CSCE 231/2303 Spring 2021
Term Project - Phase 2: Memory Page Table and Long Mode

Assigned: Thursday, 15-4-2021

Due: Monday, 3-5-2021

Delayed submission with penalty until Wednesday, 5-5-2021

Goals

This is the second phase of your Term Project that you will work on. The goal of this assignment is to write the necessary x86 assembly code of the remaining part of the second stage boot loader that will build a memory page table and then switch to the x86_64 Long mode. By the end of this assignment, you should be running 64-bit Long Mode from your third stage boot loader and having access to your full physical memory and being able to start scanning installed PCI devices and setting up your PIT timer which will take place in your next project phase.

Details

In this assignment you will work on the second and third stage boot loader within the skeleton code tree. You should build on top of the work you have done in the first phase. Initially you can use the code presented in the slides to map the first 2 MB, before you switch your processor to long mode. After being in long mode you will need to jump to your third stage boot loader code where you need to write the necessary code to map all your available physical memory (Memory Region Type 1).

The way you are required to map your memory and build you page table is through page walk. You will need to design an assembly function that takes a virtual address and a virtual page size (4KB or 2MB) as parameters. The function will divide the address accordingly and walk through the page table levels and add needed page tables at missing levels to accommodate the virtual address as needed. It is very essential to maintain a bit map of all physical frames in your environment, and the function will need to scan the physical frame bit map for available frame(s) to map it to the target virtual page. The selected free physical frame(s) need to be marked occupied. In case of mapping 2MB virtual pages, if there are no enough contiguous frames to map a 2 MB virtual page the function should fail.

Essentially, you need to map 2MB virtual pages and only map 4KB virtual pages when cannot be otherwise. This will be dependent on where physical memory type1 starts and end; meaning that if a physical memory region does not start at 2MB aligned address or its size is not multiple of 2MB then some of its physical frames need to be mapped as 4KB virtual pages. **Your mission is to use the least number of 4KB virtual page mappings as much as possible. You should**

investigate configuring BOCHS with different physical memory layouts to be able to test your code. For example, you might start by mapping 2 MB virtual pages in a loop until your page walk function fails, and then try another loop for 4KB virtual pages to map the rest of the scattered physical frames.

You are also required to write a memory tester to test your mapped memory. The tester should be a function that loops over all memory bytes that are unused (beyond the first MB and not used by your page table) and write a value and read it from each byte to verify it and to make sure that the memory is accessible. If there is a problem with your memory map page table the VM will throw a page fault exception which is not yet handled by your code and this will lead to the suspension or crash of your VM, which will be an indication that something went wrong.

Now as we are in 64-bit Long Mode, you will need to use the video RAM to display text on the screen. As soon as you switch to Long mode you need to print messages on the screen after performing each step to show the progress.

Again, you can borrow available code from the slides, but you will need to come up with the missing parts on your own. Although the mapping of the first 2 MB and the CPU long mode switching will take place in the second stage, you will need to map the whole memory, as described above, by amending the code in the third stage under sources/includes/third_stage. Unlike the first and the second stages, you will need to add new files if needed and make all arrangements needed to get the target functionalities working.

IMPORTANT NOTE: WE HIGHLY RECOMMEND USING BOCHS IN THIS PHASE OF THE ASSIGNMENT.

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 your 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.
    d. Screenshots from your running.
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 <Section_ID>_<Team_ID>_Term_Project_Phase2.zip. Finally, upload your code to blackboard.

Grade

This assignment is worth 10% of the overall course grade. The assignment will be graded on a 100% grade scale, and then will be scaled down. 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.