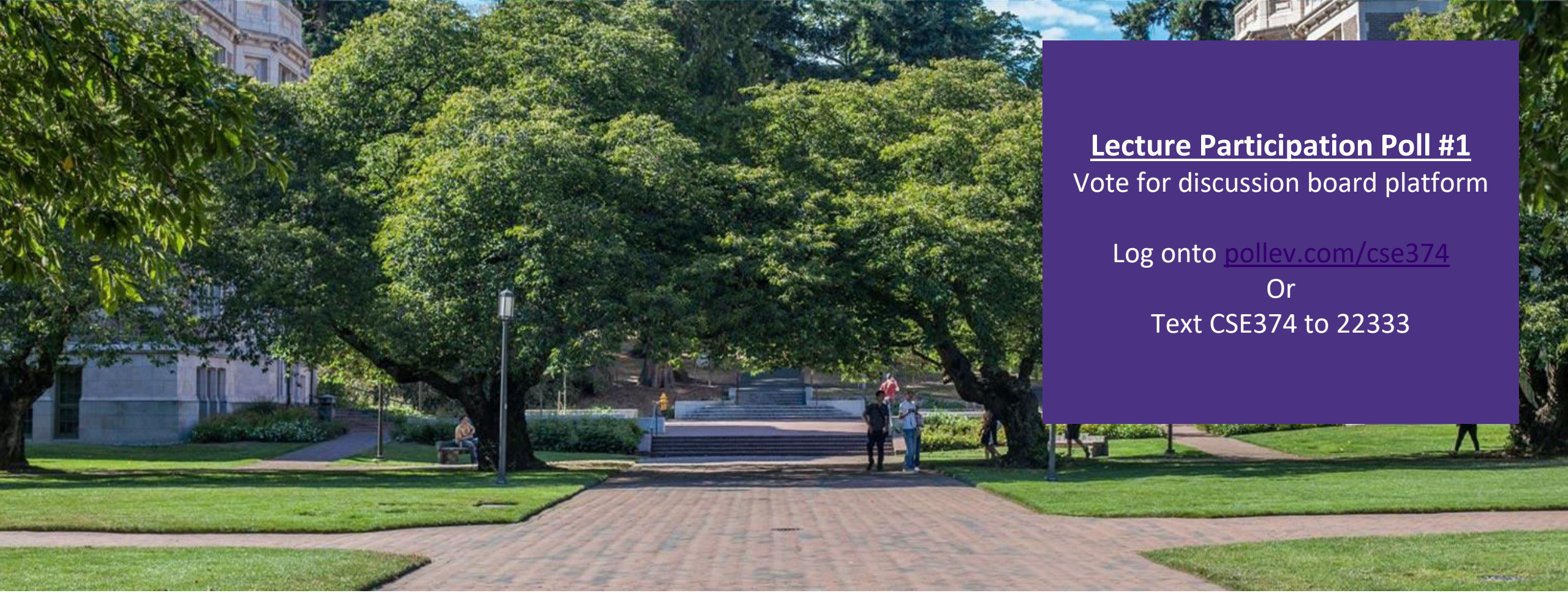


**Lecture Participation Poll #1**

Vote for discussion board platform

Log onto pollev.com/cse374

Or

Text CSE374 to 22333

# Lecture 1: Intro to Linux

CSE 374: Intermediate Programming Concepts and Tools

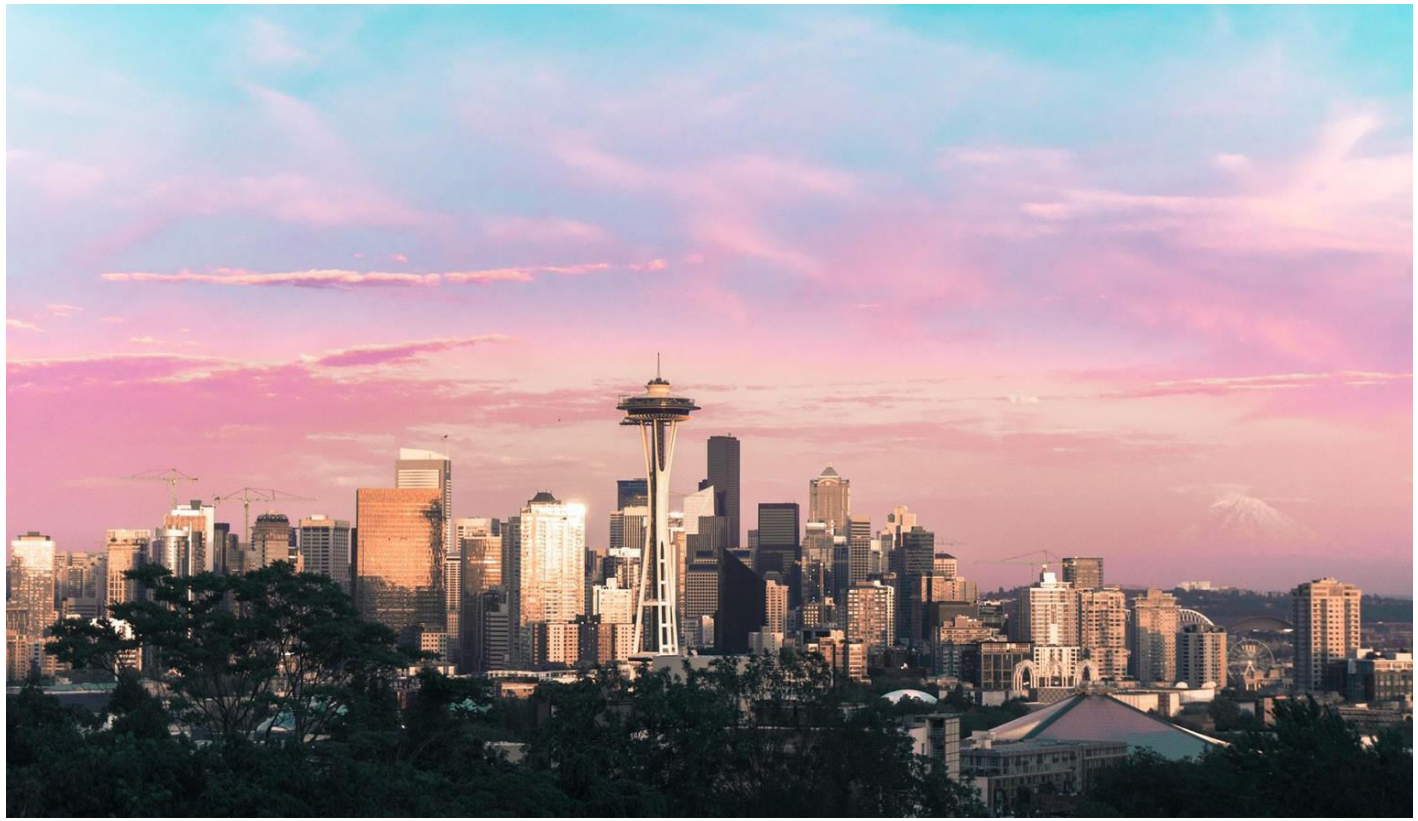
# Welcome!

- Yay, another transition to go through together!

Class Values

- Humans first, students second

- patience, vulnerability, kindness



**Land Acknowledgement**

***"We acknowledge that we are on the traditional land of the first people of Seattle, the Duwamish People past and present and honor with gratitude the land itself and the Duwamish Tribe."***

# Hello!

## *I am Kasey Champion*

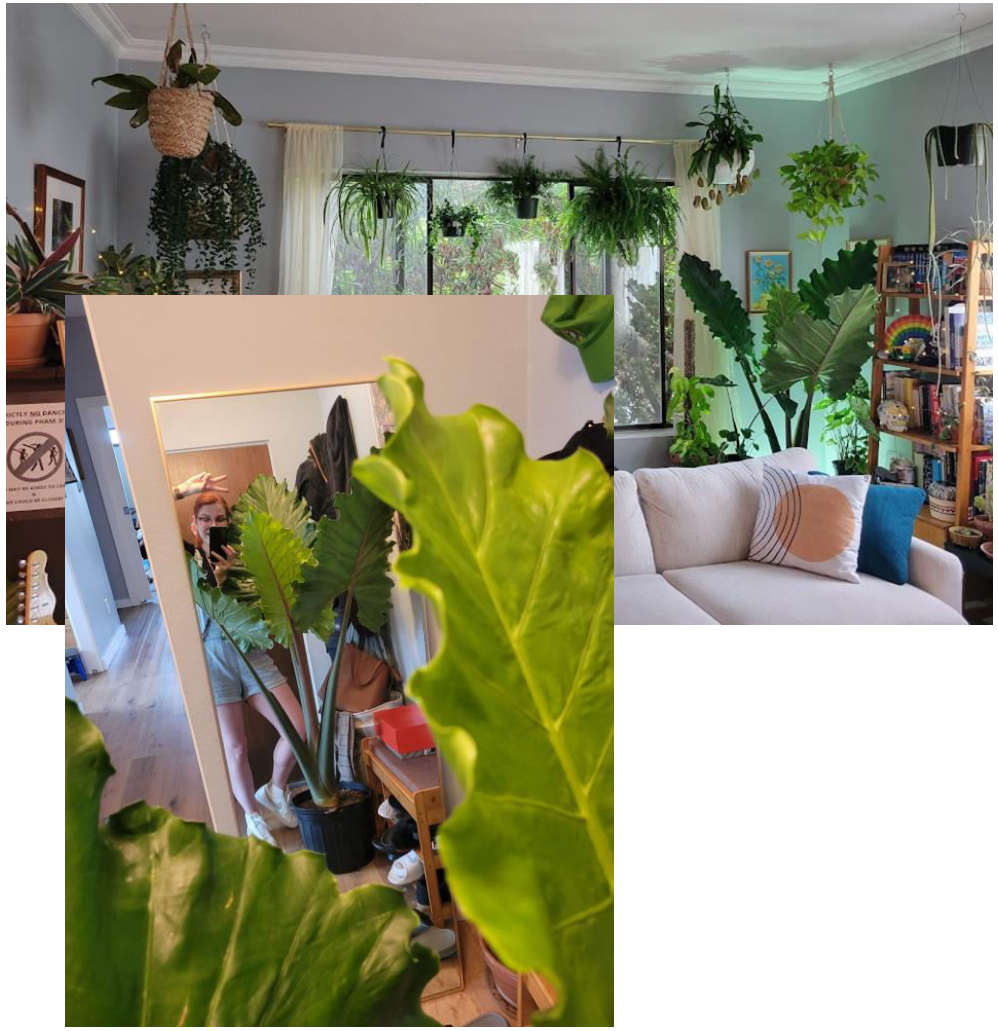
*(she/her)*

Technical Program Manager @ Google

Technical Interview Content Team Lead @ Karat

Software Engineer @ Microsoft

champk@cs.washington.edu



@techie4good

# Meet your TAs

Leah Perlmutter – lrperlmu@cs.washington.edu

Andres Paz – anpaz@cs.washington.edu

Tom Tian – zhaoyt@uw.edu

Dixon Tirtayadi – dixontir@cs.washington.edu

Staff Mailing List – cse374-staff@cs.washington.edu

# Course Overview

## Course Goals

- Gain a basic familiarity with the Linux
- Skills to automate common computing tasks
- Obtain beginning proficiency in C programming
- Familiarity with C++ programming language
- Learn the basics of programming tools
- Practice core software-engineering practices
- Understand the basics of shared-memory concurrency
- Learn how to acquire additional information and skills independently

## Course Topics

- The Linux Operating System
- Working with the Linux Shell
- Scripting with Bash
- C Programming Language
- Computer Memory
- Software Engineering Tools and Techniques
- C++ Programming Language
- Assembly Code
- Concurrency

# Course Components

## Learning Components

- Lectures
  - Recorded
- Lecture Participation Polls
  - Graded on participation NOT correctness
- Homeworks
  - Larger assignments sprinkled throughout quarter, optional groups up to 3 people
  - Individual Homework Assessments

Exams

- Office Hours
  - Please come hang out with us!

## Course Tools

- Class webpage
  - Central location for all information
- Course canvas
  - Gradebook
  - Panopto Lecture Recordings
- Poll Everywhere
  - Lecture participation
- Gradescope
  - Assignment submission
- Ed Discussion board
  - Get help
- Anonymous Feedback Tool
  - Tell us how it's going

# Course Policies

## Turn In Policies

- Homework
  - Open once content is covered
  - Due on Thursdays – 1.5 to 2 weeks later at 11:59PM PDT
  - Will remain open for 72 hours until lock to accept late assignments
  - Everyone receives **3 late days** per quarter, late days are individual
  - assignments will be docked 10% late penalty per 24 hours late after consumption of all late days
  - Group Code Solution
    - Work done in GitLab, submitted via Gradescope
  - Individual Assessments
    - Exam style questions and group work review
    - submitted via Gradescope
- Participation
  - Poll everywhere open at start of lecture
  - Due before start of next class
  - No late polls will be accepted

## Grade Breakdown

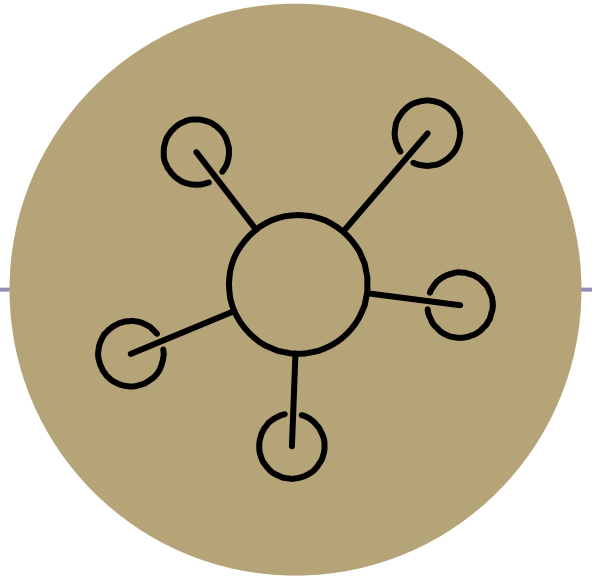
- Group Homework 65%
- Individual Homework Assessments 10%
- Midterm: 10%
- Final Exam: 15%

## Academic Misconduct

- Don't share your code
- Don't look at others code
- Don't "step by step"
- DO talk to one another about concepts and approaches
- DO look things up on the internet

## Accommodations and Extenuating Circumstances

- Make sure you get the support you are entitled to via DRS
  - If you're having issues with DRS system reach out to Kasey
- When in doubt, reach out!

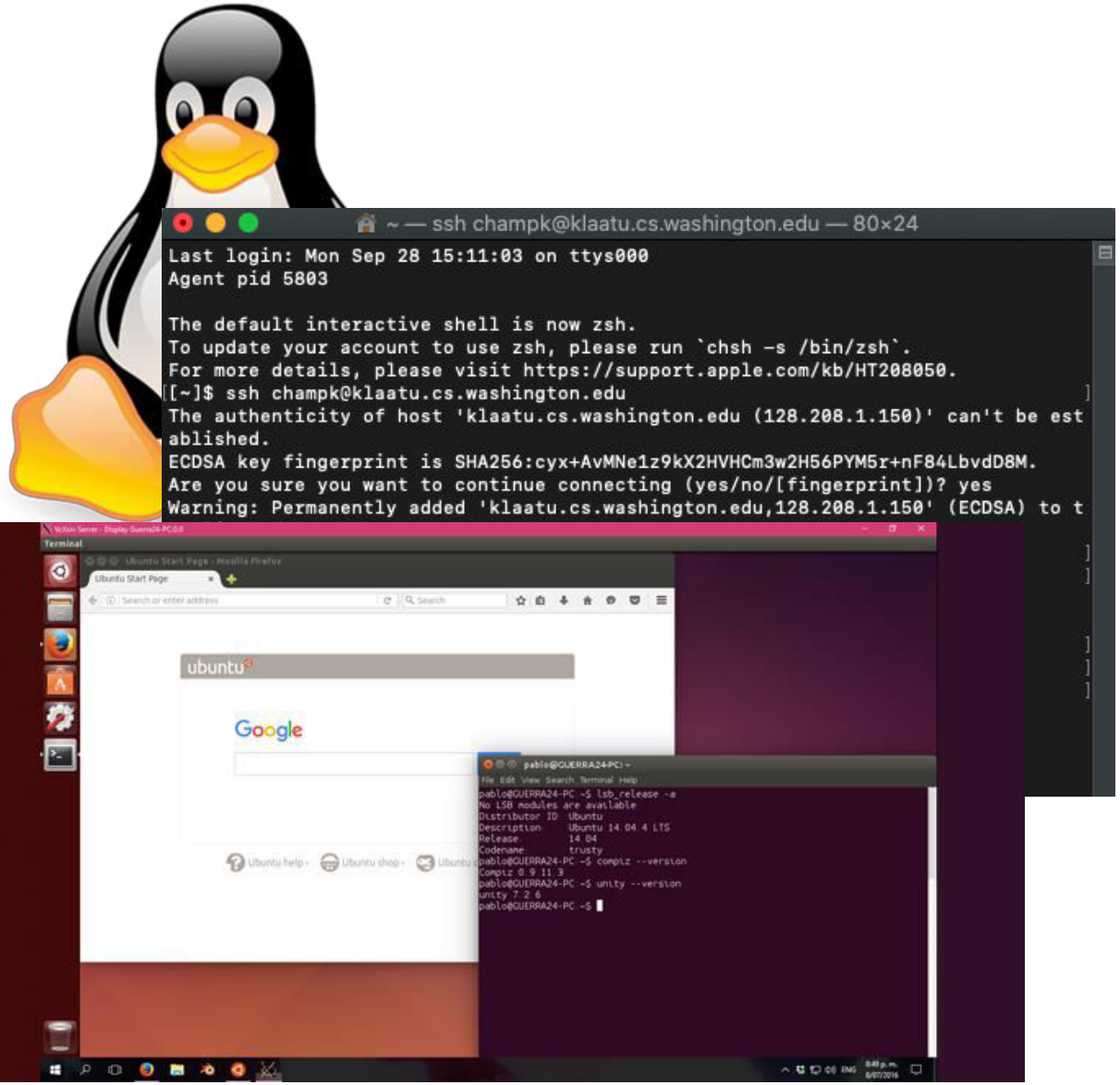


# Questions?

# Meet Linux

## What is Linux?

- Linux is an **operating system** like Windows or MacOS
  - The **operating system** manages the relationship between computer hardware, software resources and user interaction points
- Interact with the Linux machine via the **shell**
  - The **shell** is a text-based interface to the computer, for linux specifically "bash"
  - The shell is run by the **terminal** program
- Linux also has a GUI interface, just like Windows and Mac have shell interfaces
  - Text interface is harder to use, but more efficient
  - Text interface allows for task automation
- Linux was built in C using open source licensing and philosophies



# What Makes Linux "Linux"?

- Linux is a family of **open source Unix**-like operating systems based on **Linux kernel**
  - **Linux kernel** was written by Linus Torvalds using the **GNU Project open source** system
  - **GNU Project** believed in freedom
    - freedom to run a program for any purpose
    - freedom to study the mechanics of the program and modify it
    - freedom to redistribute copies
    - freedom to improve and change modified versions for public use
  - GNU Project allowed Torvalds to make the Linux Kernel free and open for development by other engineers
  - MacOS is also based on unix kernel (Steve Jobs used Carnegie Mellon Mach kernel to take NeXT to market faster)
  - Android and Chrome OS based on Linux Kernel

https://www.pcmag.com/news/

**Linus Benedict Torvalds**

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

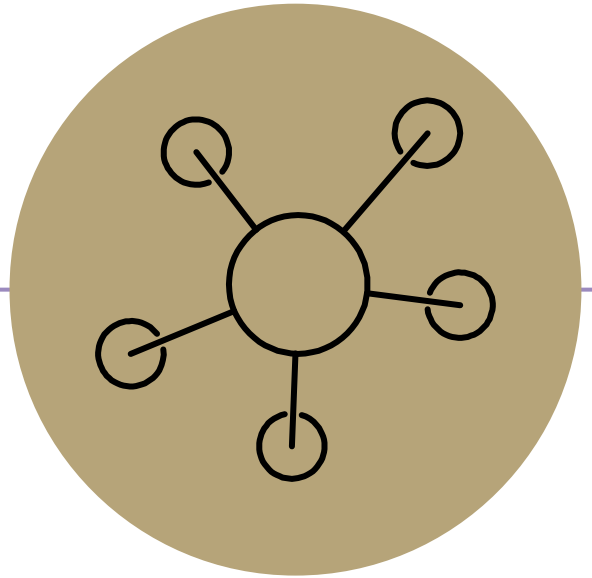
Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

**Trivia**
Linus Torvalds had wanted to call his invention Freax, a combination of "free", "freak", and "x" (as an allusion to Unix). One of the volunteer administrators for the FTP server hated the name and instead named the project "Linux" on the server without consulting Torvalds. Torvalds disliked the name Linux as he found it too egotistical, but the name stuck.
https://en.wikipedia.org/wiki/History_of_Linux#Naming

# Linux Demo

Recorded Demo from 374 Sp 20 Instructor Megan Hazen



# Questions?