UNIVERSITY OF THE WEST INDIES

Department of Computing
COMP3101 – Operating Systems (Semester I, 2020)
Lecturer: Dr. Kevin Miller

FINAL PROJECT:
Design an animated simulation of an operating system concept.

The aim of this project is to show how an OS works, in a visual manner with proper pacing that allows the viewer to follow what is happening, with the aim of helping anyone who find it difficult to see/visualize how the written and theoretical OS material really works. Rather than show all complexity simultaneously, the focus will be on one subsystem at a time while showing only the highest conceptual view of the others, i.e. keeping the others in the visual/perceptual background.

Each group (Max: 5) chooses one aspect of the OS as listed below:

The Kernel Process Handling System

This should include, the process state model diagram, context switching, kernel vs user mode. Your visualizer should show every change between kernel and user state using noticeable visual cue.

Virtual Memory Subsystem

You visualizer should have the ability to show page replacement strategies, VM to physical address translation, and including the TLB and page cache.

I/O System

How Interrupts, DMA, and device controllers/drivers work and interact with the OS and process blocking. For example, when a logical I/O call is made by a process, show how a call goes through the O/S, the device driver, the controller, and work DMA.

OS Architecture

How OS architectures work in general in terms of system calls to kernel to physical machines, and how different OS architectures vary on this, e.g. monolithic and microkernel, and their advantages/disadvantages in terms of performance.

Threads and Processes

Kernel-level threads and user-level threads and animations that show their advantages and disadvantages in terms of context-switching time, blocking, I/O, etc, and how they relate to and work with processes.

Process Scheduling Algorithms

Simulate at least four (4) process scheduling algorithms

Synchronization

Simulations of race conditions, Semaphores, classic algorithms that work and those that don’t work.

The animated simulation must flexible, i.e. there must be the ability to change the parameters of the visualizer. The visualizer should be accompanied by a design document.

You can decide on the language(s) to use for this project. Use a language that is familiar with most if not all of the members of the group. The code should be properly documented.

The following seems to be good choices:

- Python and PyGame (others are available)
- Java (there are others than the default)
- HTML5/Javascript, CSS (many animation libraries)

Since this is a group work, you must use the necessary software tools that supports collaborating on documents and source code eg. Google Docs and GitHub.

DEADLINE: November 30, 2020 @ 11:59 PM

DEMO: TBA