COMP 530H
Fall 2014

Honors Assignment 1
Module to implement 'getpinfo'
Date assigned: September 5, 2014
Date due: September 12, 2014

Modify and extend the 'getpid' example program described in class to create a new service module (and calling user program) called 'getpinfo' that retrieves additional information from the task_struct of the calling process. The added information should include the PID of the real parent of the process along with the process state, flags, and normal priority,. All the information returned to the calling program should be formatted as one character string the caller can display on stdout and have it look similar to the following:

Current PID 30571
  parent 30570
  state 0
  flags 00402100
  priority 120

Note that fields that represent multiple bit-coded values such as flags are to be displayed in hexadecimal. Your module should deal with any errors that can occur (but not be burdened with tests for errors that cannot occur). The calling program should receive sufficient information so that it can determine if the call has been successful or has failed.

Recall that task_struct is defined in the include file <linux/sched.h>. You will need to make extensive use of C functions for manipulating strings. Fortunately most of the familiar functions from the C library are implemented in the kernel. The API is documented at: https://www.kernel.org/doc/htmldocs/kernel-api/libc.html.

Submitting your program:

The materials you submit for your programming assignments should be handled as follows.

- Create a directory in your AFS home-directory space and give me (login 'smithfd') read and lookup permissions on the directory. If you are not familiar with AFS directory permissions, from the CS department home page, navigate to Computer Services Help → Help Articles → All Help Articles and read article AFS File Security.
- Place all the submitted materials into the directory. The submitted materials should include all source files ready to use with 'make' to create a module and to compile the calling program with gcc.
- Include a README file that provides ***complete*** instructions for compiling and running the module and calling program.
- Send email to me giving the pathname for the directory containing your submitted materials. The later of (1) the date on that email message, and (2) the latest file-modified time in the directory determines the submission date for purposes of deciding if the assignment is late. WARNING: this means that if you change a file after submission it may be considered late.