

- ☒ 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?

```

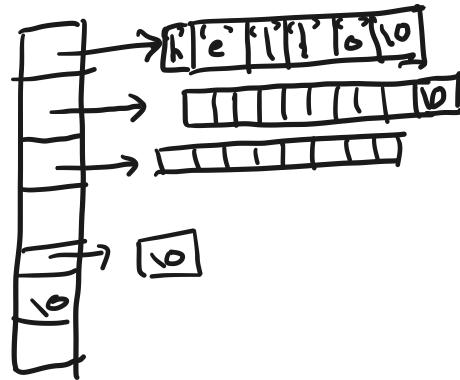
{
    ...
    char* new_argv[];
    char* new_envp[];
    // ...
    // initialize argv and envp
    // ...

```

```

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

```

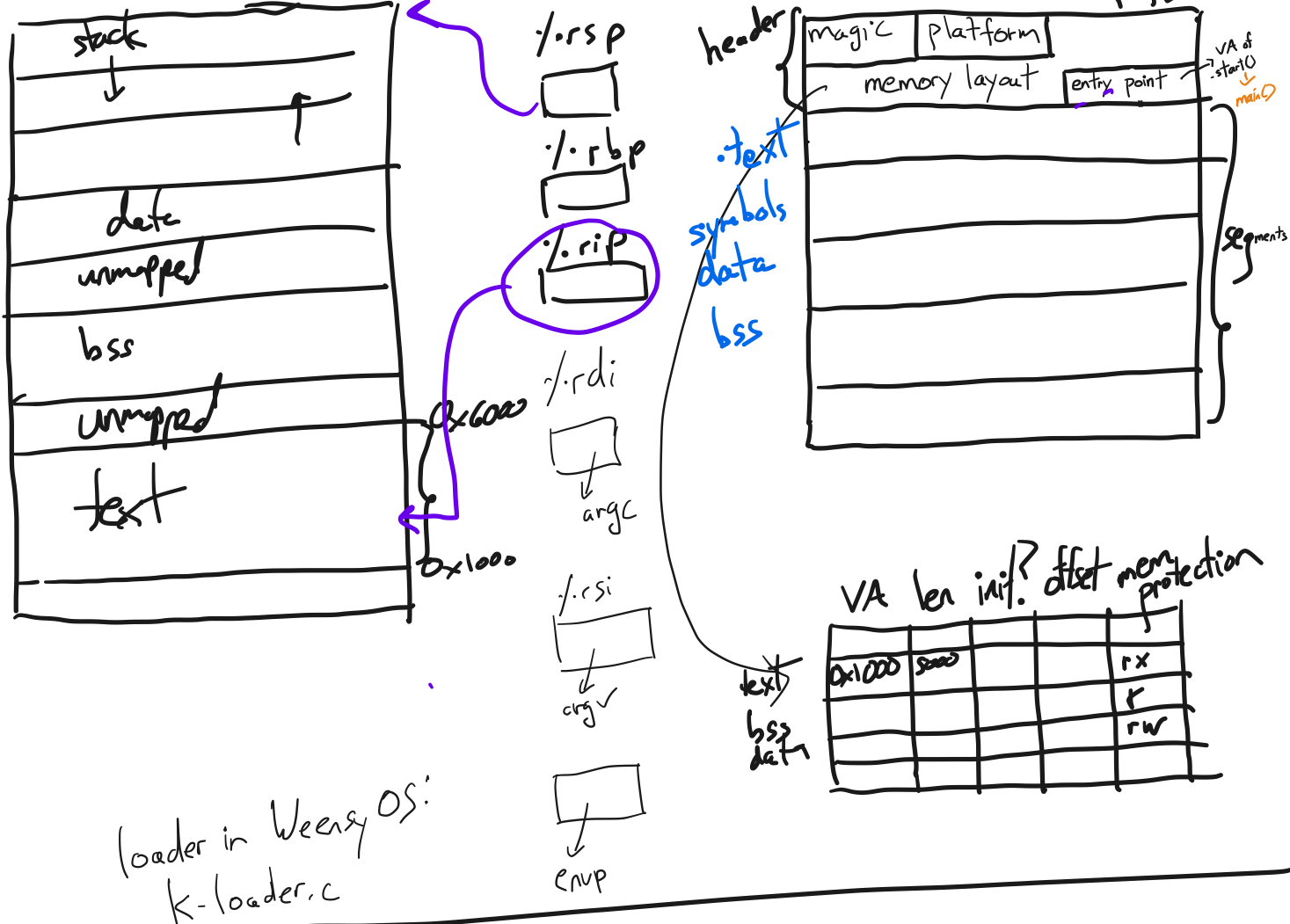


```

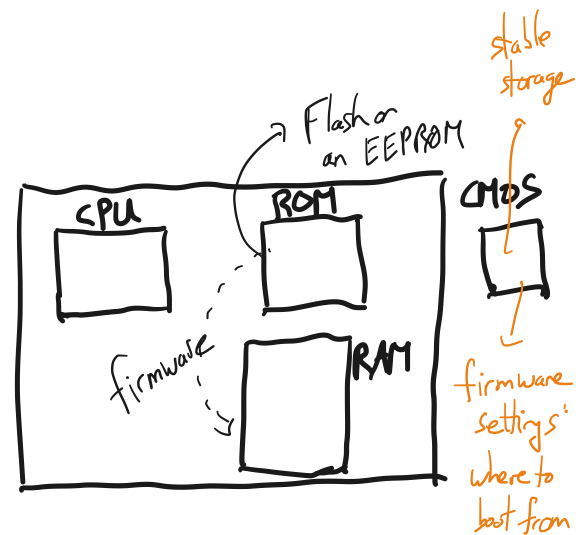
$ gcc hello.c -o hello
$ ls
hello
$ ./hello

```

executable file (ELF PE)



Step 1: power up
regs zeroed out
ctrl. 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 $\rightarrow$ long mode (64-bits, paging)
create an identity-mapped pg table
create an initial IDT
initialize other processor structures
set ctl registers

Peripherals/devices: Initialize (UEFI has drivers)

$\hookrightarrow$ Disk, USB, Display, Keyboard, Mouse, NIC

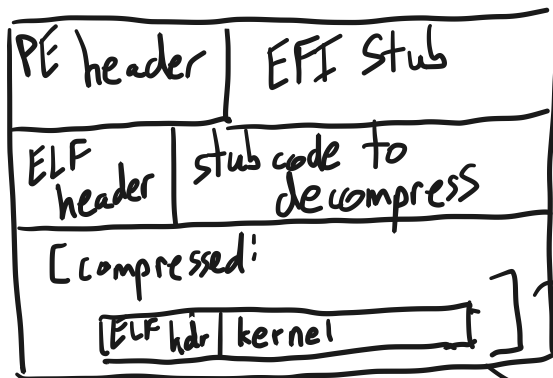
Device Tree initialization ($\equiv$ CONFIGURATION-TABLE)

Mount VFAT partition $\rightarrow$ /EFI

Load + execute boot loader

Step 3: boot loader

vmlinuxz



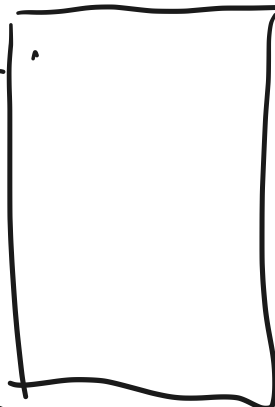
WesleyOS
boot.c

$\hookrightarrow$ kernel

kernel { CONFIG-
TABLE,

...
};

mem



Step 4: kernel

Initialize the system:

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

[fsck, ^{wait}run here, if FS using fsck]

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

Step 5: init

\$ man 8 init

Init: PID 1

Most distributions: systemd

Executes as superuser aka root

(a) Finish initializing the system

(b) Launch the "login manager"

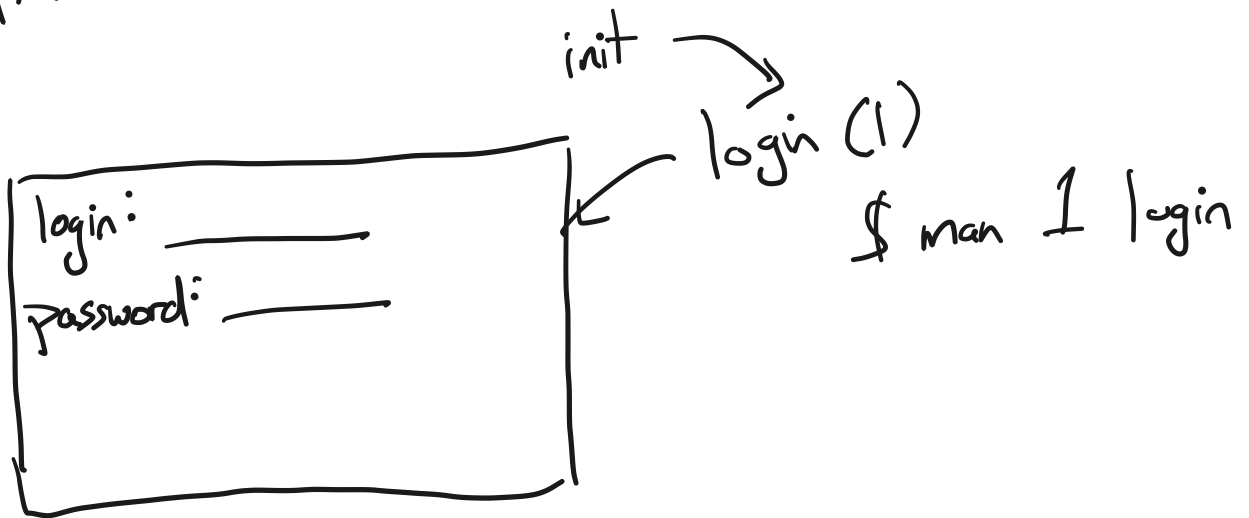
(c) Relaunch

when users log out

(a): network: Dtcp
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)

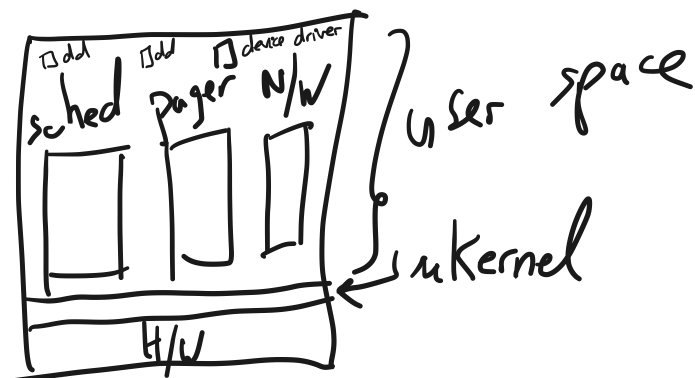
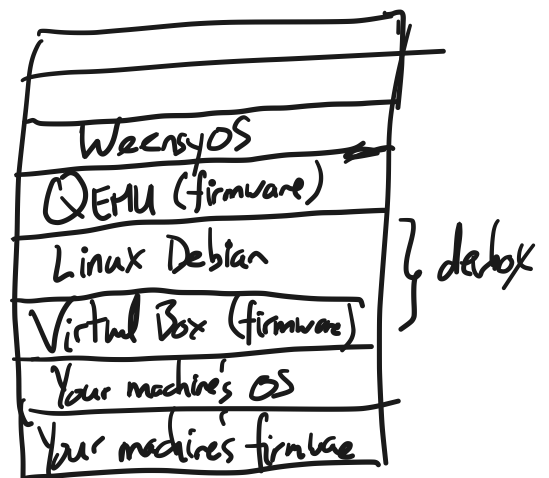
\$ man 2 setuid

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

Ex. Remarks + observations

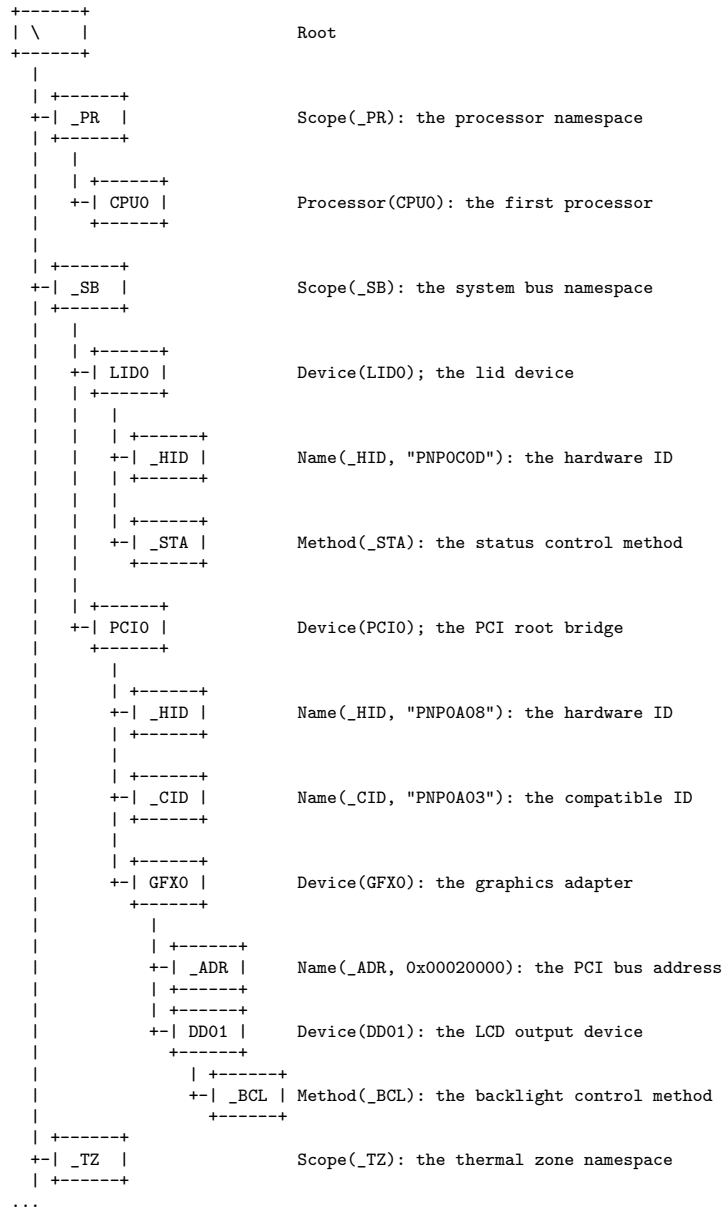
- multiple OS-like things

- monolithic vs. microkernels



CS202

Example Device Tree



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