МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

КАФЕДРА «ЭВМ»

Отчет по лабораторной работе №4

по дисциплине

«Машинно-зависимые языки программирования»

Выполнили:

Студенты группы 945

Комлева Екатерина

Соболева Елизавета

Горбачева Дарья

Проверили:

доц., к.т.н. Муратов Е.Р.

асс. каф. ЭВМ Тарасов А.С.

Рязань 2021

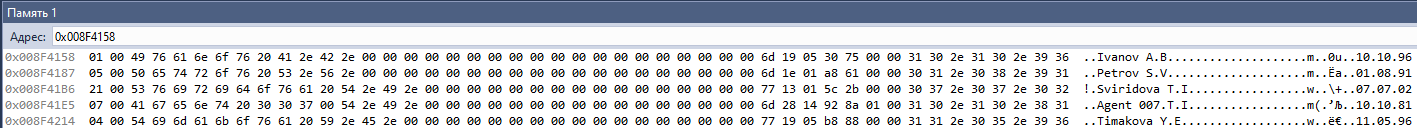
**Цель работы:** научиться размещать массивы данных в памяти и выполнять арифметические действия над разными типами данных.

**ПРАКТИЧЕСКАЯ ЧАСТЬ**

**Задание:**

1. Необходимо посчитать количество сотрудников мужского и женского пола.

Для начала необходимо прочитать файл. Файл был считан в переменную workers\_array.

Рисунок 1 – Считанный файл

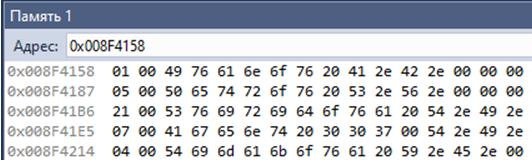


Рисунок 2 - Считанный файл

Результат выполнения подпрограммы для задания 1:

C:\Users\komle\AppData\Local\Temp\lu5596y7ilg2.tmp\lu5596y7ilgj_tmp_87f89ed77a18dfa5.png   
Рисунок 3– Выполнение задания 1

C:\Users\komle\AppData\Local\Temp\lu5596y7ilg2.tmp\lu5596y7ilgj_tmp_13371ac9ca6a8a06.png   
Рисунок 4– Выполнение задания 1

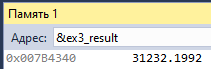
1. Вычислить средний возраст всех сотрудников, используя целочисленное деление.

Результат выполнения подпрограммы для задания 2. Средний возраст сотрудников – 0х27 – 39 лет.

C:\Users\komle\AppData\Local\Temp\lu5596y7ilg2.tmp\lu5596y7ilgj_tmp_2e24ca3bcb9b275d.png  
Рисунок 5 – Средний возраст

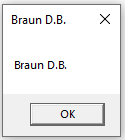
1. Вычислить средний оклад всех сотрудников, используя FPU

Результат выполнения подпрограммы для задания 3:

  
Рисунок 6 – Результат выполнения задания 3

1. Вывести ФИО сотрудника с идентификатором 7

Результат выполнения подпрограммы для задания 4:

  
Рисунок 7 – Результат выполнения задания 4

**Вывод:** в данной лабораторной работе мы научились размещать массивы данных в памяти и выполнять арифметические действия над разными типами данных.

**ПРИЛОЖЕНИЕ**

.686P

.MODEL FLAT, STDCALL

.STACK 4096

option casemap:none

include c:/masm32/include/windows.inc

include c:/masm32/include/user32.inc

includelib c:/masm32/lib/user32.lib

include c:/masm32/include/kernel32.inc

includelib c:/masm32/lib/kernel32.lib

.DATA

;file

FileName db "F:\base.dat",0

BadText db "File dont open!",0

;struct

workers struc

id dw 0h

fio db 30 dup (' ')

gender db 'm'

age db 18h

standing db 1

salary dd 1000

birthdate db 8 dup(' ')

workers ends

ex4\_Text db 30 dup(' ')

.DATA?

hFile HANDLE ?

dwBytesRead dd ?

workers\_array workers 10 dup(<?>)

ex1\_Mcount dd ?

ex1\_Wcount dd ?

ex2\_count dw ?

ex3\_divisible dd ?

ex3\_divider dd ?

ex3\_result dd ?

.CODE

START:

invoke CreateFile, addr FileName, GENERIC\_READ, 0, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL

mov hFile, eax

cmp hFile, INVALID\_HANDLE\_VALUE

jz ErrorMsg

invoke ReadFile, hFile, offset workers\_array, 470, addr dwBytesRead, NULL

or eax, eax

jz ErrorMsg

ex1:

mov eax, 0

mov ex1\_Mcount, 0

mov ex1\_Wcount, 0

cykle2:

imul ebx, eax, 47

mov cl, workers\_array[ebx].gender

cmp cl, 'm'

jz m

inc ex1\_Wcount

jmp mend

m:

inc ex1\_Mcount

mend:

inc eax

cmp eax, 10

jnz cykle2

ex2:

mov ebx,0

mov eax, 0

mov ex2\_count, 0

mov ecx, 0

c2:

imul edx, eax, 47

mov cl, workers\_array[edx].age

add ebx, ecx

inc eax

cmp eax, 10

jnz c2

mov eax, ebx

mov ebx, 10

mov edx, 0

div bx

mov ex2\_count, ax

ex3:

mov eax, 0

mov ex3\_divisible, 0

mov ex3\_divider, 0

mov ex3\_result, 0

c3:

imul ebx, eax, 47

mov ecx, workers\_array[ebx].salary

add ex3\_divisible, ecx

inc eax

cmp eax, 10

jnz c3

mov eax, 10

mov ex3\_divider, eax

fild ex3\_divider

fild ex3\_divisible

fdiv st, st(1)

fstp ex3\_result

ex4:

lea eax, workers\_array[6\*47].fio

lea edx, ex4\_Text

mov ebx, 30

c4:

mov cl, byte ptr[eax]

mov byte ptr[edx], cl

inc eax

inc edx

dec ebx

jnz c4

invoke MessageBox, NULL, addr ex4\_Text, addr ex4\_Text, MB\_OK

invoke ExitProcess, 0

invoke CloseHandle, hFile

jmp End\_code

ErrorMsg:

invoke MessageBox, NULL, addr BadText, addr BadText, MB\_OK

invoke ExitProcess, 0

End\_code:

RET

END START