

Gebze Technical University  
Department Of Computer Engineering  
CSE 312 /CSE 504 Spring 2021  
Operating Systems  
Homework #01  
Due Date: 05.04.2021

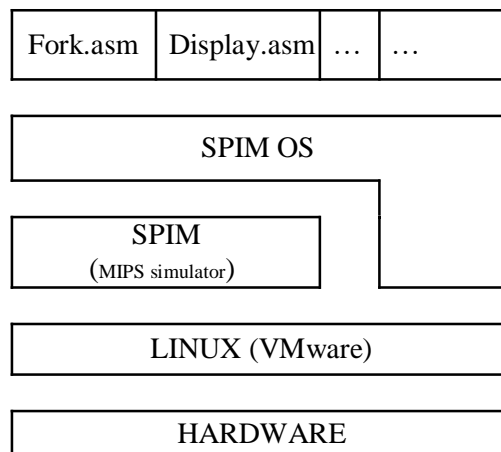
Throughout the semester, we will try to write our operating system that runs on a MIPS CPU that you are already familiar with from the CSE 331 course. The MIPS CPU will be simulated by the open source package SPIM (<http://spimsimulator.sourceforge.net/>, full source code

<https://sourceforge.net/p/spimsimulator/code/HEAD/tree/>). From the SPIM documentation

*“Spim is a self-contained simulator that runs MIPS32 programs. It reads and executes assembly language programs written for this processor. Spim also provides a simple debugger and minimal set of operating system services. Spim does not execute binary (compiled) programs.”*

The operating system that comes with SPIM is very primitive. It only supports a few trivial system calls, such as reading and printing, which are defined in /CPU/syscall.h .

The position of our SPIM OS is shown in the main software architecture diagram below, which indicates that our SPIM OS will be implemented on both SPIM and LINUX. In other words, some of our OS packages will be in MIPS assembly and some of them will be in C++.



**Figure 1: System architecture**

For this homework, you will implement 3 regular MIPS assembly programs, which can run with the simple SPIM OS without any modification. You will write another MIPS assembly program that needs a new OS service, which will be implemented by you as part of this homework.

First, download and study the SPIM package very carefully from the above link. Then, you will write and test the following MIPS assembly files. Use the provided sample assembly files to learn about how to use the MIPS simulator.

1. ShowPrimes.asm: file that prints all integer from 0 to 1000 on the screen if the number is prime than it will print “prime” next to number. Each number will be printed on a new line.
2. Factorize.asm: Given an integer from keybpard, your code will find and show integer factors of it.  
For example: factors of 20 is 1,2,4,5,10,20
3. BubbleSort.asm : your code will sort the given integers in increasing order by implementing selection sort algorithm.
4. Shell.c: You will write a shell program that will be very similar to the example given in the textbook in Fig 1-19. Note that, our current SPIM OS does not support fork, waitpid, and execve system calls, so you will have to implement some system services for this shell. Instead of implementing these three system services, you will implement the Windows system procedure call CreateProcess which loads an assembly file from the disk and executes it. After this call is processed, the OS returns to the caller. Note that Shell.c is a C program which needs to be compiled by a MIPS compiler. Read the SPIM documentation for the compilers. You may also have to write some helper assembly programs for the shell.

You will have to change the SPIM source code in two places: /CPU/syscall.h and /CPU/syscall.cpp . Do not modify any other SPIM code other than these files.

Below are the instructions for homework submission

1. Download and Install Vmware Player from official site.
2. Download and install our virtual machine from  
[https://drive.google.com/open?id=1YppX3lNkyTsHV\\_lvA4w9TomNCUkpLeEg](https://drive.google.com/open?id=1YppX3lNkyTsHV_lvA4w9TomNCUkpLeEg)
3. Carefully examine SPIM simulator and MIPS instruction set.  
[https://web.stanford.edu/class/cs143/materials/SPIM\\_Manual.pdf](https://web.stanford.edu/class/cs143/materials/SPIM_Manual.pdf) is a very good SPIM documentation.
4. Your submission includes only the files below
  - ShowPrimes.asm
  - Factorize.asm
  - shell.c (your original C program that needs to be compiled)
  - shell.asm (your shell program that needs to be produced by the MIPS compiler)
  - shellHelp.asm (your helper assembly program you may need to write)
  - syscall.cpp
  - syscall.h