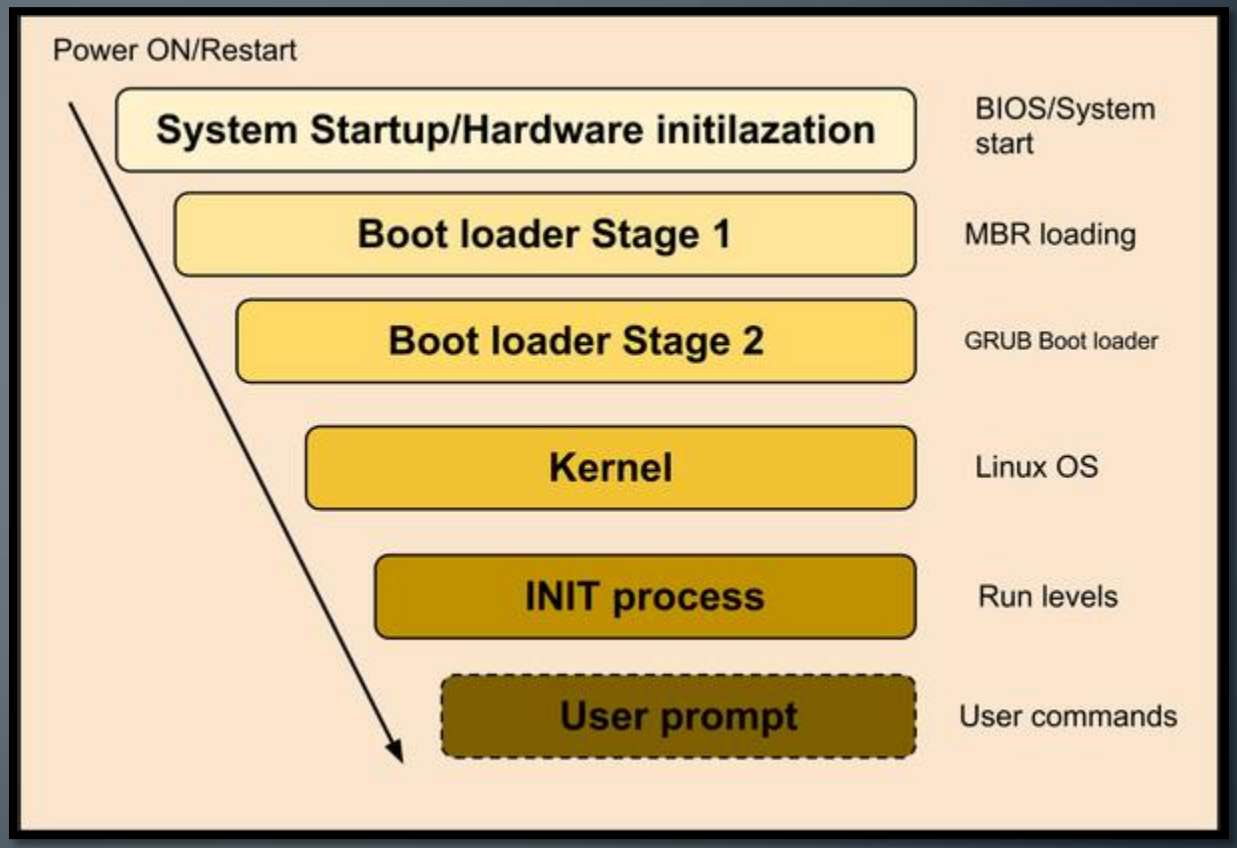
MBR

How to Write Operating System

Bootstrap


Power ON/Restart
System Startup/Hardware initilazation
BIOS/System start
Boot loader Stage 1
MBR loading
Boot loader Stage 2
GRUB Boot loader
Kernel
Linux OS
INIT process
Run levels
User prompt
User commands

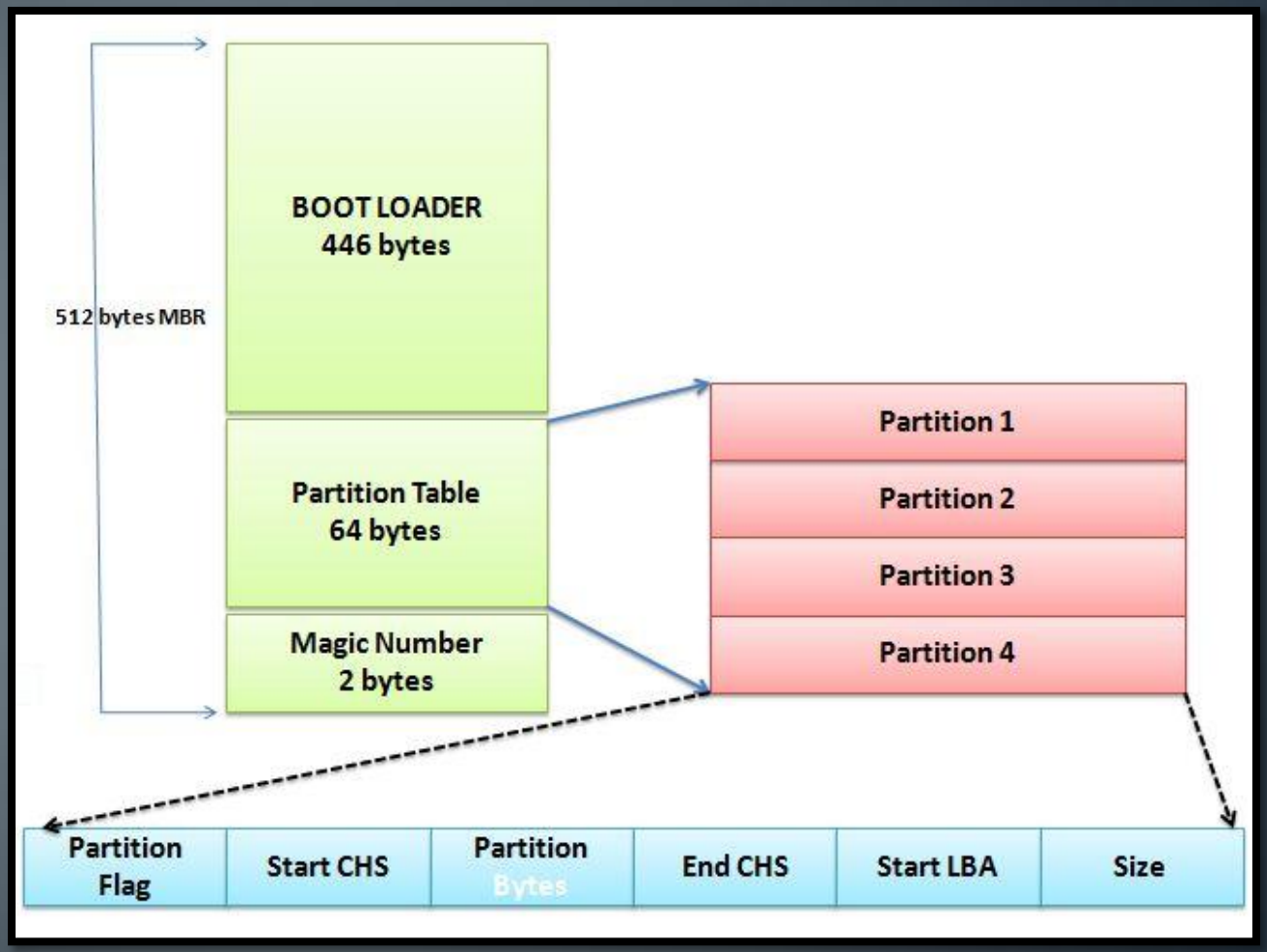
BIOS

- Sistemdeki donanımların kontrolünü
- Bootloader'ı çalıştırır
- ...

MBR

- Sistem boot edilirken BIOS'un çalıştırtığı kod parçasının büyüklüğü en fazla 512 byte olabilir.
- 512 bytes:
  - 2 byte'lık mbr imzası alanı,
  - 16 byte'lık (*4 adet*) bölüm tablosu
  - 446 byte'lık kod parçası

MBR


512 bytes MBR
BOOT LOADER
446 bytes
Partition Table
64 bytes
Magic Number
2 bytes
Partition 1
Partition 2
Partition 3
Partition 4
Partition Flag
Start CHS
Partition Bytes
End CHS
Start LBA
Size

Linux | Sistem Çağrısı

%eax	Name	%ebx	%ecx	%edx	%esx	%edi
1	sys_exit	int	-	-	-	-
2	sys_fork	struct pt_regs	-	-	-	-
3	sys_read	unsigned int	char *	size_t	-	-
4	sys_write	unsigned int	const char *	size_t	-	-
5	sys_open	const char *	int	int	-	-
6	sys_close	unsigned int	-	-	-	-

Linux | Sistem Çağrısı

```
mov eax, 4
mov ebx, 1
mov ecx, userMsg
mov edx, lenUserMsg
int 80h

;Read and store the user input
mov eax, 3
mov ebx, 2
mov ecx, num
mov edx, 5          ;5 bytes (numeric, 1 for sign) of that information
int 80h

;Output the message 'The entered number is: '
mov eax, 4
mov ebx, 1
mov ecx, dispMsg
mov edx, lenDispMsg
int 80h

;Output the number entered
mov eax, 4
mov ebx, 1
mov ecx, num
mov edx, 5
int 80h

; Exit code
mov eax, 1
mov ebx, 0
int 80h
```

Örnek bir MBR kodu

```
1   [BITS 16]
2
3       mov ax, 07C0h
4       mov ds, ax
5       mov si, msg
6
7   REPEAT:
8       mov al, [si]
9       test al, al
10      jz EXIT
11
12      mov ah, 0x0e
13      mov bl, 7
14      int 0x10
15      inc si
16
17      jmp REPEAT
18
19  EXIT:
20      jmp $
21      msg db  "Welcome YTU-CE Experimental MBR!..", 0
22      times 510 - ($ - $$) db 0
23      dw 0xAA55
```

Compile & Run

- nasm -f bin -o boot-ytuos.bin boot-ytuos.asm

  -f: flat binary

- dd status=noxfer conv=notrunc if=boot-ytuos.bin of=boot-ytuos.flp

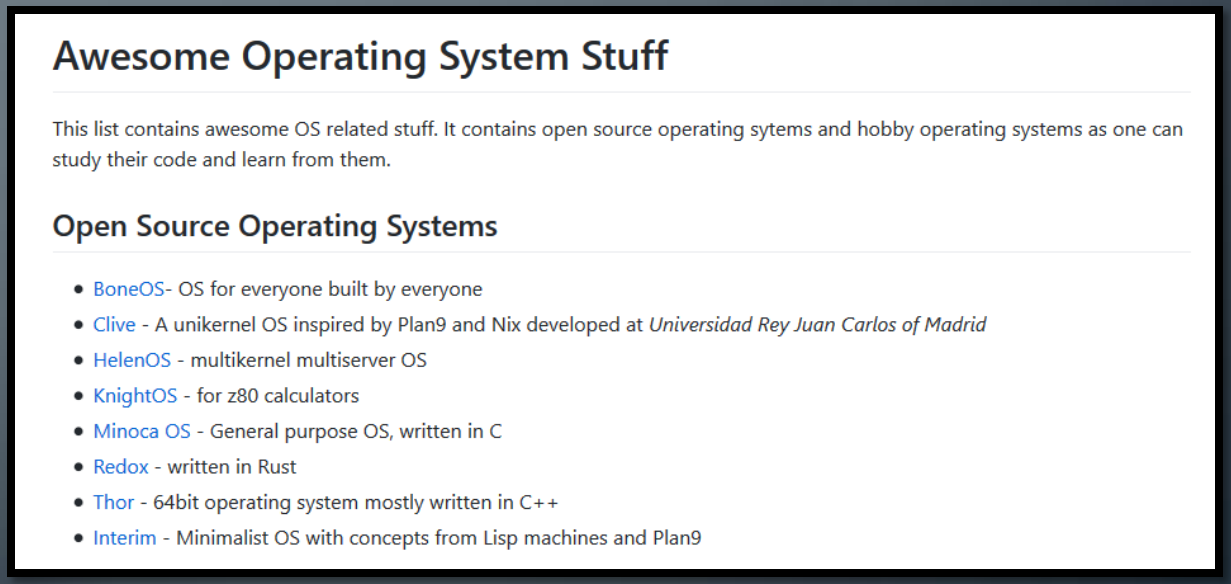
- qemu-system-i386 boot-ytuos.flp

xv6

- https://github.com/mit-pdos/xv6-public
- xv6 is a re-implementation of Dennis Ritchie's and Ken Thompson's Unix Version 6 (v6)

Kaynaklar

- https://github.com/jubalh/awesome-os
- https://www.youtube.com/watch?v=RdbyPwo4W2E
- https://www.youtube.com/watch?v=DZ0-GMtOtEc&list=PL14-cztsT7zpTikkkROTPka86zLdx7AJh
- https://github.com/sindresorhus/awesome



Awesome Operating System Stuff

This list contains awesome OS related stuff. It contains open source operating sytems and hobby operating systems as one can study their code and learn from them.

Open Source Operating Systems

- BoneOS- OS for everyone built by everyone
- Clive - A unikernel OS inspired by Plan9 and Nix developed at *Universidad Rey Juan Carlos of Madrid*
- HelenOS - multikernel multiserver OS
- KnightOS - for z80 calculators
- Minoca OS - General purpose OS, written in C
- Redox - written in Rust
- Thor - 64bit operating system mostly written in C++
- Interim - Minimalist OS with concepts from Lisp machines and Plan9