SPAASM Assignment 1 - Documentation

Ondrej Krajčovič

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Systémové Programovanie A Assemblery

xkrajcovico@stuba.sk

18. march 2025

IV.Semester Bc Štúdia

Assignment:

Number of Assignment**:**

- Assignment:
  - Number 6:

    “Vypísať všetky riadky vstupu a pred každý napísať jeho poradové číslo.”

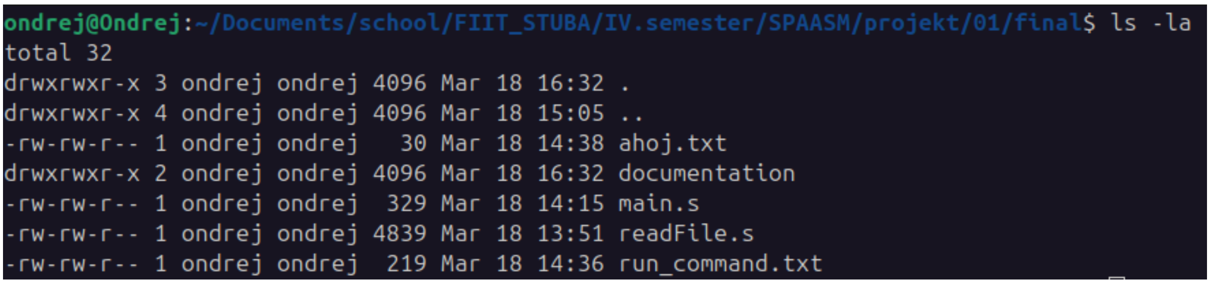
Bonuses:

- I would like to try to fulfill these bonus requirements(10 & 12):
  - *“10.Plus 2 body je možné získať ak pridelená úloha bude realizovaná ako externá procedúra (kompilovaná samostatne a prilinkovaná k výslednému programu).*
    …
  - *12.Plus 1 bod je možné získať za (dobré) komentáre, resp. dokumentáciu, v anglickom jazyku.”*

Solution:

Structure:

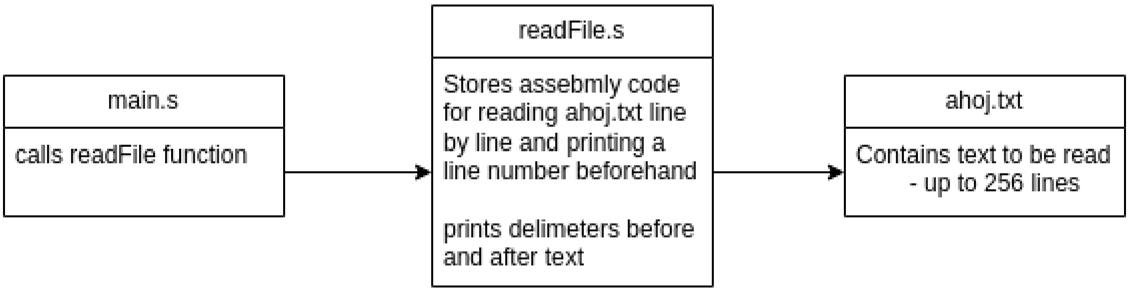
- Project is made of these elements:
    - Main - assembly code
    - readFile - assembly function
    - ahoj.txt - file being read
    - documentation



```
ondrej@Ondrej:~/Documents/school/FIIT_STUBA/IV.semester/SPAASM/projekt/01/final$ ls -la
total 32
drwxrwxr-x 3 ondrej ondrej 4096 Mar 18 16:32 .
drwxrwxr-x 4 ondrej ondrej 4096 Mar 18 15:05 ..
-rw-rw-r-- 1 ondrej ondrej   30 Mar 18 14:38 ahoj.txt
drwxrwxr-x 2 ondrej ondrej 4096 Mar 18 16:32 documentation
-rw-rw-r-- 1 ondrej ondrej  329 Mar 18 14:15 main.s
-rw-rw-r-- 1 ondrej ondrej 4839 Mar 18 13:51 readFile.s
-rw-rw-r-- 1 ondrej ondrej  219 Mar 18 14:36 run_command.txt
```

Main.s

- Is first one to be run.
- After start, it calls external function readFile stored in readFile.s
- After readFile returns, main returns 0 and safely exits


main.s
calls readFile function
readFile.s
Stores assebmly code for reading ahoj.txt line by line and printing a line number beforehand
prints delimeters before and after text
ahoj.txt
Contains text to be read - up to 256 lines

readFile.s

Simple example of run of the readfile function:

- **.data**
    - defines registers and variables

      (among others static “ahoj.txt” file destination to read from)
- **.bss**
    - defines buffers
- **.text**

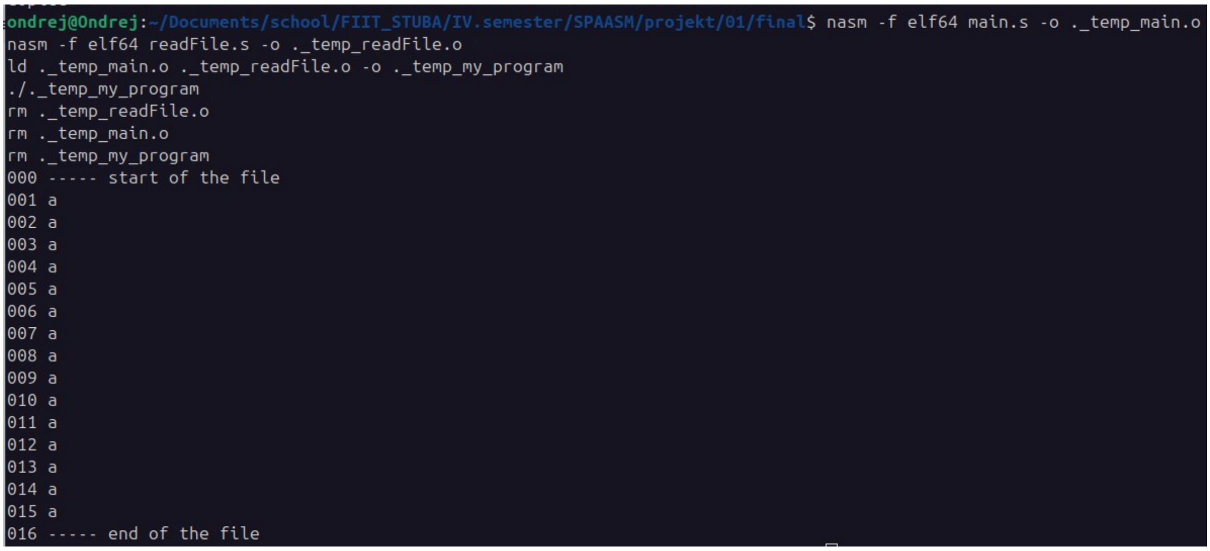
  _start readFile
- prints opening delimiter
- **read_loop**
- prints number + space
- reads line
- chcecks for EOF
- prints line
- increments number
- prints closing delimiter
- returns to super function (main)

  (other used help functions are used for converting and printing ASCII numbers before lines)

In summary readFile iterates through reading a file line by line and prints number before each one.

Running the project:

- To run the project we need i used commands necessary, stored in run_commands.txt
- Example output for ahoj.txt containing 15 lines of just letter a:



```
ondrej@Ondrej:~/Documents/school/FIIT_STUBA/IV.semester/SPAASM/projekt/01/final$ nasm -f elf64 main.s -o ._temp_main.o
nasm -f elf64 readFile.s -o ._temp_readFile.o
ld ._temp_main.o ._temp_readFile.o -o ._temp_my_program
./._temp_my_program
rm ._temp_readFile.o
rm ._temp_main.o
rm ._temp_my_program
000 ----- start of the file
001 a
002 a
003 a
004 a
005 a
006 a
007 a
008 a
009 a
010 a
011 a
012 a
013 a
014 a
015 a
016 ----- end of the file
```

Summary:

The project requires "Main.s", "readFile.s", and "Ahoj.txt" files and is compiled using the standard 64-bit x86 NASM assembler. It opens "ahoj.txt", prints a start delimiter, then iterates through the file, printing each line with a line number before it, followed by a closing delimiter. The program uses Linux syscalls for file operations. Input must be a readable file with lines, and it assumes the file exists and is accessible. Potential improvements include better error handling, performance optimizations, and more flexible formatting options for line numbering and delimiters.

Content