Operating Systems - Winter 2018

Sambuddho Chakravarty

March 28, 2018

Assignment 3 (Total points: 90)

Due date: April 12, 2018. Time: 23:59 Hrs.

1 Multiple reader writer with synchronization (Total points: 45)

You are already aware of the multiple user reader writer problem wherein multiple "writers" can write to a shared queue and multiple "readers" can read data off the shared queue. You task is to implement the multiple reader writer problem in such a way that there are no race conditions. While a reader "writes" a certain element to the queue, no "reader" can read that element. The readers may read other elements but not the one current written to. Similarly, only one "writer" can update the queue at a time. Multiple "readers" may read elements off the queue but dequeuing from the queue by the "readers" must be atomic.

You may use various system synchronization primitives for the same, *e.g.* `pthread_mutex`es, `semaphores` *etc.*. You may additionally also use IPC primitives which were used in the previous assignment, if you wish to.

What To Submit

- Program source code with Makefile
- Write-up describing the following:
  - Description of your code and how you implemented the function – the logical and implementation details.
  - Description of how to compile and test the program
  - The inputs the user should give.
  - Expected output (and how to interpret it).
  - Error values and how to interpret them.

Grading Rubric

- Successful compilation your the program – 10 points.
- Correct functioning of the chat system for multi-users – 20 points.

- Correct handling of input errors (atleast two different types of errors should be handled) – 10 points.
- Description of the systems, test cases *etc.* – 5 points.

2 System Bootup and Switching to Protected Mode (Total points: 45)

You need to implement a simple bootloader, which would be booted via UEFI (or legacy) BIOS. The bootloader would reside in the USB pen-drive. You would need make sure that the BIOS boots up the bootloader residing in the pen-drive, which would thereby switch the CPU to protected mode (64-bit mode). You would need to thereafter print "hello-world", and the contents of the CR3 register via the video buffer.

What To Submit

- Source code of the program which you have submitted and the binary, which you boot via the USB pen-drive.
- Description of your code along with the logical and implementation details.

Grading Rubric

- You would require to superimpose the **submitted** bootable image on pen-drive and demonstrate the boot-up with the expected outcomes – 35 points.
- Correct outcomes, *viz.* the "hello-word" and the CR3 register contents – 10 points.

Late Submission Policy

- Submitted after April 12, 2018 – no points shall be awarded.