

Class Number	CSC 615.01 Spring 2020 Thursday's 4:00pm-6:45pm, TH 409 (date of last update: 12-03-2019)
Class Title	Unix Programming
Instructor	Robert Bierman Office Hours: TBD Office: TBD Email: bierman@sfsu.edu
Course Description	<p>Bulletin Copy: Programming in a UNIX environment. Topics include regular expressions; utilities such as awk, sed, grep, csh, sh, ksh; system calls such as signals, sockets, POSIX IPC, and POSIX threads; kernel internal structures.</p> <p>This course is a senior elective in the area of Operating Systems and Distributed Processing.</p> <p>Spring 2020 Revised Course Description: This is a <i>physical computing</i> course focused on Embedded Linux programming. Embedded devices are all around us these days, from self-driving cars; to doorbells that recognize is someone is leaving a package at your door; to vacuum cleaners that navigate your room by themselves. The course teaches how to program directly to the operating system API's (verses stacks or other large libraries that waste resources) and specifically focuses on interacting with additional hardware such as sonar sensors, IR sensors, Lidar, and motors, with the end team-project goal of programming an autonomous robot car that can self-navigate an obstacle course.</p> <p>The course includes classroom instruction as well as in-class lab time to work with the physical devices. We use Raspberry Pi's (models 3b+ and 4) as our target system.</p>
Prerequisites	<p><u>CSC 415</u> with grade of C or better or consent of instructor.</p> <p>You should be familiar with using the Linux programming environment, and have a good working knowledge of the C programming language as well as the basics of concurrent programming (processes, threads, synchronization)</p>
Text (optional)	Title: Hands-On System Programming with Linux Author: Kaiwan N Billimoria ISBN: 978-1-78899-847-5

Course Web Site	iLearn
Reader	<p>There are no published lecture notes for this class.</p> <p>Lecture notes will be available download from the course web site during the semester.</p>
Course Objectives and Role in Program	<p>The objectives of this course include:</p> <ul style="list-style-type: none"> • Introduce the student to Unix/Linux kernel programming techniques • Teach advanced C systems programming techniques in a Unix/Linux environment • Review basic concepts covered in the core Operating Systems course prerequisite as they are realized in the Linux platform • Introduce the concepts of limited resources (limited memory, disk space and compute capacity) and how to manage them • Introduce physical computing, how to program interacting with the real world • Discuss correct synchronization techniques for both application programs and kernel code running on various platforms <p>Students will implement and evaluate several small application programs utilizing low level Linux system calls, then work through a series of progressively more difficult kernel programming tasks, culminating with the design and implementation of correctly synchronized kernel module code that can control a physical device. The knowledge of advanced programming techniques (including correct synchronization) and the Linux platform play an important role in developing our students into skilled professional programmers.</p>
Learning Outcomes	<p>At the end of this course students will be able to:</p> <ul style="list-style-type: none"> • Write correct and well documented advanced C code using low level Unix/Linux system calls that is demonstrated to execute correctly • Know where to look for platform specific programming information and be familiar with reading and using man page information as well as other standard reference materials • Clearly and accurately explain design decisions in written program documentation • Be familiar with the mechanics of connecting, utilizing and programming sensors, motors and other devices to a system

	<p>through various interfaces including USB, I2C, SPI, and UART as well as utilizing ADC and DAC devices.</p> <ul style="list-style-type: none"> • Be able to design and implement simple, but efficient, concurrent process and thread based applications. 										
Lecture Topics	<ol style="list-style-type: none"> 1. Linux Architecture 2. Virtual Memory 3. Kernel implementation of filesystems & device I/O 4. GPIO 5. Resource Limits 6. Dynamic Memory Allocation 7. Sensors and other Physical Hardware Devices 8. Processes 9. Fork & Threads 10. Semaphores 11. Code Synchronization 12. Signals 										
Assignments	<p>There will be various individual C programming assignments throughout the course following our lectures along with the Team Project of building and programming a robotic car that can self-navigate an obstacle course.</p> <p>The Team Project also includes a written Team Report and an Individual Report along with the code and results of the obstacle course as part of the final submission.</p> <p>There will be a few quizzes and a midterm examination.</p>										
Grading	<table> <tr> <td>Individual programming assignments:</td> <td>25%</td> </tr> <tr> <td>Team Project:</td> <td>40%</td> </tr> <tr> <td>Quizzes:</td> <td>10%</td> </tr> <tr> <td>Midterm:</td> <td>15%</td> </tr> <tr> <td>Attendance and Participation:</td> <td>10%</td> </tr> </table>	Individual programming assignments:	25%	Team Project:	40%	Quizzes:	10%	Midterm:	15%	Attendance and Participation:	10%
Individual programming assignments:	25%										
Team Project:	40%										
Quizzes:	10%										
Midterm:	15%										
Attendance and Participation:	10%										
Teaching Methods	<p>This course is both lecture based and hands on experimentation and learning.</p>										
Disability Access	<p>Students with disabilities who need reasonable accommodations are encouraged to contact the instructor. The Disability Programs and Resource Center (DPRC) is available to facilitate the reasonable accommodations process. The DPRC is located in the Student Service Building and can be reached by telephone (voice/TTY 415-338-2472) or by email (dprc@sfsu.edu).” (http://www.sfsu.edu/~dprc)</p>										
Student disclosures of sexual violence	<p>SF State fosters a campus free of sexual violence including sexual harassment, domestic violence, dating violence, stalking, and/or any form of sex or gender discrimination. If you disclose a personal experience as an SF State student, the course instructor is required to</p>										

notify the Dean of Students. To disclose any such violence confidentially, contact:

[**The SAFE Place** - (415) 338-2208;

http://www.sfsu.edu/~safe_plc/]

[**Counseling and Psychological Services Center** - (415) 338-2208;

<http://psyservs.sfsu.edu/>]

For more information on your rights and available resources:

[\[http://titleix.sfsu.edu\]](http://titleix.sfsu.edu)