NAME

`CS 4350` — Unix Systems Programming

SYNOPSIS

Unix Systems Programming will introduce the fundamentals of Unix operating systems, Unix file system and environment, C memory allocation, development tools, processes and signals, threads, device drivers, and programming for security.

This is the syllabus for the course.

LOGISTICS

Credits	3
Prerequisites	C or higher in CS 3358
Time	Tuesdays and Thursdays 3:30−4:50 PM
Place	Derrick 235
Final Exam	Tuesday, May 12 at 2:00 PM
Website	All content is in TRACS

INSTRUCTOR

Michael D. Ekstrand `ekstrand@txstate.edu`
Phone: (512) 245-7523
Office: Comal 307F
Office Hours:
M 10AM–noon
TR 2–3PM
F 11AM–noon

COURSE OVERVIEW

This course will introduce you to various aspects of UNIX systems programming. At the end of the course, you should be able to write 'systems' programs - programs that operate at a fairly low level and provide some kind of important service, such as file system management or network services.

We will cover a variety of topics in this course, including:

- Unix shells and scripting
- Commands and pipelines
- The organization of UNIX systems
- Basic C programing for UNIX environments
- File systems, directories, and file operations
- Network programming and sockets
- UNIX security provisions and defensive programming
- Standard Unix facilities and system daemons

MATERIALS

Textbook

If you would like a textbook to consult, we are using the following: Niel Matthew and Richard Stones, *Beginning Linux Programming, 4th Edition*, *Wrox*, 2007, ISBN 978-0-470-14762-7.

The textbook is optional.

If you want a book that goes deeper into UNIX programming, you can get and read W. Richard Stevens and Stephen A. Rago, *Advanced Programming in the UNIX Environment*, *Addison-Wesley*, 2013, ISBN 978-0-321-63773-4.

Online Reference

You will likely want to make extensive use of the `man`(1) pages that are included in any Unix system. They provide direct online access to documentation, and are accessible in the terminal. If you wish to read man pages in your web browser, try: **http://man.cx/**

On GNU/Linux systems, sometimes the texinfo system (accessible via the `info` command) has more complete documentation for a facility, such as the C compiler. BSD systems tend to have fairly complete man pages.

Software

You will need access to a Unix-like system. We will be primarily using Linux, specifically Fedora and CentOS/RHEL, although we will also be examing other Linux distributions and systems such as the BSDs and Mac OS X. You should at least use a Linux system to test your homeworks. The department provides many Linux systems, running both Ubuntu and Red Hat Enterprise Linux; some of them are remotely accessible.

If you have a Mac or Linux computer, you can do a lot of testing and exploration directly on that. If you have a Windows computer, or want to test on Linux on your Mac, you can install a Linux distribution in a virtual machine using VirtualBox, Parallels, or similar software.

More detailed instructions on setting up a Linux virtual machine, along with configuration files, are available in TRACS.

CS Remote Access:
http://cs.txstate.edu/labs/remote_access.php
VirtualBox: **http://virtualbox.org**
Fedora: **http://www.fedoraproject.org**
CentOS: **http://www.centos.org**

STRUCTURE

This course is built around several pieces:

- **Class meetings** to present and discuss the material.
- **Assignments** to give you significant experience with Unix programming.
- **Drills** to give you additional practice, and experience with material that does not fit in the assignments.
- **Exams** to assess your knowledge of the material.

Your grade will be computed as follows:

Element	**Weight**
Assignments	49%
Drills	10%
Exams	21%
Final	20%

Grades will be issued on a standard 70/80/90 scale; I may adjust the scale somewhat based on course performance, but a total weighted grade of 90 will guarantee you an A.

Class Meetings

We will use the regularly-scheduled lecture time for lectures and discussions on the class material, as well as for exams.

While I do not take attendance most class periods, I expect you to come to lecture and material in lecture is liable to show up on exams and drills.

Assignments

The assignments are an important part of this class, and are where you will get most of your experience with the material presented. Assignments will contain a mixture of C and shell programming.

There are 7 assignments in this course. The last 5 will build on each other, as you develop and extend a web server for Unix systems.

For the assignments, you have 4 ‘late days’ that you may use throughout the semester at your discretion. Using a late day extends the due date of the assignment by **24 hours**. You must state in the assignment submission if you are claiming late days for that assignment, and if so, how many.

The late days are yours to use as you will for any reason, and I do not need the justification when you use one. I do appreciate it if you let me know before the assignment is due that you will be using a late day.

Drills

The drills are small assignments that are due on the weeks with no major assignment. A drill may be a small program or script to write, or a brief quiz on TRACS.

There are 7 drills. Your lowest 2 drill grades will be dropped.

Exams

Instead of 1 or 2 large midterms, we will have 3 shorter exams throughout the semester. Each exam is weighted equally. Exams will be on Thursdays during the second half of the class period.

There will also be a make-up exam at the end of the term. This exam is optional; if you take it and turn it in to be graded, its grade **will replace** your lowest regular exam grade, *even if it is lower.* The make-up exam is to cover for cases where you need to miss one of the regularly-scheduled exams, or if you have done particularly poorly on one of the exams; taking it just to try to improve your grade is not likely to be a good idea.

Final

The final will be at its regularly schedule time, and will be cumulative.

SCHEDULE

The following is the basic schedule for the semester. Topic scheduling is subject to change and be further filled in as the semester progresses.

Week	**Monday**	**Topic**	**Sched**	Due
1	1/19	Processes, Files, and Streams		
2	1/26	C Programming for Unix		*D1*
3	2/2	Memory Management		*A1*
4	2/9	Unix File System	*Exam 1*	*D2*
5	2/16	File and Network I/O		*A2*
6	2/23	Processes in C		*D3*
7	3/2			*A3*
8	3/9		*Exam 2*	*D4*
	3/16	*Break*		
9	3/23			*A4*
10	3/30	Async I/O		*D5*
11	4/6	Threading		*A5*
12	4/13		*Exam 3*	*D6*
13	4/20	Kernel Programming		*A6*
14	4/27			*D7*
F	5/4	*Finals* (final on **May 12**)		*A7*

Each assignment is due at **noon on Tuesday** of the week in which it is due.

OTHER POLICIES

Course Announcements

From time to time, I will make announcements regarding the course materials, structure (such as assignment due dates, changes, or corrections), etc., either in lecture or via TRACS announcements. You are responsible for these announcements. Make sure that you check TRACS regularly, or have TRACS announcements delivered to your e-mail.

Late Work

Except as specified in the descriptions for individual course elements, late work will not be accepted. The course has built-in allowances that are intended to cover most ordinary need for extensions and make-up work, so individual extensions will not be granted except in exceptional circumstances.

Cheating and Academic Integrity

As both a programmer and a student, you are expected to do your own work, attribute sources, and respect the legal and moral rights of others with respect to their work; as a student, you are also required to abide by the University Honor Code [UHC]. While I aim to allow you to make reasonable use of resources, cheating (including copying code, using unauthorized resources during tests, etc.) will not be tolerated. If you are found to be cheating, the penalty may range from an F on the assignment to an F on the course, and will also be reported to the university.

External Resources

You may consult external resources such as other books and web sites for improving your understanding of the material or to overcome difficulties in completing the assignments. In your assignment solution, list all external resources you used; if they are available online, provide the URL. You do not need to cite the textbook, or man pages commonly installed on Linux, BSD, or Mac systems.

Besides the course forum, you may ask questions related to the assignments on publicly accessible discussion forums such as Stack Overflow, newsgroups, or publicly-archived mailing lists. Provide URLs to the forum discussion on the relevant web site or archive (Google Groups works well for newsgroup archives) with your project deliverable submission. When you ask your question, mention that it is for a course project and that your instructor permits you to make reasonable use of discussion forums. You may provide a link back the syllabus so that potential answerers are aware of this policy; a URL to a publicly-available version of the syllabus is under **REFERENCES**.

Restrict your questions to questions about how to go about a particular sub-portion of the problem, how something works, why something you are trying doesn't work, or other specific difficulties. Do not ask "how do I solve ⟨the problem description⟩", or similarly direct translations of the project requirements, or for specific code. Questions should be written to fill in a gap in your understanding that will then enable you to continue your work, not to get a solution to the assignment.

Conduct

You are expected to behave in a civil, respectful manner in all class interactions, both in official meetings such as lectures and out-of-classroom activities such as project group meetings and study sessions, and to contribute to a constructive learning environment. Texas State policy (PPS 4.02) describes general behaviors that are disruptive. In addition, the Hacker School Social Rules [HSSR] are a good source of guidance on how to maintain a constructive and educational environment. If you experience or witness harassment of any form, please let me know.

Disability Accommodations

If you need particular accommodations to be able to fully participate in this course, please talk with me as soon as possible. I may ask that you provide documentation from the Office of Disability Services, so if you have such documentation please bring it.

REFERENCES

HSSR *Hacker School Social Rules:*
https://www.hackerschool.com/manual#sub-sec-social-rules

UHC *University Honor Code:*
http://www.txstate.edu/honorcodecouncil/

SYL *Public syllabus:*
http://elehack.net/teaching/cs4350-s15.pdf

COPYRIGHT