Operating Systems - Winter 2018

Sambuddho Chakravarty

January 27, 2018

Assignment 1 (Total points: 45)

Due date: February 11, 2018. Time: 23:59 Hrs.

A Simple System Call

As described in the class, the OS provides operations to users via system calls. System calls are functions, natively available to users via C library functions, provide access to almost all OS level functionality – *viz.* `open()`, `write()`, `fork()` *etc.*

You have to create your own system call in C, called `sh_task_info()`, which takes argument as PID. It would need to search out the `task_struct()` corresponding to the PID and print out all the fields corresponding to it and also save it in a file. The file name also needs to be supplied as an argument to the system call.

You also would require to handle errors in user inputs, such as incorrect arguments, through appropriate `errno` and function return values (*e.g.* `0` signaling correct input, while `1` signaling incorrect input).

We are going to use Ubuntu 14.04 LTS server as the base distribution running kernel version `3.13.0`. You require to download the kernel source from Ubuntu repositories, make the appropriate changes and generate the compiled OS, supporting your changes. We downloaded the above source from the Ubuntu repository directly using the `apt-get` command. You may use the following for references:
-`https://help.ubuntu.com/community/Kernel/Compile`
-`https://wiki.ubuntu.com/Kernel/BuildYourOwnKernel`.

What To Submit

- You need to submit the `diff`, of the originally downloaded kernel source tree and the one with your changes. We would patch that `diff` with our copy of the original source tree and compile the kernel
- Write-up describing the following:
  - Description of your code and how you implemented the function – the logical and implementation detaisl.
  - The inputs the user should give.

  - Expected output (and how to interpret it).
  - Error values and how to interpret them.

- A sample C program to test out your implementation of the system call.

Grading Rubric

- Successful compilation your diff against the base kernel source – 10 points.
- Correct functioning of your system call, testable through the supplied C program – 20 points.
- Correct handling of input errors (atleast two different types of errors should be handled) – 10 points.
- Description of the systems, how to test the system through your supplied C program and what to expect *etc.* – 5 points.

Late Submission Policy

- Submitted on or before February 11, 2018 (23:59 hrs) – No points deducted.
- Submitted after February 11, 2018 but on or before February 13, 2018 (23:59 hrs) – 5 points deducted.
- Submitted after February 13, 2018 but on or before February 15, 2018 (23:59 hrs) – 15 points deducted.
- Submitted after February 15, 2018 – no points shall be awarded.