Kompilatorteknikk TDT4205 - Problem set 4

Ola Kristoffer Hoff

12^{th} April, 2023

1 Assembly programming

After battling the lack of information about mach-O I finally wrote some assembly that ran. Then it was not to difficult to create, unnecessarily, 193 lines of code it took to create the program. The basic control flow idea I implemented was a loop that continued as long as counter was less than twelve. I used $r12$ for this (both $r12$ and $r13$ is function call safe, meaning they won't change unless I do so). Inside the loop I used $r13$, sat to twelve at the start of the loop to jump to the day we were suppose to write first. The print statements was written consecutively with a label between each. This allowed me to jump to e.g. label: $num5$, then day five, four, and so on until day 1 would be printed. The code itself was a bit to big to fit neatly into the PDF, so I left it out, so check out the code file.

2 Code Generation Part I

I developed the compiler to compile to mach-O. I also had to use the *safe print* on MacOSX. Not much more to say a about the code. Have a look at it. There are some *points*/edge cases in the code that the test cases did not reach, and I did not go over all of them. Hence, there might be a lot of terminating code for cases I thought might appear.