Assignment 2 booting and printing worklog gdt

Starting by setting up the dev enviroment according to the guide on canvas, adding the multiboot.asm to the add\_executable part of the cmake list to enable the multiboot2 bootloader, as this initially led to an invalid elf error.

Afterwards an Global descriptor table was added by creating the file «descriptor\_tables.h» that contains all the nessecary function declarations to setup the Global descriptor table, and the Interrupt decriptor table used for the next assingment.

Afterwards, we set up the function «gdt\_flush» in an assembly file, TODO: add explanation

After this we created the «gdt.c» file that implements the init\_gdt, and gdt\_set\_gate function that is used to initialize the global descriptor table.

Gdt\_set\_gate sets the value of one GDT entry.

Init\_gdt begins by defining the gdt\_ptr, afterwards setting up the rest of the gdt using the gdt\_set\_gate function by setting the null, code, data segment and flushing them using the gdt\_flush function.

Afterwards, the function is called in the main loop in the «kernel.c» file.

Wordlog printf