

# 2차시 수업



# 부트 과정 - 1

- 일단 컴퓨터를 키자
- BIOS가 로딩이 됩니다 (하드웨어 초기화 등등을 하지요)
- BIOS가 부팅 순서에 따라서 디스크의 MBR을 읽어요  
그러다가 MBR 511-512(맨 마지막 부분)이 55 AA면 묻지도 따지지도  
않고 일단 그 MBR을 0x7c00으로 복사함



# MBR이 어떻게 생겼는가

결국은 우리가 코드를 넣을 부분은  
최대 446바이트 밖에 없다!!!

마스터 부트 레코드의 구조

주소		설명		크기 (바이트)
십육진수	십진수			
0000	0	코드 영역		440 최대 446
01B8	440	디스크 서명		4
01BC	444	보통 없음(Null); 0x0000		2
01BE	446	프라이머리 파티션 테이블 (4개의 16바이트 엔트리)		64
01FE	510	55h	MBR 서명; 0xAA55	2
01FF	511	AAh		
MBR, 전체 크기: 446 + 64 + 2 =				512



# 여기서 드는 의문

- 마! 446 바이트면 몇 글자 집어넣으면 꼭 찰거 같은데!
- 그래서 일반적으로 Stage 1(MBR) 부트로더에서 완전히 커널로 이동하지 않고 Stage 2(Loader)로 이동합니다...



# 어쨌든 돌아와서 부트과정 - 2

- 0x7c00으로 로드가 된 Stage 1 부트로더가 Stage 2를 부릅니다.
- Stage 2가 커널을 적재합니다.
- 커널이 시동됩니다.
- 끝



# 16비트 리얼모드

- 16비트 레지스터 사용 (ax, bx, cx, dx, si, di, ...)
- 메모리를 1MB까지 쓸수 있음
- 메모리에 대한 무제한 접근성
- 참시만, 16비트이면  $2^{16}$ 바이트 = 64kb 써야 하는 것 아닌가요?



# 세그먼트 주소지정

- 16비트 레지스터 2개가 있다!
- [세그먼트 레지스터 : 오프셋 레지스터] =  
세그먼트 레지스터 \* 16 + 오프셋 레지스터
- example ) [0x1234 : 0x1111] =  
0x12340 + 0x1111 = 0x13451
- 쉽게 생각해서 네번 쉬프트 하고 더합시다! 그러면 메모리 1MB 가능!



# 32비트 보호모드의 필요성

- 16비트에선 말이지...
- 커널이 올라갔다 -> 그 자리에 어플리케이션이 올라오고 싶다 -> 올라간다 -> 커널이 죽었습니다
- 1MB 이상 메모리를 못쓰는데여?



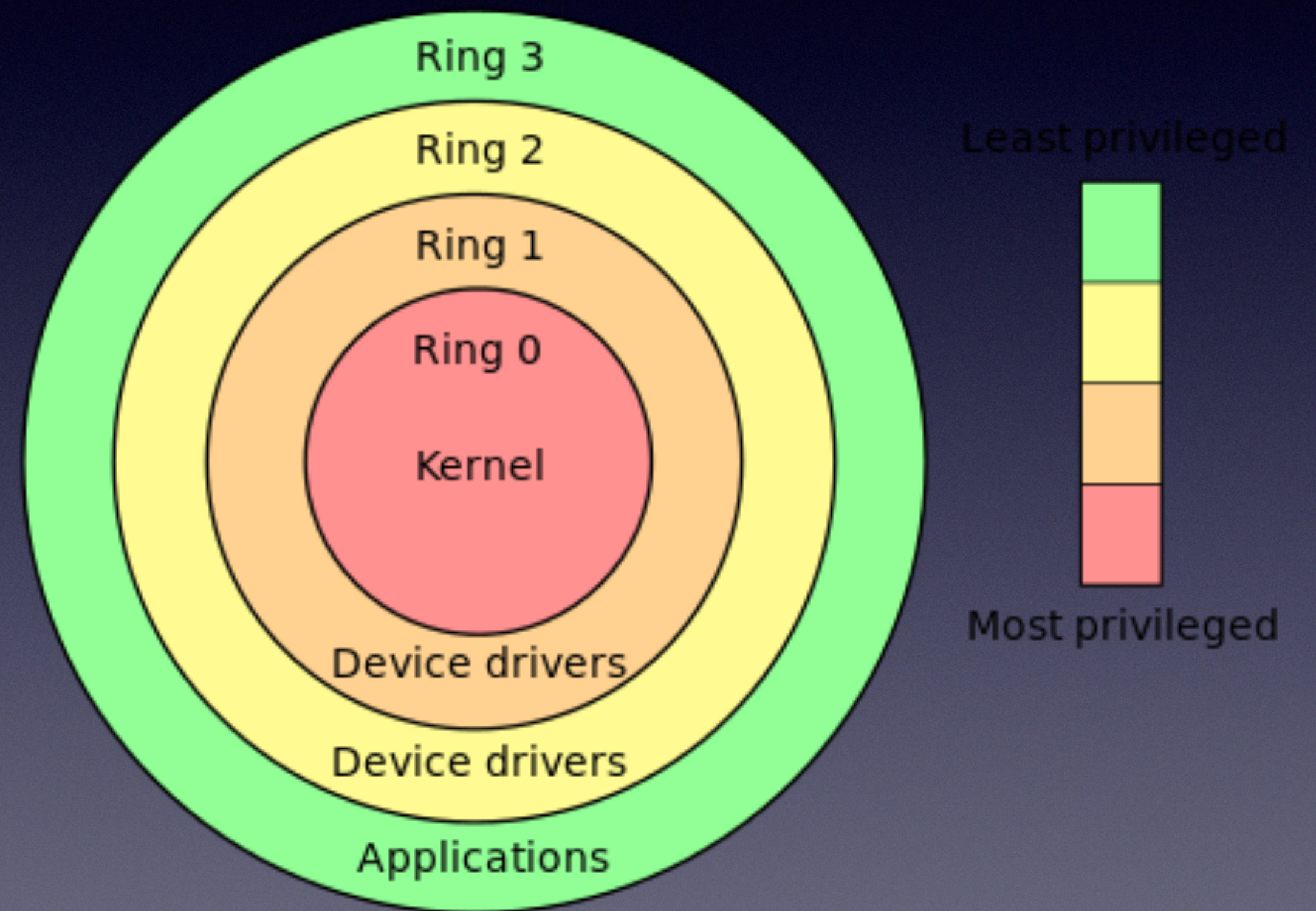
# 32비트 보호모드

- 32비트 레지스터 사용 (eax, ebx, ecx, ...)
- 일단 메모리를  $2^{32}$ 바이트 = 4GB까지 사용 가능하다!  
(사실 PAE라고 확장할 수는 있음.. 대부분 운영체제에서 불안정)
- GDT나 IDT라는 고-급 개념으로 커널 공간을 보호할 수 있다
- 하지만 메모리 접근이 어려워지고 페이징 기법이 좀 재미없음



# 마! 저번에 설명했던 Ring 0 드감?

- Intel 플랫폼에선 대부분 Ring 0, 3만 사용
- Ring 0가 커널 특권 모드
- Ring 3가 응용 프로그램 모드
- 특권 명령은 Ring 0에서만 사용 가능  
ex) HLT 등등...





# 요약

- 16비트 -> 1MB, 무제한 접근(좋은거 아니야)
- 32비트 -> 4GB, 한정적 접근
- 예고편으로 64비트 -> 16EB, 나도 자신 없는 페이징 기법...