Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №7

на тему:

**«**Загрузка и выполнение программ. Работа с памятью.**»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253502  АЛЬХОВИК Данила Игоревич |
|  |
| (дата, подпись студента) |
| Проверил старший преподаватель кафедры проектирования информационно-компьютерных систем  ЯЧИН Николай Сергеевич |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Цель работы:** Вариант 2. Подсчитать число пустых строк в файле.

**Ход работы:** на рисунке 1 представлены изначальные значения регистров, на рисунке 2 – ход выполнения задачи, на рисунке 3 - соответствующие значения после выполнения программы.

Листинг 1 – Исходный код программы задания 1

section .rodata

enterProgName:

db "Enter program name: ", 0

formatInputS:

db "%s", 0

enterRuns:

db "Enter number of runs (1-255): ", 0

formatInputD:

db "%d", 0

errInputRuns:

db "Number of runs must be between 1 and 255", 10, 0

runMsg:

db "Running %s, iteration %d", 10, 0

formatProg:

db "./%s", 0

runningErr:

db "Error running %s, iteration %d", 10, 0

finishMsg:

db "Finished running %s %d times", 10, 0

default rel

global main: function

extern printf, scanf, puts, system, sprintf

section .text align=1 exec ; executable instructions

main:

push rbp ; save base pointer

mov rbp, rsp ; set base pointer

sub rsp, 320 ; allocate 320 bytes on stack

mov edi, enterProgName ; print enter program name

mov eax, 0

call printf

lea rax, [rbp-112] ; load address of program name

mov rsi, rax

mov edi, formatInputS

mov eax, 0

call scanf

.inputRuns:

mov edi, enterRuns

mov eax, 0

call printf

lea rax, [rbp-116]

mov rsi, rax

mov edi, formatInputD

mov eax, 0

call scanf

mov eax, DWORD [rbp-116] ; check if runs is between 1 and 255

test eax, eax ; if not, print error message and ask for input again

jle .errRuns ; if yes, start loop

mov eax, DWORD [rbp-116]

cmp eax, 255

jle .startLoop

.errRuns:

mov edi, errInputRuns ; print error message

call puts ; ask for input again

jmp .inputRuns ; jump to input

.startLoop:

mov DWORD [rbp-4], 0 ; set counter to 0

jmp .cmpLoop ; jump to compare loop

.run:

mov eax, DWORD [rbp-4]

lea edx, [rax+1] ; set iteration number

lea rax, [rbp-112]

mov rsi, rax ; load address of program name

mov edi, runMsg

mov eax, 0

call printf

lea rdx, [rbp-112]

lea rax, [rbp-320]

mov esi, formatProg

mov rdi, rax

mov eax, 0

call sprintf

lea rax, [rbp-320]

mov rdi, rax

call system ; run program

mov DWORD [rbp-8], eax ; check if program ran successfully

cmp DWORD [rbp-8], 0; if not, print error message and ask for input again

je .iteration ; if yes, increment counter and jump to compare loop

mov eax, DWORD [rbp-4]

lea edx, [rax+1]

lea rax, [rbp-112]

mov rsi, rax

mov edi, runningErr

mov eax, 0

call printf

mov eax, 1

jmp .endProg

.iteration:

add DWORD [rbp-4], 1 ; increment counter

.cmpLoop:

mov eax, DWORD [rbp-116] ; compare counter to runs

cmp DWORD [rbp-4], eax

jl .run ; if counter is less than runs, run program again

mov edx, DWORD [rbp-116] ; if counter is equal to runs, print finish message

lea rax, [rbp-112]

mov rsi, rax

mov edi, finishMsg

mov eax, 0

call printf

mov eax, 0

.endProg:

leave

ret

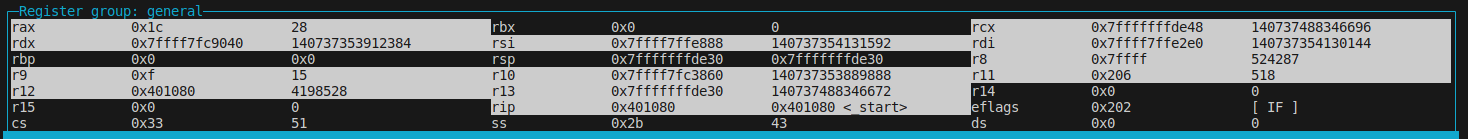


Рисунок 1 – Значения регистров перед выполнением

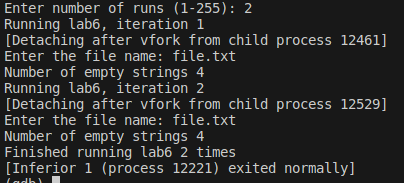


Рисунок 2 – Ход выполнения задачи

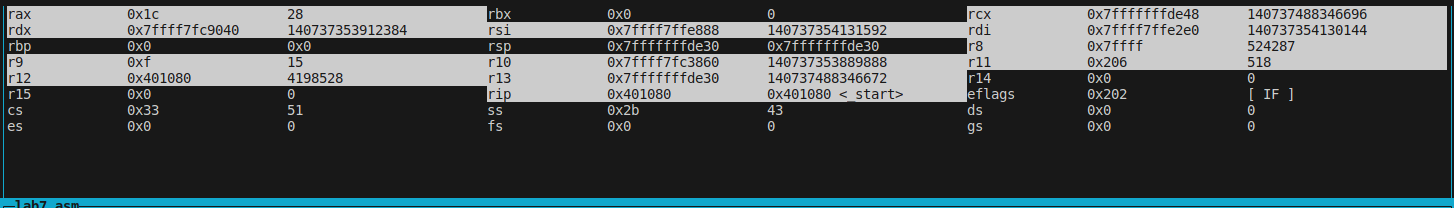


Рисунок 3 – Значения регистров после выполнения