

- ☑ A. Last time
- ☑ B. Today's class
- ☑ C. Binaries + loading (assuming static linking)
- ☑ D. From power-up to terminal
  - ☑ Step 1: power up
  - ☑ Step 2: firmware
  - ☑ Step 3: OS bootloader
  - ☑ Step 4: kernel
  - ☑ Step 5: init(8)
  - ☑ Step 6: login(1)
- ☑ E. Remarks and observations

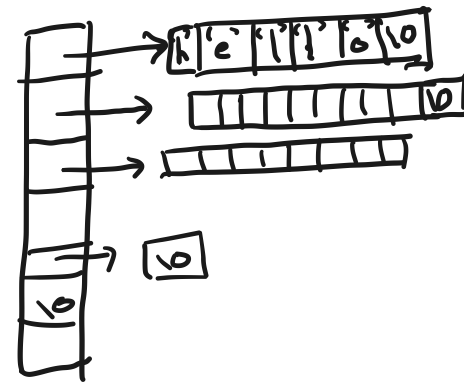
## Binaries/loading

What happens when this code executes?

```

{
  ...
  char* new_argv[];
  char* new_envp[];
  // ...
  // initialize argv and envp
  // ...
}

```



```

if (fork() == 0)
  execve("hello", new_argv, new_envp);
}

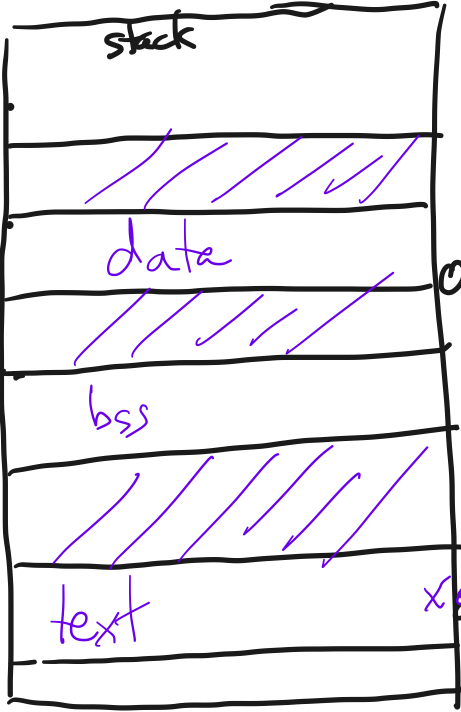
```

```

$ gcc hello.c -o hello -static
$ ls
hello
$ ./hello

```

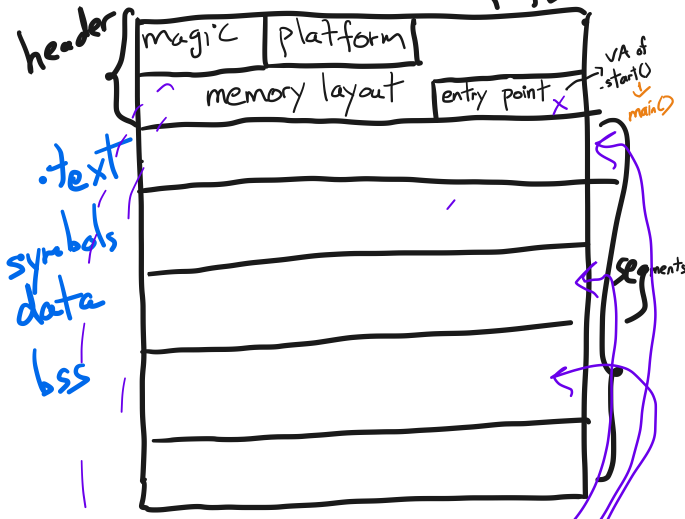
executable file (ELF/PE)



0x5000

0x5000

x 1000



.text  
symbols  
data  
bss

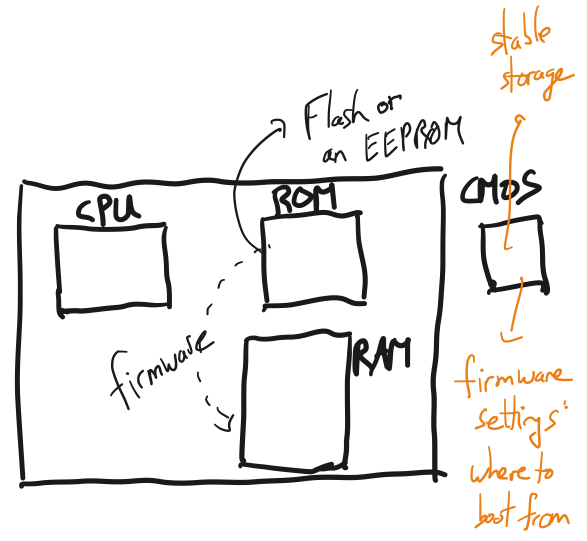
VA len init? offset mem protection

0x1000	5	Y	rx
0x2000	2	Y	rw
0x3000	1	N	rw

loader in Weensy OS:  
k-loader.c

Step 1: power up  
regs zeroed out  
ctl. regs get defaults  
real mode: 8086  
no paging  
1 MB of phys mem

processor hard-wired:  
copy from ROM → RAM  
jump to known offset



Step 2: Firmware

UEFI or BIOS  
time for the boot process

initialize hardware + provide a runtime

Core: real → long mode  
create an identity-mapped pg table  
create an initial IDT  
initialize other processor structures  
set cpl registers

Peripherals/devices: Initialize (UEFI has drivers)

↳ Disk, USB, Display, Keyboard, Mouse, NIC

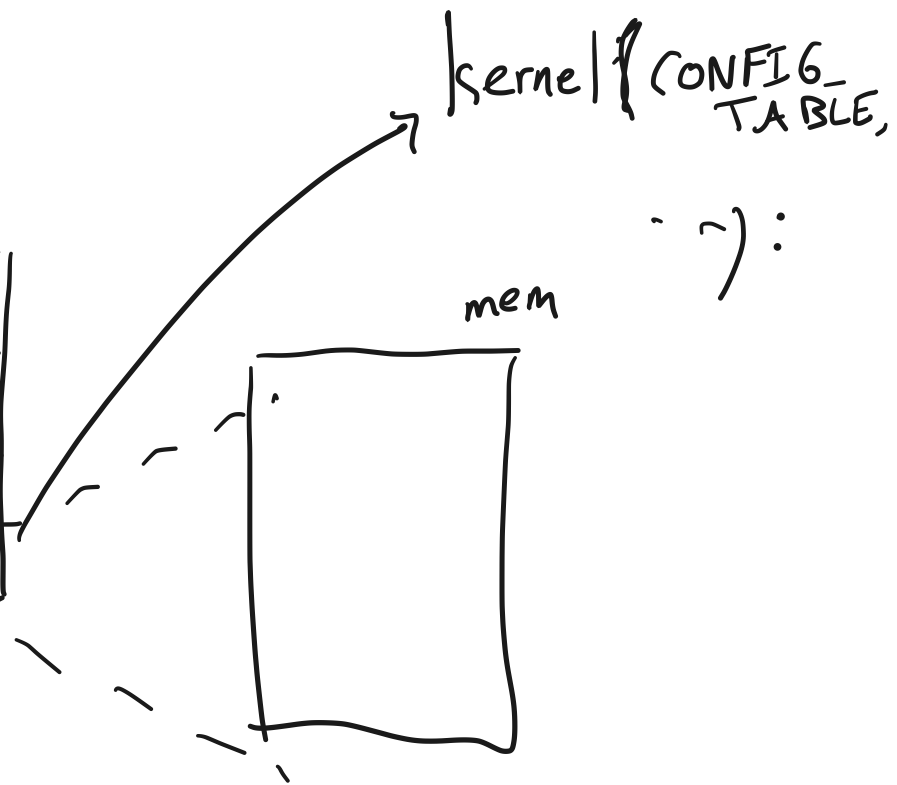
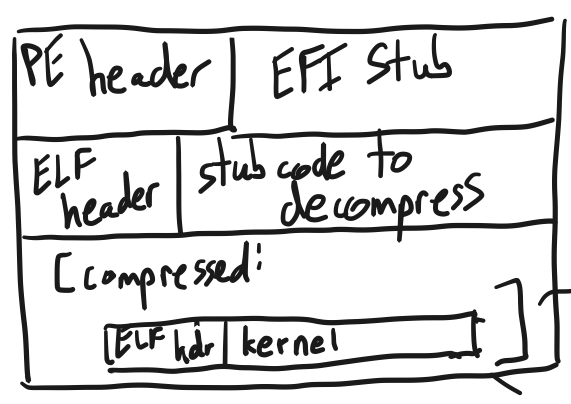
Device Tree initialization (= CONFIGURATION\_TABLE)

Mount VFAT partition → /EFI

Load + execute boot loader

### Step 3: boot loader

vmlinuz



## Step 4: kernel

Initialize the system:

- moving away from identity-mapped <sup>address</sup> space
- rewrite the interrupt descriptor table (IDT)
- load + configure device drivers
- mount the device that will contain '/'  
"root device"

[fsck, <sup>wait</sup> run here, if FS using fsck]

After initialization, kernel forks and exec init(8).

## Step 5: init

Init: PID 1

Most distributions: systemd

Executes as superuser aka root

(a) Finish initializing the system

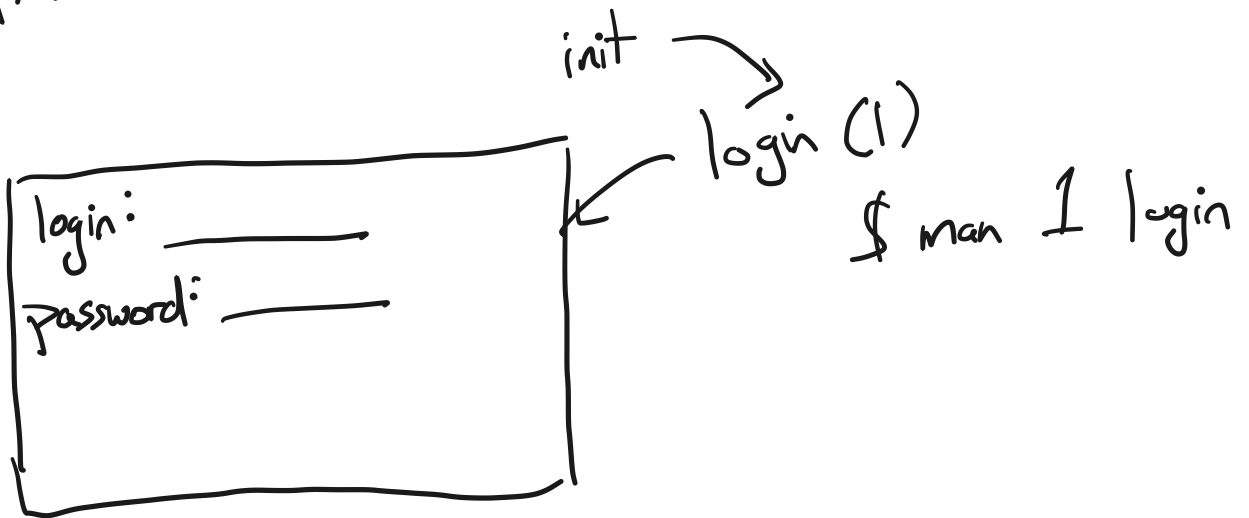
(b) Launch the login manager

(c) Relaunch " " when users log out

(a): network: DHCP  
graphics: set display resolution (GPU)  
power mgmt settings

some of the initial services bind to ports 1...1024.  
"tcpserve"

(b):



---

## Step 6: login

login runs as root

\$ login  
<error>

/etc/passwd  
/etc/shadow

if match:

fork a new process  
parent waits using waitpid().

child:

setuid(2)

\$ man 2 setuid

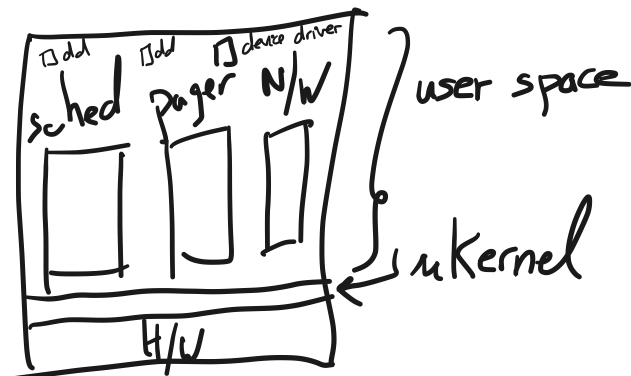
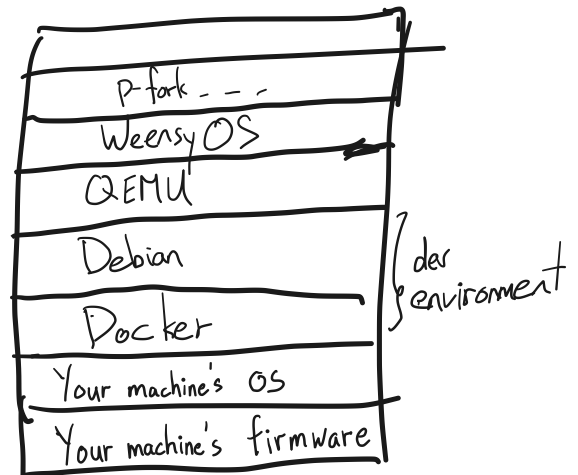
setgid  
change directory to user's home directory  
look up which shell is user's default  
exec() that shell

---

## Ex. Remarks + observations

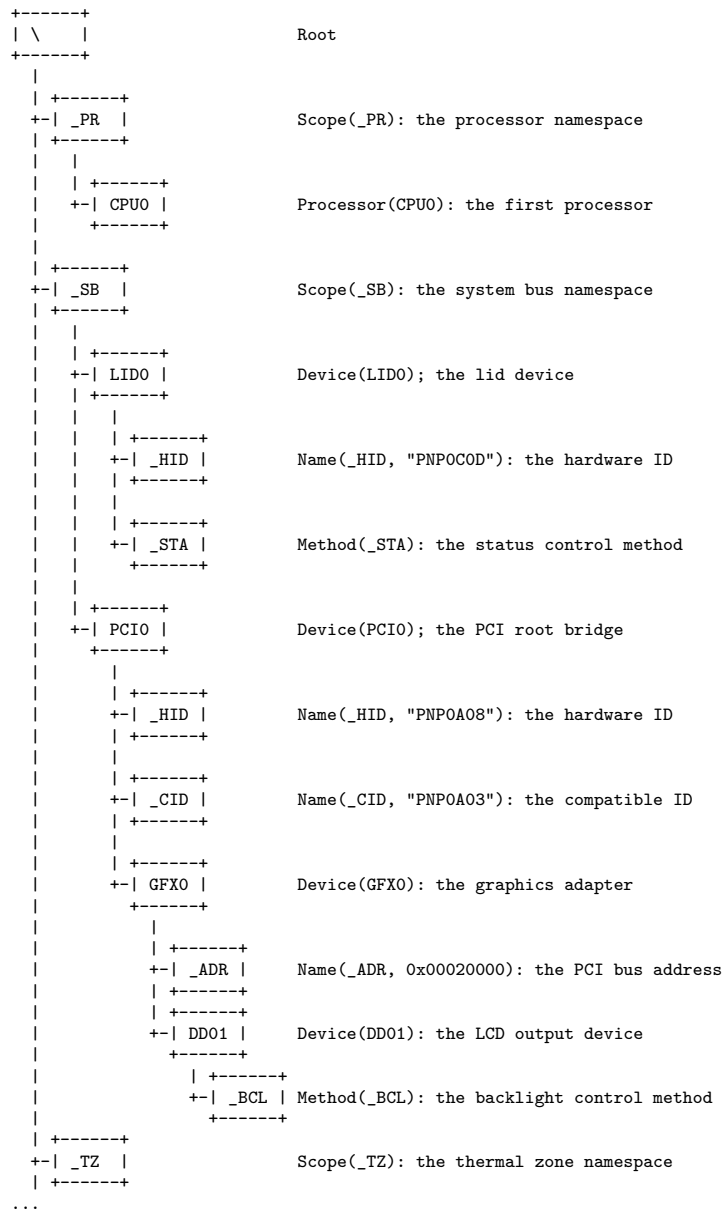
- multiple OS-like things

- monolithic vs. microkernels



# CS202

## Example Device Tree



Credit: Copied and modified from: <https://www.kernel.org/doc/html/latest/firmware-guide/acpi/namespace.html>.