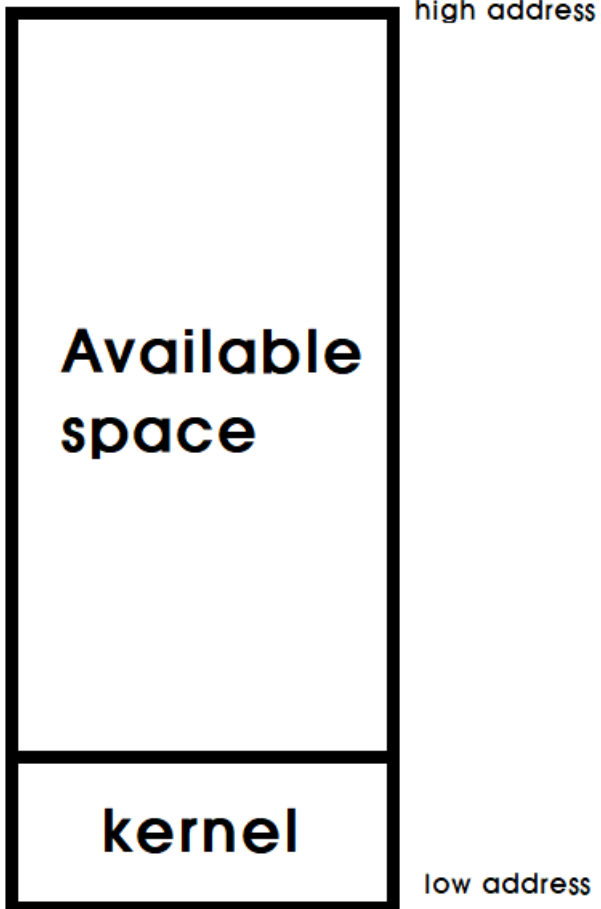
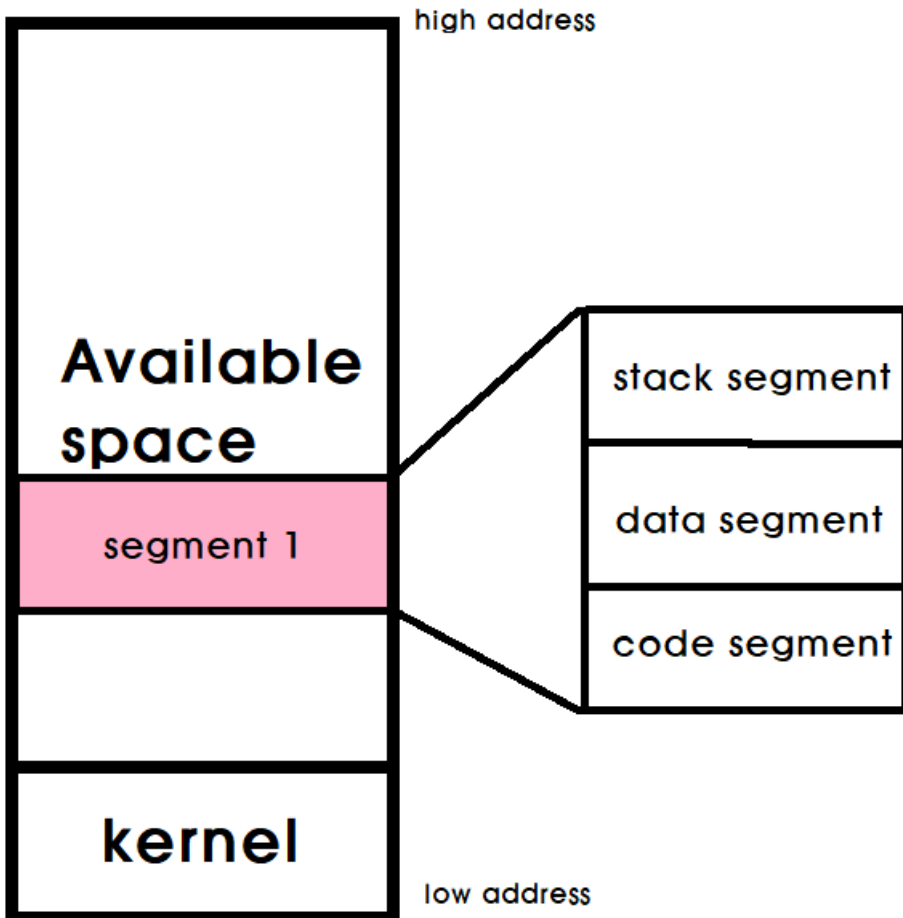


8086 Memory Architecture

01.

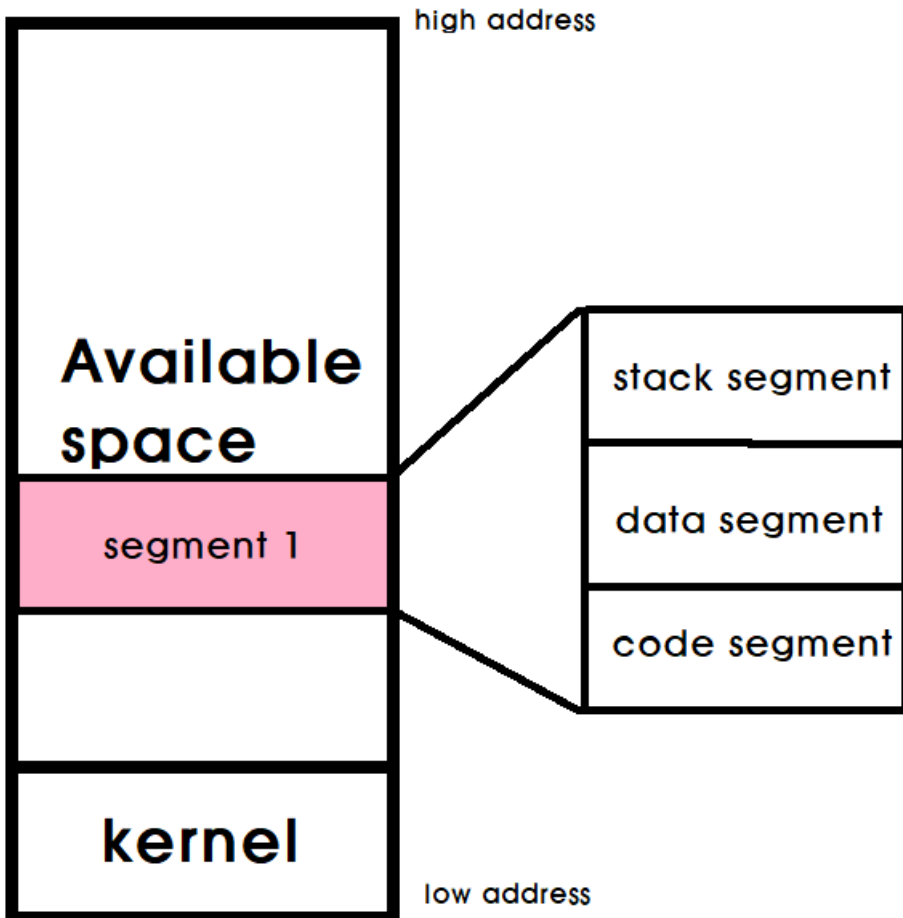


01.



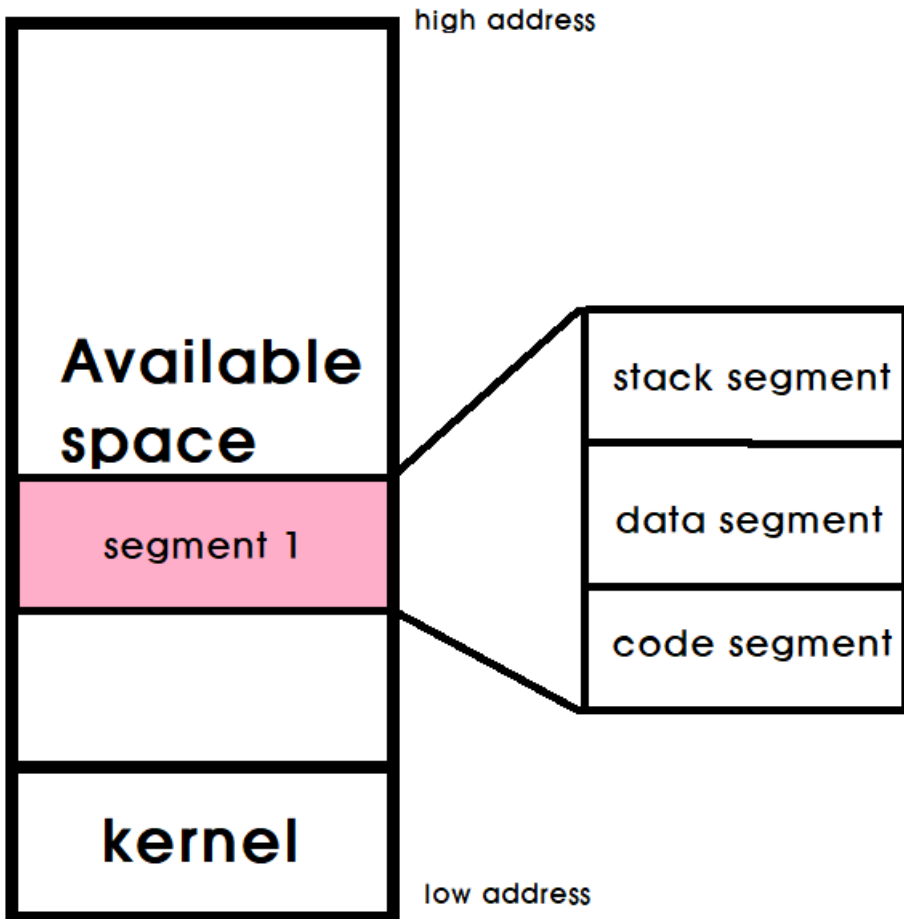
우리가 사용하는 buffer가 위치하는 곳이다.
push, pop 연산을 통해 stack 이용 가능

01.



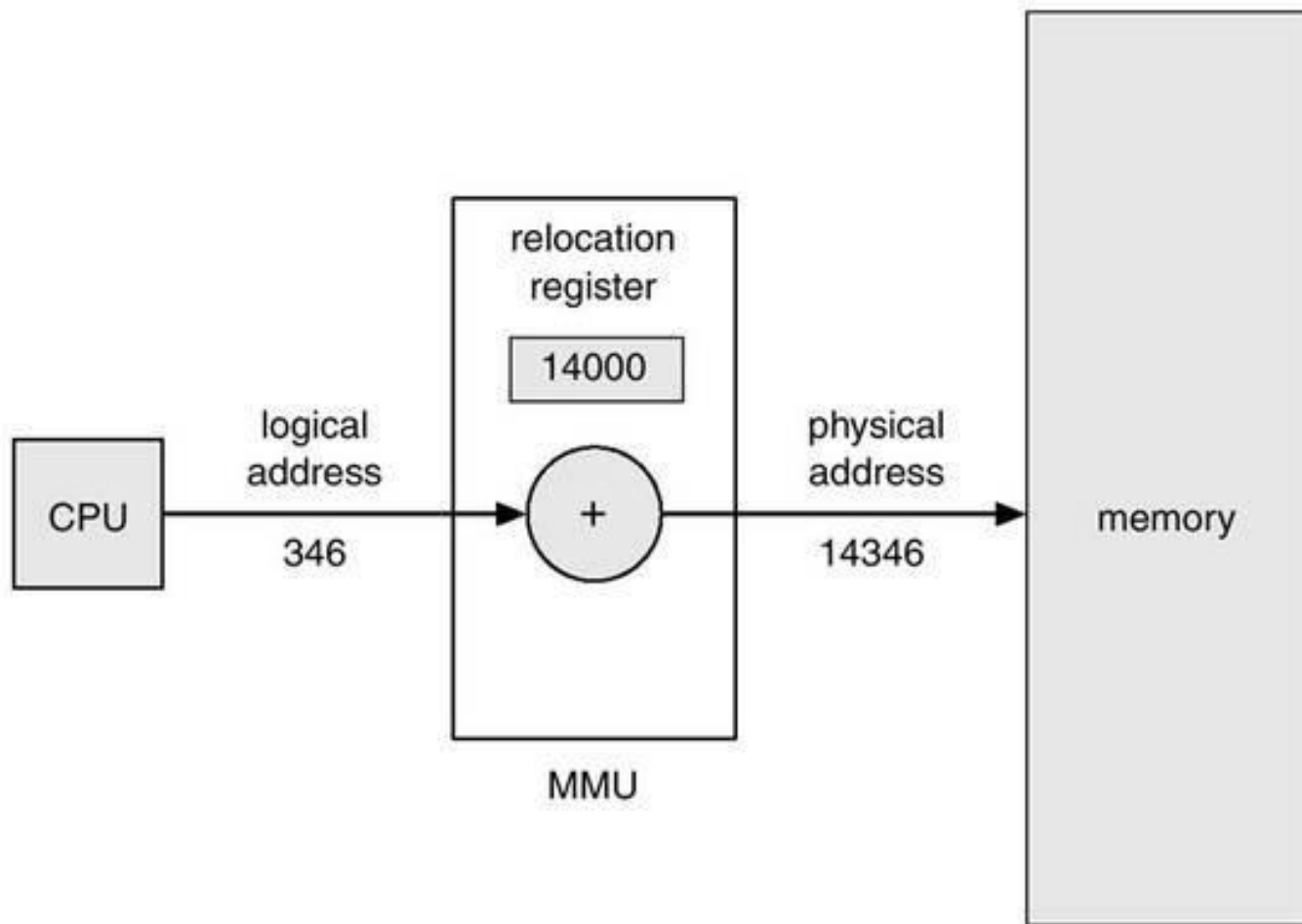
data segment에는 프로그램 실행시 사용되는 전역변수들이 자리잡게 된다.

01.

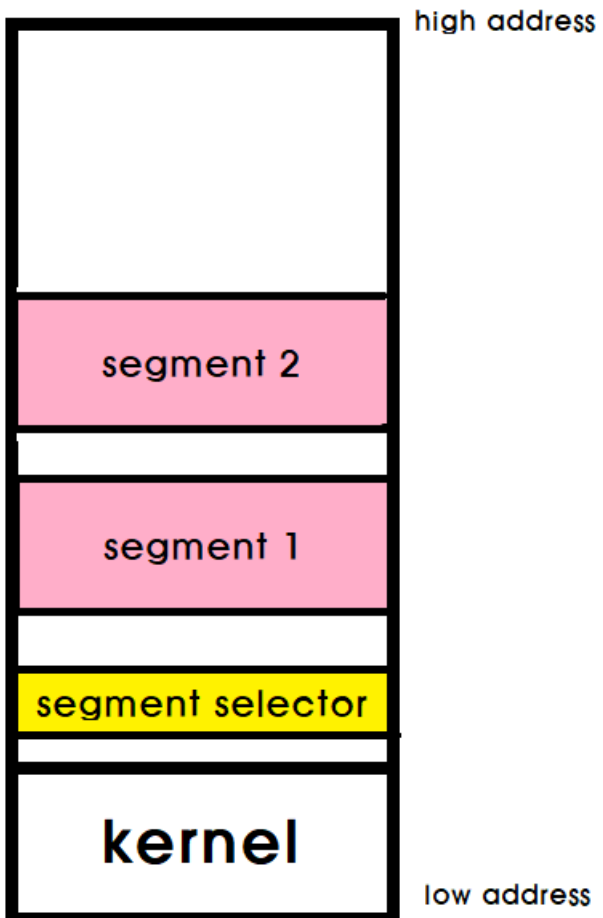


code segment에는 프로그램이 알아들을 수 있는 명령어(instruction)들이 들어있다.

01.



01.



범용 레지스터

1. EAX(Extended Accumulator Register)

: 곱셈과 나눗셈 명령에서 자동으로 사용되고, 함수의 리턴값이 저장되는 용도.

2. EBX(Extended Base Register)

: ESI나 EDI와 결합하여 인덱스에 사용된다.

3. ECX(Extended Counter Register)

: 반복 명령어 사용시 반복 카운터로 사용된다. ECX 레지스터에 반복할 횟수를 지정해 놓고, 반복 작업을 수행하게 된다.

4. EDX(Extended Data Register)

: EAX와 같이 쓰이며 부호 확장 명령 등에 쓰인다.

명령 포인터

1. EIP(Extended Instruction Pointer)

다음에 실행해야 할 명령어가 존재하는 메모리 주소가 저장된다. 현재 명령어를 실행 완료 한 후에 EIP 레지스터에 저장되어 있는 주소에 위치한 명령어를 실행하게 된다. 실행 전 EIP 레지스터에는 다음 실행해야 할 명령어가 존재하는 주소의 값이 저장된다.

2. ESP(Extended Stack Pointer)

: 하나의 스택 프레임의 끝 지점 주소가 저장된다. PUSH, POP 명령어에 따라서 ESP의 값이 4Byte씩 변한다.

3. EBP(Extended Base Pointer)

: 하나의 스택 프레임의 시작 지점 주소가 저장된다. 현재 사용되는 스택 프레임이 소멸되지 않는 동안 EBP의 값은 변하지 않는다. 현재의 스택 프레임이 소멸되면 이전에 사용되던 스택 프레임을 가리키게 된다.

세그먼트 레지스터

1.CS(Code Segment)

: 실행 가능한 명령어가 존재하는 세그먼트의 오프셋이 저장된다.

2.DS(Data Segment)

: 프로그램에서 사용되는 데이터가 존재하는 세그먼트의 오프셋이 저장된다.

3.SS(Stack Segment)

: 스택이 존재하는 세그먼트의 오프셋이 저장된다.

01.

인덱스 레지스터, 플래그 레지스터 등등

Thank you