**Computer Science & Information Systems**

**Lab Sheet - Introduction to CPU-OS Simulator**

1. **Objective:**

At the end of this lab session student will be able to:

* Install CPU-OS Simulator
* Understand three components of CPU-OS simulator

1. **Introduction :**

CPU-OS Simulator is a free program that enables you to run programs created manually.  It has a typical set of CPU instructions (not specific to any commercial CPU), addressing modes and a configurable register set.

The CPU-OS simulator can be installed and run on MS Windows based (XP, Vista, 7) system.

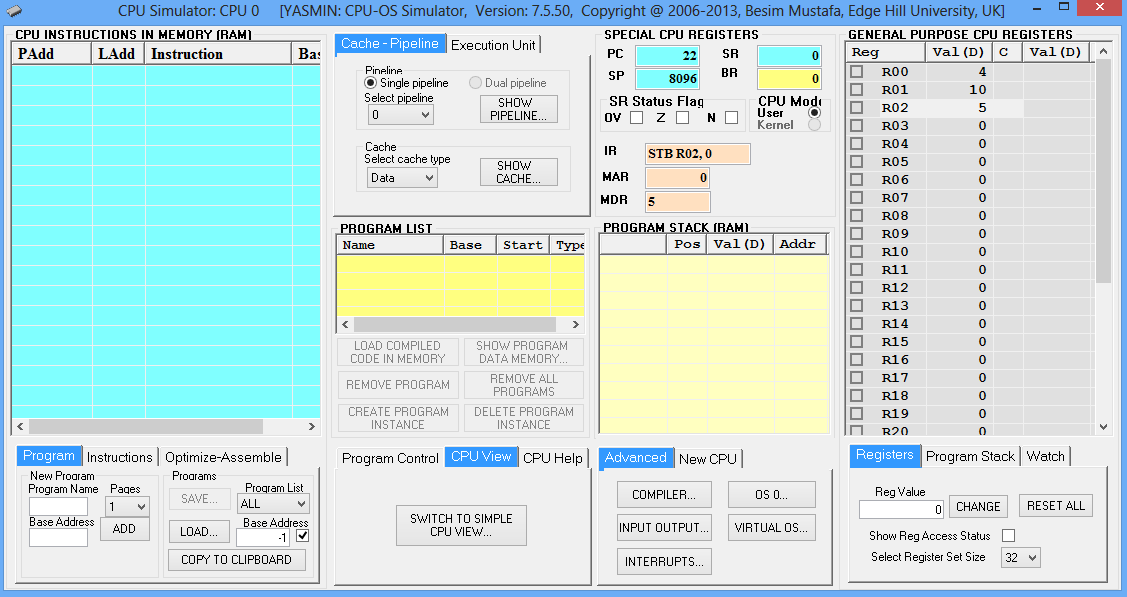
Link to down load CPU-OS Simulator Version 7.5: <http://www.teach-sim.com/>

Three components of CPU-OS simulator are:

* The CPU Simulator
* The OS Simulator
* The Compiler

**The CPU Simulator:**

When the CPU-OS simulator is started, it will be in CPU simulator mode. Full screen appearance of CPU-OS Simulator version 7.5 in simple CPU simulator mode is as shown in Figure 1. The CPU simulator is based on RISC architecture with register file comprising of 32 registers, variable length instructions and a limited number of addressing modes. CPU simulator also includes cache and pipeline. Cache placement and replacement policies can be selected. CPU simulator also provides list of vectored interrupts.

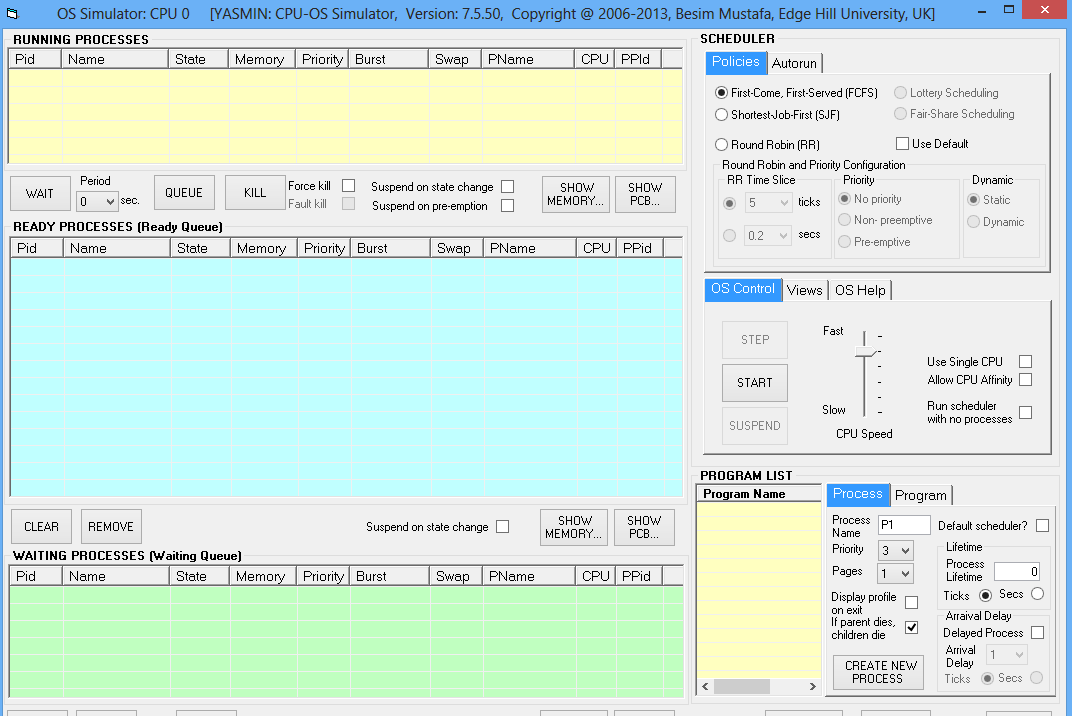
****

**Figure 1: CPU-OS Simulator Version 7.5 in simple CPU simulator mode**

**The OS Simulator:**

The OS simulator supports two important aspects of a computer system’s resource management: process management and memory management. Figure 2 shows the main user interface for OS simulator. Important features of OS Simulator is

* Once a compiled code is loaded in CPU memory, its image is also available to the OS simulator. It is then possible to create multiple instances of the program images as separate processes.
* The OS simulator displays the running processes, the ready processes and the waiting processes. Each process is assigned a separate process control block (PCB) which contains information on process state. This information is displayed in a separate window.

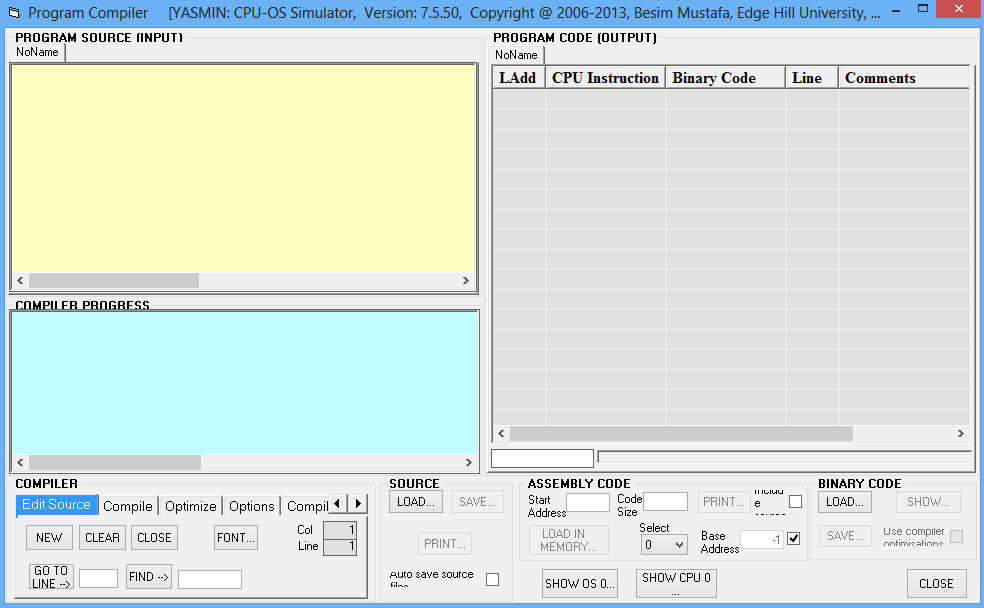
****

**Figure 2: CPU-OS Simulator Version 7.5 in simple OS Simulator Mode**

* The memory display demonstrates the dynamic nature of page allocations according to the currently selected placement policy. The OS maintains a separate page table for each process which can also be observed.
* The simulator demonstrates how data memory is relocated and the page tables are modified as the pages are moved in and out of the main memory illustrating virtual memory activity.
* The process scheduler includes various selectable scheduling policies which includes priority‐based, pre‐emptive and round‐robin scheduling with variable time quanta.
* The OS is able to carry out context‐switching which can be visually enhanced by slowing down or suspending the progress at some key stage to enable the students to study the states of CPU registers, stack, cache, pipeline and the PCB contents.
* The simulator incorporates an input output console device which is used to display text and accept input data.

**The Compiler :**

CPU-OS simulator provides a basic but complete high‐level teaching language to support the CPU and OS simulations. This language supports control structures, constructs and system calls. A compiler generates both assembly‐level language and its equivalent binary byte‐code as output. Figure 3 shows a snapshot of the main compiler user interface. The compiler includes code optimizations, support for profiling, display of compiler stages and the binary code generated as well as some statistical data. The compiler also includes an integrated tabbed source editor capable of handling multiple source code at the same time.



**Figure 3: CPU-OS Version 7.5 in compiler mode.**