ECE 331

Project 1 - Morse Code Kernel Module

See course web site for due date

Grade reduced 1% per minute after due date/time, rounded up to the next minute.

Items	Details	Value	Score
Deliverables **25%**	Morse Code Kernel Module Code		
	Printed using enscript	5%	
	4 space tabs	5%	
	Proper header with name on all pages	5%	
	Gitlab Repository		
	Code in git	5%	
	Reasonable commit comments	5%	
Design **50%**	Compiles	5%	
	Loads	5%	
	Does not cause a panic nor fault	5%	
	Test suite works/passes, Grader test suite passes	10%	
	Proper locking	10%	
	Commented	5%	
	Errors checked and handled correctly	5%	
	Organized data representation	5%	
Report **25%**	Format correct	5%	
	Engineering Style	5%	
	Grammar, spelling, and sentence structure	5%	
	Technical details correct	10%	
	TOTAL	100%	

Description

For this project, write a kernel module so that it blinks an LED on the ECE 331 S16 RPi expansion board. Use the framework code on gitlab. You may clone the repository.

Incorporate your morse code code into the kernel module. That is, a user space program should pass a string to the kernel module. All conversion and encoding should be done in kernel space.

This module should allow one or more user space processes to open and write to the morse device special file but no data corruption should occur. Once a transfer is started from one user space process, complete the transfer before operating on data from another user space process. User space processes flagged as non blocking should return without encoding any data with an appropriate error.

Once the kernel module is written, write one or more C programs (may also have perl or python helpers) that test the functionality of your kernel module. It should spawn multiple processes to sufficiently prove that your module works, is robust, and thoroughly tested. Thoroughly document your driver.

Comment all of your code and scripts. Use ECE's git repository to manage revisions for all of your code. Include the instructor of the course as a "Developer" member. Name the repository "kmorse".

ECE 331 S16 Expansion Board Interface

Two GPIO pins (GPIO4 and GPIO17) are used to control Morse Code generation. GPIO4 is the active low enable pin. Pulling this line low enables the ECE 331 LED blinker. GPIO17 is encode Morse Code data. The encoding works just like in homework 5.

Report

Write a concise one page report on the purpose, design, and testing of locking within the kernel module. The report shall have

- Margins of: left 1 inch, right 1/2 inch, top 1 inch, bottom 3/4 inch with a footer.
- The first page shall begin with your name, date, and course number, centered, one item per line, bold in 14 point Times New Roman or Nimus Roman No9 L font followed by the Introduction section
- 12 point Times New Roman or Nimus Roman No9 L font,
- Double spaced
- Page numbers in the footer
- Three sections using the following headings: A very concise introduction, Locking: Purpose, Design, and Testing, and the last section a concise Conclusion.
- Engineering style. Engineering style is in the third person. It avoids slang, contractions and sentences that contain "there is/was/were/are". Word are chosen carefully for the meaning. Design decisions are supported by engineering design reasons.
- One complete and full page.
- The term (or any of its forms) "implement" shall not appear anywhere in the document.

Code in any form is NOT allowed in this document.

Deliverables

At the due date, turn in your

- Report
- Code committed in gitlab

Grading

Grading will be done by the rubric given on the first page.