CSCE 231/2303 Spring 2021
Term Project: Phase 1

Assigned: Tuesday, 30-3-2021

Due: Tuesday, 6-4-2021

Delayed submission with penalty until Thursday, 8-4-2021

Goals

This is the first phase of your Term Project that you will work on it **individually**. The goal of this phase of the project is to build the first and second stages of the boot loader using the skeleton provided to you and presented to you in the project environment setup tutorial.

Details

This first phase of the project you will use the skeleton provided to you and that you can download from black board. You are allowed to use code from the slides to complete this phase of the project. You are required to build all the missing code in the skeleton to build the first stage and second stage boot loader and to land on the third stage boot loader.

You are required to write all the missing code in the first stage bootloader located in first_stage.asm and the folder includes/first_stage. Use the code in the slides to build that to load from disk the second stage and the third stage boot loaders. You should jump then to the second stage boot loader. Your boot loader should be able to boot for a virtual floppy as well as a virtual hard drive. You will need to be able to detect the type of device you have booted from, and you need to read the device parameters if you are booting from a hard drive. You will need to traverse all the files under the sources/includes/first_stage folder and add the necessary code whenever you find a comment that reads **"; This function need to be written by you."**

Next, you are required to work on the second stage boot loader within the skeleton code tree. Your main entry point is sources/second_stage.asm and you will add your code in the files included from the directory sources/includes/second_stage within your skeleton code tree. Your job is to add all the necessary code for checking the A20 gate and enabling it if needed, checking the support for the CPUID instruction, check if Long mode is supported, and scan the available physical memory.

You should print a message to the user before each step and wait for a key stroke from the user as an indication to start the execution of step. After each step, you should also print a message indicating failure or success of the step that has been just performed. Moreover, you are required to print all the scanned memory regions details on to the screen; mainly, start address, length, and type.

You still have the privilege of using and borrowing the code presented in the slides and explained in class to complete your mission, but again as the slides stated, it is very important to understand the code that you are going to borrow from the slides as it will build up, and hence missing the basic concepts at this level will make it difficult in the next stages to build the needed functionality.

You will need to traverse all the files under the sources/includes/second_stage folder and add the necessary code whenever you find a comment that reads " **; This function need to be written by you.**".

Finally, you should write a very small piece of code in your third stage boot loader in third_stage.asm to print a message indicating that you were able to jump to it. You need finally to jump to the third stage boot loader and then halt. Notice and be very careful that the third stage boot loader is located in a different segment which will required adjusting the appropriate selectors.

Important: the provided boot loader written and documented thoroughly will only work from within QEMU. It will definitely not work on some of the native bare metal hardware. There is something missing, that we discussed in class, in the provided boot loader that you are requested to investigate and add. After applying the amendments, you need to create a bootable flash drive with your new boot loader and boot a real computer from it.

What to submit

1. Your full in-line documented second stage assembly code for all code added to the code tree by you. **It is very important to highlight that you are not allowed to copy the code documentation presented in the slides, you need to explain the code in your own words. If you copy the documentation you will get ZERO in the assignment.**
2. All the skeleton code with you updates must be submitted on black board.
3. A PDF report that includes:
    a. A detailed description of your any assumptions you have made.
    b. List all findings that you have come up from doing this assignment.
    c. The steps needed to run your code.
4. A readme file indicating how to compile and test your code.
5. A contribution document that lists in details the contribution of each member in the group.

How to submit:

Compress all your work: source code, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Section_Number>_<ID>_Term_Prokect_Phase1.zip. Finally, upload your code to blackboard.

Grade

This assignment is worth 5% of the overall course grade. The assignment will be graded on a 100% grade scale, and then will be scaled down to the 5% its worth. The grading of the assignment will be broken down as follows:

1. 10 % for just submitting a meaningful assignment before or on the due date. This 10% does not account for the correctness of your assignment but submitting an empty assignment without code will definitely results in loosing this 10% and consequently the whole grade of this assignment.
2. 65 % for the correctness and the quality of your code.
3. 25 % for the quality of your inline documentation, the report, and the readme file.

Delays

You have up to 2 working days of delay, after which the assignment will not be accepted and your grade in that case will be ZERO. For every day (of the 2 allowed days), a penalty of 10% will be deducted from the grade. And of course, you will lose the 10% mentioned in point 1 above under the "Grade" section.