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 №6

на тему:

**«**Работа с файлами**»**

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 .data align=1 noexec ; section number 2, data

section .bss align=1 noexec ; section number 3, bss

section .rodata align=1 noexec ; section number 4, const

EnterMsg:

db "Enter the file name: ", 0

formatInput:

db "%s", 0

fileFlag:

db "r", 0

ErrorMsg:

db "Error opening the file", 0

term:

db 10, 0

ResultMsg:

db "Number of empty strings %d", 10, 0

default rel

global main: function

extern \_\_stack\_chk\_fail, fclose, fgets, strcmp, exit, puts, fopen, \_\_isoc99\_scanf, printf

section .text align=1 exec ; section number 1, code

main: ; Function begin

endbr64

push rbp ; save old base pointer

mov rbp, rsp ; set new base pointer

sub rsp, 240 ; allocate 240 bytes for local variables

mov rax, qword [fs:abs 28H] ; stack canary

mov qword [rbp-8H], rax ; save stack canary

xor eax, eax ; set eax to 0

lea rax, [rel EnterMsg] ; load address of EnterMsg to rax

mov rdi, rax ; set rax to first argument of printf

mov eax, 0 ; set eax to 0

call printf ; call printf with 1 argument (EnterMsg)

lea rax, [rbp-0E0H] ; load address of file name to rax

mov rsi, rax ; set rax to second argument of scanf

lea rax, [rel formatInput] ; load address of formatInput to rax

mov rdi, rax ; set rax to first argument of scanf

mov eax, 0 ; set eax to 0

call \_\_isoc99\_scanf ; call scanf with 2 arguments (formatInput, file name)

lea rax, [rbp-0E0H] ; load address of file name to rax

lea rdx, [rel fileFlag] ; load address of fileFlag to rdx

mov rsi, rdx ; set rdx to second argument of fopen

mov rdi, rax ; set rax to first argument of fopen

call fopen ; call fopen with 2 arguments (file name, fileFlag)

mov qword [rbp-0E8H], rax ; save file pointer to local variable

cmp qword [rbp-0E8H], 0 ; compare file pointer to 0

jnz counterInit ; if file pointer is not 0, jump to counterInit

lea rax, [rel ErrorMsg] ; load address of ErrorMsg to rax

mov rdi, rax ; set rax to first argument of puts

call puts ; call puts with 1 argument (ErrorMsg)

mov edi, 1 ; set edi to 1

call exit ; call exit with 1 argument (1)

counterInit:

mov dword [rbp-0ECH], 0 ; set counter to 0

jmp fileRead ; jump to fileRead

cmpLoop:

lea rax, [rbp-70H] ; load address of buffer to rax

lea rdx, [rel term] ; load address of term to rdx

mov rsi, rdx ; set rdx to second argument of strtok

mov rdi, rax ; set rax to first argument of strtok

call strcmp ; call strcmp with 2 arguments (buffer, term)

test eax, eax ; test eax with eax

jnz fileRead ; if eax is not 0, jump to fileRead

add dword [rbp-0ECH], 1

; increment counter by 1

fileRead: mov rdx, qword [rbp-0E8H] ; load file pointer to rdx

lea rax, [rbp-70H] ; load address of buffer to rax

mov esi, 100 ; set esi to 100

mov rdi, rax ; set rax to first argument of fgets

call fgets ; call fgets with 3 arguments (buffer, 100, file pointer)

test rax, rax ; test rax with rax

jnz cmpLoop ; if rax is not 0, jump to cmpLoop

mov eax, dword [rbp-0ECH]

mov esi, eax ; set eax to second argument of printf

lea rax, [rel ResultMsg] ; load address of ResultMsg to rax

mov rdi, rax

mov eax, 0

call printf ; call printf with 2 arguments (ResultMsg, counter)

mov rax, qword [rbp-0E8H]

mov rdi, rax

call fclose ; call fclose with 1 argument (file pointer)

mov eax, 0

mov rdx, qword [rbp-8H] ; load stack canary to rdx

sub rdx, qword [fs:abs 28H] ; compare stack canary with stack canary from before

jz endProg

call \_\_stack\_chk\_fail

endProg:

leave

ret

; main End of function

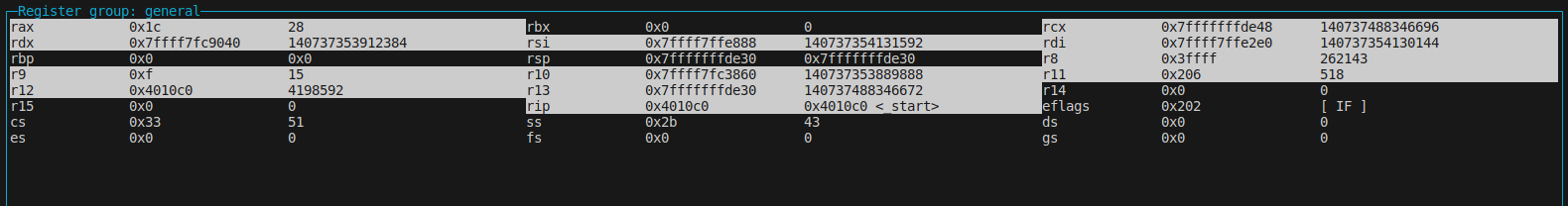


Рисунок 1 – Значения регистров перед выполнением

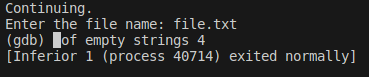


Рисунок 2 – Ход выполнения задачи

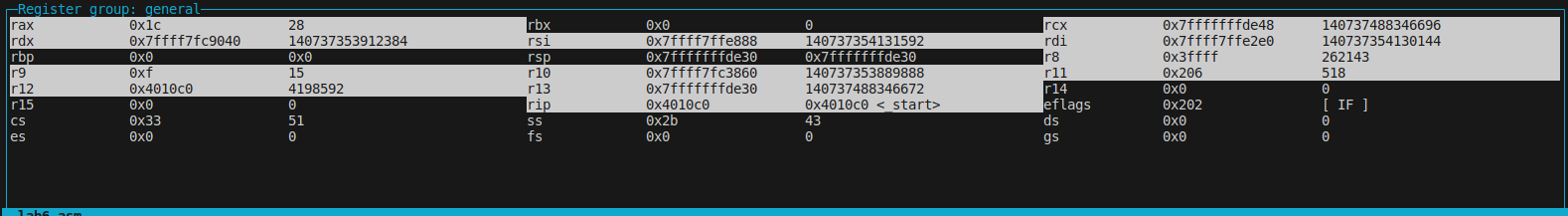


Рисунок 3 – Значения регистров после выполнения