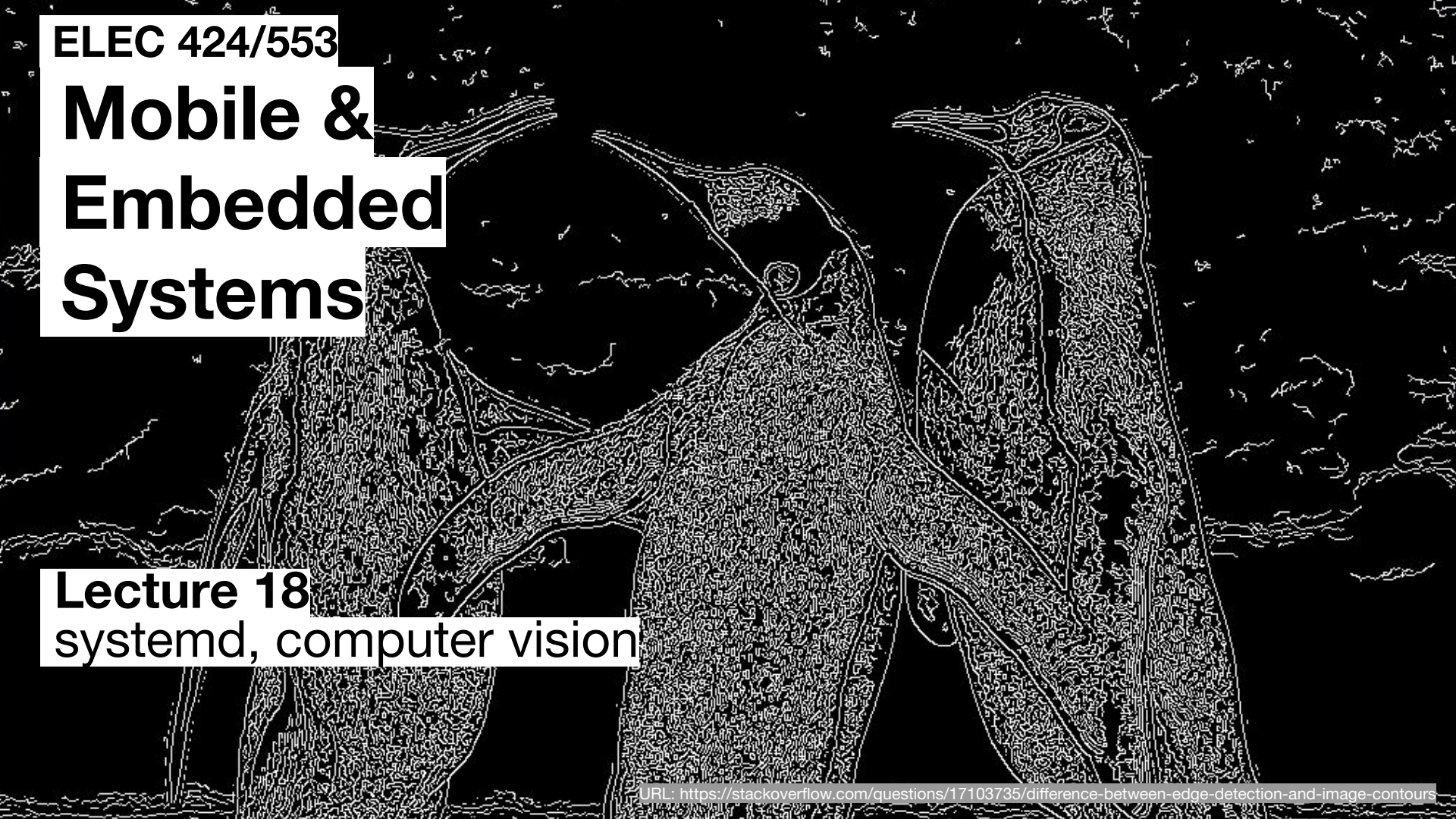
ELEC 424/553

Mobile & Embedded Systems



Lecture 18

systemd, computer vision

URL: https://stackoverflow.com/questions/17103735/difference-between-edge-detection-and-image-contours

Install php while you are waiting:

- **`sudo apt update`**
  - (NOT UPGRADE!!)
- **`sudo apt install php`**
- **`sudo apt install python3-opencv`**

If on Mac, also install XQuartz

If on Windows, install MobaXTerm

The Goal

- Learn how to use the device tree
  - Project 2
- Learn about general operation of kernel (and more)
  - Midterm
- (optionally) Learn how to use Rust
  - (optional) Assignment 3
- Put everything together by building an advanced mobile & embedded system
  - Final Project (with project 3 as a milestone towards the final project)
- This is a lot
- But with diligence, you can be done before the end of the last day of classes

Housekeeping - Project 2

- Due this Friday Nov 3 at 11:59pm
  - ONLY in person demos allowed
  - Demo and help times:
    - Before and after class
    - Joe’s office hours Tuesday 3-4pm in Duncan Hall 2098
    - Lucy’s office hours Wednesday 5:30-6:30pm in Ryon B10
    - Friday 3-5pm in Ryon B10
- Any questions?

Project 2 Demo

Housekeeping - Midterm

- **Midterm released Monday Nov 6**
  - Topics to be sent out today or tomorrow
  - 30 questions, 1 hour
  - All multiple choice/true or false/etc
  - Entirely in Canvas, online
  - Due Sunday Nov 13 at 11:59pm
  - Open slides, notes taken during class, assignments, projects
  - NOT open internet

- NO lecture Wed Nov 8, I will be on zoom to help with any midterm question clarification during class time [zoom link to be announced on Canvas]

Picture from *Spy x Family* (2022-present), from studios Wit Studio & CloverWorks, distributed by Crunchyroll Home Entertainment. Found on *Anime Reviews* from DoubleSama at URL: https://doublesama.com/spy-x-family-part-2-episode-18/

Housekeeping - (optional) Assignment 3

- Assignment 3 optional
  - Can replace lowest assignment or project score - your choice
- Rust programming
- Keep an eye out for announcements on posting and due date

Housekeeping - Final Project

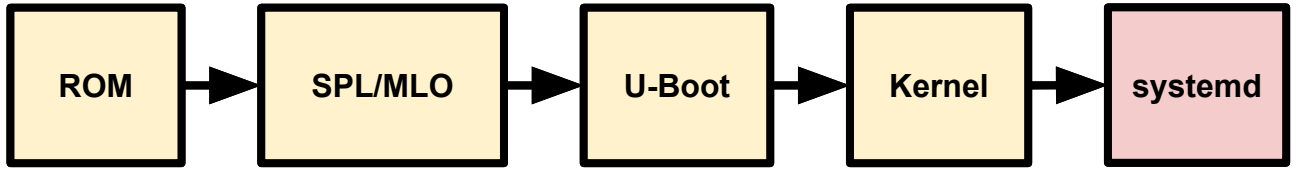
Housekeeping - Final Project

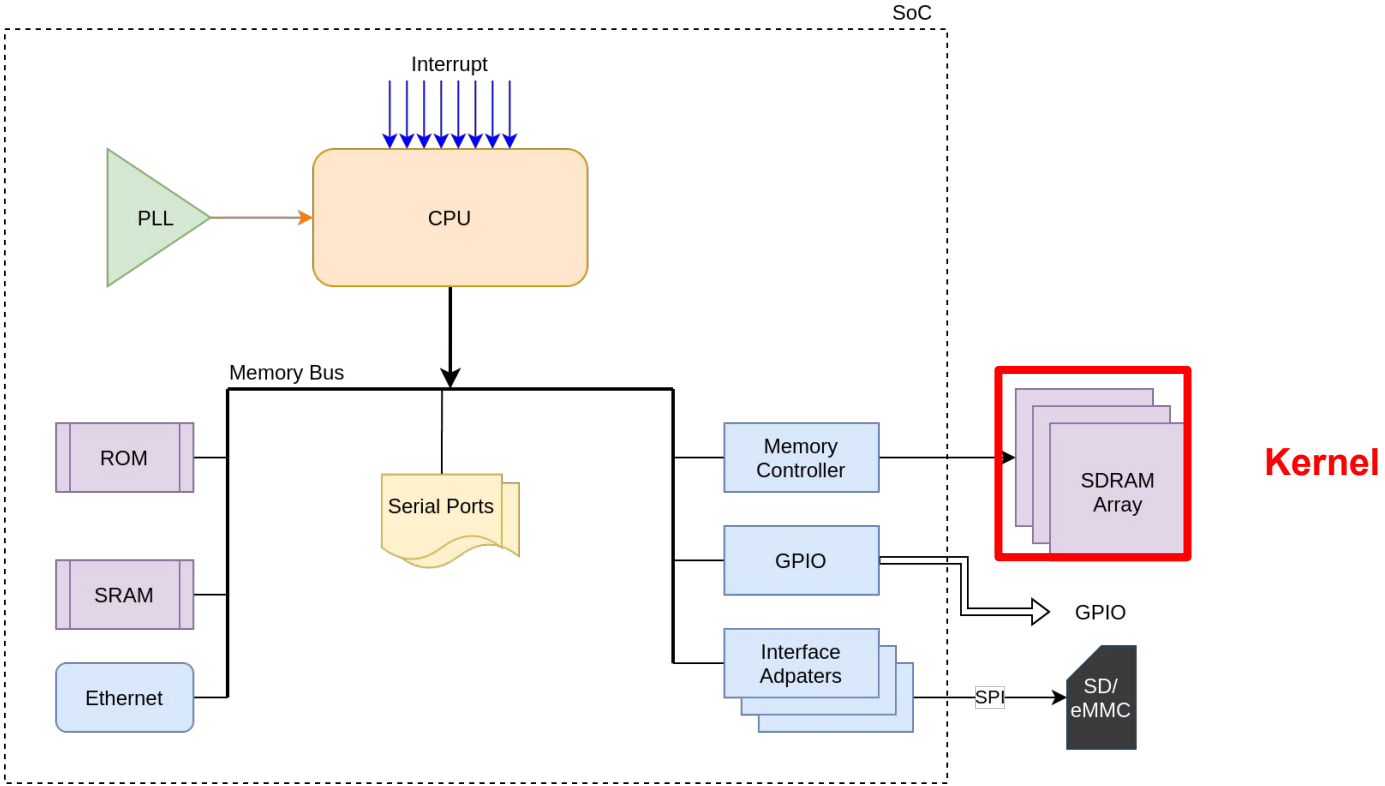
- Due last day of classes - Friday Dec 1 at 11:59pm
- Goal: Webcam-based lane-keeping RC car
- Project 3 as a milestone - Due date to be announced, likely Tuesday Nov 14
- Teams of 4 [both for final project and project 3]
  - Teams must be exclusively undergraduate students or graduate students, not both
  - Sign up here by end of this Wed Nov 1
- Graduate students: ML

5 Minutes For Team Discussion

Link to final project team sign up has been posted in Canvas announcement

SoC & Boot-Up

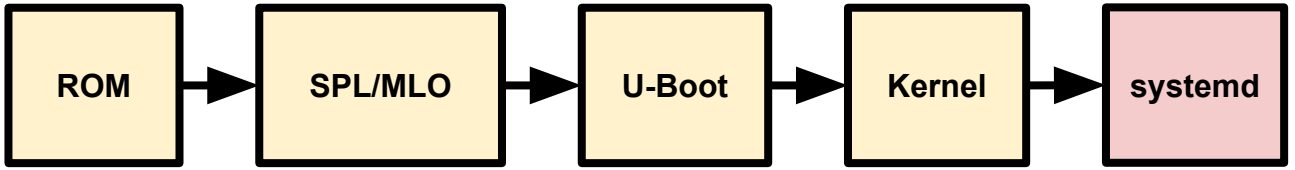




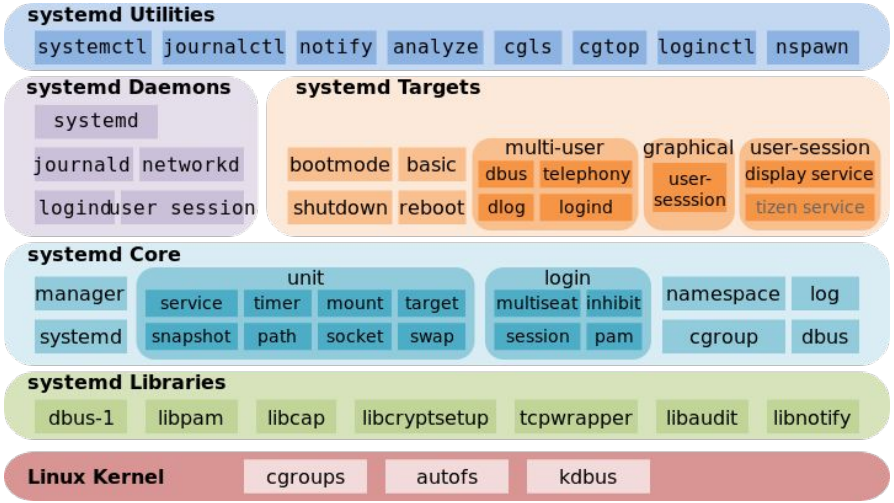
Kernel

Image from: “Device Tree and Boot Flow”. From EdWiki. URL: http://shukra.cedt.iisc.ernet.in/edwiki/Device_Tree_and_Boot_Flow

[● ◀] systemd



- System & service manager
- d stands for daemon (in Unix fashion)
- Init process (daemon) PID 1
    - Universal common ancestor of all processes
- Parallelism
- *Controversial*



Carla Schroder, “Understanding and Using Systemd”. URL: https://www.linux.com/training-tutorials/understanding-and-using-systemd/

“Working with Linux and U-boot”. Embedded Artists AB. URL: https://www.embeddedartists.com/wp-content/uploads/2019/03/iMX_Working_with_Linux_and_uboot.pdf

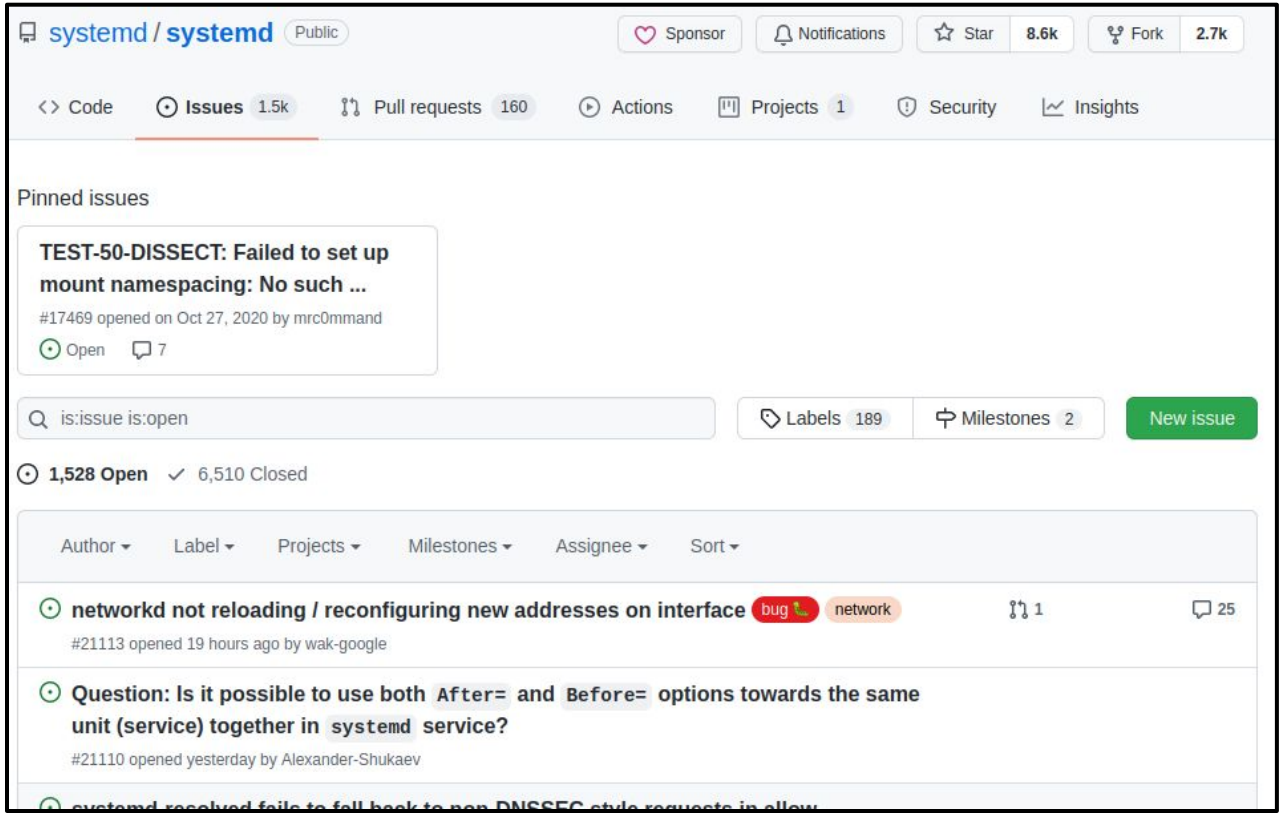
A Word From Our Best Friend

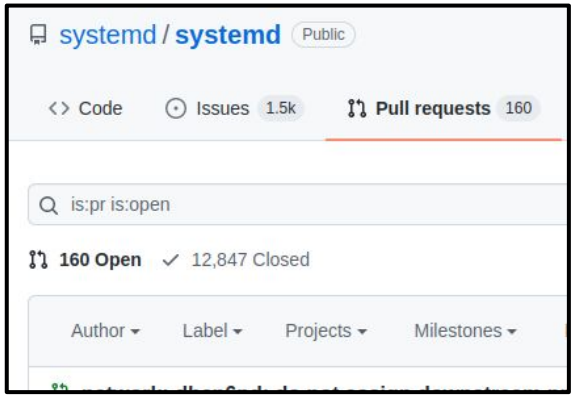
(language warning)

https://en.wikipedia.org/wiki/XML#Criticism

https://www.youtube.com/watch?v=7SofmXIYvGM&t=1118s

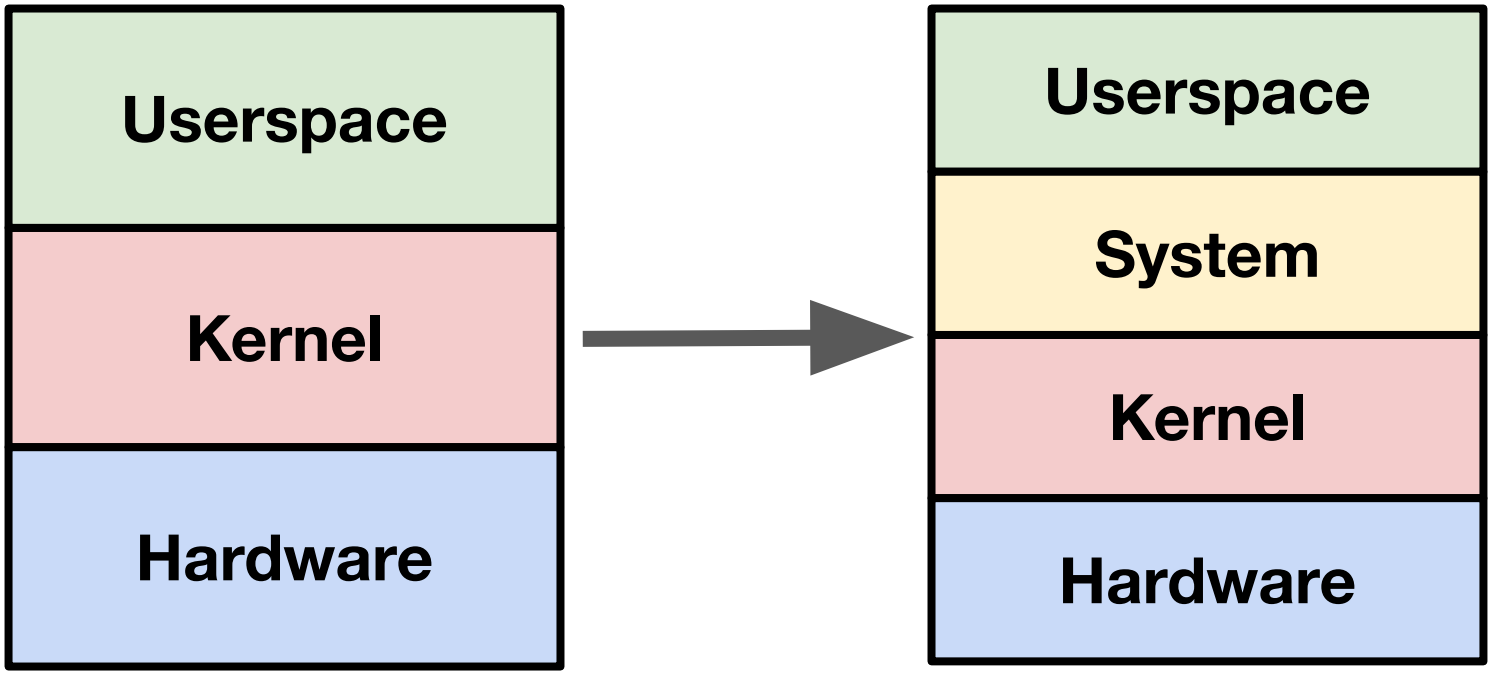
“bug reports have been basically ignored in some cases”





https://github.com/systemd/systemd

A Change In The Winds



Inspired by figure at time 15:13 from: “The Tragedy of systemd”. Benno Rice. URL: https://www.youtube.com/watch?v=o_AIw9bGogo&t=2089s

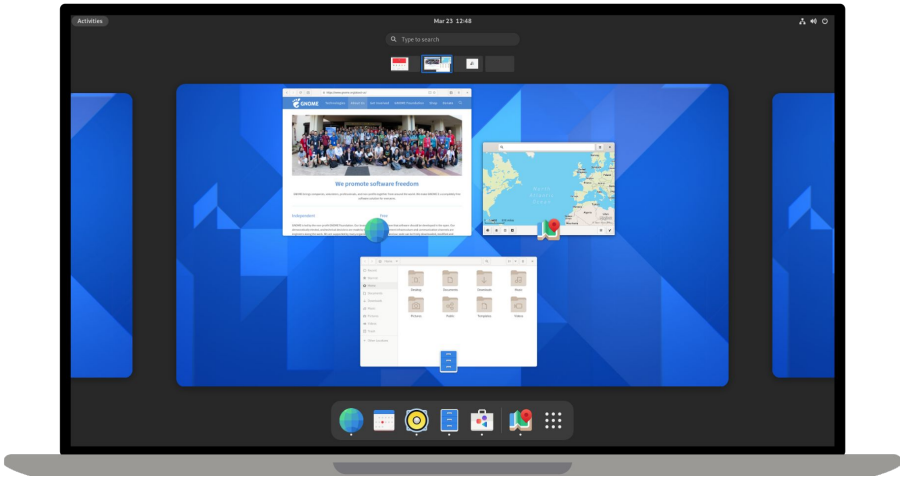
Example of Userspace Depending On Supposed System Layer

- GNOME
- From Nov. 2012 meeting notes:

  ‘- No hard compile time dep on systemd for "basic functionality"

  - example of basic functionality: active session tracking

  - example of non-basic functionality: power management’



https://www.gnome.org/

“20121104 meeting minutes”. URL: https://mail.gnome.org/archives/release-team/2012-November/msg00015.html

Adoption of systemd By Distributions

Fedora 15	May, 2011
openSUSE 12.2	September, 2012
CentOS 7.14.04	April, 2014
Red Hat Enterprise Linux 7.0	June, 2014
SUSE Linux Enterprise Server 12	October, 2014
Debian 8	April, 2015
Ubuntu 15.05	April, 2015

Screenshot from: “The Tragedy of systemd”. Benno Rice. URL: https://www.youtube.com/watch?v=o_AIw9bGogo&t=2089s

Motivation

- Automated service management
  - Start
  - Stop
  - Restarting
  - Autorestarting
- Windows has long history of service management
- MacOS has launchd

“The Tragedy of systemd”. Benno Rice. URL: https://www.youtube.com/watch?v=o_AIw9bGogo&t=2089s

Joe Show launchd

Terminal
top
man top
q
top -o +pid

What Can systemd Do For Us?

- We can create a service
- Webcam streaming
- Web server
- Can start automatically on system boot

What Can systemd Do For Us?

- We can create a service
- **Webcam streaming**
  - We won’t do this, but we will get a preview of computer vision
  - This is a side story but also an appetizer
- Web server
- Can start automatically on system boot

Select image and copy to Pi

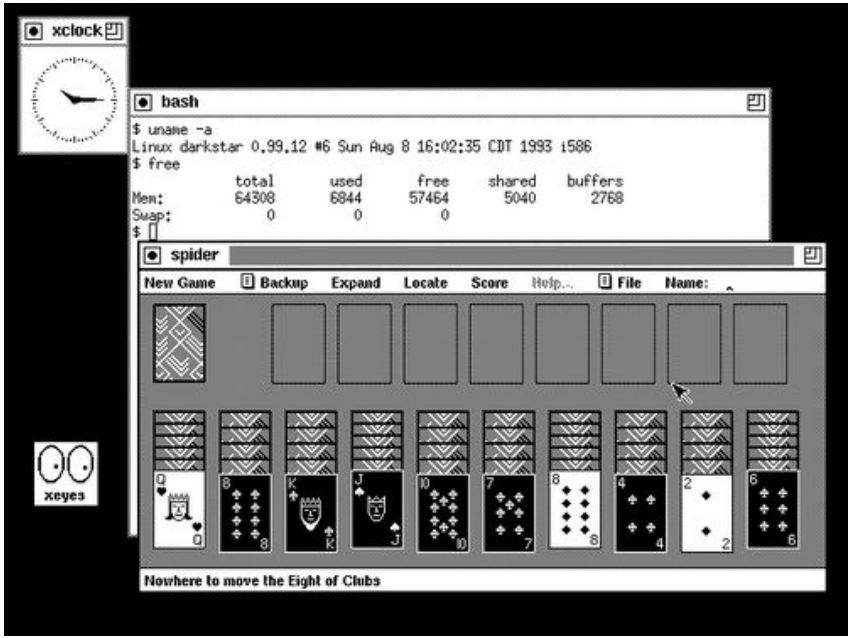
```
cd Downloads

scp img.jpg pi@raspberrypi.local:~/
```

Remember?

1992: A Big Year for Linux

- X Window System ported to Linux
  - First GUI on Linux



https://doublslash.com/blog/2013/05/11/linux-user-base-is-below-1-and-it-is-changing/

https://frontpagelinux.com/articles/guide-through-history-of-unix-linux-everything-you-need-to-know/

Login To Pi With X Forwarding

```
cd Downloads

scp img.jpg pi@raspberrypi.local:~/
```

openCV

```
# Modified from: https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/
# import opencv
import cv2

# Load the input image
image = cv2.imread('img.jpg')
cv2.imshow('Original', image)
cv2.waitKey(0)

# Use the cvtColor() function to grayscale the image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Grayscale', gray_image)
cv2.waitKey(0)

# Window shown waits for any key pressing event
cv2.destroyAllWindows()
```

Molloy, D. [DerekMolloyDCU]. (2013, May, 25). Beaglebone: Video Capture and Image Processing on Embedded Linux using OpenCV [Video file]. Retrieved from http://www.youtube.com/watch?v=8QouvYMfmQo . GitHub: https://github.com/derekmolloy/boneCV

openCV - Edge detection

```
# Edge detection
edges = cv2.Canny(image=gray_image, threshold1=100, threshold2=200)

cv2.imshow('Edges', edges)
cv2.waitKey(0)
```

Molloy, D. [DerekMolloyDCU]. (2013, May, 25). Beaglebone: Video Capture and Image Processing on Embedded Linux using OpenCV [Video file]. Retrieved from http://www.youtube.com/watch?v=8QouvYMfmQo . GitHub: https://github.com/derekmolloy/boneCV

v4l2

- Video 4 Linux 2
- Video capture API for Linux
- Works with a variety of webcams
- v4l2-ctl -h
- v4l2-ctl --list-devices
- v4l2-ctl --all
- V4l2-ctl --set-ctrl=brightness=200

Molloy, D. [DerekMolloyDCU]. (2013, May, 25). Beaglebone: Video Capture and Image Processing on Embedded Linux using OpenCV [Video file]. Retrieved from http://www.youtube.com/watch?v=8QouvYMfmQo . GitHub: https://github.com/derekmolloy/boneCV

v4l2 (2)

- v4l2-ctl --list-formats
- v4l2-ctl --set-fmt-video=width=1920,height=1080,pixelformat=1
- v4l2-ctl --all

Molloy, D. [DerekMolloyDCU]. (2013, May, 25). Beaglebone: Video Capture and Image Processing on Embedded Linux using OpenCV [Video file]. Retrieved from http://www.youtube.com/watch?v=8QouvYMfmQo . GitHub: https://github.com/derekmolloy/boneCV

What Can systemd Do For Us?

- We can create a service
- **Webcam streaming**
- Web server
- Can start automatically on system boot

What Can systemd Do For Us?

- We can create a service
- Webcam streaming
- **Web server**
- Can start automatically on system boot

Exercise 14 Code: web3

Creating a Service - My Modified Example From Benjamin Morel On Medium

- Let's make a simple web server
- **`nano server.php`**

```php
<?php

$sock = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);
socket_bind($sock, '0.0.0.0', 10000);

for (;;) {
    socket_recvfrom($sock, $message, 1024, 0, $ip, $port);
    $reply = str_rot13($message);
    socket_sendto($sock, $reply, strlen($reply), 0, $ip, $port);
    $reply2 = str_rot13($reply);
    socket_sendto($sock, $reply2, strlen($reply2), 0, $ip, $port);
}
```


Benjamin Morel, "Creating a Linux service with systemd". Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Try Out The Server

- Run the server:
  - **`php server.php`**
- *You may need to install php:*
  - ***`sudo apt update`***
    - ***`NOT UPGRADE!!`***
  - ***`sudo apt install php`***
  - ***`(again) php server.php`***
- From a different terminal:
  - **`nc -u 127.0.0.1 10000`**

Benjamin Morel, “Creating a Linux service with systemd”. Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Let's Make This a Service (Runs In Background) Managed By systemd

- Make a service file in the location for system services:
  - **`sudo nano /etc/systemd/system/rot13.service`**
- Put configuration details in file:

```
[Unit]
Description=ROT13 demo service
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=pi
ExecStart=/usr/bin/env php /path/to/server.php

[Install]
WantedBy=multi-user.target
```


Benjamin Morel, "Creating a Linux service with systemd". Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

What Do All These Configuration Details Mean?

```
[Unit] (link to explanations for this section)
Description=ROT13 demo service
After=network.target
StartLimitIntervalSec=0

[Service] (link to explanations for this section)
Type=simple
Restart=always
RestartSec=1
User=debian
ExecStart=/usr/bin/env php /path/to/server.php

[Install] (link to explanations for this section)
WantedBy=multi-user.target (special note)
```

The Service Exists - Let’s Get It Started

- `systemctl -h`
- `man systemctl`
- Stop current server from running if it is: **control+c**
- `systemctl start rot13`
- `nc -u 127.0.0.1 10000`

Benjamin Morel, “Creating a Linux service with systemd”. Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Taking Your Service To The Next Level: On Boot

- Making service autostart when Pi powered on:
  - **`systemctl enable rot13`**
- Undoing autostart:
  - **`systemctl disable rot13`**

Benjamin Morel, “Creating a Linux service with systemd”. Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Taking Your Service To The Next Level: On Boot

- Waiting for something else to happen
  - E.g., MySQL server needs to be alive before service
  - **`After=mysqld.service`**
- The services dies! (*exits*)
  - No autorestart by default
  - **`Restart=always`**
  - **`on-failure`** in place of **`always`** causes autorestart for exit status~=0

Benjamin Morel, “Creating a Linux service with systemd”. Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Taking Your Service To The Next Level: Patience

- Autorestart has default delay of 100ms
- Can change to (for example) 2 seconds:
  - **`RestartSec=2`**
- But wait, there's more!
- If >5 autorestart failures occur in <10s, systemd throws in the towel
- Who did this?
  - **`StartLimitBurst=5`**
  - **`StartLimitIntervalSec=10`**

Benjamin Morel, "Creating a Linux service with systemd". Medium. URL: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6