**CEG 2350 - OS Concepts and Usage**

**Fall 2023**

**Lecture**: TR 9:30 – 10:50 AM in-person in Oelman Hall 320

**Lab**: MW 1:25 – 2:20 PM, in-person in Oelman Hall 320

**Instructor**: Ms. Kayleigh Duncan

**Office Location:** 340 Russ Center Engineering

**Instructor Office hours**:

In-person: MWF 10:30 – 11:30 AM or by appointment in Russ 340

Remote: via Discord, send message in course chat. Do not expect answers after 8:00 PM

**Contact info**: [kayleigh.duncan*@*wright.edu](mailto:vanessa.starkey@wright.edu); 775-3682

**TAs**:

* Ankita Agarwal ([agarwal.15@wright.edu](mailto:agarwal.15@wright.edu))
  + Office Hours: TR 2:00 – 3:00 PM on Discord / Webex
* Harrison Aunspaw ([aunspaw.3@wright.edu](mailto:aunspaw.3@wright.edu))
  + Office Hours: MW 3:00 – 6:00 PM on Discord. In-person by request.

**Midterm: 9:30 – 10:50 AM Thursday, October 12th**

**Final: 8:00 – 10:00 AM Tuesday, December 12th**

**Course description:**

Provides introduction to Linux and Windows operating systems and system administration. Covers files and directories, ownership and sharing, programs and processes, system calls, libraries, dynamic linking, command line shells, scripting, regular expressions and secure network protocols.

**Prerequisites**:

Undergraduate level CS 1160 Minimum Grade of D or Undergraduate level CS 1180 Minimum Grade of D or Undergraduate level CEG 2170 Minimum Grade of D

**Recommended Resources**:

Students will be provided with online reference material through the course of the class. Textbook(s) can be purchased at the student's discretion.

The Linux Command Line, 2nd Edition, no starch press, William Shotts <https://linuxcommand.org/lc3_learning_the_shell.php>

How Linux Works, 3rd Edition: What Every Superuser Should Know, no starch press, Brian Ward

Pro Git, Apress, Scott Chacon and Ben Straub <https://git-scm.com/book/en/v2>

UNIX and Linux System Administration Handbook 5th Edition, Addison-Wesley Professional, Evi Nemeth, Garth Snyder

A Practical Guide to Linux: Commands, Editors, and Shell Programming, Prentice Hall, Mark G. Sobell <https://proquest.safaribooksonline.com/book/operating-systems-and-server-administration/linux/9780134774626> (This book can be read online – use your Wright State Library credentials. Edition does not matter.)

**Software**:

WSL2 Ubuntu (for Windows users), AWS Virtual Servers (via Education Account)

**Pilot*/*campus email**:

[https:*/*/pilot.wright.edu](https://pilot.wright.edu) Pilot will be used in this course for submitting labs and projects and for accessing course materials and grades. It is the student's responsibility to check the Pilot site, as well as his/her WSU email, for course announcements, updates to project requirements, etc.

**Grading**:

Lab assignments: 60%

Quizzes: 10%

Exams: 30% (15% for midterm and 15% for the final exam).

No makeup exams are given unless there is a verifiable emergency.

The grading scale for the course is [90-100] A; [80-90) B; [70-80) C; [60-69) D; [0-60) F.

**Assignment \ Quiz \ Exam Policy:**

Students are responsible for keeping up with due dates. Specific dates are given in the Dropbox section of Pilot.

All labs must be submitted via the Dropbox on Pilot. Be sure you upload the files and then click submit. If you do not receive an email confirmation, then your work was not properly submitted. Be sure to allow enough time to submit your work before the Dropbox closes. Personal computer issues/loss of data/not allowing enough upload time are not valid justifications to merit an extension of the deadline.

Quizzes and Exams will be given in an online format. Dates for both will be announced with ample time. If the student requires any change / extension they are responsible for notifying the instructor as soon as possible.

**Late Policy**:

Late work will receive a penalty of 10% off per day. Work more than 3 days late will not be accepted.

Exams and quizzes will only be open for the time frame specified.

**Students with disabilities**:

Any student with a disability must register with the Office of Disability Services to determine what accommodations, if any, are appropriate. The student must inform the instructor of the special accommodations needed as soon as possible.

**Technology Requirements**:

This course requires you to have access to a computer and the internet. If you do not own or have access to any of these items, please see the “Remote Support Information” options through CaTS and contact them for assistance here: wright.edu/it. In the event CaTS does not have support options for any of the listed items above after contacting them, please contact me via email as soon as possible.

**Help Room**:

The Department of Computer Science and Engineering maintains a help room, staffed by upper-level students, for students in introductory programming classes.

A virtual help room is available via the Department’s Discord server. In the “WSU Computer Science Department” Discord server, select the “CS Help Room” channel. Here you can make a post to “cs-help-room" to list your questions or request one-on-one support with voice or screen sharing.

**Tutoring Services & Study Coaches:**

Tutoring Services provides individual and small group academic support for a wide range of courses. Students can meet with an individual tutor by appointment for frequently requested courses numbered 3000 and below.

Students who want to develop strategies and skills specific to their needs for all of their courses can meet by appointment with a Study Coach. Study Coaching is an excellent way to get personalized solutions to the challenges of online and in-person courses.

For students to sign up for appointments, follow the link on the website (www.wright.edu/tutoring) to the Tutoring Services Online Request form. Once the form is received, their front desk staff will be in contact regarding their appointment status. Appointments are free and are set for one hour each week through the last week of classes.

**Academic misconduct**:

Oelman 320 is a computer lab. There should be no food or drink allowed. Do not be disruptive during lecture or lab. Violations of either of these rules can be cause for dismissal from lecture or lab for that period.

Labs are to be completed individually. Sharing your work or copying someone else's work or plagiarizing from online resources will result in a 0 for the assignment(s) and a formal Academic Integrity Violation will be submitted. The university policy on academic misconduct can be found at <http://www.wright.edu/students/judicial/integrity.html>

**Lecture Topics**:

The topics are grouped based on coherence. This is not necessarily the weekly schedule, more of a guideline for topics this class will cover.

Note: Commands are referenced in Unix/ Linux terminology. Windows equivalents will also be discussed.

Getting started

1. Operating systems as jargon – Windows, Linux, Mac
2. Linux distributions: Debian, RedHat, etc
3. Intro to course tools (AWS & MobaXTerm)
4. Intro to shells (bash, sh, zsh, powershell, ssh) and commands (man, vim, history)
5. Intro to ssh & key pairs
6. Intro to git (clone, commit, push, add)

Files, directories, and OS structure

1. OS structure in Linux (and Mac) vs Windows
2. Files vs directories
   * Commands: ls, cp, mkdir, rmdir, vim, mv, rm, cat, touch, head/tail, less/more
3. Importance of names and extensions (file)
4. Ownership & sharing
   * rwx permissions in Linux (and Windows counterpart)
   * user vs. group vs. other
   * chmod, chown
   * User types and groups: sudo vs. admin vs. “other” (sudo, adduser, deluser)
5. Hard links & soft links (ln)
6. Inodes
7. File locks

Scripting

1. Aliases & customizations (.bashrc, .vimrc)
2. Commands: grep, diff, find, file, wc, sort, uniq, whereis, which
3. IO redirection (<, >, |, tee)
4. PATH (order of execution)
5. Scripting languages: bash, python
6. Regular expressions (grep, sed, test)
7. Alternate data streams (od, /dev/null)
8. Requiring root to run

Command line programming

1. Assembly/machine language programming
   * Bits (1), bytes (8 bits), word (16, 32, or 64 bits)
2. Interpreters
3. Libraries & dynamic linking (loads and links the shared libraries needed by an executable when it is executed)
4. Memory handling in programming (static vs. stack vs. heap)
5. Makefiles
6. Compiling programs (Linux is not Windows)
   * Linux ELF (Executable & Linkable Format)
   * Windows EXE
7. Overview of command line debugging
8. Commands: ldd, make

Data storage and access:

1. RAM vs. ROM (SSDs vs HDDs)
   * Suspend vs hibernation of an OS
   * Virtual memory & swap tables (one go) / paging (chunking)
2. Booting of OS
   * Von Neumann machine: Control unit; instruction fetch, decode, and execution
   * Power on self-test (POST)
   * BIOS vs UEFI
   * Boot loaders: NTLDR, GRUB
3. Fragmentation
4. Sequential and random access
5. Compression (gzip, tar)
6. File systems: vfat, ntfs, ext
   * implementations / fragmentation / performance
7. Network file share: SFTP, NFS, Samba
8. Commands: df, du, mount, umount, etc/fstab
9. MBR vs GPT
10. Devices and drivers

Processes

1. Task Manager (Linux [top] vs. Windows)
2. init process / kernel level vs user level processes
   1. parent vs child process
3. System calls (way for programs / APIs to interact with the operating system)
   1. systemd - systemctl & journalctl
   2. Fork & exec
4. Commands: kill, ps, top, nice, bg, fg, jobs, nohup, killall, screen
5. Preemptive (interruptible) and non-preemptive scheduling, priorities, signals
6. CPU & kernel memory management
7. Thread vs Fork
8. Single CPU multi-tasking
9. Multiple CPUs and SMP (Symmetric multiprocessing)

Git (version control)

1. Git servers vs. clients
2. Basic commands: push, pull, commit, add, init, clone, branching, custom configurations

Networking

1. Host names, IP addresses, Subnets, DNS, URLs
2. Protocols: HTTP vs HTTPS
3. TCP and UDP
4. Ports vs sockets, clients and servers, secure shell, sftp
   * Application opens a socket which is connected to a port
5. X11 in terms of graphical “forwarding” and role of a GUI
6. Web browsers (URLs, SSL certs) and clients (ping, wget, traceroute)
7. Physical pieces of a network
   * hosts, routers, switches, ISPs, wireless, LAN
8. Firewalls
9. OSI layers (encapsulation / packets, application, transport, network, datalink, physical)

System Administration

1. File system & hardware integrity
2. Updates (Linux vs. Windows)
3. Installing packages with make
4. systemctl & journalctl
5. Virus scanning & types of computer diseases
6. Creating archives, password locking files
7. System restore (automatic, manual, backups)
8. Windows Registry
9. Passwords: /etc/passwd, /etc/shadow
10. Boot scripts & init.d
11. Access Control List (ACL)
12. Open source movement
13. Torrents?
14. Commands: write, finger, who, mesg

Computer-ception

1. Virtual machines (GUI style) and hypervisors
2. Containers
3. Cloud computing (Google, AWS, Azure) & basic terminology
   * Storage buckets, cloud networking, heavy computation

Extra Topics

1. Building a Linux kernel from source