

# Grading Rubric: COS 318 Project 1

Fall 2012

This project is worth a total of **10 points**, plus 1 point extra credit.

A01 A02: Both `bootblock.s` and `createimage.c` should compile with no errors and no warnings. Not even one. If you don't get any warning by using the given `Makefile`, it should be fine. 1 pt. for each file

B01: The bootloader must be properly located (or relocated) and transfer control to the kernel. Specially, on entering to the kernel, `%CS=0x0000`, `%IP=0x1000` and `%DS=0x0000`. 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, it should be the case that the machine does not crash, and that the stack pointer `%SP` does 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 only of zero bytes, and the file must be padded properly (both before each program segment, and at the end). 1 pt.

C02 When `createimage` is given the `--extended` flag, it must output additional information from the ELF file. Specially, it is required to display the entry offset (from the ELF Header), and must display the Offset, Virtual Address, File Size and Memory Size fields from each ELF Program Header, i.e. roughly in the same format of the printing output of `createimage.given`. 1 pt.

D01: Your bootloader must be able to successfully load and execute a specially crafted small kernel. This kernel is similar to the kernel supplied in the project. It is 2 sectors long. 2 pts.

D02: Your bootloader must be able to successfully boot the provided, 9-sector kernel on a real machine. Specially, we will use your implementation of `createimage` and `bootblock`, as well as the supplied kernel, to create an image. We will burn that image onto a USB hard drive, and then will 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 a specially crafted medium kernel. This kernel is 85 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 a specially crafted large kernel. This kernel is similar to the large kernel provided on the course website. It is 160 sectors long. 0.5 pts.

Additionally, we reserve the right to remove as much as 1 point for submissions which are extremely confusing, obfuscated or overcomplicated. Please write simple, readable code with comments. Don't 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.