



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS

Virtual Machine

4 laboratory work

Lecturers: dr. Pavel Stefanovič, Rokas Štrimaitis, dr. Tomas Petkus

The main aim of laboratory work

- Create a virtual machine (VM), which will be able to read a given binary file instructions and save/run given tasks.
- To learn how to read/work with binary files (*.BIN).
- Improving programming skills.

4 laboratory work (part 1) (1)

- Create a virtual machine, which will have:
 - 16 general registers R0-R15 (8 bits);
 - 8 bits program addressing;
 - 256 bytes program memory;
 - Flag register.
- Two files are given:
 - input data (*q1_encr.txt*);
 - instructions binary file (*decryptor.bin*).

4 laboratory work (part 1) (2)

- The task is considered completed when VM by executing the given program (**decryptor.bin**) and using the input data (**q1_encr.txt**) will decode the text into a separate file.

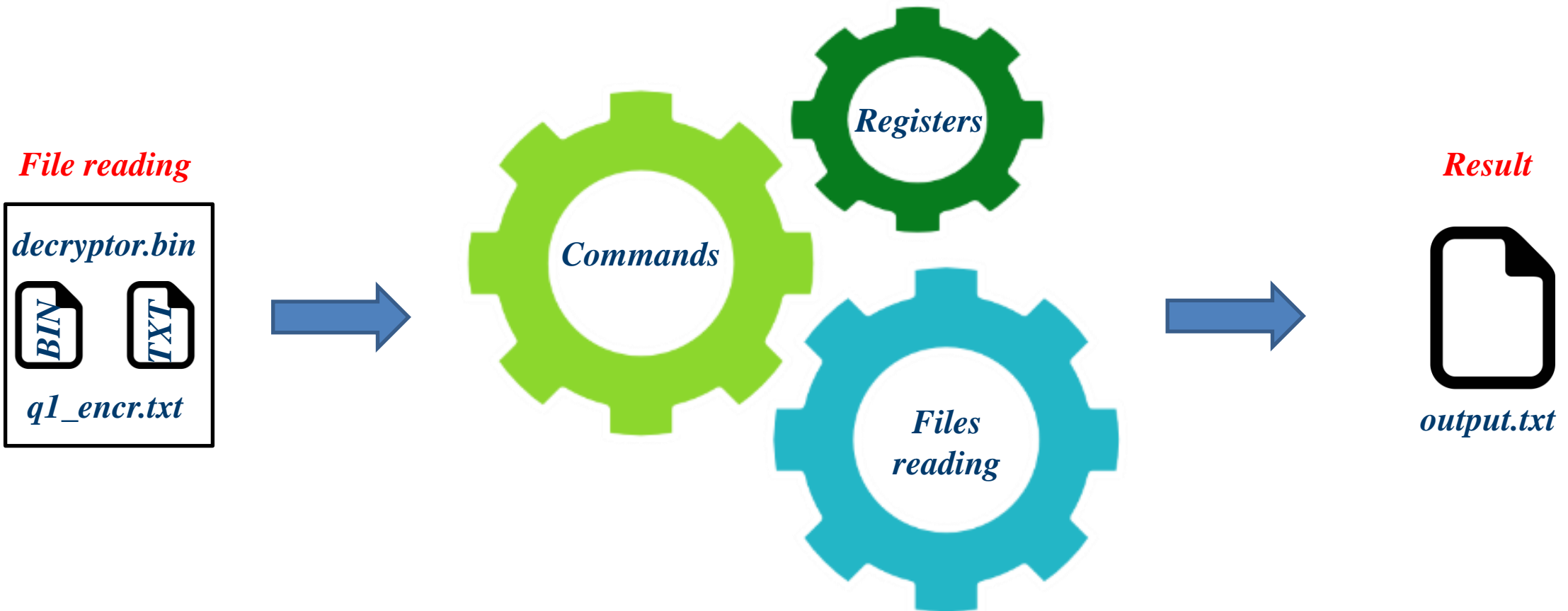
4 laboratory work (part 2)

Create your own decryptor and encryptor files to encrypt and decrypt your chosen short message.

Requirements:

- at least 8 different VM commands are used;
- one character is encoded with at least two characters, decoding – from two characters one character is obtained;
- at least two different jumps should be used.

Scheme of the program



Virtual machine

Binary file

- To see the content of binary file (*decryptor.bin*) you can use the hex dump programs or simply visit the website <https://hexed.it/>

File Information			-Untitled- x	decryptor.bin x
File Name	decryptor.bin		00000000	04 40 10 01 0A 1A 10 02 10 03 0D 02 0D 03 05 03 .@.....
File Size	32 bytes		00000010	05 03 05 03 05 03 0F 32 0E 12 11 02 07 E6 0B 002.....µ..
			00000020	
Data Inspector (Little-endian)				
Type	Unsigned (+)	Signed (±)		
8-bit Integer	4	4		
16-bit Integer	16388	16388		
24-bit Integer	1064964	1064964		
32-bit Integer	17842180	17842180		

Input data

- Text file (*q1_encr.txt*) is used as input data, inside of him is totally 717 letters and symbols.



q1_encr.txt

1 QHAIMCUEGYNBRMCADCQMCWCCSCGA@BQABEECU@CR

Structure of VM commands (1)

I type (command with two parameters):

- These commands are used to work with registers:

8 bits command code							
R_y				R_x			
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

- Some of commands (*LSL*, *INC*) doesn't have the R_y register, so in this case the 7-4 bits are ignored.

Structure of VM commands (2)

2 type (command with one parameter):

- 1 byte command code and 1 byte constant (MOVC) or jump (JZ, JNZ, JFE) size.



Command list of VM (1)

COMMAND	CODE	TYPE	COMMENT
INC	01	1	Register R_x increased by one.
DEC	02	1	Register R_x decreased by one.
MOV	03	1	Copy the register R_y content to the register R_x .
MOVC	04	2	Copy the byte constant to the register R_0 .
LSL	05	1	Register R_x shifting by one bit to the left.
LSR	06	1	Register R_x shifting by one bit to the right.
JMP	07	2	Jump command, which are obtained after address constant are added to the command counter with the sign.
JZ	08	2	Jump command, which are obtained after address constant are added to the command counter with the sign, if flag is on.
JNZ	09	2	Jump command, which are obtained after address constant are added to the command counter with the sign, if flag is off.

Command list of VM (2)

COMMAND	CODE	TYPE	COMMENT
JFE	0A	2	Jump command, which are obtained after address constant are added to the command counter with the sign, if input file ends (IN).
RET	0B	–	Virtual machine stop working.
ADD	0C	1	ADDITION: $R_x = R_x + R_y$
SUB	0D	1	SUBTRACTION: $R_x = R_x - R_y$
XOR	0E	1	XOR operation: $R_x = R_x \oplus R_y$
OR	0F	1	OR operation: $R_x = R_x \vee R_y$
IN	10	2	Reads one byte from the input data file, assigns value to the register R_x and set up the file end trigger.
OUT	11	1	Prints the content of register R_x to the file.

ADVICES

- Realize VM registers as **CHAR** array: *unsigned char regs[16]*.
- Realize VM program memory as a **CHAR** array: *char prog_mem[256]*.
- Flag register can be a simple **INT** variable.
- Inside the binary file *decryptor.bin* there are **ASCII** control codes, so you will need to open/read this file in binary mode (“**rb**”).
- For commands realization use **IF**, **ELSEIF** or **SWITCH**.

Evaluation



General requirements (1 point)

1. VM specification is maintained (0.5).
2. Program code is optimized (0.5).

I part (5 points)

1. All VM commands are programmed (1).
2. Jumps are universal (1).
3. The text is decoded correctly (2).
4. Present the result in a separate file (1).

II part (4 points)

1. At least 8 different VM commands were used (1).
2. At least 2 jumps used (1).
3. Correct encoding / decoding (2).