

Gebze Technical University
Department Of Computer Engineering
CSE 312 /CSE 504
Operating Systems

Homework #01
Due Date: March 1th 2018
8080 CPU Emulation and Simple OS

Homework Instructions

1. Download and Install Vmware Player from official site.
1. Download and install our virtual machine from https://drive.google.com/open?id=1YppX3INkyTsHV_lvA4w9TomNCUkpLeEg
2. Carefully examine 8080 cookbook and instruction set.
3. Use the assembler at <http://sensi.org/~svo/i8080/> to develop your assembly code. You can download the output .com file directly using this assembler.

Your homework template includes several code and sample files. Here is a description for them. Do not change Emulator Files and do not send these files with your homework. Study these files to understand the 8080 assembly code and the emulator.

8080emu.cpp : Emulator implementation for 8080. You will not change this file

8080emuCPP.h: Emulator interface for 8080. You will not change this file

Memory.h and memoryBase.h : You will not change these files

gtuos.h : sample header for GTUOS. You will rewrite this file

gtuos.cpp : sample implementation for GTUOS. You will rewrite this file

main.cpp : sample main function for this HW. You will rewrite this file

sum.asm: Adds numbers from 0 to 10 and prints the results on screen.

sum2.asm : Adds numbers from 0 to 10 and prints the results on screen using a different system call.

printstr.asm : 8080 Assembly file that prints "Hello world" on screen.

Homework Tasks

There are two main tasks for you to do in this homework. First, you need to implement a number of GTUOS systems calls as described below

Call	Params	Function	Cycle
PRINT_B	REGISTER A =4	Prints the contents of Reg B the screen as decimal.	10
PRINT_MEM	REGISTER A =3	Prints the of contents of memory pointed by Reg B and C as decimal	10
READ_B	REGISTER A =7	Reads an integer from the keyboard and puts it in to Reg B	10
READ_MEM	REGISTER A =2	Reads an integer from the keyboard and puts it in to address Reg BC	10
PRINT_STR	REGISTER A =1	Prints the null terminated string at the address pointed by B and C	10 per char
READ_STR	REGISTER A =8	Reads the null terminated string from the keyboard and puts it at the memory location pointed by B and C	10 per char

You will rewrite the two files (gtuos.h, gtuos.cpp) to implement the above system calls and you will write another file (main.cpp) to use your OS with the 8080 CPU. Use the sample main file as a guide. **Do not touch original emulator code files.**

You will also write and test the following assembly files. Use the provided sample assembly files to learn about how to use the assembler and how we call the OS.

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, your code will find and show integer factors of it. For example: factors of 20 is 1,2,4,5,10,20
3. Palindrome.asm: Show whether a string given from keyboard is a palindrome by printing the "string" semicolon: "Palindrome" or "Not Palindrome"
Ex:
ada: Palindrome
arthas: Not Palindrome
4. **Expected Files:** gtuos.h, gtuos.cpp ,main.cpp and Assembly files above
5. **Please carefully inspect the guidelines!!**

General Homework Guidelines

1. No cheating, No copying, No peaking to other people homeworks
2. Follow the instructions very carefully.
3. Send required files only. Do not share your whole file system with us.
4. If you fail to implement one of the requirements, leave it be. Do not send an empty file
5. When homework mentions about reading from keyboard it means that we are reading from a file called **"input.txt"**.
6. All your outputs should be written into a file called **"output.txt"**. Whenever we talk about printing to screen we are referring to writing to **"output.txt"**
7. Respect the file names! Our HW grading is case-sensitive.
8. Failing to comply any of the warnings above will result in getting a **0** for your current homework.
9. You should use the shared virtual machine to build your code in order to achieve consistent behavior. Since we will grade your assignments on top of this virtual machine
10. You can get in touch with TA for any misunderstandings or confusions about assignments.