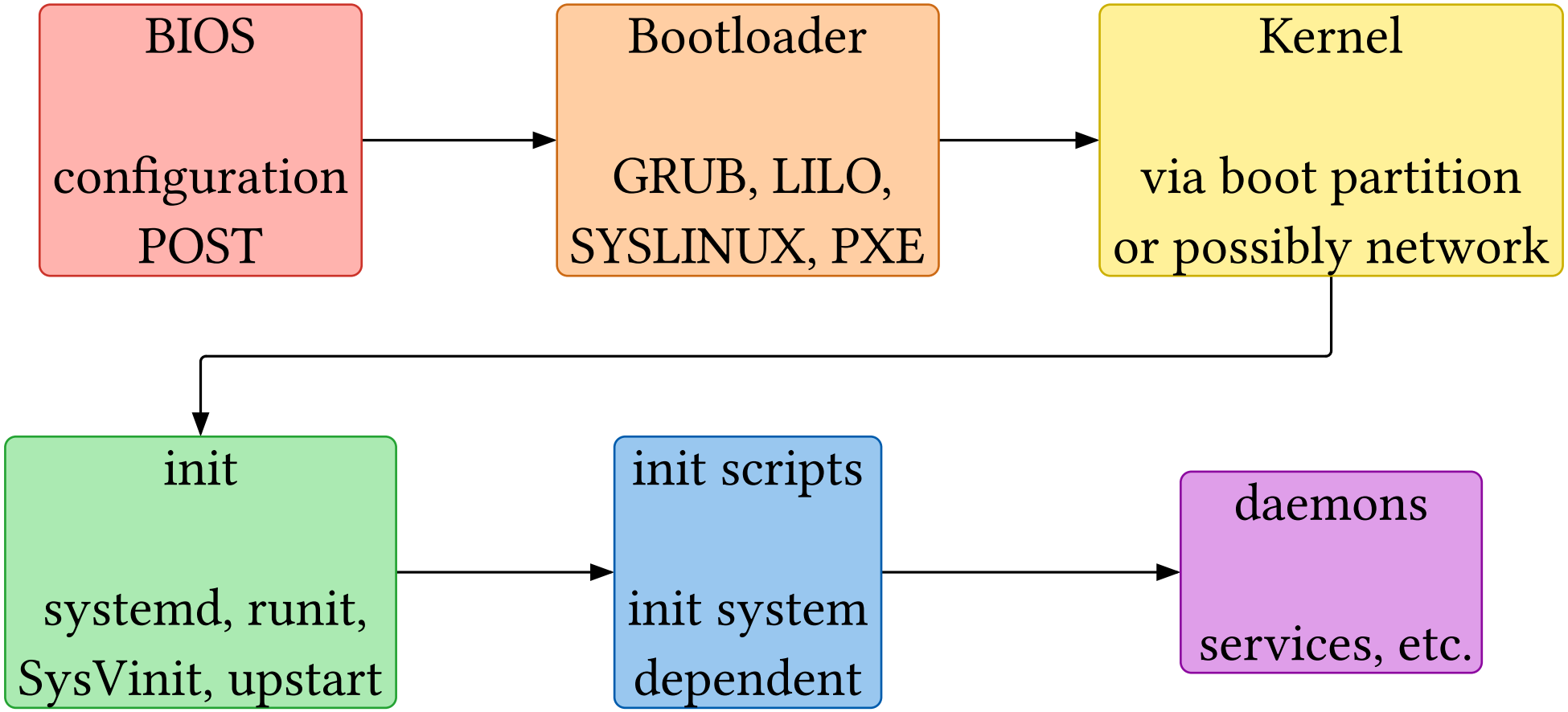
What's missing?

Ryan Tolboom

New Jersey Institute of Technology

We've been learning in a virtualized environment. What haven't we had a chance to work with?

The Boot Process



Init System

- PID 1
- SysV was the old way of doing it
- Most modern systems run sytemd (it's conentious)
- The init system brings up and monitors daemon processes

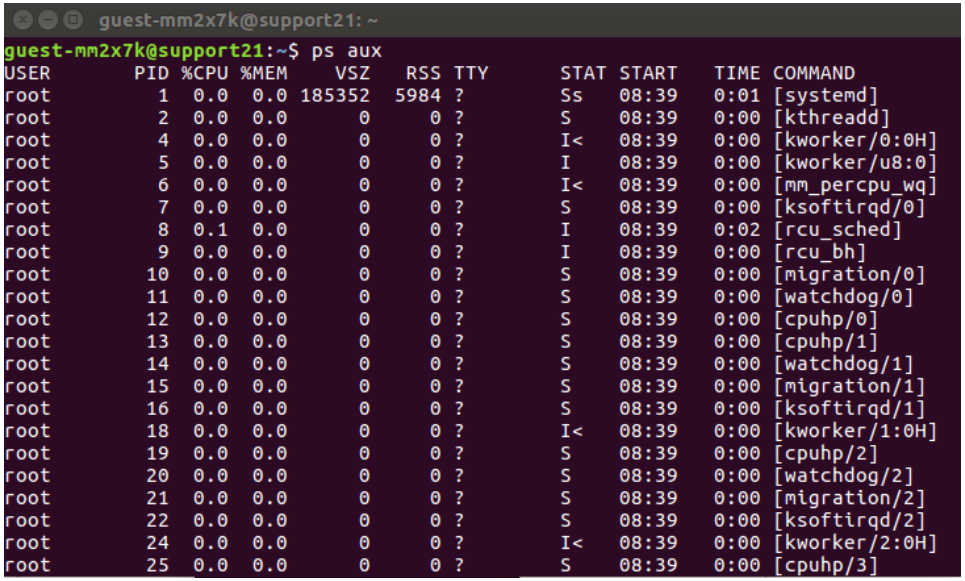
Basic systemctl Commands

[● ◀] systemd

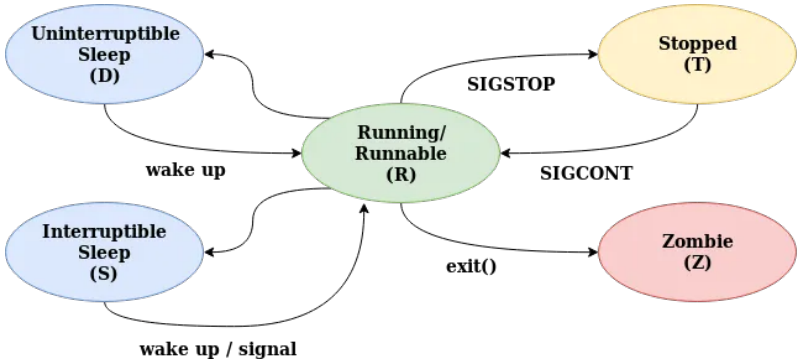
- `systemctl list-units --type=service`
- `systemctl start <servicename>`
- `systemctl stop <servicename>`
- `systemctl restart <servicename>`
- `systemctl enable <servicename>`

Processes

- OS kernel allows for multiple things to run at once
- A process is one of those things
- The kernel scheduler splits time between them
- This can be adjusted!



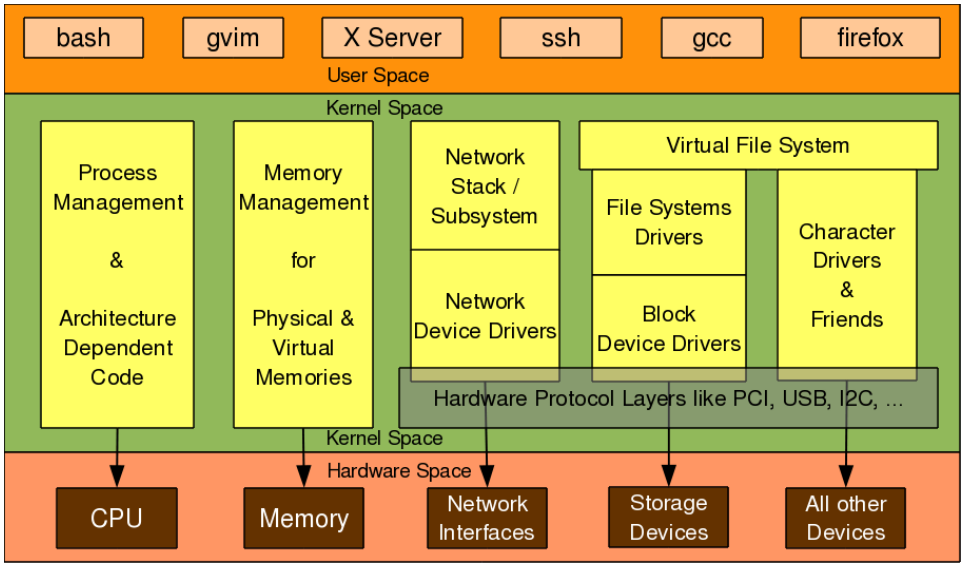
Tuning the Process Scheduler



- `/proc/sched_debug` shows all tunable variables
- `sysctl` (not systemctl!) can be used to adjust them
- `chrt` shows the real-time attributes of a running process
- If you make changes, don't forget to make them permanent! (/etc/sysctl.conf)

Devices

- Real systems have real devices
- I/O is a pretty standard bottleneck in production systems
- Sysfs allows for tuning of I/O devices
- I/O also has schedulers



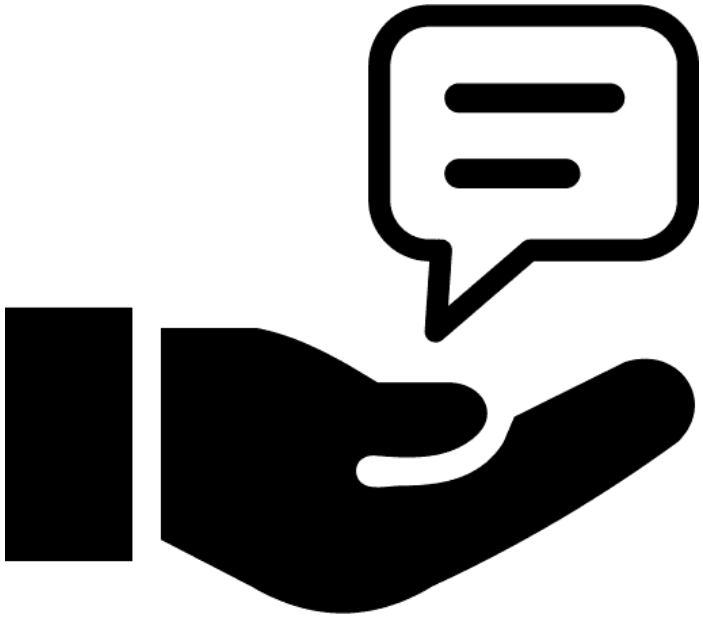
udev

- A list of rules that determines what to do/create when a device is added
- Devices can have persistent names through devfs (can be very useful for USB)
- Initialization can take place automatically

/lib/udev/rules.d/80-usb.rules

```
KERNEL=="sd*", SUBSYSTEMS=="scsi", ATTRS{model}=="USB 2.0 Storage Device", SYMLINK+="usbhd%n"
```

General Advice for Tuning Linux



- Determine your metric in advance!
- Take slow steps and monitor changes
- Be prepared to walk-back changes