 시스템 데이터, 하드웨어에 직접적인 접근이 불가능
- 시스템 전체의 안전성과 보안을 지키기 위해 접근이 불가능함
 - 운영체제 데이터에 접근, 수정하지 못하게 함으로써 오동작을 유발하는 유저 애플리케이션이 시스템 전체의 안정성을 해치지 않게 함



메모리 구조

- 메모리 영역
 - 유저 영역(User Space)
 - 프로그램이 동작하기 위해 사용되는 메모리
 - Code, Data, BSS, Heap, Stack영역으로 나누어짐
 - Code(Text) 영역(Code(Text) Segment)
 - 작성한 소스 코드 저장
 - 읽기 전용 데이터 (Read Only)
 - CPU가 이 영역에 있는 명령을 읽고 처리
 - 기계어 저장
 - 컴파일 타임에 크기 결정, 크기가 변하지 않음



메모리 구조

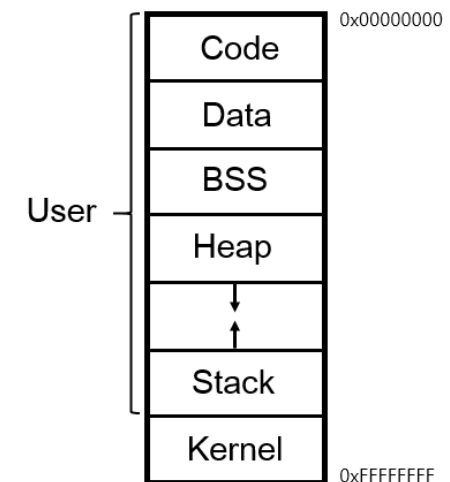
- 메모리 영역

- Data 영역(Data Segment)

- 초기화 된 전역변수, 정적(static)변수
- 프로그램 시작과 동시에 할당되고 프로그램 종료 시 소멸
- 컴파일 타임에 크기 결정, 크기가 변하지 않음

- BSS 영역(Block Stated Symbol Segment)

- 초기화 되지 않은 전역변수, 정적(static)변수
- 데이터 영역과 BSS 영역을 나누는 이유
 - 실행 파일(프로그램)의 크기를 줄이기 위해서임
 - BSS 영역은 할당 필요없이 0으로 채움

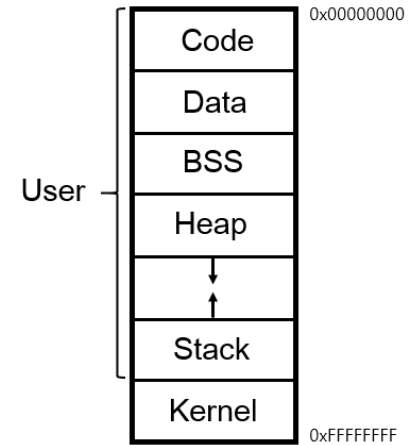


메모리 구조

- 메모리 영역

- Heap 영역(Heap Segment)

- 사용자가 직접 관리할 수 있는 메모리 영역
- 동적 할당
 - 프로그램 실행 도중 크기가 결정되는 할당 방법
 - e.g., malloc(), calloc(), realloc()
- 낮은 주소에서 높은 주소로 할당 (↓)
- 런타임에 크기 결정

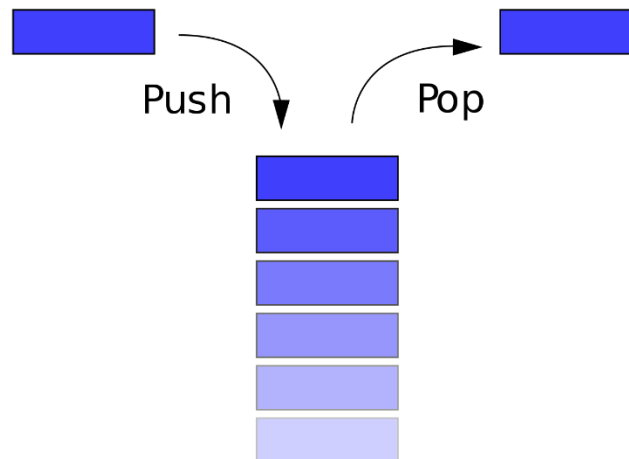
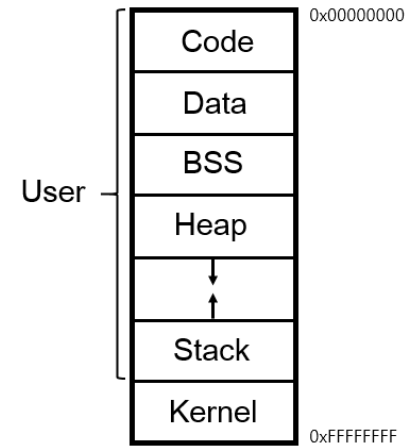


메모리 구조

- 메모리 영역

- Stack 영역(Stack Segment)

- 지역변수, 매개변수, return값 등 저장
- 높은 주소에서 낮은 주소로 할당 (↓)
- 컴파일 타임에 크기 결정
- LIFO(Last In First Out)
 - 후입선출, 나중에 들어 데이터가 가장 먼저 나가는 방식
- push 명령어로 데이터 저장, pop 명령어로 데이터 인출



목 차

- 메모리 구조

- 메모리 영역
- 스택 프레임

- 프로세스

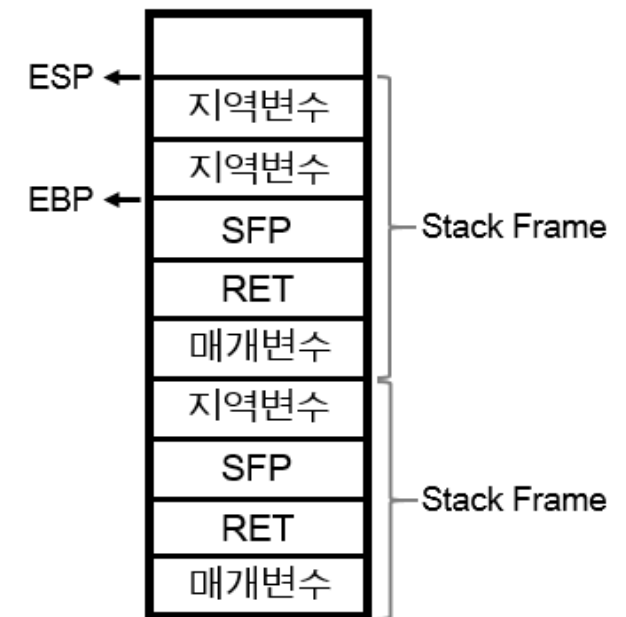
- 프로세스란
- 프로세스 구성

- 가상 메모리

- 가상 메모리란
- 페이징(Paging)
- 세그먼테이션(Segmentation)
- 페이지화된 세그먼테이션(Paged Segmentation)

메모리 구조

- 스택 프레임(Stack Frame)
 - 함수 호출 시 사용되는 스택 공간
- ESP(Extend Stack Pointer)
 - 스택 프레임의 가장 최상부, 현재위치
- EBP(Extend Base Pointer)
 - 스택 프레임이 시작된 주소
- RET
 - 복귀할 함수의 주소
- SFP(Saved Frame Pointer)
 - EBP 복귀 주소 저장



목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

프로세스

- 프로세스란
 - 프로그램(Program)
 - 보조기억장치에 저장되어있는 명령어 파일
 - 생명 X, 실행될 수 있는 파일인 상태
 - 프로세스(Process)
 - 주기억장치에 적재되어 CPU에 의해 처리되는, 실행 중인 프로그램
 - 생명 O, 실행되고 있는 상태
- 프로그램 -> 실행되지 않은 것
- 프로세스 -> 실행된 것

목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

프로세스

- 프로세스 구성

- 프로세스 제어 블록(PCB, Process Control Block)
- 프로세스에 대한 정보를 가지고 있는 자료구조
- 각 프로세스가 생설될 때마다 고유의 PCB를 가짐
- 커널 영역에 저장

프로세스 식별자
프로세스 상태
프로그램 카운터
레지스터 저장 영역
프로세서 스케줄링 정보
계정 정보
...

프로세스

1. 프로세스 식별자(PID, Process Identifier)

- 운영체제가 각 프로세스를 식별하기 위해 부여된 식별번호

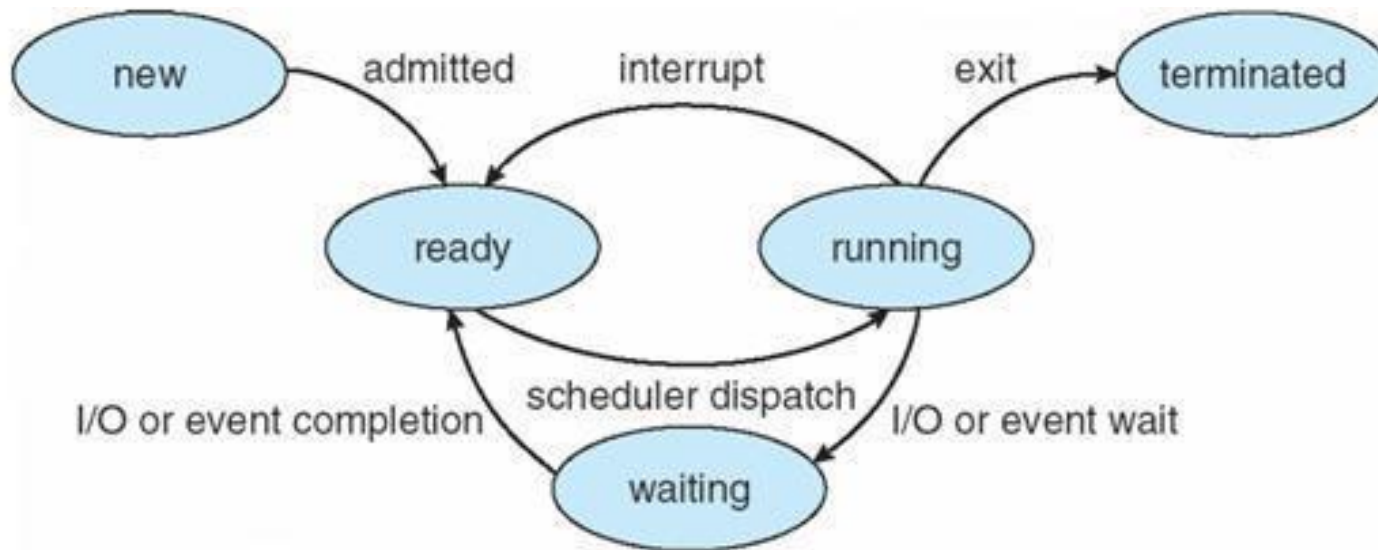
2. 프로세스 상태(Process State)

- 생성(new)
 - 프로세스가 막 만들어진 상태, 메모리에 올라가기 전의 상태
- 준비(ready)
 - 프로세스가 메모리로 올라간 상태, CPU에 올라가기 전 상태
- 실행(running)
 - CPU를 할당 받아 실제 수행되고 있는 상태
- 대기(waiting)
 - I/O(Input/Output) 또는 이벤트로 인해 잠시 대기상태로 전환된 상태
 - 이벤트
 - 어떤 반응이나 동작을 수행하도록 사용자가 생성시킨 동작
 - e.g., 마우스 클릭
 - 이벤트가 종료되면 준비상태로 돌아감

프로세스

2. 프로세스 상태(Process State)

- 완료(terminated)
 - 실행상태인 프로세스가 종료된 상태



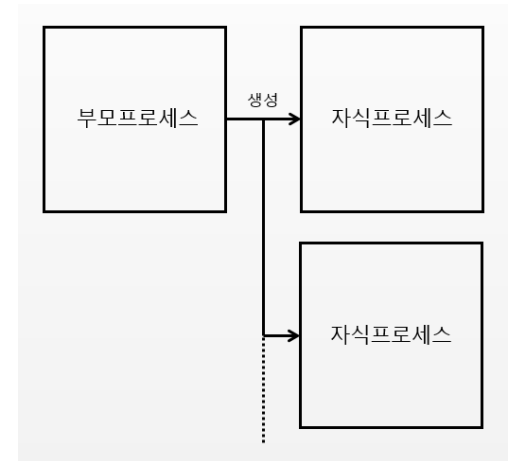
프로세스

3. 프로그램 카운터(PC, Program Counter)
 - 다음으로 실행할 명령어가 저장된 메모리 주소 값
4. CPU 레지스터(Register) 정보
5. 스케줄링(Scheduling) 우선순위
 - 여러 프로세스들의 실행될 순서를 결정한 값
6. 계정 정보
 - CPU 사용 시간, 계정 번호 등
 - 계정 번호(UID, User ID)
 - 프로세스를 실행한 특정 사용자에게 부여한 번호

프로세스

7. 부모 프로세스, 자식 프로세스

- 부모 프로세스와 자식프로세스에 대한 포인터
 - 부모 프로세스
 - 프로그램 실행에 의해 처음 생성된 프로세스
 - 자식 프로세스
 - 부모 프로세스에 의해 복사되어 생성된 프로세스



8. 입출력 정보

- 프로세스에 할당된 입출력 장치와 파일 목록

9. 메모리 관리 정보

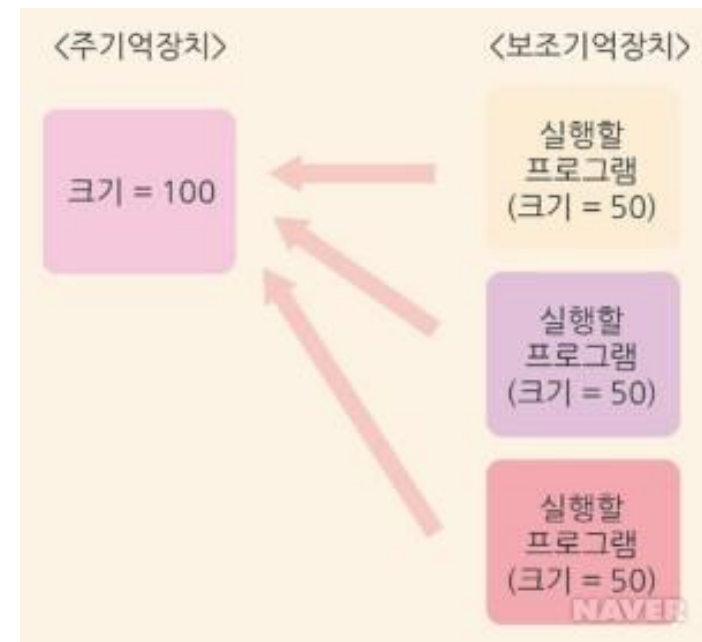
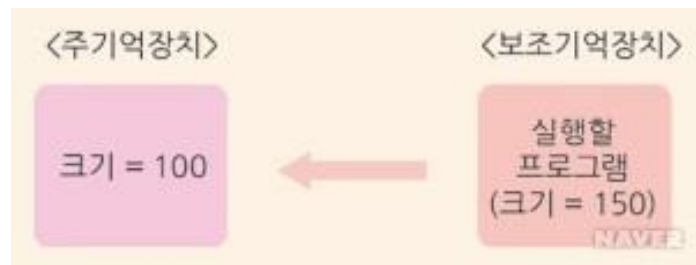
- 기준 레지스터, 경계(한계) 레지스터
- 페이지 테이블, 세그멘테이션 테이블

목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

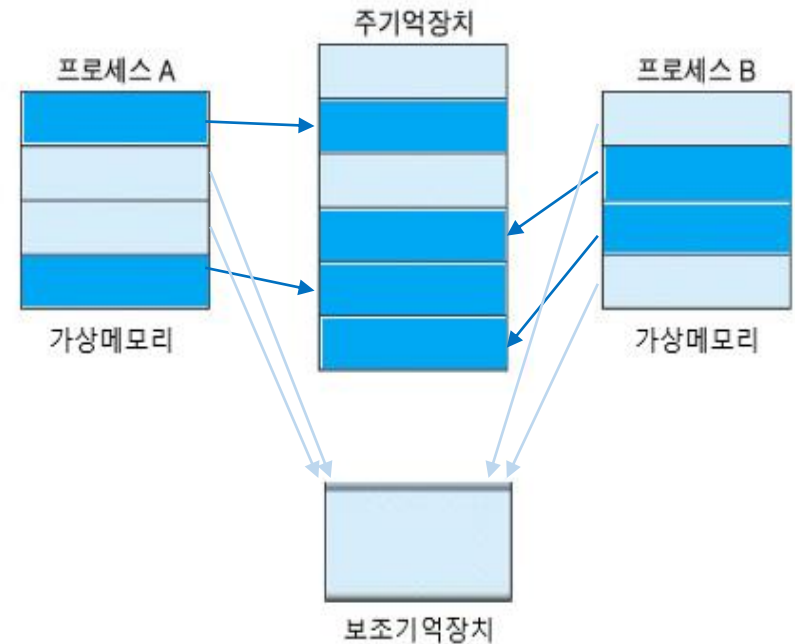
가상메모리

- 가상 메모리란(Virtual Memory)
 - 실행할 프로세스의 크기가 주기억장치보다 클 때 문제가 발생
 - 프로세스의 필요한 부분만 주기억장치에 적재하고, 나머지는 보조기억장치에 넣어 동작하는 방법으로 해결



가상메모리

- 프로세스가 사용하는 주소가 주기억장치에 접근하는 주소와 다름
- 프로세스가 사용하는 주소
 - > 가상 주소 공간, 논리적 주소
- 주기억장치에 접근하는 주소
 - > 실제 주소 공간, 물리적 주소
- 메모리 관리 장치(MMU, Memory Management Unit)
 - 논리적 주소에서 물리적 주소로 변경해주는 장치



목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

가상메모리

- 페이징(Paging)

- 가상 메모리와 실제 메모리를 특정한 크기 단위인 페이지와, 프레임으로 관리하는 방식

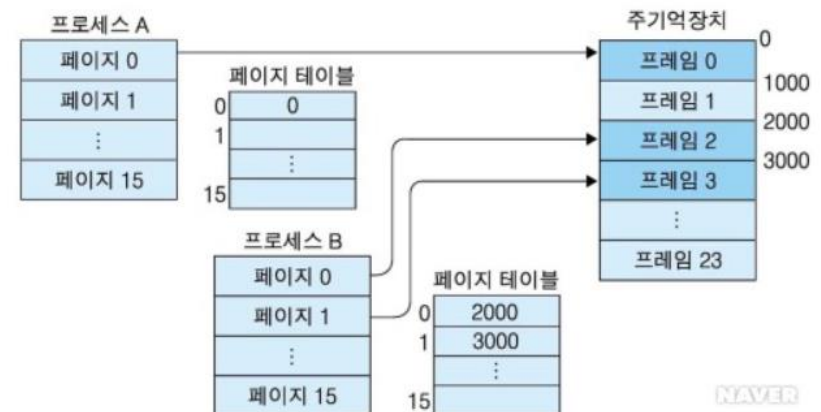
- 페이지(Page), 프레임(Frame)

- 논리적인 의미와 관계없이 동일한 크기로 자른 블록

- 페이지->가상메모리, 프레임->실제메모리

- 페이지테이블(Page Table)

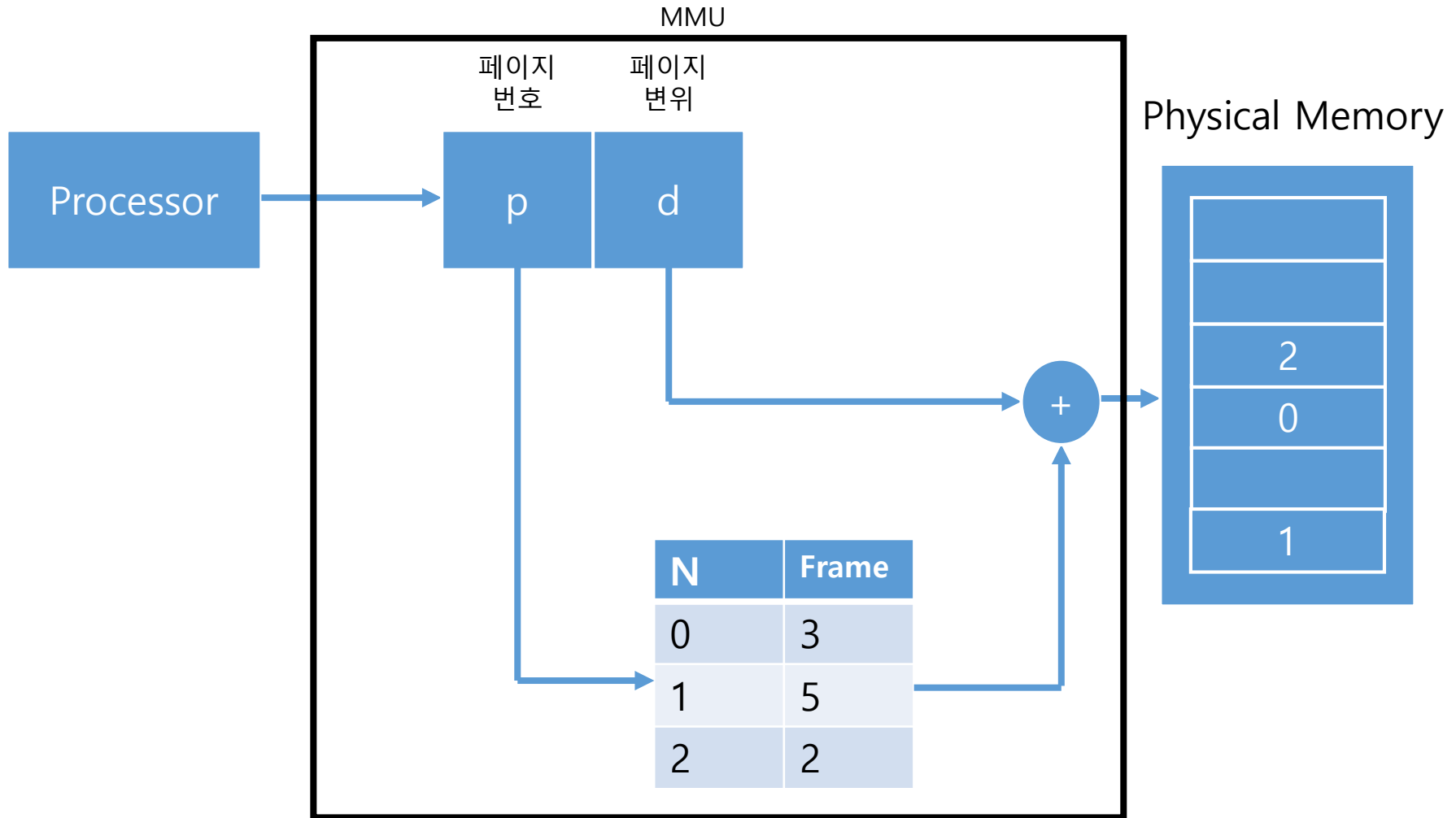
- 각각의 페이지가 주기억장치의 어느 프레임에 저장될지에 대한 위치 정보를 나타냄



NAVER

가상메모리

- 페이징(Paging)



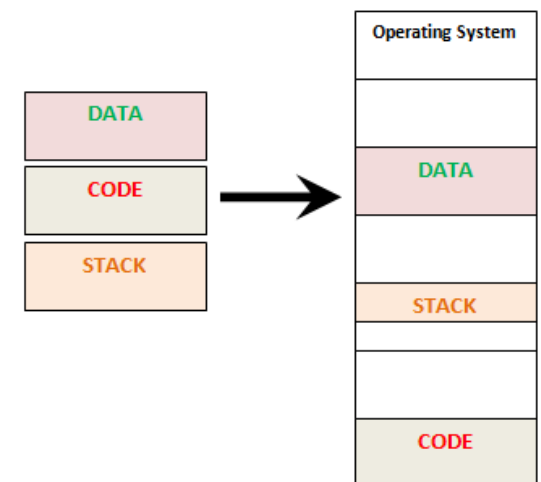
목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

가상메모리

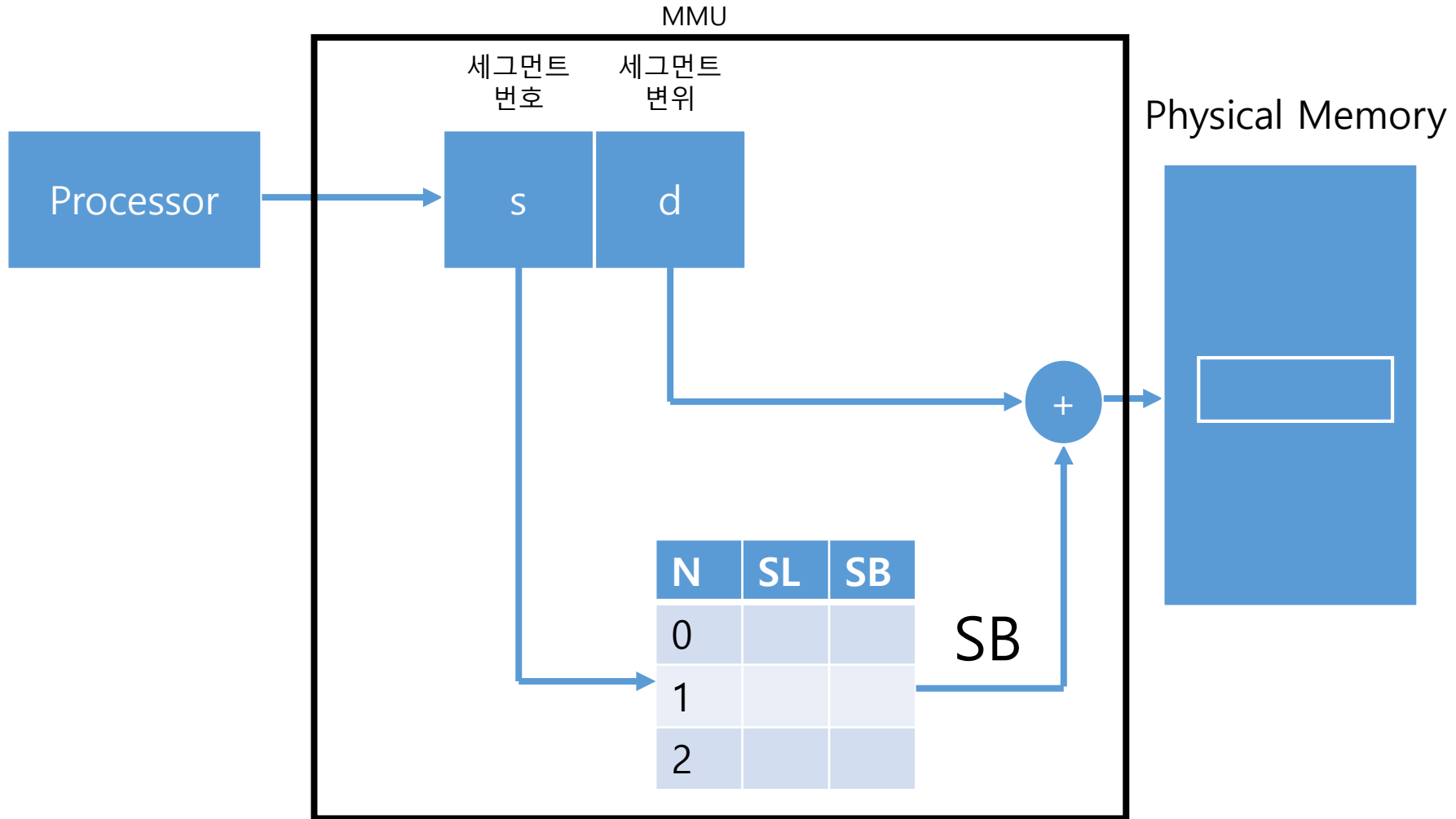
- 세그먼테이션(Segmentation)
 - 가상 주소 공간을 논리적 단위(영역, Segment)로 분할해 실제 주소 공간에 올리는 방식
 - 페이징과 다르게 세그먼트들은 크기가 서로 달라 미리 분할해두기 불가능
 - 메모리에 적재될 때 빈 공간을 찾아 할당하는 가상 메모리 관리 기법
- 세그먼트 테이블(Segment Table)
 - 실제 메모리 영역의 위치 정보 제공
 - 세그먼트 한계(SL, Segment Length), 세그먼트 기준(SB, Segment Base)

Figure 3: Processes Loaded To Memory



가상메모리

- 세그먼테이션(Segmentation)



목 차

- 메모리 구조
 - 메모리 영역
 - 스택 프레임
- 프로세스
 - 프로세스란
 - 프로세스 구성
- 가상 메모리
 - 가상 메모리란
 - 페이징(Paging)
 - 세그먼테이션(Segmentation)
 - 페이지화된 세그먼테이션(Paged Segmentation)

가상메모리

- 페이지화된 세그먼테이션(Paged Segmentation)
 - 세그먼테이션 기법 + 페이징 기법
 - 프로세스를 세그먼트 단위로 적재하되,
세그먼트를 적재할 때는 페이지 단위로 적재하는 방법
 - 한 프로세스는 하나의 세그먼테이션 테이블을 가지고,
한 세그먼테이션은 하나의 페이지 테이블을 가짐

감사합니다!