Educational information systems and networks

Matthew X. Curinga

Tom Jennings

Christian Correa

David Frackman

Table of Contents

**0858-606, Summer 2022**

**Description:** From a foundation of computer networks and systems, this course expands to cover instructional technology infrastructure: file systems, users, wired and wireless networks, email, web servers, computer labs, and common educational software services. This course focuses on Free Software; where the source code is free to use, study, or modify. To explore these topics in this hands on class all students will be configuring their own Raspberry Pi computers and using them to complete a software/hardware project.

**Keywords:** linux, bash, systems, networks, lamp, free software, trouble shooting, technical project management, rasberry pi, physical computing, debian, ubuntu

# Goals & Objectives

This course introduces students to the key concepts in current networked computing in order to develop a conceptual framework for configuring and troubleshooting computing systems. Upon completing this course they will be able to:

* set up a secure, network computing environment
* effectively use the basic tools of GNU/Linux computing environments
* implement techniques for administering group and user permissions
* install and troubleshoot hardware and software infrastructure for networked and internet computing
* configure a client/user computer for specific purposes
* configure various server-side applications to support teaching and learning
* identify the ethical and legal concerns surrounding educational information systems

# Class Information

* **Instructor:** Matt Curinga [mcuringa@adelphi.edu](mailto:mcuringa@adelphi.edu)
* **Class dates:** May 26 - July 1
* **Live sessions:** Wednesdays: 4:30pm - 6:20pm; zoom link on moodle

# Course Communications

This online course will have a mix of synchronous and asynchronous assignments. The synchronous sessions, “live lab,” will be held each Wednesday, via Zoom, 4:30pm to 6:20pm.

Participants in this course must actively participate in our suite of online communications tools, including Slack (<https://auedtech.slack.com>), Adelphi email, and the course website.

You *must* check your Adelphi email and the #code channel on Slack **at least once a day during this short summer course.** It is highly recommended that you install the Slack mobile client and an email client on your mobile phone so that you receive “push notifications” of course announcements.

The best place to post general course questions and any content-related questions is the #code Slack channel. The instructor, as well as other students and alums, monitor this channel and often provide immediate support. You are encouraged to contact the instructors at any time via email or direct message on Slack.

### Online tools:

* **Moodle:** will be used to post the syllabus and links to weekly readings, videos, and discussions.
* **Zoom:** for class meetings and help (via screen sharing), please join Zoom from your laptop so that you can share your screen to demo work and troubleshoot problems. You have should have a calendar invite to our meetings with the link. The link is also on Moodle in the course header.
* **Slack:** will be our main channel for online communications. Please [Join our Slack team with your Adelphi email](https://auedtech.slack.com/signup). If you run into trouble or have a question, post it here to our channel, #code, or send a message to @mxc. During the weeks of the class, we recommend running the Slack app for you phone.
* **Trello:** is project management software that you will use to track/plan your two projects, and to share your progress with the course instructors. Create a [Trello Account](https://trello.com/signup) before the first week of class, and (optionally) [install the mobile app](https://trello.com/platforms).
* **mail.adelphi.edu email:** we will use your official adelphi student email for class email communications. Please check this email regularly.

*You should already be comfortable with these skills, or should be prepared to learn them during the course. Teaching them is not part of the course content.*

* Taking screen shots [mac](https://support.apple.com/en-us/HT201361) [win](https://www.cnet.com/tech/services-and-software/7-ways-to-take-screenshots-in-windows-10-and-11/) [linux](https://itsfoss.com/using-gnome-screenshot-tool/)
* Make a screen recording [Quicktime / Mac](https://support.apple.com/guide/quicktime-player/record-your-screen-qtp97b08e666/mac) [OBS / all platforms](https://obsproject.com/)

### Getting remote help:

Because of the hands on nature of working with computers and networks, you will have to use a variety of tools to get help. Your first stop for help should be #code on Slack. You are most likely to get quick help if you post here because anyone in the channel (instructors, other students, other faculty, ed tech alums) can lend a hand.

For more complicated problems, you will need to be able to:

* [post formatted code in slack](https://get.slack.help/hc/en-us/articles/202288908-Format-your-messages)
* take a screenshot from your computer (*unless absolutely necessary, don’t send photos of your screen taken with your phone*)

# Required Materials

## Textbooks

There are no required

## Desktop/Laptop

You must have access to a desktop or laptop computer running Linux, Windows, or MacOS to participate in this course. Further, you must have an administrative account where you can fully configure the computer and install software. “Work” computers where you are not an admin are not sufficient. If your hard drive doesn’t have at least 10GB of free space, you should start clearing space or purchase an external drive.

You will use this computer to participate in the course online and to complete the course projects and activities. You can use a phone or tablet for some online elements of the course, however you will not be able to complete course assignments without access to a computer where you are able to install software.

## Software

* **VirtualBox**. Windows and Intel Mac users will install VirtualBox. VirtualBox is Free Open Source Software that you can install for no cost. Please make sure that you have a computer that meets its [minimum requirements](https://www.virtualbox.org/wiki/End-user_documentation).
* **Parallels**. M1 Mac users will have to purchase [Parallels](https://www.parallels.com/products/desktop/buy/) student edition for $40. If you have a Windows laptop or an Intel-based Mac, you do not need Parallels.

# Schedule

| Week | Date | Topic |
| --- | --- | --- |
| 1 | Wed, Jun 1 | Computers & Operating Systems |
| 2 | Wed, Jun 8 | The Command Line |
| 3 | Wed, Jun 15 | Users, Groups, Files, & Permissions |
| 4 | Wed, Jun 22 | Networks |
| 5 | Wed, Jun 29 | Ethics |

# Grades & Assignments

| Assignment | Pct | Due |
| --- | --- | --- |
| Participation | 20% | ongoing |
| Quiz Design | 20% | ongoing |
| Web Software Project | 30% | 6/18 |
| Final Group Project | 30% | 7/1 |

## Participation

Because this is a short summer course (3 credits in 6 weeks), you should budget 10-15 hours each week for course work. Wednesday live lab sessions are required, but you will also spend significant time working through course materials at your own pace.

Your participation grade will consider preparation for live labs, timely management of your Trello projects, completion of the comprehension quizzes, and participation in Slack and Moodle discussions. Smaller (ungraded) weekly assignments will also be considered for your participation grade. You may earn bonus participation points if you go “above and beyond” in a week, especially by providing help or making full use of our classroom tools to ask good questions and share work.

## Quiz Design

Working in a team of 3, you will be responsible for designing a 15 question quiz (via Google Forms) to check the basic understanding of the videos and readings for the week. The quizzes should consist of multiple choice (one answer or multiple “all that apply”) questions that can be automatically scored. You must provide a detailed answer key for every question. The questions should not be overly challenging, but should be based on the readings (not general knowledge). Anyone who carefully completes the readings should be able to score 13/15 points or better.

Quizzes are **ungraded** for the students taking them. Your team will receive a group grade for the quiz considering:

* *coverage:* did the quiz cover all of the big ideas in the readings?
* *depth:* are the questions challenging without being tricks or requiring detailed synthesis of the materials? do they avoid being to specific or two general?
* *explanations:* do the answer keys for each questions appropariately explain the correct and incorrect answers?

[See the docs page for making quizzes with Google Forms.](https://support.google.com/docs/answer/7032287?hl=en)

Your quiz will be due by end of day on Tuesday and posted to the Moodle “Weekly Quiz” discussion as a new topic for your week. Please, also, leave comments and feedback for the quiz authors – but not until after the quiz is closed. Everyone must attempt the quiz before the start of class at 4:30 on Wed.

## Web Software Project

For the first project, you will install the [Ubuntu](https://ubuntu.org) OS as a *guest* operating system on your computer. You will use Ubuntu to configure a server running one of the following popular web-based software systems:

1. [WordPress](https://wordpress.org/)
2. [Moodle](https://moodle.org/)
3. [Ghost](https://ghost.org/docs/)

Once you have the software installed, you will customize it for a specific use (that you decide). For example, you might configure WordPress as a classroom website, Moodle to support a summer course, or Ghost to host a personal newsletter or blog.

*Submission Guidelines*

To submit this project, you will upload a narrated screencast video of your project to VoiceThread (you can sign in with your Adelphi email account). If you do not want the video to be public, change the permissions so that it is an “unlisted” video. Only people with the link will be able to see it. Create a new thread in the discussion forum with a title for your project. Post the link to your video here, along with any supporting documentation (screenshots, code, config files).

Begin the video with a description of who the target users are of this software and what your goals are for them. Next, show your project from the user’s perspective, showing how it meets the needs of the use cases that guided you in designing the project. Next, you can, if appropriate, give a brief tour of the configuration files and scripts that make your project possible. Your video should be 5-6 minutes long.

Please take time to watch your classmate’s videos and post comments, questions, and suggestions.

*This is an individual project.*

## Final Group Project

One of the key technical tasks of an educational technologist or instructional designer is to research and evaluate possible solutions to a problem and then implement a plan to test a possible solution. Real world problems often require the combination of several systems in a new way, suffer from incomplete or inaccurate documentation, and are hindered by time or resources/financial constraints.

With this in mind, you will choose a software project that you find interesting and engaging to pursue for your final project. You will be responsible for gathering/purchasing the materials to complete the project.

You will be evaluated on:

* skill with the linux operating system
* ability to customize your solution to meet the project’s goals
* creativity of the project chosen
* risk/scope of the project
* reflection on the process

This is a *group project* with either 2 or three people per team. You and your team will work on the same project. You will share a Trello and set goals together. It will be up to you to decide who work is divided and the most effective way to collaborate and share resources.

Here are a few raspberry project ideas that will give you a sense of the size and scope of what you can do for your final project.

1. **MyCroft Open Source personal assistant** <https://mycroft.ai> MyCroft is a Free Software alternative to systems such as Alexa and Google Assistant. MyCroft is developed for Linux. It requires some attention to details to get all of the media components working properly. For this project, you will download and configure MyCroft for your VM; create and connect to your online account, configure and test the microphone and audio output, and customize the features and functionality of your RPI assistant. Start at the [mycroft site](https://mycroft.ai) for all of the information you need to get started.
2. **Jupyter Hub collaborative programming site** <https://jupyter.org/hub> Jupyter Hub is a collaborative version of the Jupyter Notebook software development environment. Set up a Jupyter Hub installation on your VM and configure it for use in a programming, math, or other learning environment that requires a shared coding environment.
3. **Ghost/Moodle/WordPress** While you *had* to set up a basic version of these systems for the first project, they are complex softare systems with many templates, plug-ins, and customizations. Your team may choose to work on one of these to create a more complete website.
4. **Mattermost chat server** <https://mattermost.org> Mattermost is a FOSS chat sever (similar to Slack) that you can host on your own sever for privacy and customizability.
5. **MagicMirror digital kiosk** <https://magicmirror.builders/> The magic mirror or smart mirror runs a customizable web server that serves as a digital kiosk (and, if desired, a magic mirror – a mirrored LCD monitor). Install the base
6. **Slack chatbot** Using the [Slack API](https://api.slack.com/), set up a server (e.g. Node or Python) and create an “app” that interacts, automatically, with Slack chat.
7. **Jekyll website** <https://jekyllrb.com/> Create a simple, static website using the Jekyll system and markdown syntax. Host it live on your own Github Pages site.

These are only a few suggestions. If you have something that you want to build, but aren’t sure the best way to accomplish it, please post your idea in Slack and we can investigate solutions as a team.

*Submission Guidelines*

To submit this project, you will upload a narrated screencast video of your project to VoiceThread. Begin the video with a description of who the target users are of this setup and what your goals are for them. Next, show your project from the user’s perspective, showing how it meets the needs of the use cases that guided you in designing the project. Next, you can, if appropriate, give a brief tour of the configuration files and scripts that make your project possible. Your video should be 8-10 minutes long. Submit one video per team.

# Class Sessions

## Week 1: Computers & Operating Systems

This week we will learn about core computer hardware, and the key software (Operating Systems) that drive computers. We’ll also learn about the specific hardware of our RPIs and install an operating system so we can take them out for a test drive. By the end of the week you should have either VirtualBox or Parellels installed on your computer and have created and installed a Virtual Machine (VM) running the Ubuntu operating system.

### Guiding Questions:

1. How does a modern computer work?### To Do:
2. What makes a computer a *computer*?
3. How is a computer an abstraction of many parts and processes?

### Watch & Read:

1. Watch the Week 1 Introduction Voice Thread (see Moodle)
2. [Computer Hardware](https://www.youtube.com/watch?embed=0&v=ExxFxD4OSZ0) (Computer parts Explained) [7:48]
3. [What is an OS?](https://www.youtube.com/watch?embed=0&v=26QPDBe-NB8) (Crash Course Computer Science) [13:35]
4. [Virtualization As Fast As Possible](https://www.youtube.com/watch?embed=0&v=XItj08D5KPk) (Techquickie) [4:44] (the last 2 minutes are ads, skip it)
5. Klint Finley. (April 24, 2019) [The WIRED Guide to Open Source Software](https://www.wired.com/story/wired-guide-open-source-software/). *WIRED*.

### To Do:

*Complete all the videos/readings before the first live session on Wednesday.*

1. [Join Slack](https://auedtech.slack.com) and install the desktop and mobile apps ([slack support](https://get.slack.help/hc/en-us))
2. Create an account on [Trello](https://trello.com) and install the mobile app ([Trello Support](https://help.trello.com/))
3. Install VirtualBox or Parallels
4. Create a new virtual machine (VM) and install [Ubuntu Desktop](https://ubuntu.com/desktop) as the “guest” operating system.
5. Get to know your new Ubuntu computer by installing software from the Software Shop and using it for some of your every day computing tasks.
6. Read the syllabus and post any questions on Slack
7. Take the “Computer and Operating Systems Quiz” – link on Moodle

### Live Lab Agenda

1. Welcome & Introductions
2. Discussion of videos/readings
3. Breakout Rooms
   1. Room A: Troubleshooting the VM install
   2. Room B: Configuring and using Ubuntu
4. Getting ready for next week: terminal basics

## Week 2: The Command Line

This week we will learn how to interact with our computer using the command line (aka terminal). The command line interface (CLI) offers an alternative to the graphical user interface (GUI) we are more familiar with. Wile the terminal hearkens back to the early days of computing, its still very much alive for systems administrators, software developers, and others. In particular, complex tasks can be accomplished with a few lines of text, remote computers can be easily accessed, and all manner of tasks can be automated (and scheduled). Specifically, after gaining fluency with the command line, you will be able to connect to a remote Linux server from your regular laptop or desktop computer in order to configure and control it. To get started on the command line we’ll check out Terminus, a game developed at MIT to introduce the players to the command line.

In addition to jumping into the command line, by the end of the week you will have customized your VM by installing software packages through apt and the GUI. You will also choose your topic for the first project and set up a Trello with a Board for your project and milestones for each week until it’s due.

### Guiding Questions:

1. How is a command line interface different from the more familiar graphical user interface?
2. What is gained from a CLI? What is lost?

### Watch, Play, & Read:

1. Read [What is Free Software?](https://www.gnu.org/philosophy/free-sw.en.html)
2. Watch [Keyboards & Command Line Interfaces: Crash Course Computer Science #22](https://www.youtube.com/watch?embed=0&v=4RPtJ9UyHS0)[11:23]
3. Play [Terminus](http://web.mit.edu/mprat/Public/web/Terminus/Web/main.html) for at least an hour. Post on Slack how far you get.
4. Read/review [Ubuntu Command Line For Beginners](https://ubuntu.com/tutorials/command-line-for-beginners#1-overview)

### To Do:

1. Update the software on your VM with apt (hint sudo apt update then sudo apt dist-upgrade)
2. Install at least 2 programs on your RPI using apt or the Software center.
3. Post in the FOSS Apps Discussion which apps you installed and a brief review of them.
4. Create a Trello Board for your project and invite the course instructor as collaborator.
5. Complete the “Command Line” quiz

### Live Lab Agenda

1. Welcome, questions
2. Command line Quizlet
3. Breakout rooms: A. WordPress working group B. Moodle working group C. Ghost working group
4. Project ideas and Trello Updates

## Week 3: Users, Groups, Files, & Permissions

As you’re working on finishing up your first project, we’ll take a deeper look at how files, permissions and security works in linux/unix operating systems. To better understand how to work with files, we’ll take a look at file archives (multiple files and directories combined in a single file) and compression (reducing the size of a file). Because your programs all run as a “user” (either root or their own user), it’s important to understand files and permissions to troubleshoot problems.

### Guiding Questions:

1. How do you secure digital resources?
2. Is the unix approach of users and groups sufficient for all security needs?

### Watch & Read:

1. Watch [Files & File Systems: Crash Course Computer Science #20](https://www.youtube.com/watch?embed=0&v=KN8YgJnShPM) [12:02]
2. Watch [Compression: Crash Course Computer Science #21](https://www.youtube.com/watch?embed=0&v=OtDxDvCpPL4) [12:47]
3. Read [Understanding Basic File Permissions and ownership in Linux](https://www.thegeekdiary.com/understanding-basic-file-permissions-and-ownership-in-linux/)
4. Watch [Linux Terminal 201: How To Use tar, gzip, bzip2, and zip - HakTip 156](https://www.youtube.com/watch?embed=0&v=f8-7lhs4ky0) [11:32]
5. Docs [Ubuntu user security](https://ubuntu.com/server/docs/security-users)
6. Docs [How To Extract Zip, Gz, Tar, Bz2, 7z, Xz and Rar File in Linux](https://tecadmin.net/extract-archive-file-linux/)

### To Do:

1. Finish the first project
2. Complete the “Users & Groups” quiz

### Live Lab Agenda

*project working session*

## Week 4: Networks

How does the internet work? What happens after you hit enter on a search term on [Duck Duck Go](https://duckduckgo.com/spread)? How does Spotify get music to your phone then to your wireless earbuds? A deeper understanding of different networking hardware, software, and protocols will help us better understand the networked software we’re installing, configuring, and troubleshooting.

### Guiding Questions:

1. How does the internet work as a collection of networks?
2. How does a multi-layered model allow for abstraction between different layers?

### Watch:

1. [How the Internet Works in 5 Minutes](https://www.youtube.com/watch?embed=0&v=7_LPdttKXPc) [4:48]
2. [The Internet: IP Addresses & DNS](https://www.youtube.com/watch?embed=0&v=5o8CwafCxnU) [6:44]
3. [The Internet: Wires, Cables & Wifi](https://www.youtube.com/watch?embed=0&v=ZhEf7e4kopM&list=PLzdnOPI1iJNfMRZm5DDxco3UdsFegvuB7&index=2) [6:41]
4. [The Internet: Packets, Routing & Reliability](https://www.youtube.com/watch?embed=0&v=AYdF7b3nMto&list=PLzdnOPI1iJNfMRZm5DDxco3UdsFegvuB7&index=4) [6:25]

### To Do:

1. Post (with a partner) a written report on one of the network topics (see full instructions below).
2. Choose a topic and form a team for the final project. Create a Trello board for it and share it with your partners and the instructor.
3. Complete the “Networks & Internet” quiz.

### Networking Topic Report Instructions

For a more in-depth look at networking, you will work with a partner to write a mini report on a related topic. Please: a) choose a partner, and b) choose a topic from the list below. If you have an idea for a different topic, you may write about that with approval from the instructors. By the end of the week, please post your report directly into our *Networking Topics* forum. Your report should be roughly 500-800 words. It should give an overview of the topic, summary of how it works, and discussion of how it’s used and why it is (or isn’t) important. Read through the other posts, and ask any follow-up questions. The instructor and topic authors will do their best to answer your questions.

Possible topics:

1. Bluetooth
2. Mesh network
3. Near Field Communication (NFC)
4. HTTPS/SSL
5. 4g/5g/6g
6. BitTorrent
7. Dark Web
8. Radio-frequency identification (RFID)
9. Bluetooth Beacons
10. Web3

## Week 5: Ethics

[As Spider-Man learned](https://www.marvel.com/comics/issue/17610/spider-man_with_great_power._2008_1), with great power comes great responsibility. We trust the people who run our networks to keep our information safe, and to not violate our trust while they do it. Along the way, they may face some hard choices, usually weighing the benefits of individuals versus the group. We’ll take a look at some of the ethical concerns in “big tech”, and also investigate how these same concerns appear in the context of schools and other educational institutions.

### Guiding Questions:

1. How do school systems balance student privacy and autonomy, with safety and learning efficiency?
2. What are the key ethical concerns of embedding corporate interests (and technology) into public spaces like schools?

### Read:

1. **Monitoring Student Social Media**

* Karen Turner. April 22, 2016. [Schools are helping police spy on kids’ social media activity](https://www.washingtonpost.com/news/the-switch/wp/2016/04/22/schools-are-helping-police-spy-on-kids-social-media-activity/). *The Washington Post*.
* Aaron Leibowitz. September 6, 2018. [Could Monitoring Students on Social Media Stop the Next School Shooting?](https://www.nytimes.com/2018/09/06/us/social-media-monitoring-school-shootings.html). *The New York Times*.
* Tom Simonite. August 20, 2018. [Schools Are Mining Students’ Social Media Posts for Signs of Trouble](https://www.wired.com/story/algorithms-monitor-student-social-media-posts/). *Wired*.

1. **Student Data**

* [Student Privacy](https://www.eff.org/issues/student-privacy). *Electronic Frontier Foundation*.
* Natasha Singer. May 13, 2017. [How Google Took Over the Classroom](https://www.nytimes.com/2017/05/13/technology/google-education-chromebooks-schools.html). *The New York Times*.
* Gennie Gebhart. March 28, 2017. [Privacy By Practice, Not Just By Policy: A System Administrator Advocating for Student Privacy](https://www.eff.org/deeplinks/2017/03/privacy-practice-not-just-policy-system-administrator-advocating-student-privacy). *Electronic Frontier Foundation*.

### To Do:

1. Complete ethics readings and come to class prepared to discuss and debate.
2. Update your Trello project with the work you’ve completed this week. Identify any issues and get help as needed.
3. Complete the “Privacy & Ethics” quiz.

## Week 6: Final Project Due

There is no class meeting, finish up your project and turn it in. Have a great summer!