

# CSCE 231/2303 Fall 2018

## Assignment 7: A20 Gate, Check Long Mode, and Scan Physical Memory

Assigned: Wednesday, November 7<sup>th</sup> in Class

Due: Wednesday, November 14<sup>th</sup> at 5:00 pm

Delayed submission with penalty until Friday, November 16<sup>th</sup> at mid-night.

### Goals

This assignment is an individual assignment and you will work on it on your own. The goal of this assignment is to write the necessary x86 assembly code of the second stage boot loader that will check for the prerequisites needed to be able to switch to the x86\_64 Long mode. Precisely, you will check the A20 gate and enable it if needed, check for the support of CPUID instruction, check for LM-bit, and finally scan the physical memory installed in your virtual machine. By the end of the assignment you should be ready to build your first page table and switch to 64-bit Long Mode, which will obviously be the deliverables of your next assignment.

### Details

In this assignment you will 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.**”.

### 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 came up from doing this assignment.
  - c. The steps needed to run your code.
4. A read me file indicating how to compile and test your code.

### How to submit:

Compress all your work: source code of full skeleton source tree, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Student\_ID>\_<Name>\_Assignment7.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 each 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.