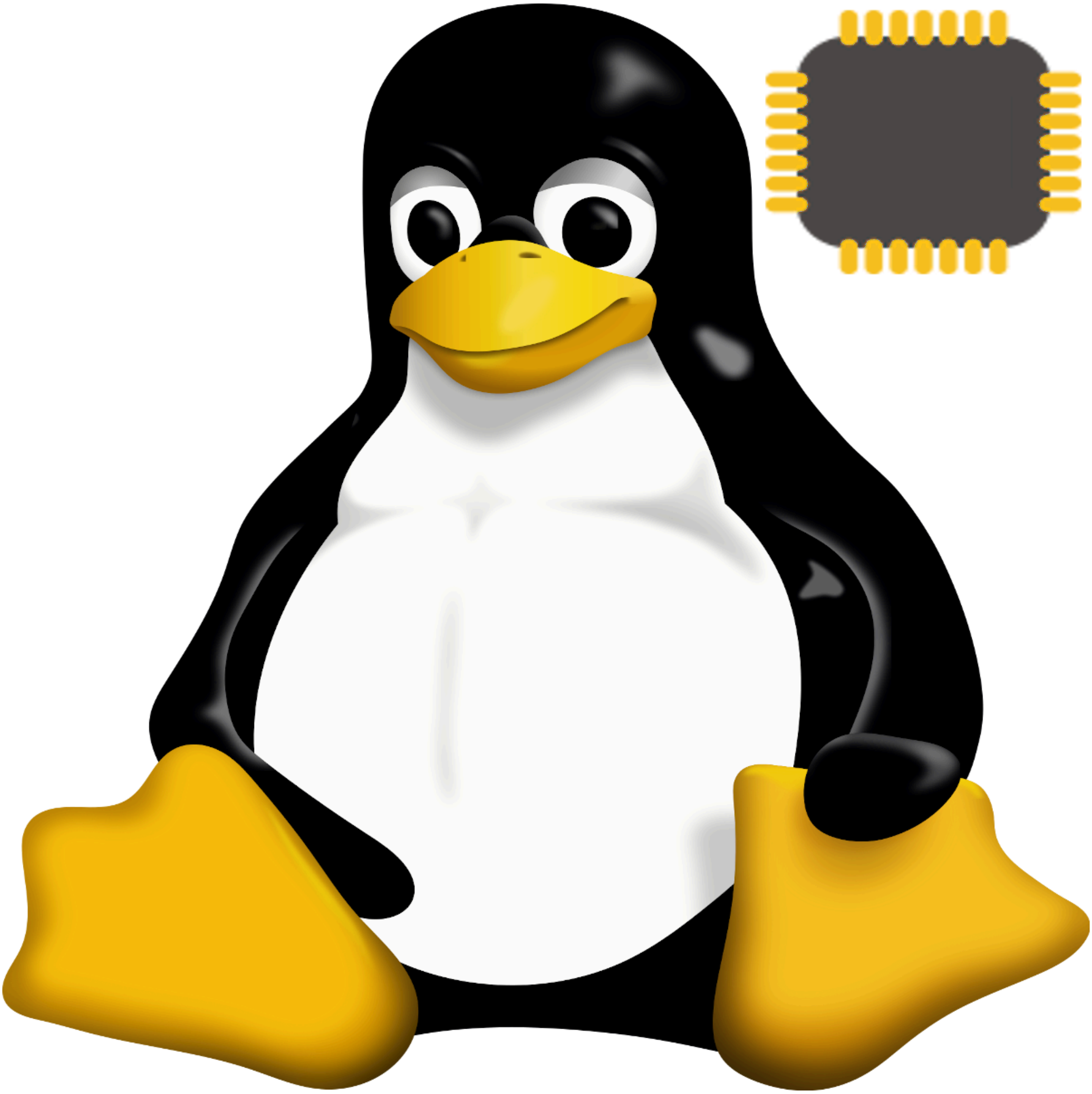
Linux System Programming and Introduction to Buildroot



by University of Colorado Boulder

About this Course

This course provides an overview of System Programming for the Linux operating system, or software which is interfacing directly with the Linux Kernel and C library. The basic components of a Linux Embedded System, including kernel and root filesystem details are discussed. The Buildroot build system is introduced, which students use to build their own custom Embedded Linux system through programming assignments. Linux System Programming and Introduction to Buildroot can also be taken for academic credit as ECEA 5305, part of CU Boulder’s Master of Science in Electrical Engineering.

Basic Info	Course 1 of 3 in the Advanced Embedded Linux Development Specialization
Level	Advanced
Commitment	4 weeks of study, 10-15 hours per week
Language	English, **Subtitles:** Arabic, French, Ukrainian, Chinese (Simplified), Greek, Italian, Portuguese (Brazil), Dutch, Korean, German, Russian, Thai, Indonesian, Swedish, Turkish, Spanish, Hindi, Japanese, Kazakh, Polish
How To Pass	Pass all graded assignments to complete the course.
User Ratings	4.3 stars Average User Rating 4.3

Syllabus

Module 1

Intro to Linux System Programming and Toolchains

In this module, you will learn about Linux shell scripting and System Programming concepts. You will also understand the basics of an Embedded Linux toolchain and setup a cross-compiler.

17 videos, 4 readings

1. **Reading:** Non-Credit Students: Welcome and Where to Find Help
2. **Reading:** Introducing Honorlock for For-Credit Exam Proctoring
3. **Video:** Linux Overview
4. **Video:** Linux Command Line
5. **Video:** Advanced Linux Command Line
6. **Video:** Scripts
7. **Video:** Assignment 1 Overview
8. **Reading:** Assignment 1 Instructions
9. **Video:** Course Introduction
10. **Video:** Course Format
11. **Video:** Licensing
12. **Video:** Environment Setup
13. **Video:** System Programming
14. **Video:** Linux Filesystems
15. **Video:** Processes and Threads
16. **Video:** Users and Groups
17. **Video:** System Programming and Error Handling
18. **Video:** Embedded Linux Toolchain Overview
19. **Video:** Logging and Syslog
20. **Video:** Assignment 2 Overview

21. **Reading:** Assignment 2 Instructions

Graded: Academic Integrity and Getting Help

Graded: Assignment 1: Bash Scripting Basics

Graded: Intro to System Programming

Graded: Assignment 2: File Operations and Cross Compiler

Module 2

File IO, Process Management, Basics of Embeddd Linux

In this module, you will learn the basics of File IO for Linux. You will understand the basics of Process Management, process forking, and Linux daemons. You will build your own Linux kernel and root filesystem, boot and run on QEMU.

11 videos, 2 readings

1. **Video:** Linux File I/O
2. **Video:** Linux File I/O: Reading and Writing Files
3. **Video:** Linux File I/O: Additional Topics
4. **Video:** Process Management
5. **Video:** Processes and Daemons
6. **Reading:** Assignment 3 Part 1 Instructions
7. **Video:** Assignment 3 Part 1: System Calls Overview and Demo
8. **Video:** Linux Kernel Overview
9. **Video:** Embedded Linux and Emulation Overview
10. **Video:** Building the Linux Kernel
11. **Video:** Linux Root Filesystems
12. **Reading:** Assignment-3 Part-2 Instructions
13. **Video:** Assignment 3 Part 2: Manual Linux Overview and Demo

Graded: File I/O

Graded: Processes

Graded: Assignment 3 Part 1

Graded: Assignment 3 Part 2

Module 3

Threads and Multithreading, Introduction to Buildroot

In this module, you will learn the basics of Threading and Multi Thread Synchronization in Linux system programming using POSIX. You will also learn about the Buildroot build system and build a QEMU Image using Buildroot.

10 videos, 2 readings

1. **Video:** Threads and Multithreading
2. **Video:** Synchronization
3. **Graded Assignment:** Threading
4. **Reading:** Assignment 4 Part 1 Instructions
5. **Video:** Assignment 4 Part 1: Threading Overview and Demo
6. **Video:** Introduction to Buildroot
7. **Video:** Buildroot Builds
8. **Video:** Buildroot Tips
9. **Video:** Buildroot Tips Part 2
10. **Reading:** Assignment 4 Part 2 Instructions
11. **Video:** Assignment 4 Part 2: Buildroot Demo
12. **Video:** Signals Overview
13. **Video:** Signal Management

Graded: Assignment 4 Part 1
Graded: Assignment 4 Part 2
Graded: Signals

Module 4

Time and Sockets

In this module, you will learn the fundamentals of signal handling and time management in Linux System Programming. You will learn the basic components of Embedded Linux debugging. You will implement a socket server application and deploy on a QEMU based Embedded System using Buildroot.

8 videos, 2 readings

1. **Video:** Time
2. **Video:** Sleeping and Timers
3. **Video:** Sockets
4. **Reading:** Assignment 5 Part 1 Instructions
5. **Video:** Assignment 5 Part 1: Native Socket Server
6. **Video:** Assignment 5 Tips
7. **Video:** Linux System Initialization
8. **Video:** Embedded Debug Strategies
9. **Video:** Assignment 5 Part 2: Embedded Socket Server
10. **Reading:** Assignment 5 Part 2 Instructions

Graded: Time
Graded: Assignment 5 Part 1
Graded: Assignment 5 Part 2

How It Works

General

What do start dates and end dates mean?

Once you enroll, you'll have access to all videos, readings, quizzes, and programming assignments (if applicable). If you choose to explore the content without purchasing, you may not be able to access certain assignments. If you don't finish all graded assignments before the end of the session, you can reset your deadlines. Your progress will be saved and you'll be able to pick up where you left off.

What are due dates? Is there a penalty for submitting my work after a due date?

Within a course, there are suggested due dates to help you manage your schedule and keep work from piling up. Quizzes and programming assignments can be submitted late without consequence. However, it is possible that you won't receive a grade if you submit your peer-graded assignment too late because classmates usually review assignment within three days of the assignment deadline.

Can I re-attempt an assignment?

Yes. If you want to improve your grade, you can always try again. If you're re-attempting a peer-graded assignment, re-submit your work as soon as you can to make sure there's enough time for your classmates to review your work. In some cases you may need to wait before re-submitting a programming assignment or quiz. We encourage you to review learning material during this delay.

Peer-graded assignments

Peer-graded assignments require you and your classmates to grade each other's work.

How do peer graded assignments work?

After you submit your assignment, you will review some of your peers' assignments. The number of assignments you must review is set by the instructor of the course.

I reviewed my peers' assignments! What happens next?

While you're reviewing your peers' assignments, they'll review yours. If you submit your assignment on time, you'll get your grade within a week, as long as at least one peer reviews your assignment. If you submit late, you'll need all of the peer reviews the instructor requires. Learn more about Peer Graded Assignments. ↗

How are grades calculated?

You and your classmates will be asked to provide a score for each part of the assignment. Final grades are calculated by combining the median scores you received for each section.

What kind of feedback should I give?

Use the instructor's criteria in the rubric to grade honestly and fairly. If your peers' answers are excellent, score them highly and tell them what they did well. If their answers aren't as good, give the score they deserve, and be sure to provide respectful, useful feedback ↗ so they can do better next time they attempt the assignment.

Is there a penalty for submitting my work late?

No, but it's important to submit your work as close to the due date as you can. Classmates grade most of the assignments within three days of the due date. If you submit yours too late, there may not be anyone to review your work.

If I fail an assignment, can I try again?

Yes! You can always try again, but you'll need to resubmit your work as soon as possible to make sure your classmates have enough time to grade your work.

Can I edit my assignment?

Yes, but you'll need to re-submit your work and any grade you've already received will be deleted.

Course 1 of Specialization

Deploy and Develop Software for Embedded Linux

Learn how to develop and deploy software for Embedded Linux based devices

Advanced Embedded Linux Development

University of Colorado Boulder

View the course in catalog

Linux Kernel Programming and Introduction to Yocto Project

University of Colorado Boulder

Linux Embedded System Topics and Projects

University of Colorado Boulder