

CpS 250: Unix Programming

Spring Semester 2021-2022

Instructor:	Stephen Schaub	Email:	sschaub@bju.edu
Office:	Alumni 70	Telephone:	Ext. 2264
Office Hours:	MTWF 10:00-10:50am		
	Th Electronic		

Course Information

An introduction to tools and languages used to develop software for Unix-based operating systems. Presents a spectrum of techniques ranging from shell scripting to systems programming in C, with an emphasis on secure programming practices. Second semester, three hours. *Prerequisite: CpS 230.*

Overview: This course provides a comprehensive introduction to the Unix family of operating systems from a programmer's perspective. We discuss Unix text processing commands, shell scripting, and systems programming with C and Python. The goal is for you to gain an appreciation for "alternative" operating systems and programming tools, and to obtain hands-on experience with an important and popular Unix incarnation, Linux.

Language proficiency: Students are expected to have basic familiarity with C (from CpS 230) and be reasonably proficient in Python (from CpS 110). See the instructor for resources to help you prepare if you do not have prior experience with these languages.

Course Resources

Textbook: There is no textbook for this class. A variety of online readings will be assigned. See the course schedule.

Announcements: Students are expected to use the Microsoft Teams CpS 250 team to receive course announcements and to engage in the online experiences of this course.

Website: The course website contains links to assignments and required readings: <https://cs.bju.edu/cps250>

Grading

Grading:			
Qty	Item	Points	Total
5	Quizzes	10	50
5	Labs	10	50
5	Exercises	50	250
2	Projects	100	200
2	Lab Tests	75	150
2	Written Tests	100	200
1	Final Exam	100	100
Total Points:		1000	

Scale:	approximate
A	90-100%
B	80-90%
C	70-80%
D	60-70%
F	<60%

There are three types of assignments in the class: labs, exercises, and projects. **Labs** are small-scale assignments that typically take an hour or two to complete. **Exercises** are small programming assignments that might take a few hours. **Projects** are larger-scale assignments that will likely take many hours (10-20). Projects are graded as follows:

- **60% Correctness:** Program produces correct results; runs according to specification. Attention to the program specification is very important here.
- **20% Style:** Code is written according to style guidelines and instructor's requirements. Consistency and attention to detail are important. The goal is clear, easily understandable code, thoughtfully commented. **Programs which do not receive at least 20 of the 60 correctness points will receive no style credit.**
- **15% Reports:** Each program and project submission is accompanied with a written report.
- **5% Presentation:** Program assignment submitted according to instructions.

Course Policies

Assignments can receive full credit only if submitted by 11:59 p.m. on the day due. A 25% penalty will be applied if the assignment is not turned in on time. No credit is possible after one week. I waive one late penalty for students who are punctual in their class attendance. If you anticipate trouble on an assignment, see me as soon as possible for assistance.

Students are required to submit a reasonable attempt for all exercises and projects, even if the attempt is too late to receive credit. Failure to submit a reasonable attempt for one or more assignments may result in a penalty of up to one letter grade on the final course grade.

Programs may be submitted late only by approval of the instructor. I will allow this only for students who formally request permission to submit the program late. The request must be made by email, and should report the number of hours invested and include a description of problems encountered. Your current program effort should be uploaded to the submission system. The request must be submitted by the original assignment deadline. Requests indicating little effort invested will be denied.

Electronic devices may not be used during class except by special arrangement with the instructor.

Gum chewing in professional settings is inappropriate and therefore not permitted in class. Compliance with student handbook policies is expected during class.

University Policies

Attendance Policy

You are expected to attend class and be on time per the standard University attendance policy: <https://home.bju.edu/bju-policies/>. If you come late or leave early, I will record a partial attendance mark if you missed at most 15 minutes of class. If you miss more than 15 minutes of class, you will be marked absent. If you exceed the 3 allowed absences for this class, you may be withdrawn from class.

For planned absences, notify me a week ahead of time by e-mail. Written assignments and scheduled tests should be completed before your planned absence; please contact me to make arrangements for doing so. It is your responsibility to check in advance of a planned absence to verify what is due.

For absences due to illness or emergency, contact me by the end of the day of your absence to indicate the reason for your absence and to arrange for making up any graded work without penalty. In these situations, you will be able to make arrangements for making up tests without penalty for the first occurrence. Each subsequent time a test is missed because of incapacitating illness or emergency, an additional 10 percent grade penalty for that test will be incurred.

Accommodations for Students with Disabilities

If you have a documented learning disability or if you are impaired in some way (auditory, visual, cognitive, neurological, or physical), please let the instructor know this within the first week of the course so that any necessary adjustments can be made before you get behind.

Academic Honesty and Integrity Policy

See the Computer Science Department's Academic Integrity Policy:

<https://cs.bju.edu/academics/policies/academic-integrity-policy/>

Copyright 2022 Stephen Schaub. Students are prohibited from selling (or being paid for taking) notes during the course to, or by any person, or commercial firm without the express written permission of the professor teaching the course. Students are prohibited from distributing course materials. This course uses copyrighted material from other sources. This material is being used by permission.

Curriculum Information

Context

This course supports the following objectives of the Computer Science and Information Technologies programs:

- CS 1. Design and implement solutions to practical problems
- CS 2. Use appropriate technology as a tool to solve problems in various domains
- CS 3. Create efficient solutions at the appropriate abstraction level
- CS 6. Demonstrate an ability to acquire new knowledge in the computing discipline

Learning Objectives

Objective	Content	Assessment
Complete basic programming-related tasks in Linux (CS 2)	Week 1, 2	Labs 1-4
Use Unix tools to solve text processing problems (CS 1, 2, 3)	Week 2	Exercise 1 Program1 Test 1
Solve programming problems by writing Unix shell scripts and C programs (CS 1, 2, 3, 6)	Week 3, 5-end	Programs 1-3 Stringy, Dumpit Exercises
Apply secure programming techniques to harden existing C code against attacks by malicious users (CS 1, 2, 3)	Lectures on Secure Programming, Buffer Overflows	Program 2