Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 5

по дисциплине «Программирование на языке Ассемблера»

Вариант 2

Выполнил студент гр. 150502: Альхимович Н.Г.

Проверил:        Туровец Н.О.

Минск 2022

Цель работы:

Ознакомиться с основными операциями обработки файлов, получить понятие о работе с параметрами командной строки.

Вариант задания:

Подсчитать число пустых строк в файле.

Теоретические сведения:

1. Работа с файлами.

Для работы с файлами в данной лабораторной работе лучше всего ис- пользовать функции DOS, которые обращаются к файлу через 16-битный иден- тификатор (дескриптор) файла. Такой подход более прост, чем использование более старых описателей файла (37-байтного блока управления файлом FCB) или функций низкого уровня доступа к диску (прерывание BIOS 13h).

Первые пять значений идентификаторов такого формата инициализиру- ются системой следующим образом:

-- 0 – STDIN – стандартное устройство ввода (клавиатура),  
-- 1 – STDOUT – стандартное устройство вывода (экран),  
-- 2 – STDERR – устройство вывода сообщений об ошибках (всегда экран), -- 3 – AUX – последовательный порт (СОМ1), -- 4 – PRN – параллельный порт (LPT1).

Работа с файлами выполняется через функции DOS в стандартном поряд- ке:

1) создание или открытие существующего файла;  
2) выполнение файловых операций чтения или записи данных; 3) закрытие файла.

Дополнительно доступны операции: удаление, поиск и управление.

Для создания или открытия существующего файла рекомендуется ис- пользовать функции:

-- Функция DOS 3Ch (INT 21h) – создать файл: Ввод: AH = 3Ch,

СХ =атрибутфайла(биты):  
-- 7 – файл можно открывать разным процессам, -- 6 – не используется,  
-- 5 – архивный бит (1, если файл не сохранялся), -- 4 – каталог (должен быть 0 для функции 3Ch), -- 3 – метка тома (игнорируется функцией 3Ch), -- 2 – системный файл,  
-- 1 – скрытый файл,  
-- 0 – файл только для чтения.

DS:DX = адрес ASCIZ-строки с полным именем файла.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден,

-- 04h – слишком много открытых файлов,

-- 05h – доступ запрещен.  
Если CF = 0, то операция выполнена успешно: АХ =идентификаторфайла.

Особенности: Если файл уже существует, функция все равно открывает его, присваивая ему нулевую длину.

-- Функция DOS 5Bh (INT 21h) – создать и открыть файл:

Ввод: Вывод:

AH = 5Bh,  
СХ =атрибутфайла,  
DS:DX = адрес ASCIZ-строки с полным именем файла.

Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден,  
-- 04h – слишком много открытых файлов,  
-- 05h – доступ запрещен,

-- 50h – файл уже существует.  
Если CF = 0, то операция выполнена успешно:

АХ = идентификатор файла, открытого для чтения/записи в режиме совместимости.

-- Функция DOS 5Ah (INT 21h) – создать и открыть временный файл: Ввод: AH = 5Ah,

СХ =атрибутфайла,  
DS:DX = адрес ASCIZ-строки с путем, оканчивающимся симво- лом «\», и тринадцатью нулевыми байтами в конце.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден,

-- 04h – слишком много открытых файлов,

-- 05h – доступ запрещен.  
Если CF = 0, то операция выполнена успешно:  
АХ = идентификатор файла, открытого для чтения/записи в ре- жиме совместимости (при этом в строку по адресу DS:DX допи- сывается имя созданного файла).

Особенности: Функция создает файл с уникальным именем, который не являет- ся на самом деле временным, его следует специально удалять, для чего его имя и записывается в строку в DS:DX.

-- Функция DOS 3Dh (INT 21h) – открыть существующий файл:

Ввод:

= 3Dh,

= режим доступа (биты): 0 – открыть для чтения, 1 – открыть для записи,  
3 – 2 – 00 (резерв),

Вывод:

AH AL --

-- -- --

011 – чтение запрещено

100 - запрещений нет.  
-- 7 – файл не наследуется порождаемыми процессами.

DS:DX = адрес ASCIZ-строки с полным именем файла. CL = маска атрибутов файла.

Если CF = 1, то ошибка операции (в AX код ошибки): -- 02h – файл не найден,  
-- 03h – путь не найден,  
-- 04h – слишком много открытых файлов,

-- 05h – доступ запрещен,

-- 0Ch – неверный режим доступа.  
Если CF = 0, то операция выполнена успешно:

6 – 4 – режим доступа для других процессов: 000 – режим совместимости,  
001 – все операции запрещены,  
010 – запись запрещена

АХ =идентификаторфайла.

Во всех случаях имя файла (если диск или путь отсутствуют в описании, то системой используются их текущие значения) описывается ASCIZ-строкой (строкой ASCII-символов, оканчивающейся нулем), которая, например, имеет следующий вид:

filename db 'с:\data\filename.txt',0

Для выполнения файловых операций чтения или записи данных рекомен- дуется использовать функции:

-- Функция DOS 3Fh (INT 21h) – чтение из файла или устройства: Ввод: АН = 3Fh,

ВХ =идентификаторфайла,  
СХ = число байт для чтения,  
DS:DX = адрес буфера для приема данных.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 05h – доступ запрещен,

-- 06h – неверный идентификатор.  
Если CF = 0, то операция выполнена успешно: АХ = число считанных байт.

Особенности: Каждая следующая операция чтения, так же как и записи, начи- нается не с начала файла, а с того байта, на котором остановилась предыдущая операция чтения/записи.  
Если при чтении из файла число фактически считанных байт в АХ меньше, чем заказанное число в СХ, то при чтении был до- стигнут конец файла.

-- Функция DOS 42h (INT 21h) – переместить указатель чтения/записи:

Ввод:

Вывод:

АН =42h,  
ВХ =идентификаторфайла,  
CX:DX = расстояние, на которое надо переместить указатель (знаковое число),  
AL = перемещение относительно:

0 – начала файла,  
1 – текущей позиции, 2 – конца файла.

Если CF = 1, то ошибка операции (в AX код ошибки): -- 06h – неверный идентификатор.

Если CF = 0, то операция выполнена успешно:

CX:DX = новое значение указателя (в байтах от начала файла). Особенности: Указатель можно установить за реальными пределами файла:

-- если указатель устанавливается в отрицательное число, то следующая операция чтения/записи вызовет ошибку;  
-- если указатель устанавливается в положительное число, большее длины файла, следующая операция записи увеличит

размер файла.  
Эта функция также часто используется для определения длины файла:достаточновызватьеесСХ = 0,DX = 0,AL = 2,ив CX:DX будет возвращена длина файла в байтах.

-- Функция DOS 40h (INT 21h) – запись в файл или устройство: Ввод: АН = 40h,

ВХ =идентификаторфайла,  
СХ = число байт для записи, DS:DX = адрес буфера с данными.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 05h – доступ запрещен,

-- 06h – неверный идентификатор.  
Если CF = 0, то операция выполнена успешно: АХ =числозаписанныхбайт.

Особенности: Если при записи в файл указать СХ = 0, то файл будет обрезан по текущему значению указателя.

При записи в файл на самом деле происходит запись в буфер DOS, данные из которого сбрасываются на диск при закрытии файла или если их количество превышает размер сектора диска.

-- Функция DOS 68h (INT 21h) – сброс файловых буферов DOS на диск: Ввод: АН = 68h,

ВХ =идентификаторфайла.  
Вывод: Если CF = 1, то ошибка операции (в AX код ошибки).

Если CF = 0, то операция выполнена успешно.  
-- Функция DOS 0Dh (INT 21h) – сброс всех файловых буферов на диск:

Ввод: АН = 0Dh  
Для закрытия файла рекомендуется использовать следующую функцию:

-- Функция DOS 3Eh (INT 21h) – закрыть файл: Ввод: АН = 3Eh,

ВХ =идентификаторфайла.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 06h – неверный идентификатор.

Если CF = 0, то операция выполнена успешно.

Особенности: Если файл был открыт для записи, то все файловые буферы сбра- сываются на диск, устанавливается время модификации файла и

записывается его новая длина. Дополнительные операции с файловой системой:

-- Функция DOS 41h (INT 21h) – удалить файл: Ввод: АН = 41h,

DS:DX = адрес ASCIZ-строки с полным именем файла.

Вывод: Если CF = 1, то ошибка операции (в AX код ошибки): -- 02h – файл не найден,

-- 03h – путь не найден,

-- 05h – доступ запрещен.  
Если CF = 0, то операция выполнена успешно.

Особенности: Удалить файл можно только после того, как он будет закрыт, так как DOS будет продолжать выполнять запись в несуществующий файл, что может привести к разрушению файловой системы. Данная функция не позволяет использовать маску имени файла (символы «\*» и «?») для удаления сразу нескольких файлов (только с DOS 7.0).

-- Функция DOS 4Eh (INT 21h) – найти первый файл:

Ввод:

Вывод:

АН = 4Eh,  
AL используется при обращении к функции APPEND,  
СХ = атрибуты файла (биты 0 и 5 игнорируются, если бит 3 установлен, то все остальные биты игнорируются),  
DS:DX = адрес ASCIZ-строки с именем файла, которое может включать путь и маску для поиска (символы \* и ?).  
Если CF = 1, то ошибка операции (в AX код ошибки):

-- 02h – файл не найден,  
-- 03h – путь не найден,  
-- 12h – неверный режим доступа.

Если CF = 0, то операция выполнена успешно и область DTA

заполняется данными о найденном файле.

Особенности:Вызов этой функции заполняет данными область памяти DTA (область передачи данных), которая начинается по умолчанию со смещения 0080h от начала блока данных PSP (при запуске com- и exe-программ сегменты DS и ES содержат сегментный адрес начала PSP), но ее можно переопределить с помощью функции DOS 1Ah.

Функции поиска файлов заполняют DTA следующим образом: +00h (байт) – биты 0 – 6: ASCII-код буквы диска,

бит 7: сетевой диск;  
+01h (11 байт) – маска поиска (без пути);

+0СН (байт) – атрибуты для поиска;

+0Dh (слово) – порядковый номер файла в каталоге; +0Fh (слово) – номер кластера начала внешнего каталога; +11h (4 байта) – зарезервировано;  
+15h (байт) – атрибуты найденного файла;  
+16h (слово) – время создания файла в формате DOS:

биты 15 – 11: час (0 — 23),  
биты 10 – 05: минута,  
биты 04 – 00: номер секунды, деленный на 2 (0 – 30).

+18h (слово) – дата создания файла в формате DOS: биты 15 – 09: год, начиная с 1980,

биты 08 – 05: месяц,

биты 04 – 00: день. +1Ah (4 байта) – размер файла;

+1Eh (13 байт) - ASCIZ-имя найденного файла (с расширением).

-- Функция DOS 1Ah (INT 21h) – установить область DTA: Ввод: АН = 1Ah,

DS:DX = адрес начала DTA (128-байтный буфер). -- Функция DOS 4Fh (INT 21h) – найти следующий файл:

Ввод:

Вывод:

АН = 4Fh,  
DTA должна содержать данные от предыдущего вызова функции 4Е или 4F.

Если CF = 1, то ошибка операции (в AX код ошибки).  
Если CF = 0, то операция выполнена успешно и область DTA заполняется данными о следующем найденном файле.

Особенности:Для продолжения поиска следует вызывать функцию 4Fh, пока

не будет возвращена ошибка.  
-- Функция DOS 39h (INT 21h) – создать каталог:

Ввод:

Вывод:

АН = 39h,  
DS:DX =адресASCIZ-строкиспутем,вкоторомвсекаталоги,кроме

последнего, существуют (для версии DOS 3.3 и более ранних длина всей строки не должна превышать 64 байта).

Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден,  
-- 05h – доступ запрещен.

Если CF = 0, то операция выполнена успешно.

-- Функция DOS 3Ah (INT 21h) – удалить каталог: Ввод: АН = 3Ah,

DS:DX = адрес ASCIZ-строки с путем, последний каталог в котором будет удален (только если он пустой, не является текущим, не занят

Вывод:

командой SUBST).  
Если CF = 1, то ошибка операции (в AX код ошибки):

-- 03h – путь не найден,  
-- 05h – доступ запрещен,  
-- 10h – текущий каталог нельзя удалить.

Ввод:

АН = 47h,  
DL = номер диска (00h — текущий, 01h — А: и т.д.),  
DS:SI = 64-байтный буфер для текущего пути (ASCIZ-строка без имени диска, первого и последнего символа «\»).  
Если CF = 1, то ошибка операции (в AX код ошибки):  
-- 0Fh – указан несуществующий диск.  
Если CF = 0 и АХ = 0100h, то операция выполнена успешно.

Если CF = 0, то операция выполнена успешно.  
-- Функция DOS 47h (INT 21h) – определить текущий каталог:

Вывод:  
-- Функция DOS 3Bh (INT 21h) – сменить каталог:

Ввод: Вывод:

АН = 3Bh,  
DS:DX = адрес 64-байтного ASCIZ-буфера с путем, который станет текущим каталогом.

Если CF = 1, то ошибка операции (в AX код ошибки): -- 03h – путь не найден.

Если CF = 0, то операция выполнена успешно.  
Фрагмент кода для чтения данных из файла и вывода этих данных на

экран:

...

f\_name db 'c:\data.txt',0 ; имя файла данных

...

; открытие файла

mov dx,offset f\_name ; адрес ASCIZ-строки с именем файла

mov ah,3Dh

mov al,00h

int 21h

jc exit

mov bx,ax

mov di,01

; функция DOS 3Dh

; 00 - только чтение

; открыть файл

; если ошибка – выход

; BX = идентификатор файла

; DI = идентификатор STDOUT

; чтение данных из файла и запись их в STDOUT

read\_data:

mov cx,1024 ; размер блока для чтения файла

mov dx,offset buffer ; буфер для данных

mov ah,3Fh

int 21h

jc close\_file

mov cx,ax

; Функция DOS 3Fh

; прочитать 1024 байта из файла

; если ошибка - закрыть файл

; CX = число прочитанных байт

jcxz close\_file ; если CX=0 - закрыть файл

mov ah,40h ; Функция DOS 40h

xchg bx,di ; BX = 1 - устройство STDOUT

int 21h ; вывод данных в STDOUT

xchg di,bx ; BX = идентификатор файла

jc close\_file ; если ошибка - закрыть файл

jmp short read\_data ; вывод следующей порции данных файла

; закрытие файла

close\_file:

mov ah,3Eh ; функция DOS 3Eh

int 21h ; закрыть файл

; завершение программы

exit:  
 ...

; буфер данных для работы с файлом

buffer:

2. Работа с командной строкой.

Передача параметров программы через командную строку при запуске программы – решение, которое позволяет построить гибкую по входным дан- ным программу без дополнительных диалогов с пользователем.

При запуске программы DOS помещает всю командную строку (включая последний символ 0Dh) в блок PSP запущенной программы по смещению 81h и ее длину в байт 80h. Длина командной строки, хранящейся в PSP, не может быть больше 126 символов (командная строка большей длины доступна начи- ная с DOS 4.0 в переменной среды CMDLINE).

При загрузке программы, в начале отводимого для нее блока памяти, со- здается структура данных PSP (префикс программного сегмента) размером 256 байт (100h). Затем DOS создает копию текущего окружения для загружаемой программы, помещает полный путь и имя программы в конец окружения, за- полняет поля PSP, сама программа записывается в память, начиная с адреса PSP:0100h.

При запуске com-программы регистры устанавливаются следующим об- разом:

1) AL = FFh, если первый параметр командной строки содержит непра- вильное имя диска (например, z:/something), иначе – AL = 00h.

2) АН = FFh, если второй параметр содержит неправильное имя диска, иначеАН = 00h.

3) CS = DS = ES = SS = сегментный адрес PSP.

4) SP = адрес последнего слова в сегменте (обычно FFFEh или меньше, ес- ли не хватает памяти).

При запуске exe-программы регистры SS:SP устанавливаются в соответ- ствии с сегментом стека, определенным в программе.

Затем в стек помещается слово 0000h и выполняется переход на начало программы (PSP:0100h для com, собственная точка входа для exe). Описание полей блока PSP приведено в таблице 4.



Ниже приводится фрагмент com-программы для работы с командной строкой:

.model tiny

.code

org 80h

cmd\_length db ?

cmd\_line db ?

; смещение 80h от начала PSP

; длина командной строки

; сама командная строка

org 100h

start:

; начало программы 100h от начала PSP

cld  
mov bp,sp

mov cl,cmd\_length

cmp cl,1

jle exit

; для команд строковой обработки

; сохранить текущую вершину стека в ВР

; проверка длины командной строки

; выход из программы

; преобразовать список параметров в PSP так, чтобы

; каждый параметр заканчивался нулем (ASCIZ-строка)

; адреса всех параметров поместить в стек

; в переменную argc записать число параметров

mov cx,-1 ; для команд работы со строками

mov di,offset cmd\_line ; начало командной строки в ES:DI

find\_param:

mov al,' '

; AL = пробел

; искать не пробел

; DI = адрес начала параметра

; сохранить адрес в стек

repz scasb

dec di

push di

inc word ptr argc ; увеличить argc на 1

mov si,di ; SI = DI для следующей команды lodsb

scan\_params:

lodsb ; прочитать символ из параметра

cmp al,0Dh ; если 0Dh - последний параметр

je params\_ended ; параметры закончились

cmp al,20h ; сравнение с пробелом

jne scan\_params ; текущий параметр не закончился

dec si ; SI = первый байт после параметра

mov byte ptr [si],0 ; записать в него 0

mov di,si ; DI = SI для команды scasb

inc di ; DI = следующий после нуля символ

jmp short find\_param ; продолжить разбор командной строки

params\_ended:

dec si ; SI = первый байт после конца

exit:

; последнего параметра

mov byte ptr [si],0 ; записать в него 0

; работа с параметрами

...

ret ; конец программы

; данные программы

argc dw 0 ; число параметров

end start

Код программы:

.model small

.stack 100h

.data

file db "c:\lw5\strings.txt", 0

handle dw ?

result db 10, 13, 10, 13, "The number of blank lines in the file: ", '$'

buffer db ?

counter dw ?

ten dw 10

.code

start:

mov ax, @data

mov ds, ax

open\_file:

mov ah, 3dh

mov al, 0

mov dx, offset file

int 21h

jc error

mov handle, ax

read\_the\_byte:

mov bx, handle

mov ah, 3fh

mov cx, 1h

mov dx, offset buffer

int 21h

print\_the\_byte:

push ax

mov dl, buffer

mov ah, 02h

int 21h

pop ax

cmp ax, 0 ;if the end of the file is reached

je print\_the\_result

cmp buffer, 0ah ;if '\n' is met

je possible

jmp read\_the\_byte

possible:

mov bx, handle

mov ah, 3fh

mov cx, 1h

mov dx, offset buffer ;check, whether the next line is blank

int 21h

mov ah, 42h

mov al, 1

mov cx, 0

mov dx, -1

int 21h

cmp buffer, 0dh

jne read\_the\_byte

found:

inc counter

jmp read\_the\_byte

print\_the\_result:

mov ah, 9h

lea dx, result

int 21h

mov ax, counter

call print

close\_file:

mov ah, 3eh

mov bx, handle

int 21h

exit:

mov ax, 4c00h

int 21h

ret

error:

nop

print proc near

cmp ax, 0 ;if a quotient is 0

jne not\_zero

add al, '0' ;printing '0' out

mov ah, 0eh

int 10h

ret

not\_zero:

push ax

push bx

push cx

push dx

mov bx, 10000

begin:

cmp bx, 0 ;if devider equals zero

jz end

cmp cx, 0 ;avoid printing zeros before numbers

je calc

cmp ax, bx ;if ax<bx, the result is 0

jb skip

calc:

xor cx, cx

xor dx, dx ;remainder

div bx

add al, 30h ;printing the last digit

mov ah, 0eh

int 10h

mov ax, dx ;remainder is stored in ax

skip: ;bx/=10

push ax

xor dx, dx

mov ax, bx

div ten

mov bx, ax

pop ax

jmp begin

end:

pop dx

pop cx

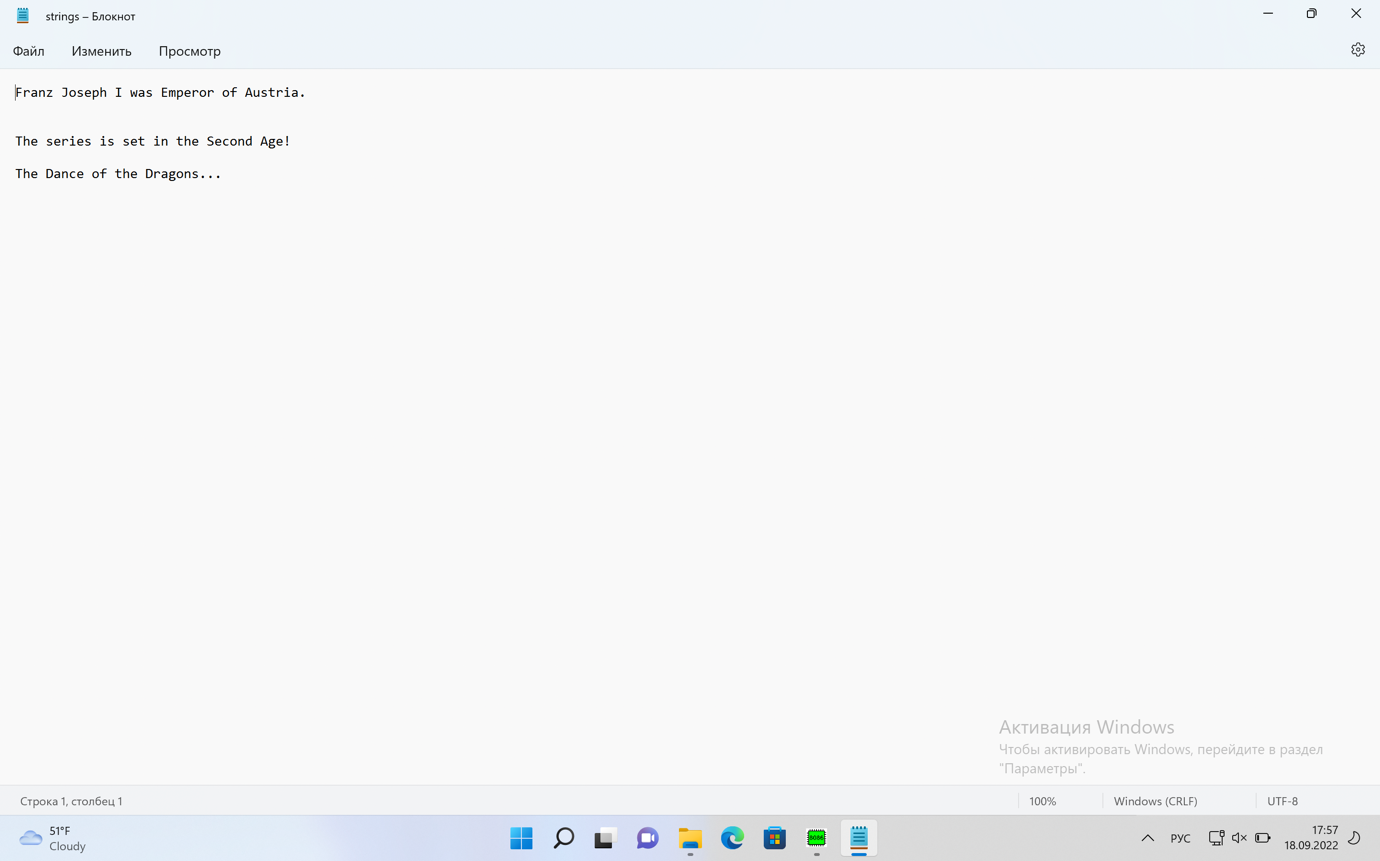
pop bx

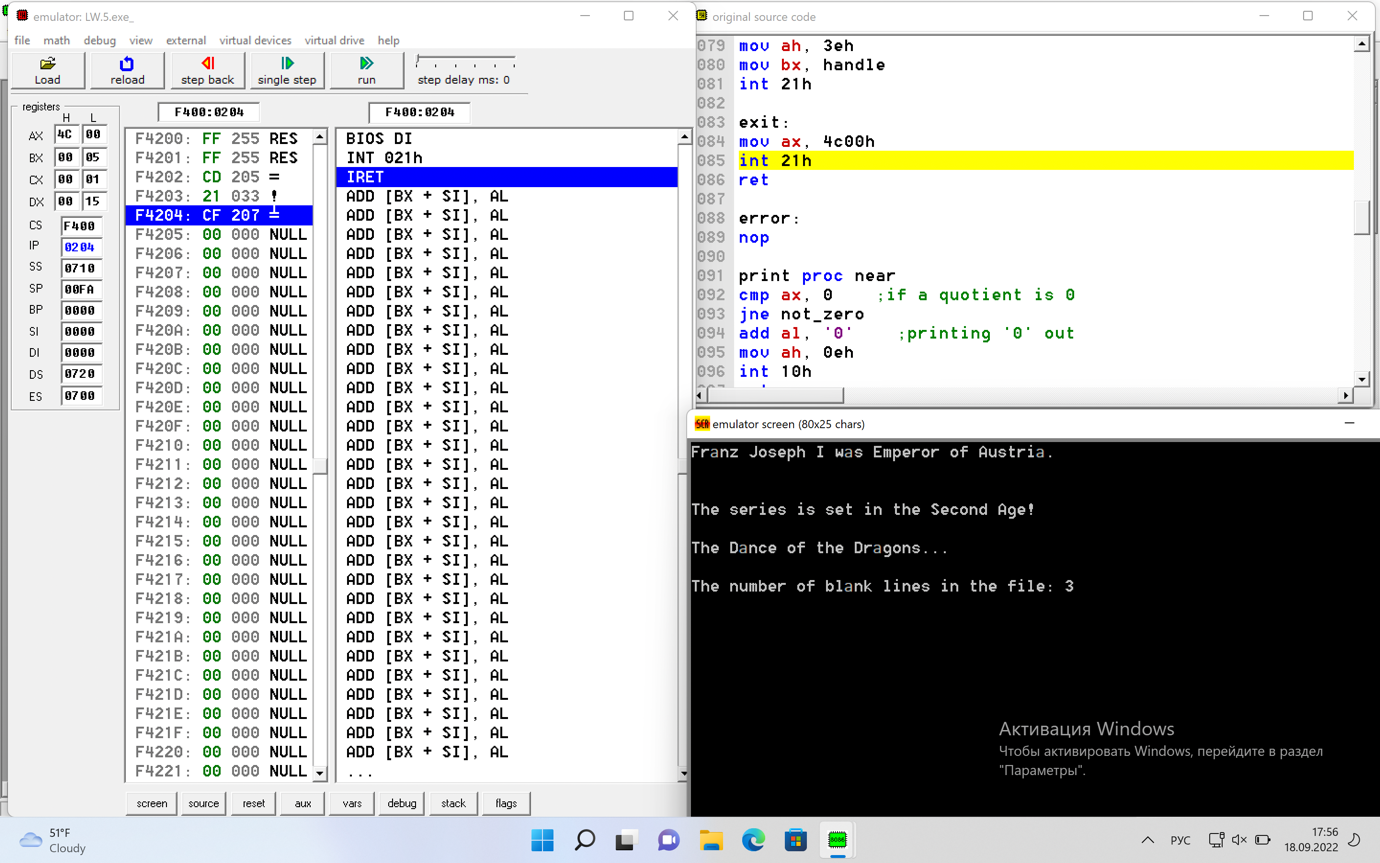
pop ax

ret

print endp

Примеры работы программы:





Вывод:

В ходе лабораторной работы было написано приложение, выполняющее пустых строк, содержащихся в файле. Я ознакомилась с основными операциями обработки файлов, получила понятие о работе с параметрами командной строки.