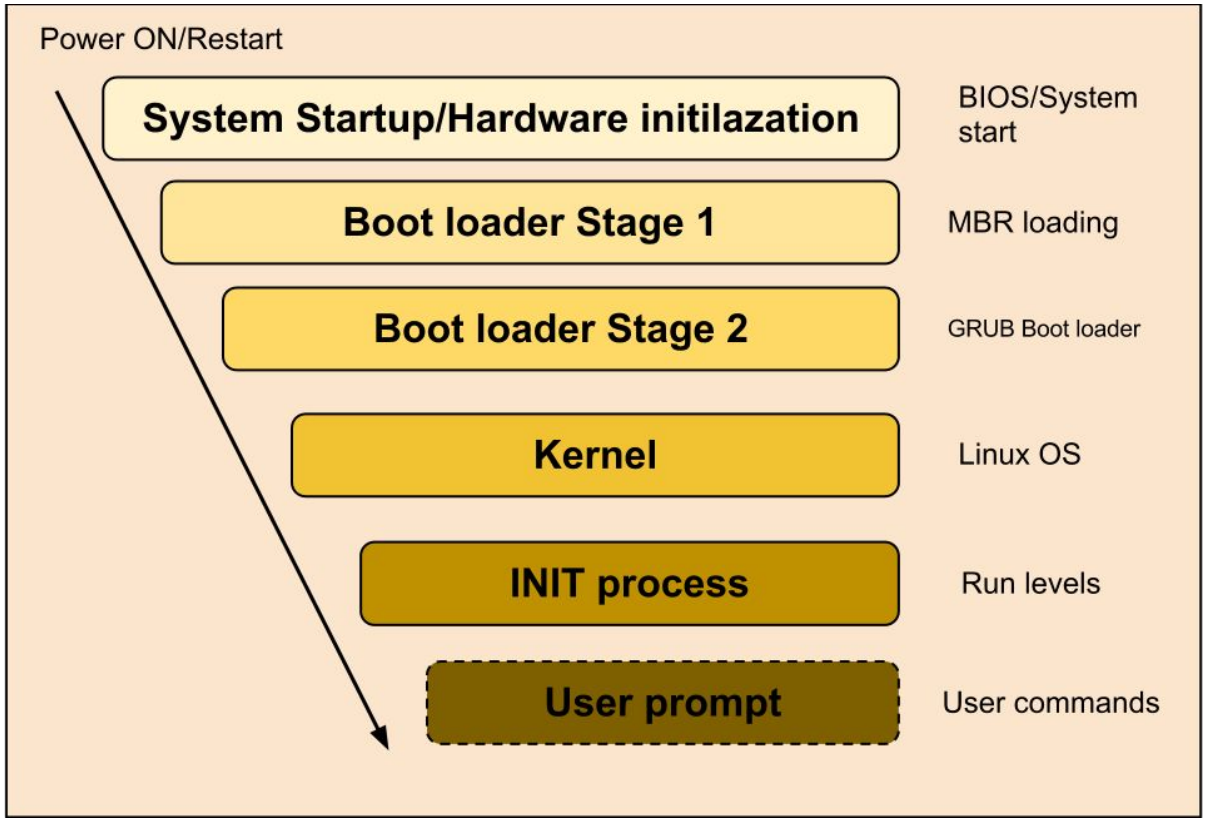
Operating System Development

IEEE

UAlbany Student Branch

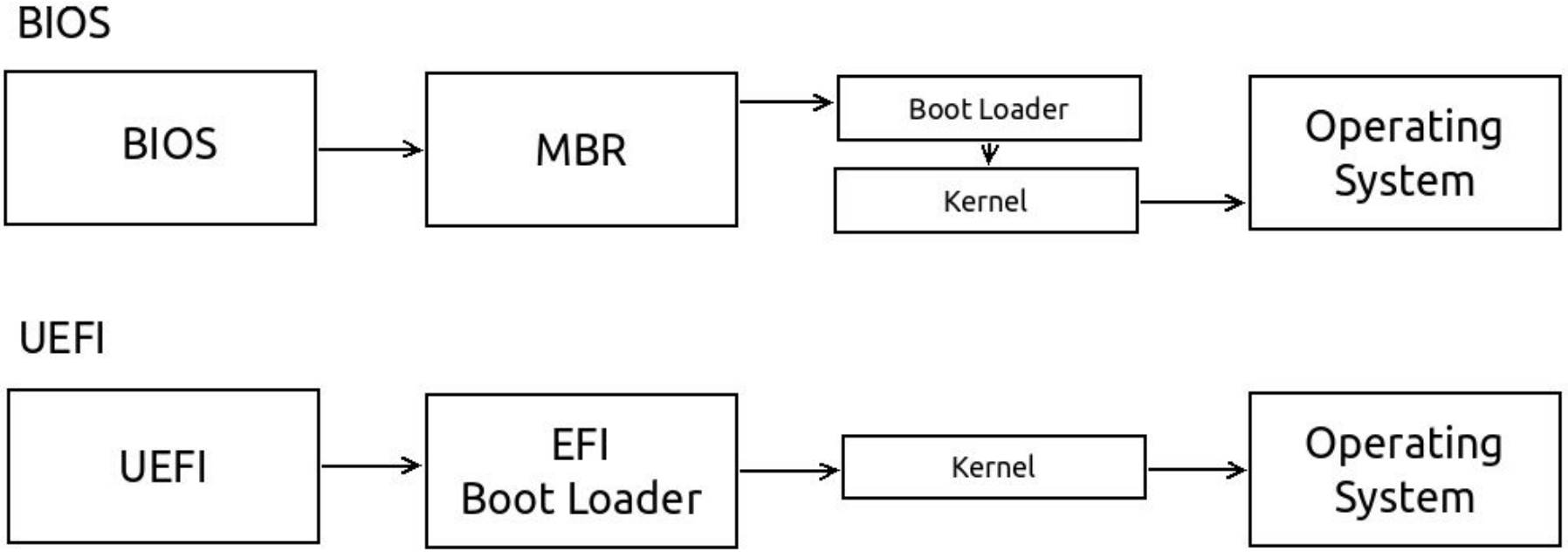
What Happens when we turn on a PC?



Building a Cross Compiler and making an ISO

https://wiki.osdev.org/GCC_Cross-Compiler

Booting, Old VS New



https://www.gnu.org/software/grub/manual/multiboot/multiboot.html

Understanding the state of the processor after boot

3.2 Machine state

When the boot loader invokes the 32-bit operating system, the machine must have the following state:

'EAX'

Must contain the magic value '0x2BADB002'; the presence of this value indicates to the operating system that it was loaded by a Multiboot-compliant boot loader (e.g. as opposed to another type of boot loader that the operating system can also be loaded from).

'EBX'

Must contain the 32-bit physical address of the Multiboot information structure provided by the boot loader (see Boot information format).

'CS'

Must be a 32-bit read/execute code segment with an offset of '0' and a limit of '0xFFFFFFFF'. The exact value is undefined.

'DS'
'ES'
'FS'
'GS'
'SS'

Must be a 32-bit read/write data segment with an offset of '0' and a limit of '0xFFFFFFFF'. The exact values are all undefined.

'A20 gate'

Must be enabled.

'CR0'

Bit 31 (PG) must be cleared. Bit 0 (PE) must be set. Other bits are all undefined.

'EFLAGS'

Bit 17 (VM) must be cleared. Bit 9 (IF) must be cleared. Other bits are all undefined.

All other processor registers and flag bits are undefined. This includes, in particular:

'ESP'

The OS image must create its own stack as soon as it needs one.

'GDTR'

Even though the segment registers are set up as described above, the 'GDTR' may be invalid, so the OS image must not load any segment registers (even just reloading the same values!) until it sets up its own 'GDT'.

'IDTR'

The OS image must leave interrupts disabled until it sets up its own IDT.

Control Registers

CR0

bit	label	description
0	pe	protected mode enable
1	mp	monitor co-processor
2	em	emulation
3	ts	task switched
4	et	extension type
5	ne	numeric error
16	wp	write protect
18	am	alignment mask
29	nw	not-write through
30	cd	cache disable
31	pg	paging

X86 Operating Modes

Operating Mode		Operating System Required	Application Recompile Required	Defaults		Register Extensions	Typical
				Address Size (bits)	Operand Size (bits)		GPR Width (bits)
Long Mode	64-Bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16	16		16
Legacy Mode	Protected Mode	Legacy 32-bit OS	no	32	32	no	32
				16	16		
	Virtual-8086 Mode			16	16		16
	Real Mode	Legacy 16-bit OS					

Writing the kernel wrapper in assembly

```
.section .text
.global _start
.type _start, @function
_start:
  mov $stack_top, %esp       /* Set up the stack */
  mov %ebx, multibootTable   /* save the multiboot info structure pointer */
  call kernelInit            /* Initilize the kernel */
  call kernelMain            /* Run the kernel */
  cli
.deathLoop:
  hlt
  jmp deathLoop
```

Writing Text to the screen from C

VGA text mode is memory mapped, we can access the terminal buffer at address 0xB8000

Text buffer [edit]

Each screen character is represented by two bytes aligned as a 16-bit word accessible by the CPU in a single operation. The lower, or character, byte is the actual code point for the current character set, and the higher, or attribute, byte is a bit field used to select various video attributes such as color, blinking, character set, and so forth.[6] This byte-pair scheme is among the features that the VGA inherited from the EGA, CGA, and ultimately from the MDA.

Attribute								Character							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Blink[n 1]	Background color			Foreground color[n. 2][n. 3]				Code point							

1. ^ Depending on the mode setup, attribute bit 7 may be either the blink bit or the fourth background color bit (which allows all 16 colors to be used as background colours).
2. ^ Attribute bit 3 (foreground intensity) also selects between fonts A and B (see below). Therefore, if these fonts are not the same, this bit is simultaneously an additional code point bit.
3. ^ Attribute bit 0 also enables underline, if certain other attribute bits are set to zero (see below).

Colors are assigned in the same way as in 4-bit indexed color graphic modes (see VGA color palette). VGA modes have no need for the MDA's reverse and bright attributes because foreground and background colors can be set explicitly.

Writing text to video memory

```c
#include<string.h>
#include<stddef.h>
#include<stdint.h>

void alert(char* str, uint8_t color){
    uint16_t* buffer = (uint16_t*)0xB8000;
    for(size_t i = 0; i < strLen(str); i++)
        buffer[i] = str[i] | color << 8;
}

void alertError(char* str){
    alert(str, 0x4f); //white text red background
}

void alertWarning(char* str){
    alert(str, 0xef); //white text yellow background
}
```

Building

- Compile the asm wrapper:
  - i686-elf-as boot.s -o boot.o
- Compile the C kernel:
  - i686-elf-gcc -c kernel.c -o kernel.o -std=gnu99 -ffreestanding -O2 -Wall -Wextra
- Link them:
  - i686-elf-gcc -T linker.ld -o myos.bin -ffreestanding -O2 -nostdlib boot.o kernel.o -lgcc
- Build the ISO
  - grub-mkrescue -quiet -o OSName.iso build/isoSrc>/dev/null

Thank You for Coming!