

Grading Rubric: COS 318 Project 1

Fall 2019

This project is worth a total of **12 points**, plus 1 point extra credit.

DR: Functioning `print_char` and `print_string` functions as well as correct descriptions of the main concepts of the project. **2 pts.**

A01 A02: Both `bootblock.s` and `createimage.c` should compile with no errors and no warnings. Not even one. **1 pt for each file.**

B01: The *bootloader* must be properly located (or relocated) and transfer control to the kernel. Specifically, before entering the kernel, the `%cs`, `%ip`, and `%ds` registers must be set to the correct values. **1 pt.**

B02: The *bootloader* must initialize sufficient stack space for the kernel. Specifically, if the kernel issues as many as 256 `pushw` instructions in a row, the machine should not crash, and the stack pointer `%sp` should not overflow. **1 pt.**

C01: The output of *createimage* (the image file) must match the output of *createimage.given* exactly. In other words, all padding must consist of only zeros, and the file must be padded properly (both at the end of each program segment, and to align the segments with the disk sector size). **1 pt.**

C02: When *createimage* is given the `--extended` flag, it must output additional information from the ELF file. Specifically, it is required to display the program header's offset (from the ELF header), the virtual address, the file size and memory size fields, and output the information roughly in the same format as the output of *createimage.given*. **1 pt.**

D01: Your *bootloader* must be able to successfully load and execute a small test kernel we have created for grading purposes. This kernel is similar to the kernel supplied in the project, and is 2 sectors long. **2 pts.**

D02: Your *bootloader* must be able to successfully boot the provided, 9-sector kernel on a real machine. We will use your implementation of *createimage* and *bootblock*, as well as the supplied kernel to create an image file. We will then transfer the image onto the USB drive and boot one of the lab PCs off of that drive. We will only attempt this procedure if your submission passes **D01**. **2 pts.**

X01: (Extra credit): Your *bootloader* must be able to successfully load and execute the medium-sized kernel provided in the extra credit materials. This kernel is the 84 sectors long. If your *bootloader* performs only a single read from disk, it may fail. If your *bootloader* does not relocate itself, it will fail. **0.5 pts.**

X02: (Extra credit): Your *bootloader* must be able to successfully load and execute the large-sized kernel provided in the extra credit materials. This kernel is 160 sectors long. **0.5 pts.**

Additionally, we reserve the right to **remove as much as 1 point** for submissions that are confusing, obfuscated or overcomplicated. Please write simple, readable code with comments.

Dont forget to attach a brief README in text format.

Recall that this class has a firm late submission policy, which is detailed on the COS 318 website.