# Announcements

# Extension policy: clarified

**Last day to turn in labs for this half of the semester: 1159pm TOMORROW**

**We will have to submit midsemester grades.**

Get your extensions in by today.

# General things we can do better

- More interactive lectures!
  - All these commands that we are running…you can follow along
- Better office hours (feel free to reach out to us personally if you have any questions)
  - We will be wearing GPI hoodies for office hours
  - GPI Commons
- Posting the lecture slides before the lecture!

# Fun things

- Bee movie scripts
- Rickrolld twice
- Wikipedia page for bread
- Gettysburg Address
- Some Jacobo Memes that I won’t show on the big screen

# Jobs, Signals, File Descriptors, and More

Shruti Chidambaram
Aditya Shekar

# This is important

# How does Unix work?

- Basics of the OS
  - Kernel
    - Memory Management
    - Multitasking
    - I/O
  - Programs
- Unix is a much leaner version of an OS called Multics
- Importance
  - An Operating System abstracts away the hardware details of your computer.
  - It serves as an intermediary between you and your computer.

# Jobs

Run a command in a shell -> any programs that it starts are added to a “job”

# Job Control

What if you want to run more than one program at the same time?

What if you want to temporarily stop running a program and resume it later?

*Foreground job:* The job you are currently running and interacting with

*Background job:* Any other job

# Job Control: Commands

| Command | Description |
|---|---|
| `jobs` | List all jobs |
| `[command] &` | Start a job in the background |
| `fg [number]` | Bring a backgrounded (stopped or running) program into the foreground (takes control of the terminal) |
| `bg [number]` | Start a stopped process, but put it in the background |

Can omit the number if job has + next to it in listing (this represents the current job)

# Process ID (PID)

- Each process is assigned a PID.
- How the operating system identifies and keeps track of processes
- Get the PIDs with **ps  u**

# Signals

- Notification to a process that an event has occurred
- **`CTRL-C`** sends an “interrupt” signal ➔ interrupts process, process will terminate (usually)
- **`CTRL-Z`** sends an “terminal stop” signal ➔ process suspended and put in background

# kill

- Command used to send signals to processes
- Most often used to terminate a process

# File Descriptors

- Non - negative numbers that are assigned to *opened* files by the kernel to help keep track of things.
- As new files are opened, the kernel returns a file descriptor to the process.
- As files are closed, these descriptors are freed.
- STDIN, STDOUT, and STDERR have file descriptors.