ECE 331

Homework 6

See course web site for due date

Place your typed homework answers in `vim`. Print single sided with your name using a **mono space font**. No need to restate questions. Fully investigating questions is required for a higher grade. Please use the kernel coding style for all code. Please use your RPi for developing answers. Although code should be written and run on a RPi, it should run on ANY POSIX compliant OS. As always, all code shall be comment, conform to the Linux Kernel Coding Style, and error conditions shall be checked and appropriately handled.

NO PARTIAL CREDIT FOR QUESTIONS 1-6.

1. Write a complete Makefile to compile the source files `trump.F`, `spicer.c`, `kellyanne.s`, and `pence.CC` into an executable named `thebest`.

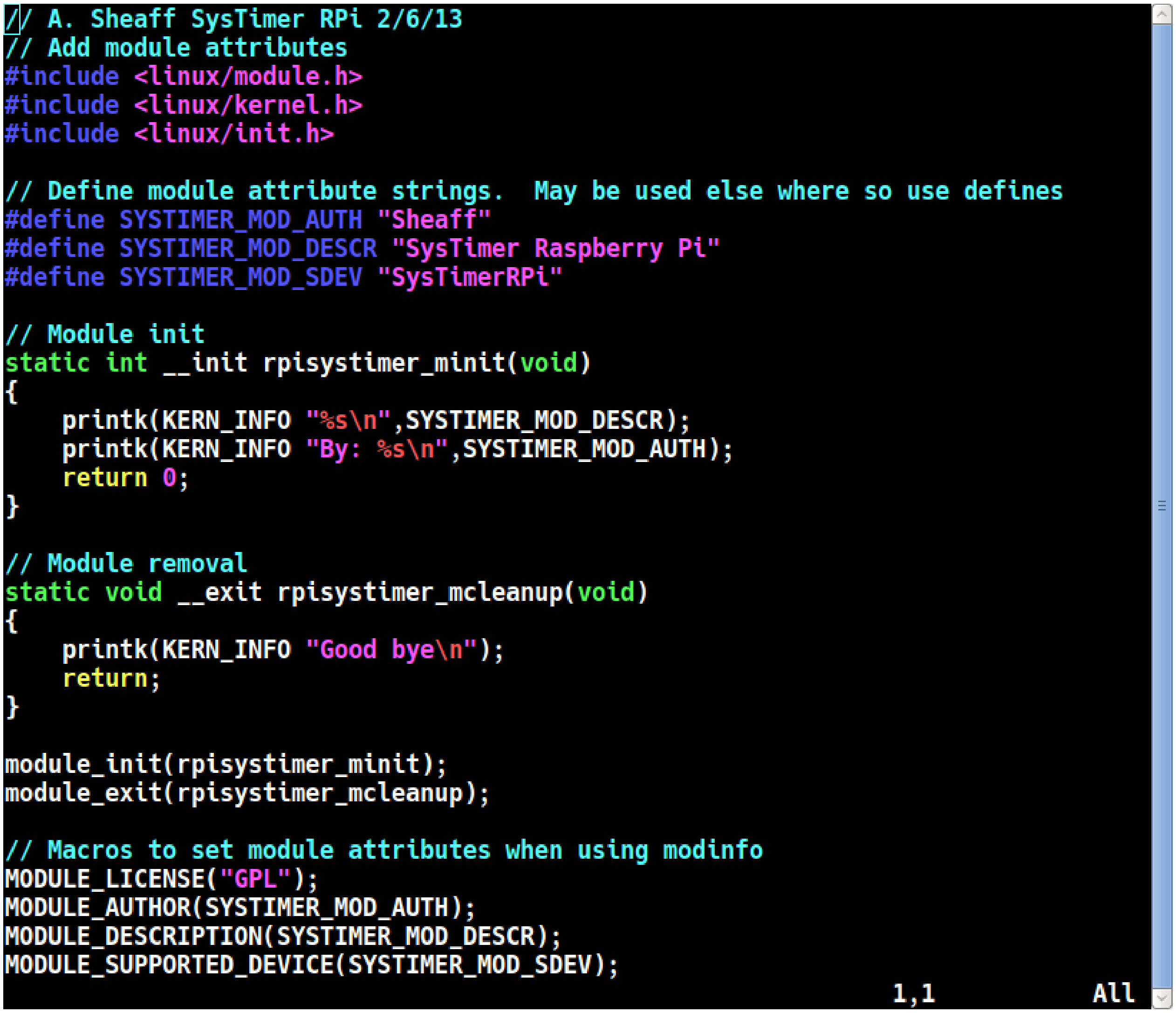
2. Write a concise perl script (no external commands) that parses all lines in `/etc/passwd` file (documented at `passwd(5)`). From the parsed `passwd` file data, have a single variable that associates each user's username with each user's shell for all users. After parsing the entire file, print all usernames and shells, one username and shell per line.
   For your answer, include the perl script.

3. Give a perl style extended regular expression validates, but does not interpret, a `passwd` file entry (one line). The `passwd` file uses encrypted passwords. Validate ONLY the first four fields.

4. Write a C code that prints the resident set size of the running program itself by calling `getrusage()`. For your answer, include the C program source code.
5. Random command line. Give concise command(s) and assume that completes disabled.
   a) Give the concise vim command to delete a word at the current cursor location.
   b) Give the concise command to display lines 20 to 30, inclusive, of a file named `morse` in the directory `/usr/local/project/from/lleh` without changing directories. The current working directory is `/var`.
   c) Give the fewest and shortest command(s) that will create a link to an existing file named `snow`. The file `snow` is located in `/usr/share/jon`. Name the link `$throne` and place it in the current directory. The current working directory is NOT `/usr/share/jon`. The current working directory is on a different filesystem than `/usr/share/jon`.
   d) All files, directories, and subdirectories in the current directory have incorrect group and other permissions. Without modifying the user's permissions, add read permission for group and other, and, as appropriate for the filetype, execute permission. The user's execute permissions are already set appropriately for the filetype. Give the command to accomplish the task.
   e) Without using the '-c' or '--count' flag, determine the number of sleeping processes. Give the command to complete the task.
   f) Give all the commands needed (complete) to install the Atari 2600 Emulation system.

6. Mystery Hunt! Give **all successful** commands that logically lead to the next step to get your message. Give the message. Copy the starting file from the flash drive. Start with placing the file in a writable empty directory. You may request the next command for a reduction of 10% on this problem for each request. When you find your message, you will know. When writing commands, **assume your shell does NOT have completes.**
7. Be sure your kernel is ready for compiling kernel modules.
   a) Write a kernel module Makefile. Include the makefile for your answer to this part.

b) Enter the following kernel code, compile it, and make sure that it loads into your running kernel on your RPi. Fix any and all errors. For this part, indicate that is have been successfully completed. Project 1 depends upon this working. Be sure to get it working.



```c
// A. Sheaff SysTimer RPi 2/6/13
// Add module attributes
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

// Define module attribute strings.  May be used else where so use defines
#define SYSTIMER_MOD_AUTH "Sheaff"
#define SYSTIMER_MOD_DESCR "SysTimer Raspberry Pi"
#define SYSTIMER_MOD_SDEV "SysTimerRPi"

// Module init
static int __init rpisystimer_minit(void)
{
    printk(KERN_INFO "%s\n",SYSTIMER_MOD_DESCR);
    printk(KERN_INFO "By: %s\n",SYSTIMER_MOD_AUTH);
    return 0;
}

// Module removal
static void __exit rpisystimer_mcleanup(void)
{
    printk(KERN_INFO "Good bye\n");
    return;
}

module_init(rpisystimer_minit);
module_exit(rpisystimer_mcleanup);

// Macros to set module attributes when using modinfo
MODULE_LICENSE("GPL");
MODULE_AUTHOR(SYSTIMER_MOD_AUTH);
MODULE_DESCRIPTION(SYSTIMER_MOD_DESCR);
MODULE_SUPPORTED_DEVICE(SYSTIMER_MOD_SDEV);
                                                            1,1           All
```