Дополнение к « Инструктивно-справочным материалам
по практикуму на MASM’е. Часть 1 » от 2012 г.

Инструкция для первого выхода на машину

Матвеева Т.К., Матвеев Ф.В.

март 20**14**

Инструкция первого выхода на машину с программой на ассемблере (март 2014).

I. ВЗЯТИЕ файла zip–архива на рабочий стол.		
1.	Запускаем **Internet–браузер**. Например, Microsoft Internet Explorer двойным кликом левой клавишей мыши на рабочем столе (2L)	см. рис.1а.
2.	http:// al.cs.msu.su / – адрес сайта кафедры АЯ набираем в адресной строке	см. рис.1б
	и “переходим” на сайт кафедры АЯ, нажав ➔ или Enter↵	см. рис.2.
3.	Пункт Материалы по ассемблеру выбираем в Карте сайта (слева) в пункте Учебная работа. Получаем	рис.3.
	Находим текст « *Для написания ассемблерных программ в среде Windows можно скачать* **архив**, *содержащий, кроме транслятора, редактор Notepad++* ». Для скачивания архива нажимаем правой клавишей мыши на гиперссылку архив (1R). В появившемся контексном меню выбираем пункт **“Сохранить объект как ..”**	см. рис.3
	Начинается процесс загрузки файла архива *assembler.zip.* *(по завершении загрузки файл assembler.zip появится на рабочем столе)* Откроется окно “динамики процесса загрузки”. Поверх него откроется окно для выбора места размещения сохраняемого файла архива assembler.zip	см. рис.4а
	Выбираем “куда сохранять” – **“На рабочий стол”** (и под каким именем).	
	Теперь нажимаем кнопку **“Сохранить”** (справа внизу)	см. рис.4б.
	Далее, окно “динамики процесса загрузки” примет окончательный вид	см. рис.5а.
	Файл архива *assembler.zip* появился на рабочем столе.	см рис.5б
II РАСПАКОВКА файла zip–архива на рабочий стол.		
4.	В окне “динамики процесса загрузки” видим, что “Загрузка завершена” и стали активными **кнопки действий.** Нажимаем кнопку «**Открыть**» (2L)	см рис.5б,в
	Запустится (откроется) окно WinRAR’а для работы с файлом архива (assembler.zip) *(видим, что в архиве находится папка Assembler)*	см. рис.6
5.	Поместим **на рабочий стол папку Assembler** со всеми ее подпапками. **Нажмём** в окне WinRAR на кнопку **« Извлечь в » .**	см. рис.6
	Уточним предложенный путь, указав что папку **Assembler** следует разместить на рабочий стол.	см. рис.7а
	Нажмем кнопку **«ОК»**, согласившись со стандартными параметрами извлечения содержимого архива.	см. рис.7б
	Папка **Assembler** на рабочем столе. *Можно закрыть окна менеджера работы с архивами WinRAR и браузера Microsoft Internet Explorer.* *Полезно сохранить файл архива assembler.zip на личном диске.*	см. рис.8

III СОДЕРЖИМОЕ ПАПКИ ASSEMBLER.		
6.	Открываем папку Assembler двойным кликом (2L) на рабочем столе. папка Assembler содержит папки: masm и npp ; файлы: **assembler.cmd** – комплексная “запускалка” ; io.asm, ioproc.asm, ioproc.obj – файлы для ввода/вывода; readme.txt – *сопроводительный текст - можно прочесть о разном*; **schem.asm** – основа для создания ассемблерной программы.	см рис.9а,б
IV. ПОЛУЧЕНИЕ программного файла и работа с ним.		
7.	Запускаем командный файл assembler (*.cmd) двойным кликом (2L). Получаем окно текстового редактора Notepad++, в котором открыт файл schem (*.asm).	см рис.10
8.	Устанавливаем режим работы Notepad++ с нужной кодировкой OEM866 (DOS кодировка), для чего выполним пункт в меню Кодировки - Кодировки - Кириллица - OEM866. [1]*)	см рис.11, рис.12.
9.	Сохраняем schem.asm под другим именем в Notepad++, для чего выполняем последовательно «Файл – Сохранить как	см рис.13.
	– новое имя (pr1) – Сохранить»	см рис.14.
	В результате в Notepad++ новый файл будет открыт в той же закладке, где был schem.asm	см рис.15.
10.	Редактируем программный файл	рис.16,17.
11.	Запускаем программу на ассемблере MASM-4 из текущего окна Notepad++ . Например, нажатием клавиш Ctrl + F9 [2] **).	см рис.18.

[1] В приложении 1 описано, как можно настроить комбинацию “горячих клавиш”, для убыстрения таких часто выполняемых действий, как выбор кодировки OEM866.

[2] В приложении 2 показана настройка Notepad++ для использования MASM 4.0.

V. Дальнейшая работа с программными файлами.		
12.	При новом вызове командного файла assembler.cmd «каркас» новой программы **schem.asm** открывается в дополнительной закладке. Для получения файла следующей программы, например, pr2.asm, следует повторить пп.7–11.	см.рис.19, 20,22,23 и рис.26
	Если при компиляции обнаружены ошибки, то в строке закладок появится одноименный файл с расширением **.lst**. В нем ошибки изучаются. Внимание! Изменения вносятся в asm-файл и программа повторно запускается на выполнение.	см рис.24
	Если во время выполнения ассемблерной программы требуется вводить данные на русском языке переключение кодировок следует производить при помощи правого Ctrl . См. readme.txt на рис.9б .	
	Если при закрытии Notepad++ несколько файлов оставались открытыми, то они откротся при новом вызове командного файла.	
13.	В папке Assembler сохраняются все (открытые и закрытые в редакторе) файлы.	см.рис.25

Можно использовать разные интернет браузеры 

, но 

есть всегда.

Для распаковки (*.zip) архива можно использовать разные средства, включая мастер, входящий в состав среды Windows, но WinRAR.exe в компьютерном классе по умолчанию.

1. **Запускаем Internet–браузер**.

Например, Microsoft Internet Explorer двойным кликом левой клавишей мыши на рабочем столе (2L) см. рис.1а

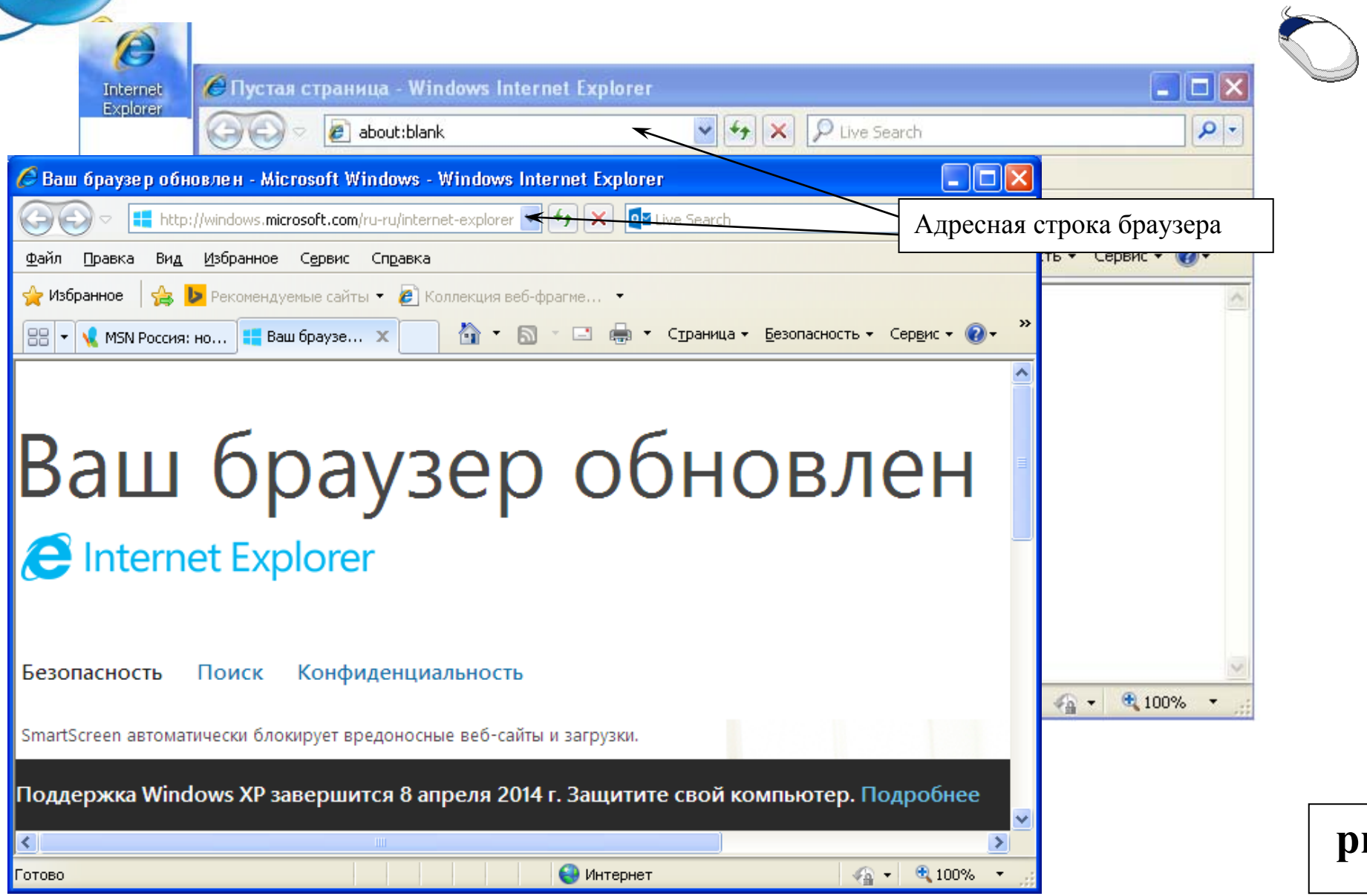


рис.1а.

2. http:// **al.cs.msu.su** / – **адрес сайта кафедры АЯ** набираем в адресной строке см. рис.1б. и “переходим” на сайт кафедры АЯ, нажав кнопку → или клавишу Enter ↵. Получаем рис.2.

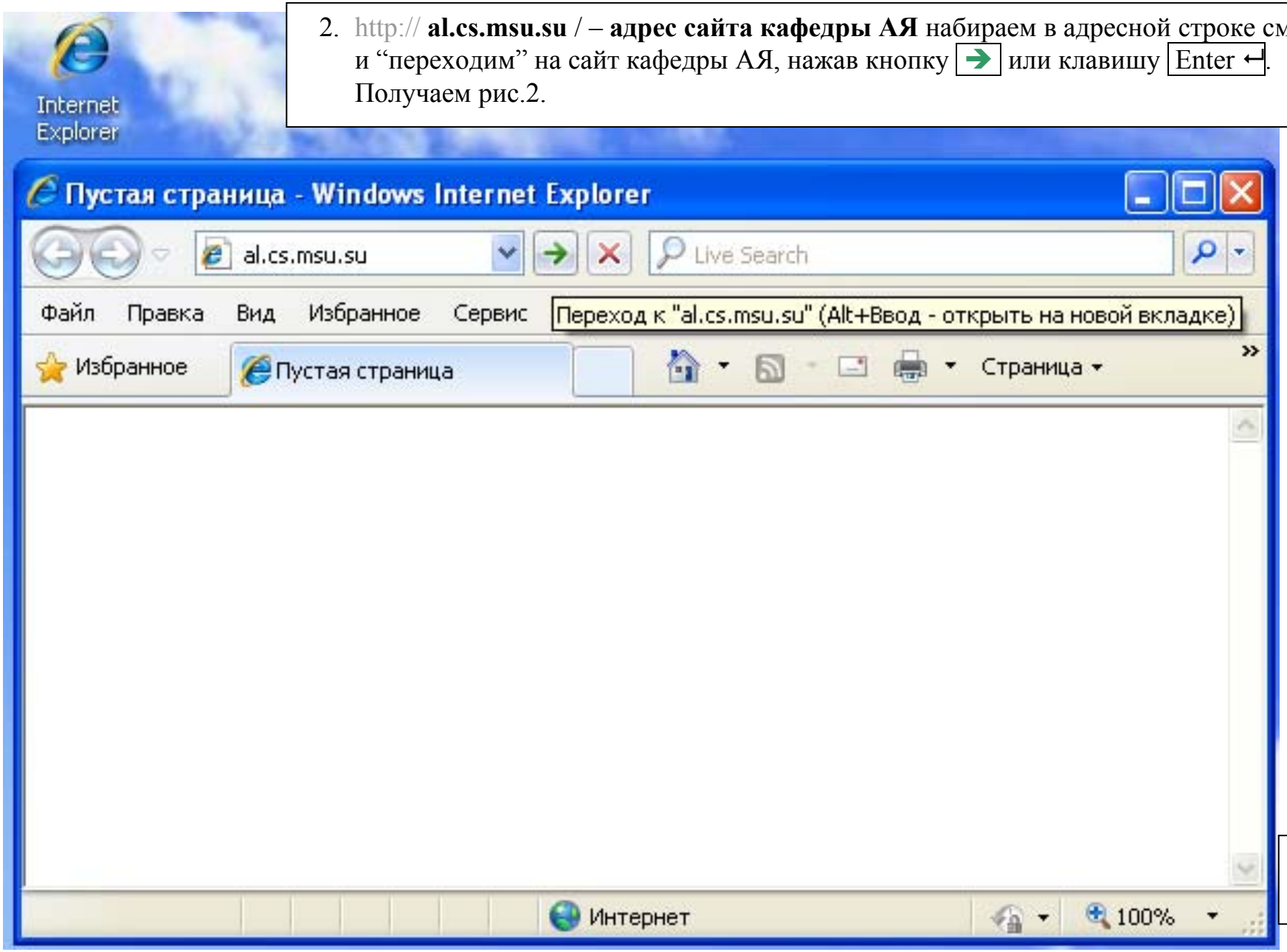
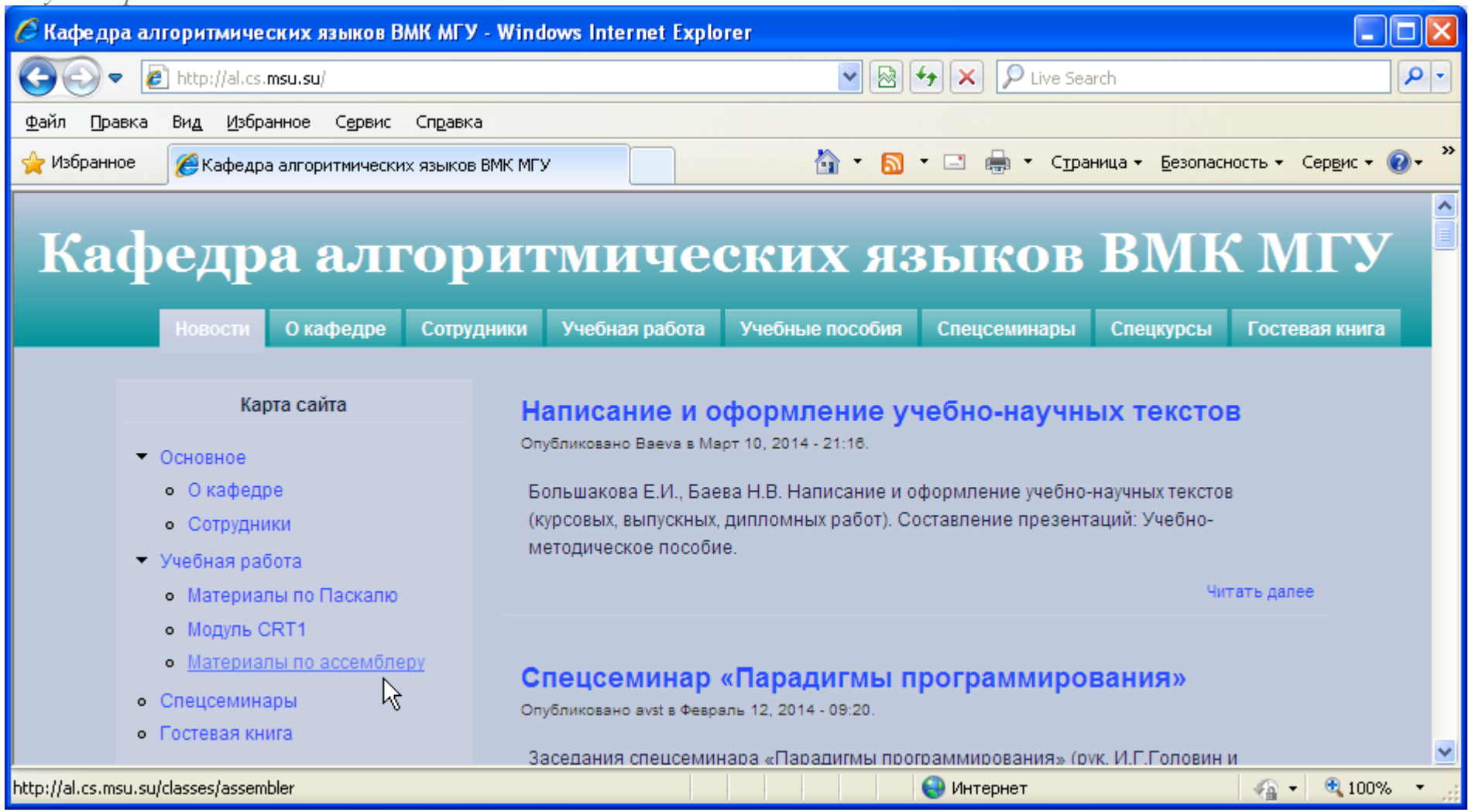


рис.1б.

получаем рис.2.

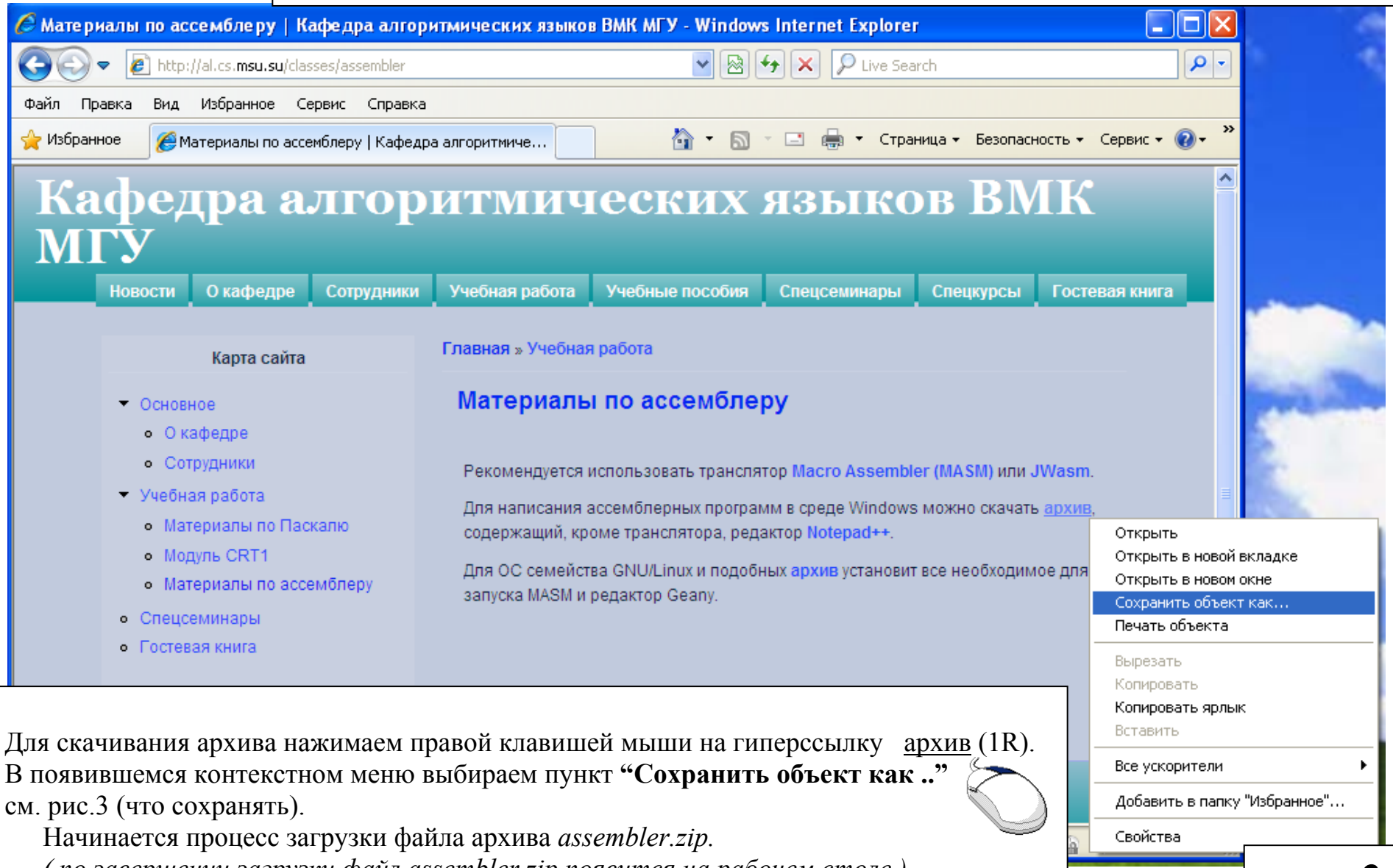


3. **Пункт <u>Материалы по ассемблеру</u>**
выбираем в Карте сайта (слева) в пункте Учебная работа.
Получаем рис.3.

рис.2.

Получаем рис.3.

Находим текст « Для написания ассемблерных программ в среде Windows можно скачать **архив**, содержащий, кроме транслятора, редактор Notepad++ ».



Для скачивания архива нажимаем правой клавишей мыши на гиперссылку архив (1R).
В появившемся контекстном меню выбираем пункт **“Сохранить объект как ..”** см. рис.3 (что сохранять).

Начинается процесс загрузки файла архива *assembler.zip.*

(по завершении загрузки файл assembler.zip появится на рабочем столе)

рис.3.

Откроется окно “динамики процесса загрузки”. Поверх него откроется окно “Сохранить как” для выбора места размещения сохраняемого файла архива assembler.zip см. рис.4а .

Выбираем “куда сохранять” – **“На рабочий стол”** (и под каким именем).

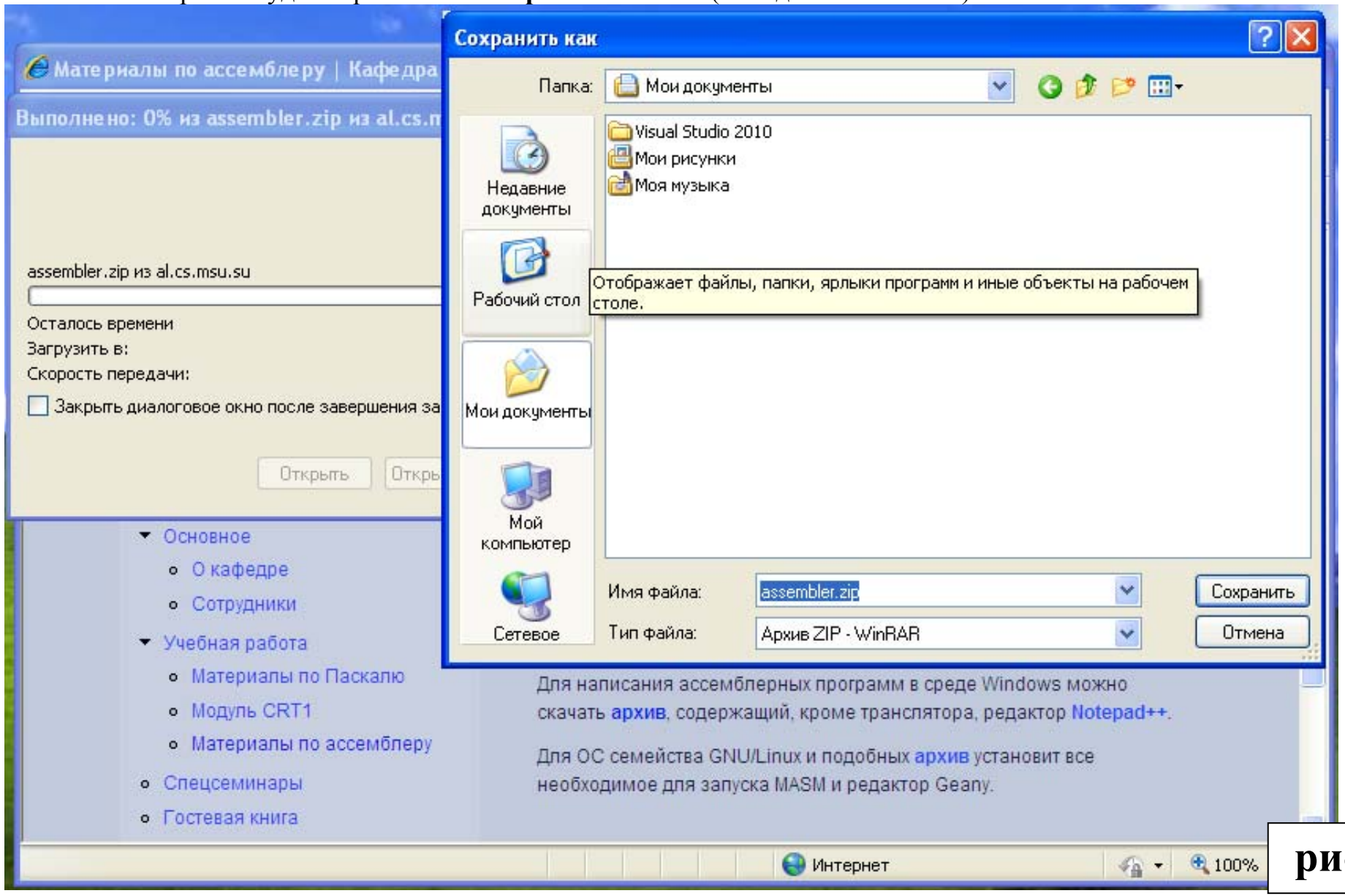


рис.4а.

Теперь нажмем кнопку **“Сохранить”** (справа внизу) см. рис.4б .

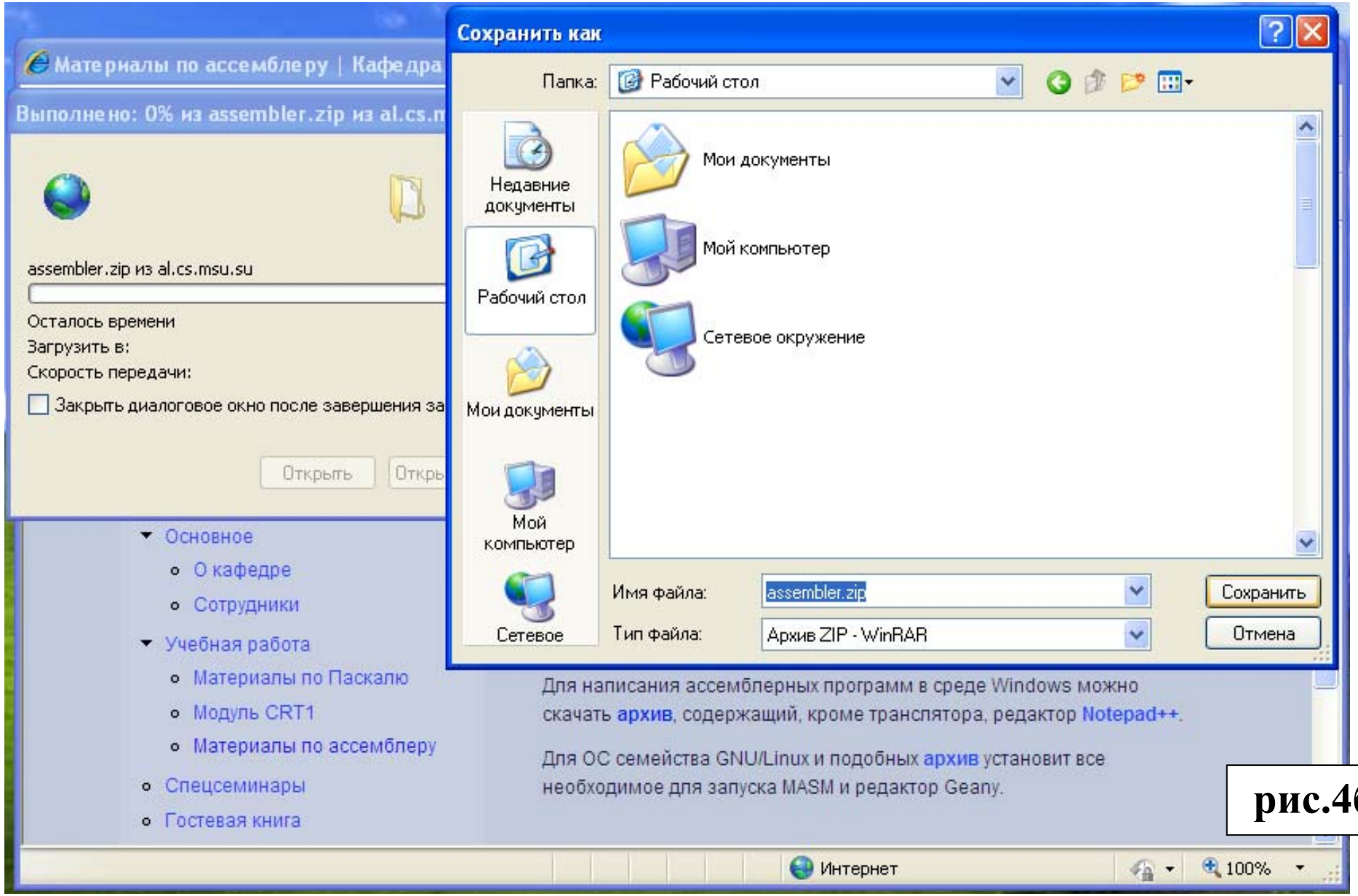


рис.4б.

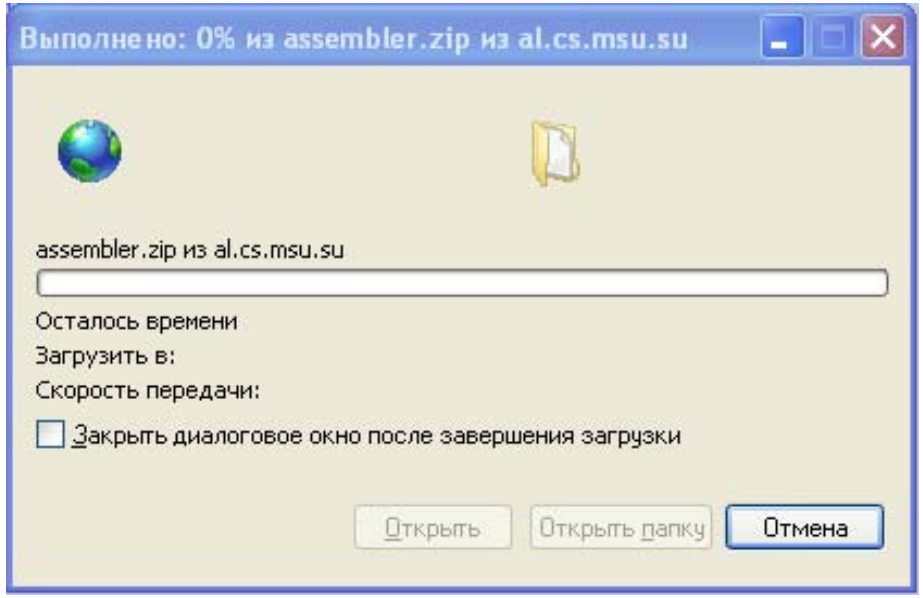


рис.5а.

Далее, окно “динамики процесса загрузки” см. рис.5а примет окончательный вид см. рис.5б “Загрузка завершена”.

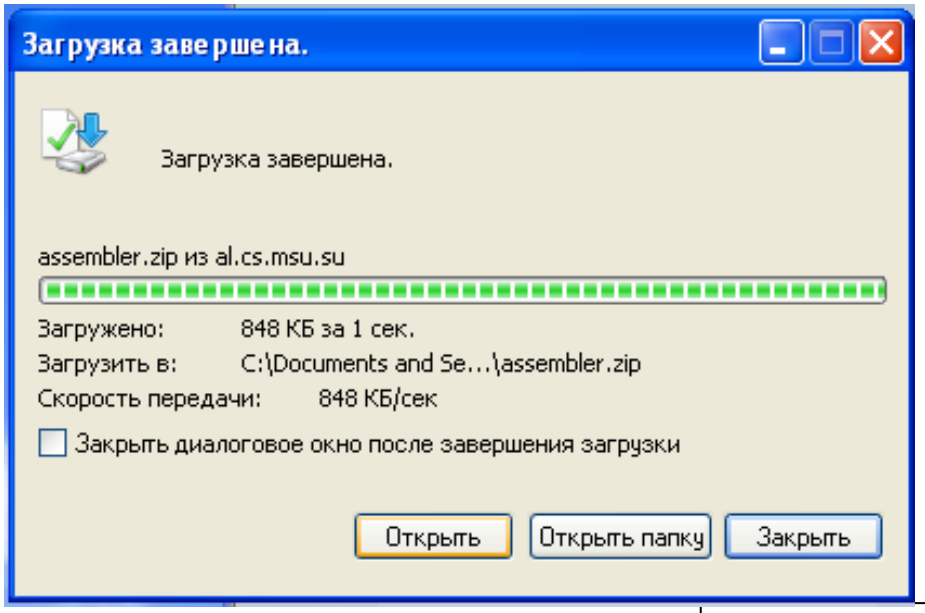
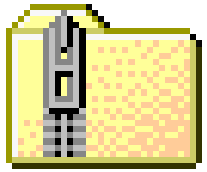


рис.5б.

Файл архива *assembler.zip* появился на рабочем столе.



Стали активными **кнопки действий.** Воспользуемся кнопкой «открыть» (2L)

assembler.zip (или assembler.zip) . ,

4. Файл архива *assembler.zip* находится на рабочем столе.
В окне “динамики процесса загрузки” видим, что “Загрузка завершена” и активны **кнопки действий** см. рис.5в**.**
Нажимаем кнопку «**Открыть**» (2L)

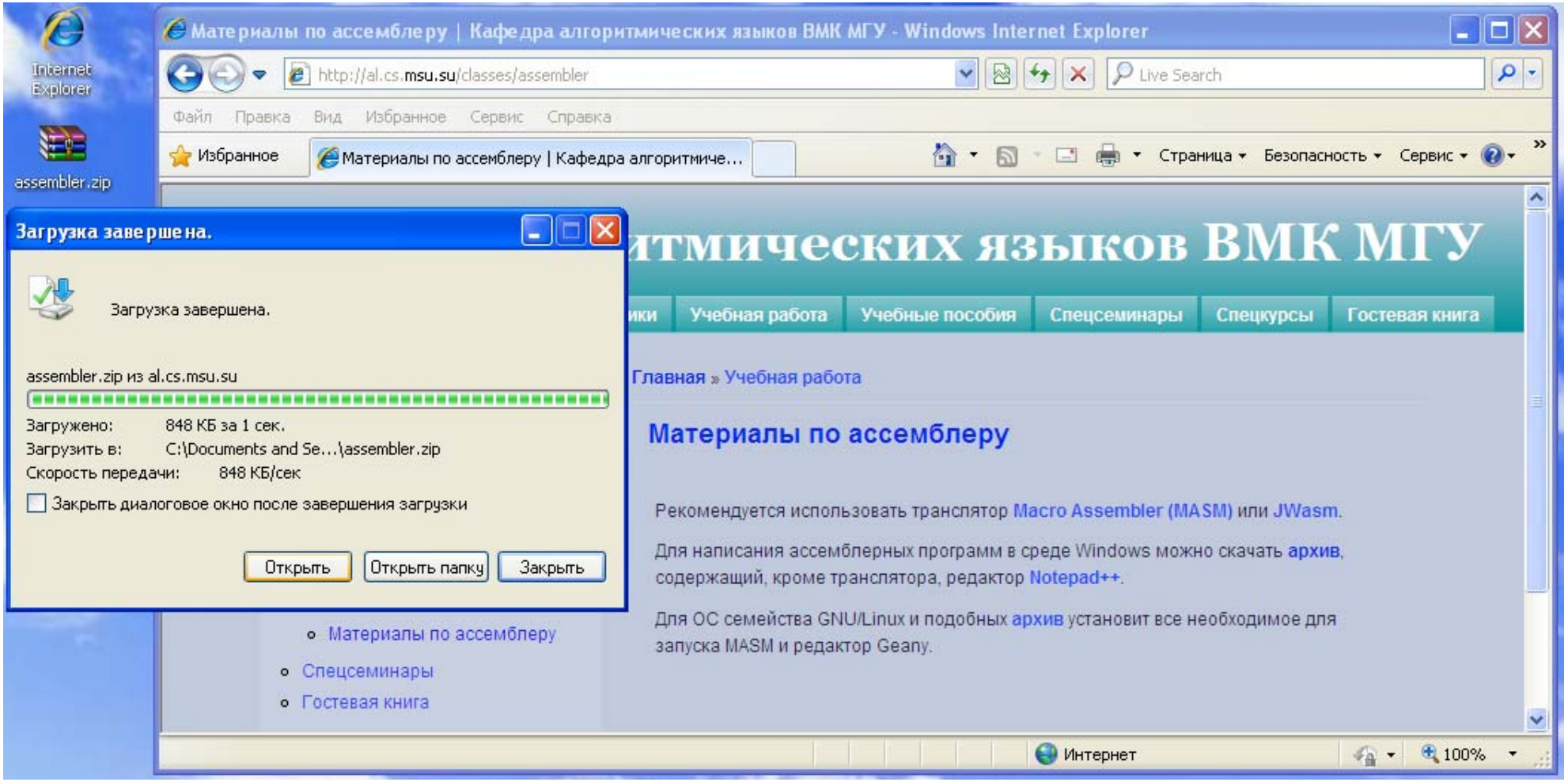


рис.5в.

Запустится (откроется окно) WinRAR для работы с файлом архива (assembler.zip) см. рис.6
(видим, что в архиве находится папка Assembler).

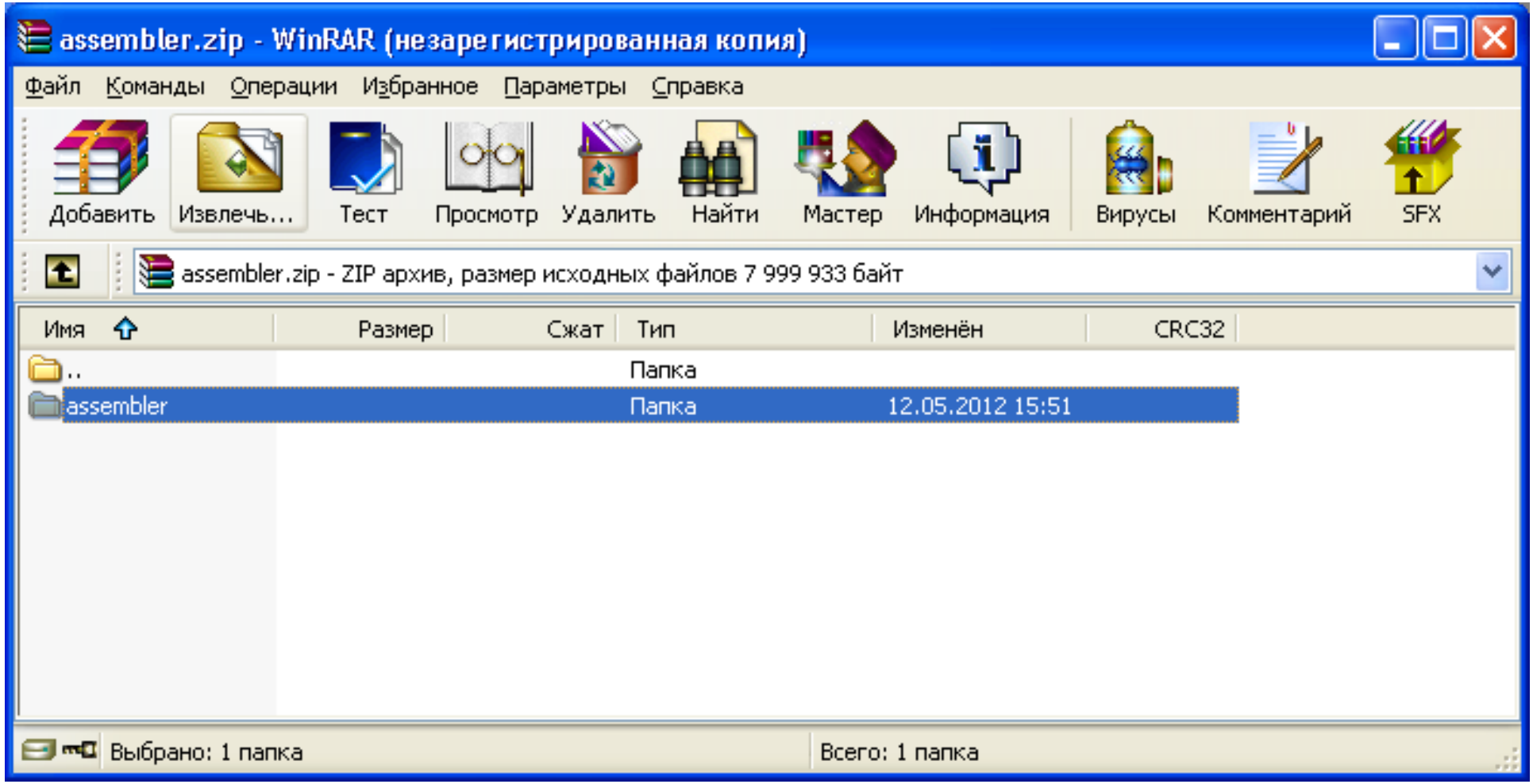


рис.6.

Воспользуемся кнопкой **« Извлечь в »**, чтобы поместить **на рабочий стол папку Assembler** со всеми ее подпапками.

5. **Поместим на рабочий стол** содержащийся в архиве папку со всеми его под папками.
Нажмём в окне WinRAR на кнопку **«Извлечь в»** см. рис.6 . В окне, изображённом на рис.7а (слева),
уточним путь – куда следует разместить папку **Assembler**. Результат показан на рис.7б (справа).
Нажмем кнопку **«ОК»**, согласившись со стандартными параметрами извлечения содержимого архива.

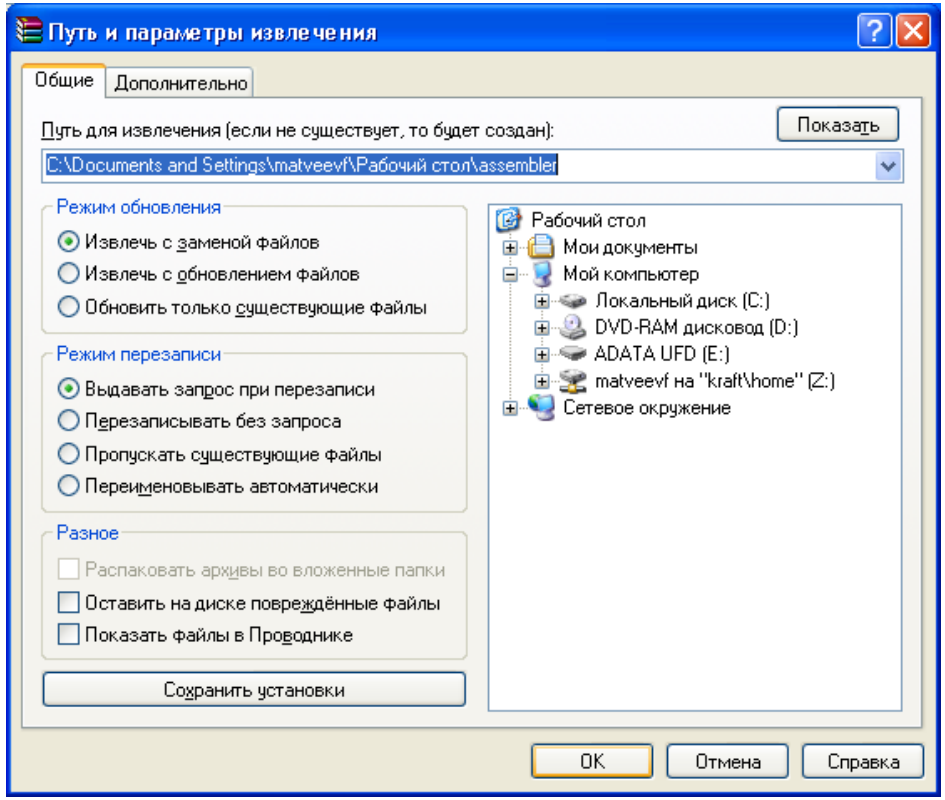


рис.7а.

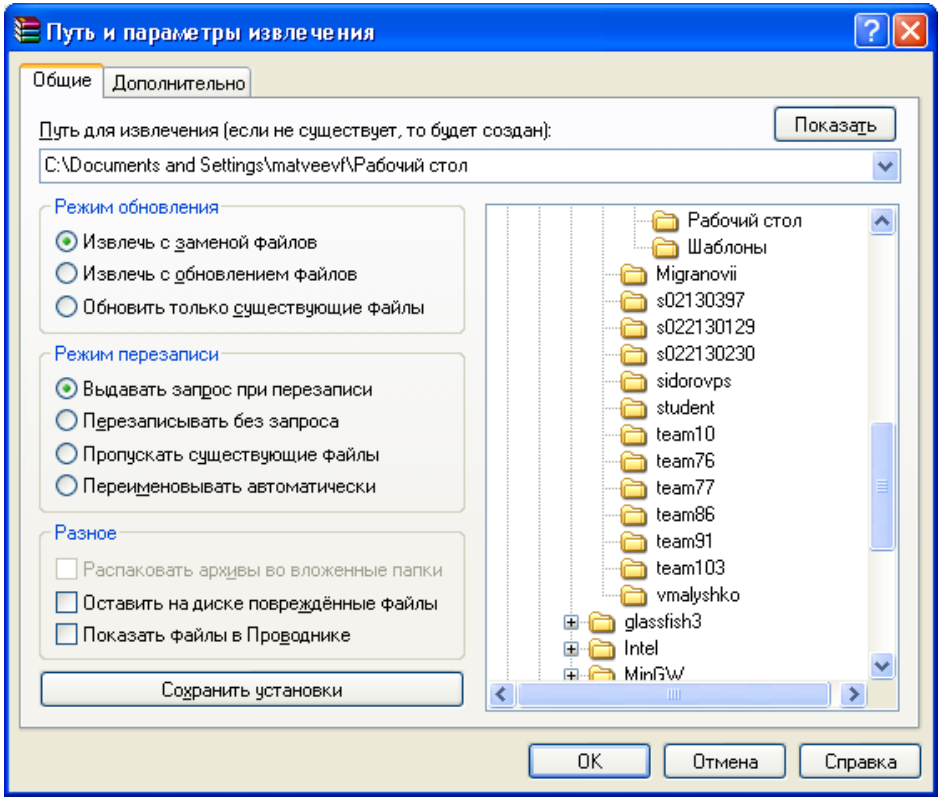
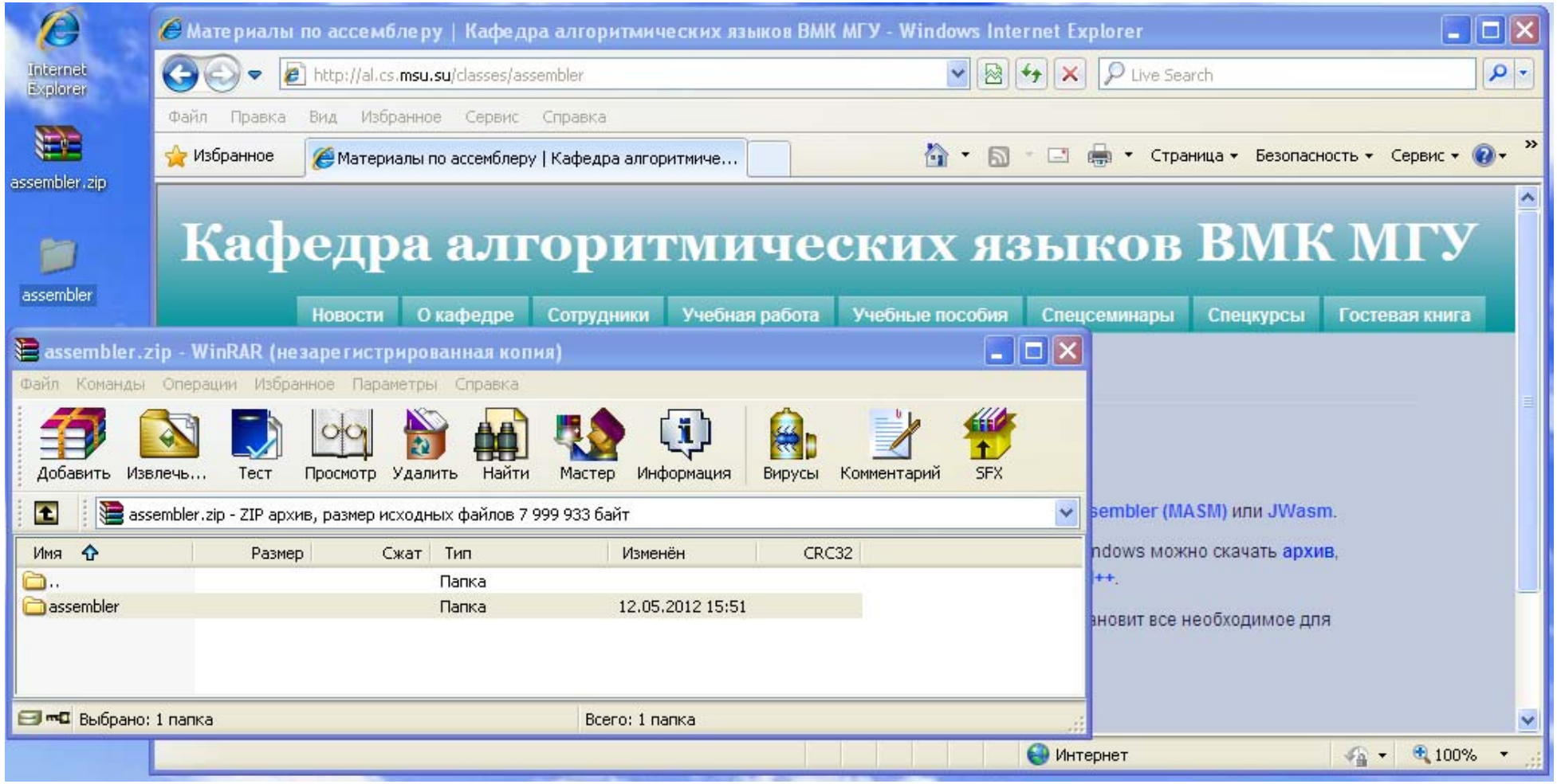


рис.7б.

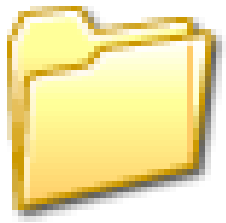
Папка **Assembler** на рабочем столе.

Можно закрыть окна менеджера работы с архивами WinRAR и браузера Microsoft Internet Explorer .



Полезно сохранить файл архива assembler.zip на личном диске.

рис.8.



assembler

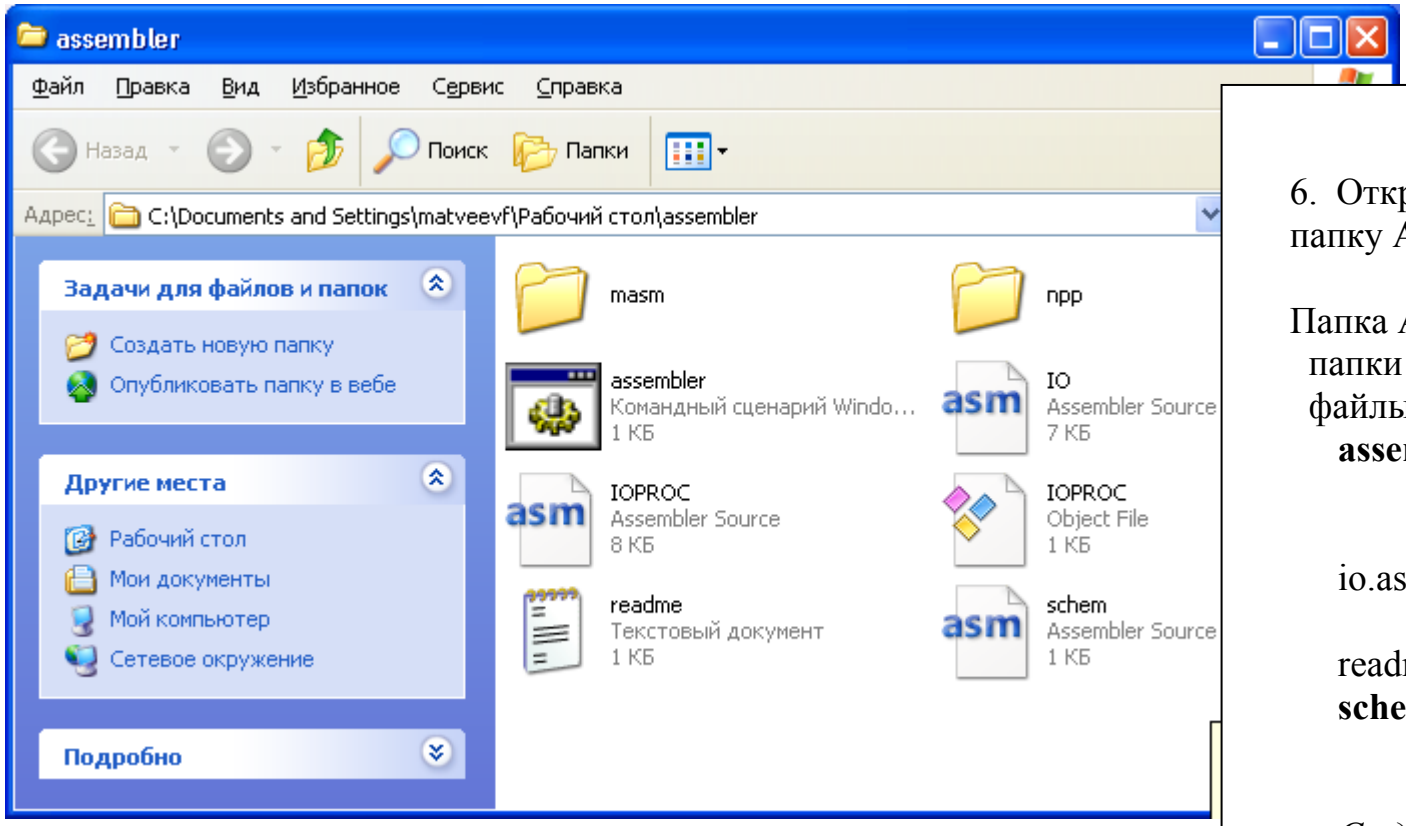


рис.9а.

6. Открываем на рабочем столе папку Assembler двойным кликом (2L) .

Папка Assembler содержит
папки : masm и npp ;
файлы :
assembler.cmd – для запуска Notepad++, настроенного для работы с ассемблером ;
io.asm, ioproc.asm, ioproc.obj – файлы для ввода/вывода ;
readme.txt –*см.* рис.9б ;
schem.asm – основа для создания ассемблерной программы .

Сюда же по умолчанию будут сохраняться файлы разрабатываемых программ

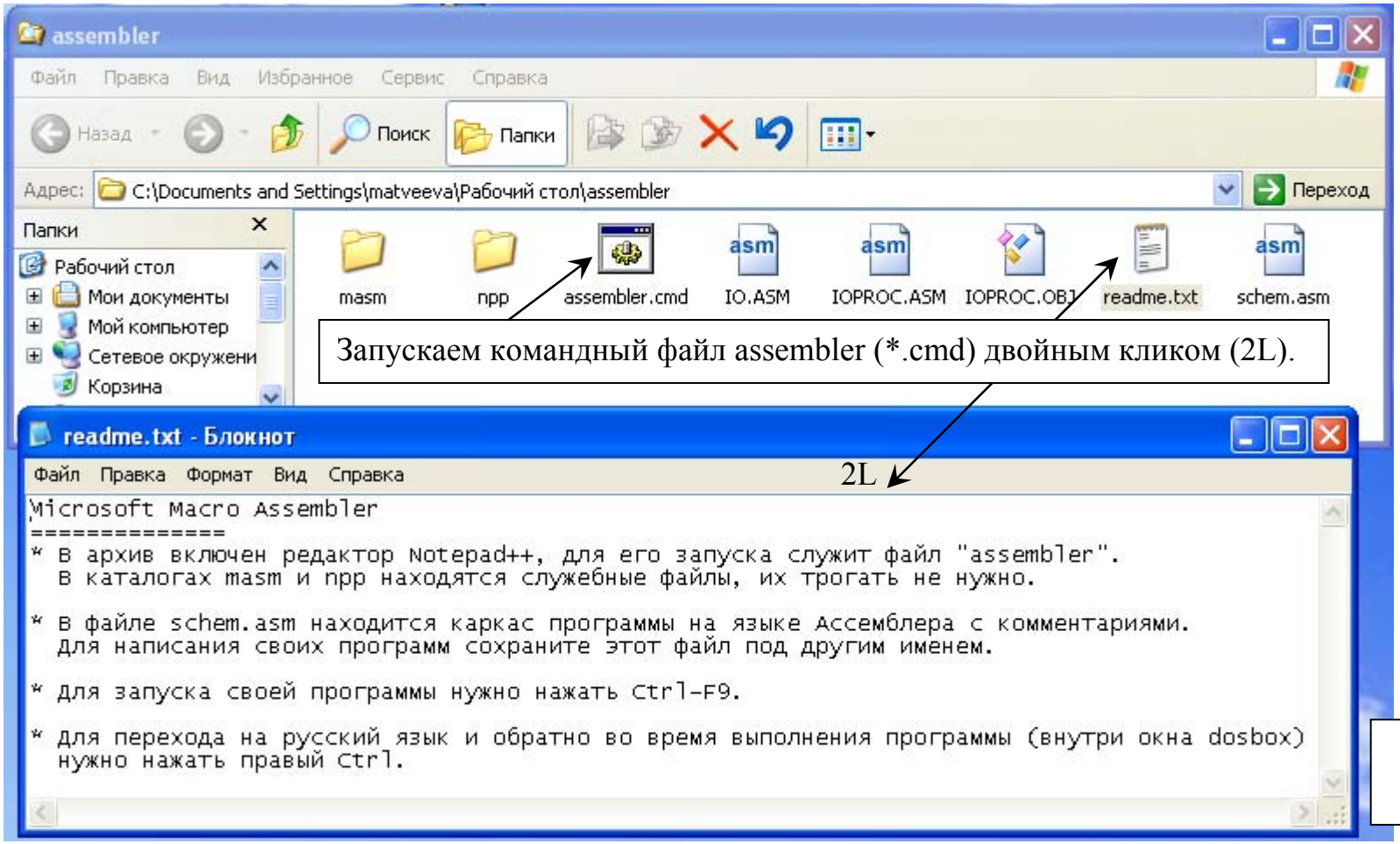

assembler
Файл Правка Вид Избранное Сервис Справка
Назад Поиск Папки
Адрес: C:\Documents and Settings\matveeva\Рабочий стол\assembler
Переход
Папки
Рабочий стол
Мои документы
Мой компьютер
Сетевое окружени
Корзина
masm
npp
assembler.cmd
IO.ASM
IOPROC.ASM
IOPROC.OBJ
readme.txt
schem.asm
Запускаем командный файл assembler (*.cmd) двойным кликом (2L).
2L
readme.txt - Блокнот
Файл Правка Формат Вид Справка
Microsoft Macro Assembler
==========
* В архив включен редактор Notepad++, для его запуска служит файл "assembler".
В каталогах masm и npp находятся служебные файлы, их трогать не нужно.
* В файле schem.asm находится каркас программы на языке Ассемблера с комментариями.
Для написания своих программ сохраните этот файл под другим именем.
* Для запуска своей программы нужно нажать Ctrl-F9.
* Для перехода на русский язык и обратно во время выполнения программы (внутри окна dosbox)
нужно нажать правый Ctrl.

рис.9б.

Запустив командный файл assembler (*.cmd) двойным кликом (2L) ; **получаем окно текстового редактора Notepad++,**

в котором открыт файл schem (*.asm)

см. рис. 10 .

В нижней (статусной) строке окна видим, что по умолчанию предполагается использование в открытом файле кодировки ANSI, принятой в Windows.

В комментариях файла schem.asm используется OEM 866 (кодировка DOS).

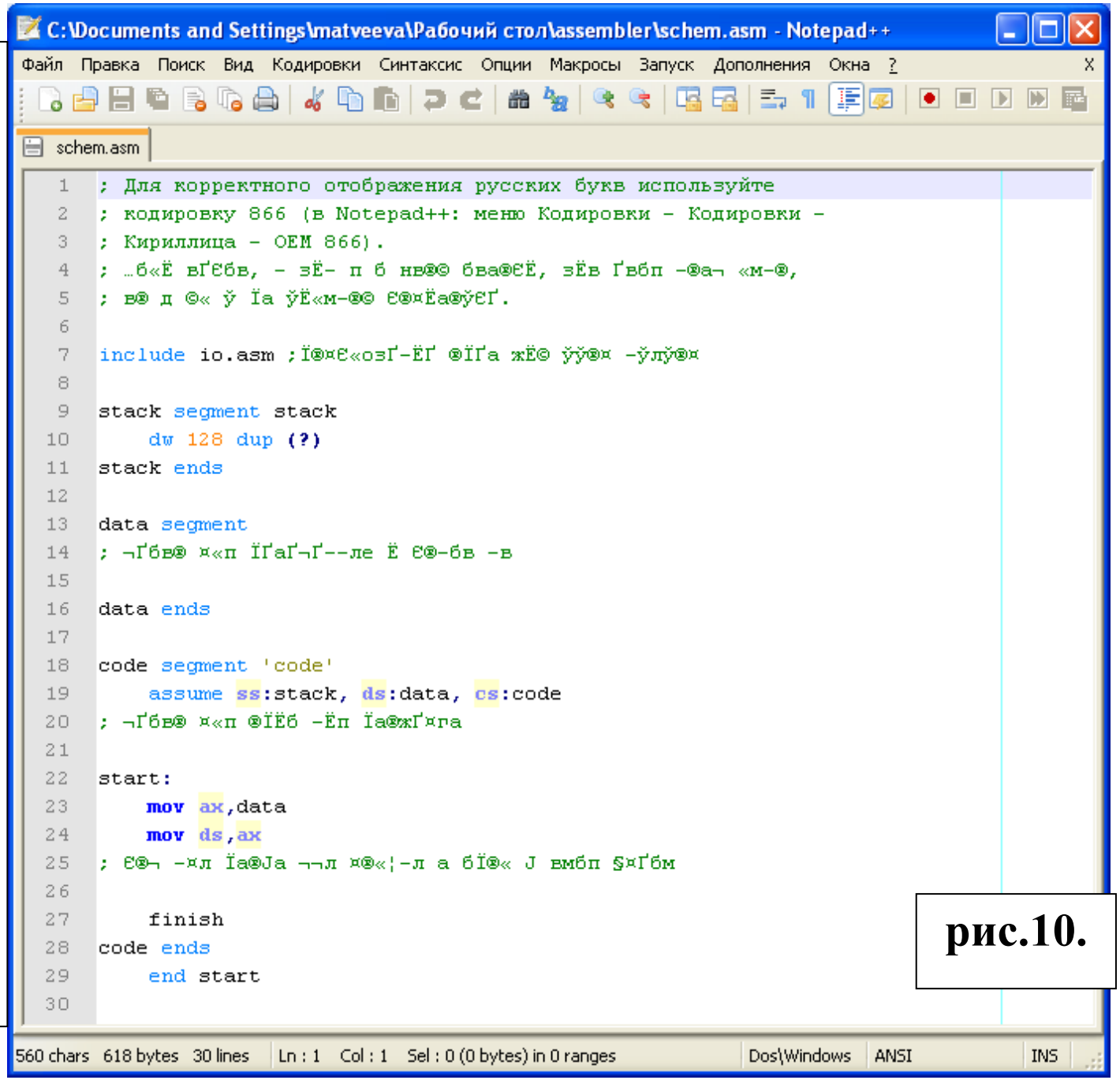


рис.10.

Устанавливаем режим работы Notepad++ с нужной кодировкой OEM866, для чего
в меню **Кодировки**
подменю **Кодировки**
в котором подпункт **Кириллица** открывает подменю, содержащее пункт **OEM866**
см рис.11.

Вызовем пункт **OEM866** одним нажатием левой клавиши мыши (1L).

Результат показан на рис.12.

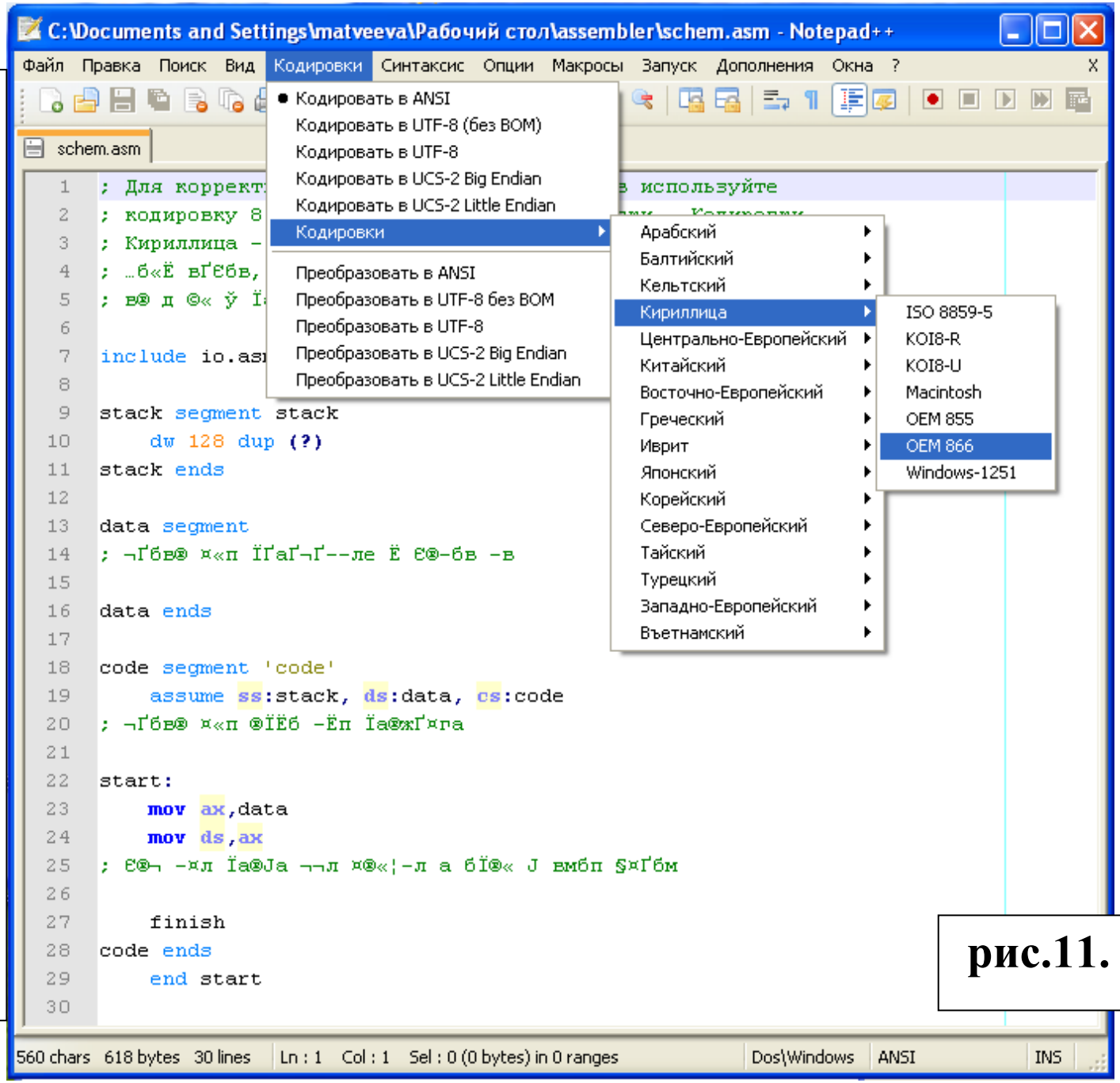


рис.11.

На рис.12 показан результат переключения Notepad++ в режим работы с кодировкой **OEM866** .

В нижней (статусной) строке окна видим, что “текущая” кодировка сменилась на **OEM866** .

Отображение латинских букв не изменилось. Комментарии, написанные в кодировке **OEM866** на русском языке, стали читаемыми.

Перестал быть читаемым только комментарий в первых трёх строках файла, написанный в кодировке ANSI.

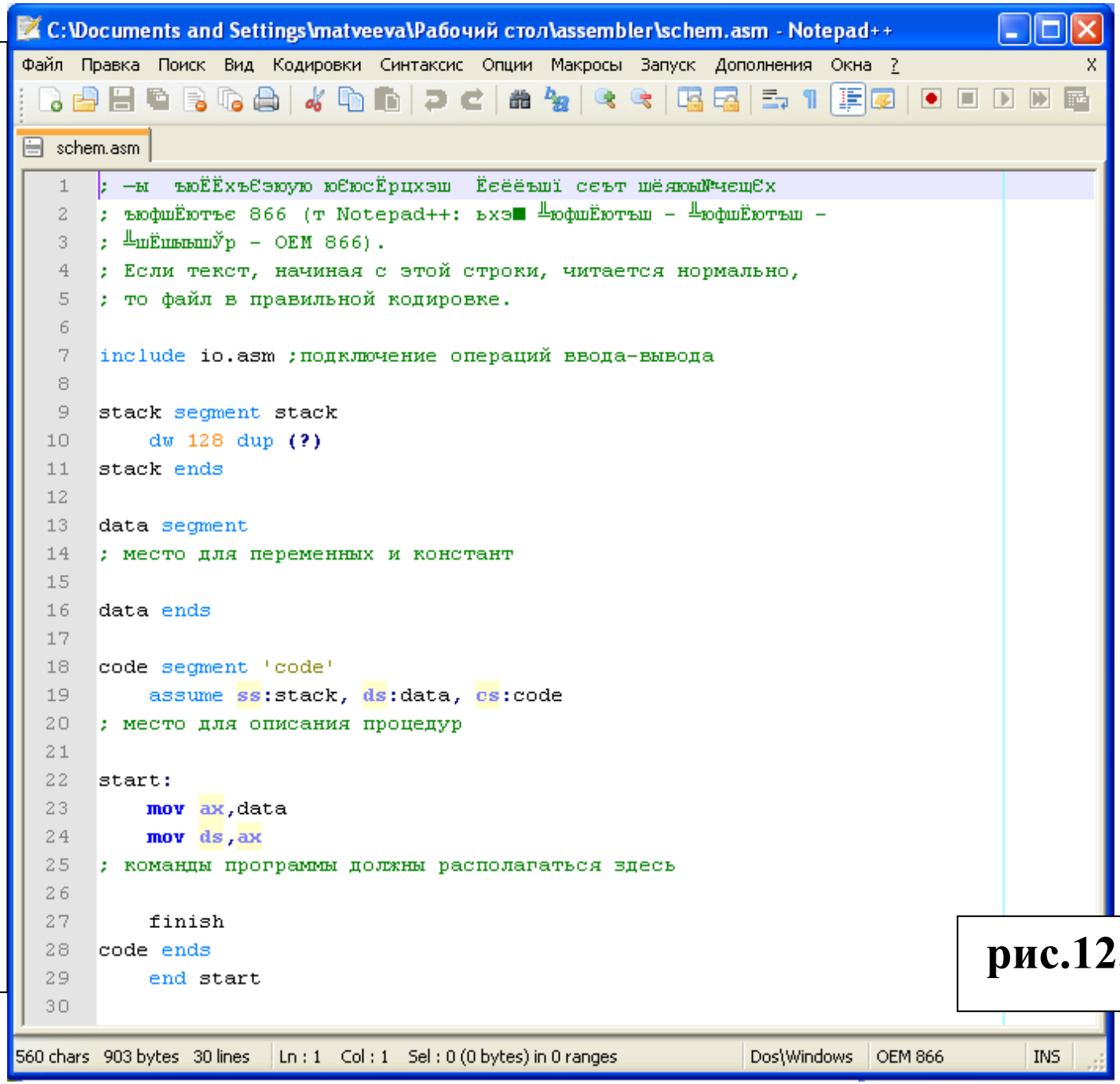


рис.12.

При запуске командного файл assembler.cmd в новой закладке открывается файл schem.asm с установкой признака запрета редактирования (атрибут read only).

Сохраним schem.asm под другим именем с тем же расширением (*.asm) .

При помощи пункта Сохранить как в меню Файл или комбинацией клавиш Ctrl + Alt + S (см рис.13)

получим окно, показанное на рис.14, в котором требуется указать новое имя файла

(и можно изменить предлагаемое по умолчанию место его размещения).

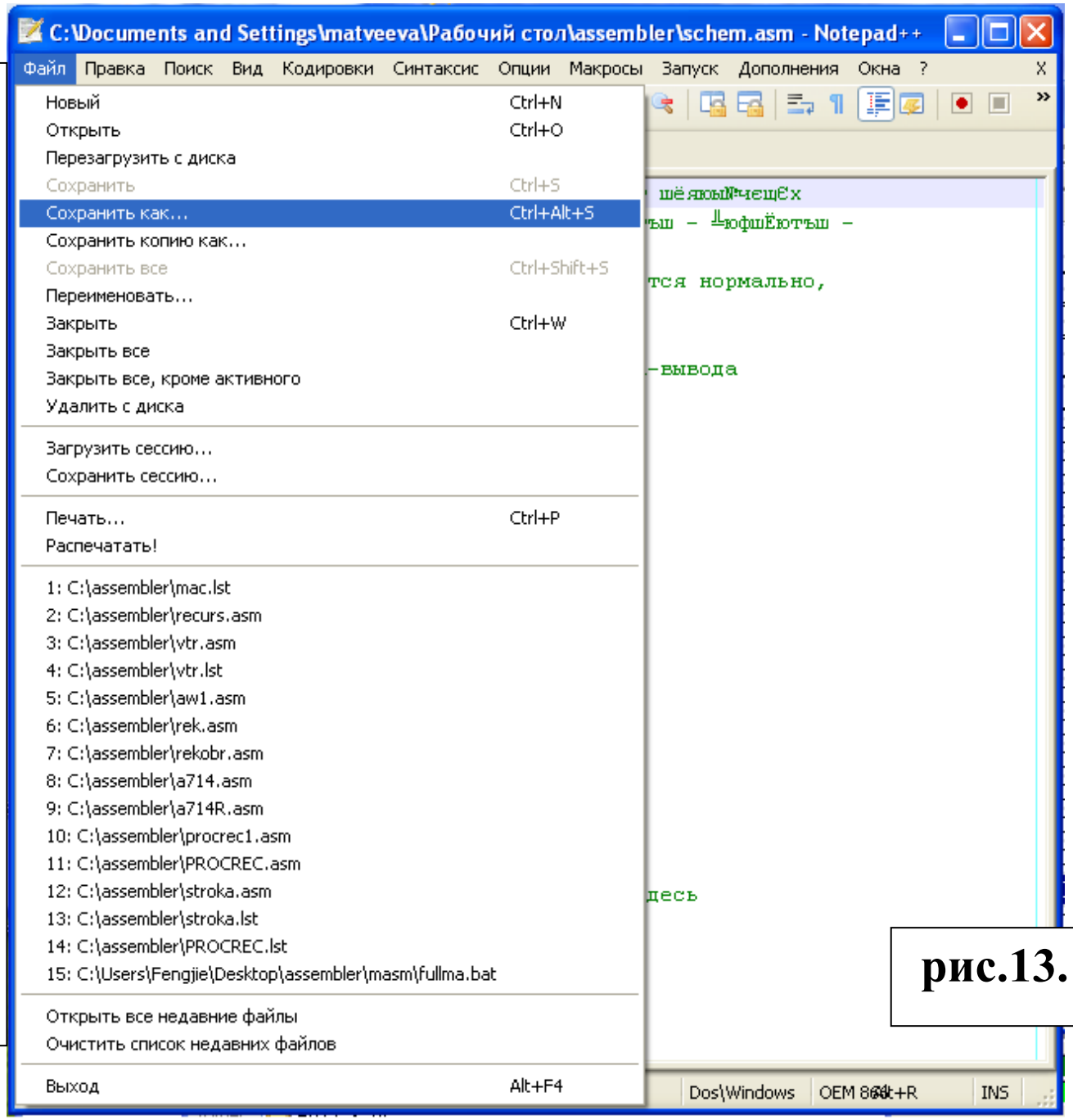
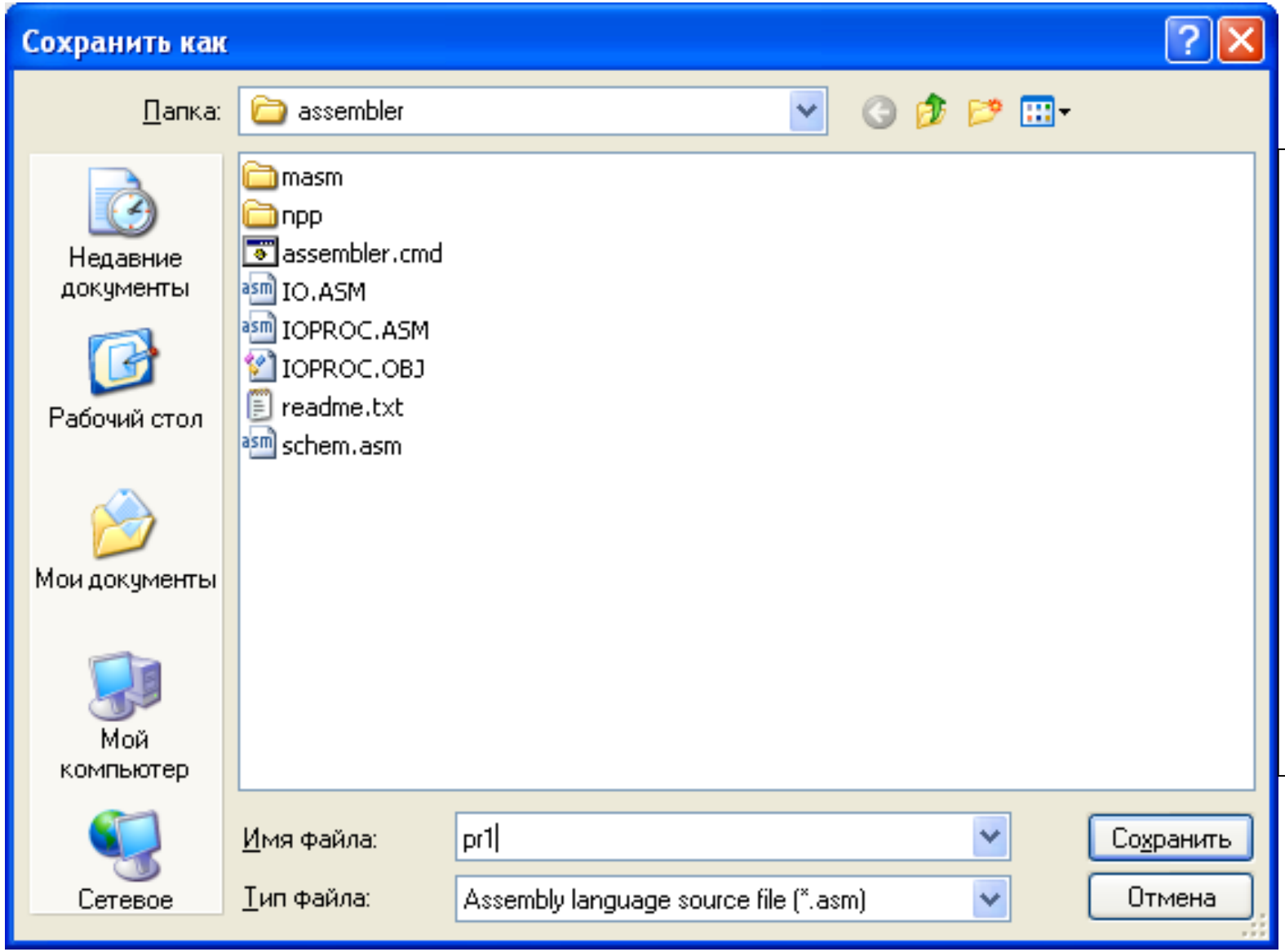


рис.13.



В окне
«Сохранить как»
меняем имя файла
на pr1
и нажимаем кнопку
Сохранить

см.рис.14.

рис.14.

В результате в Notepad++ новый файл будет открыт в той же закладке, где был schem.asm см. рис.15 .

Синий цвет стилизованного символа “дискетки” на “ярлыке” закладки левее имени файла pr1.asm свидетельствует о том, что отображаемый на экране текст был сохранён.

На рис.16 показан красный символ “дискетки” в закладке, свидетельствующий о выполненном редактировании.

Результаты редактирования следует сохранять, например, при помощи кнопки в виде “дискетки” см.рис.16.

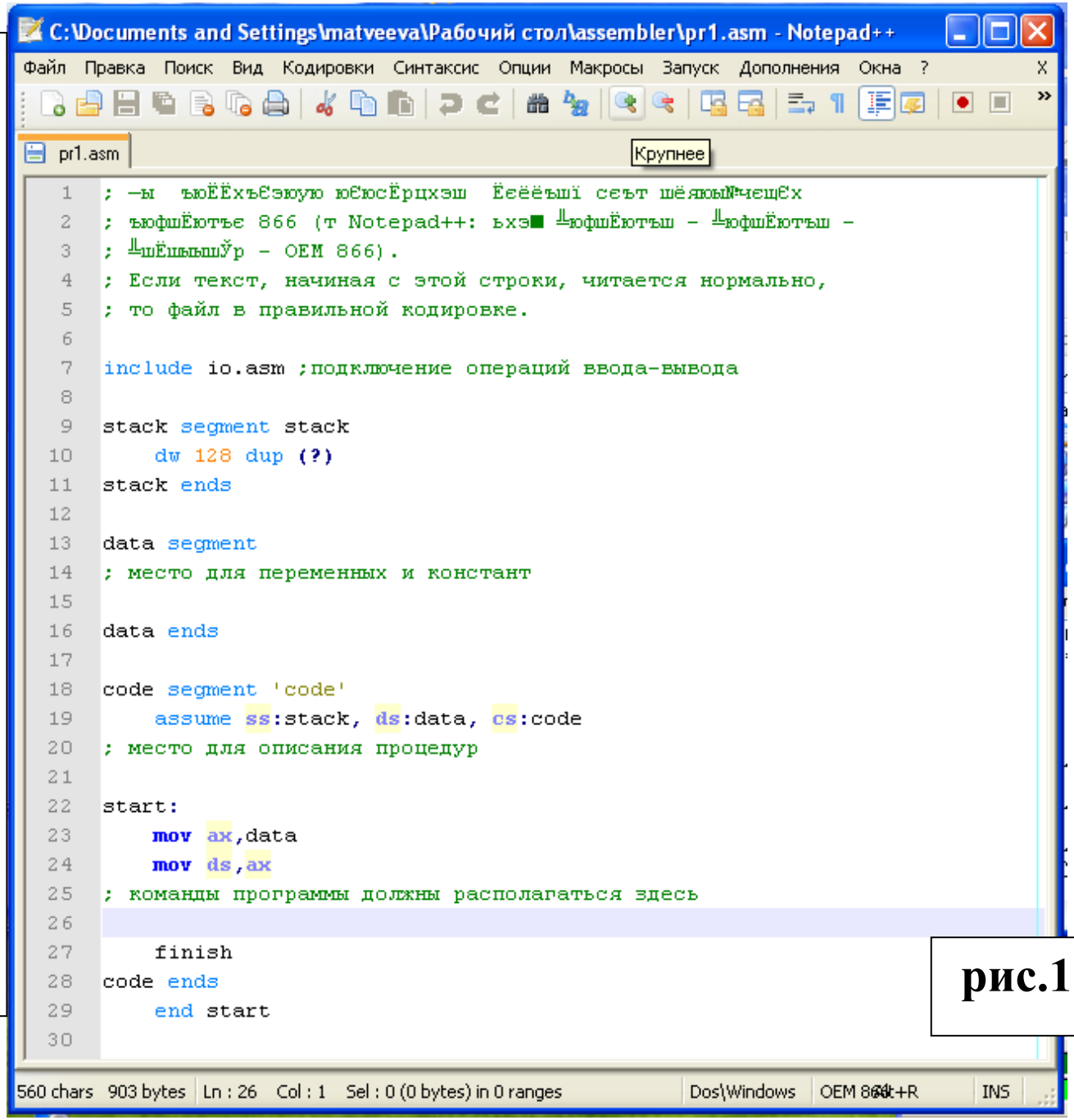
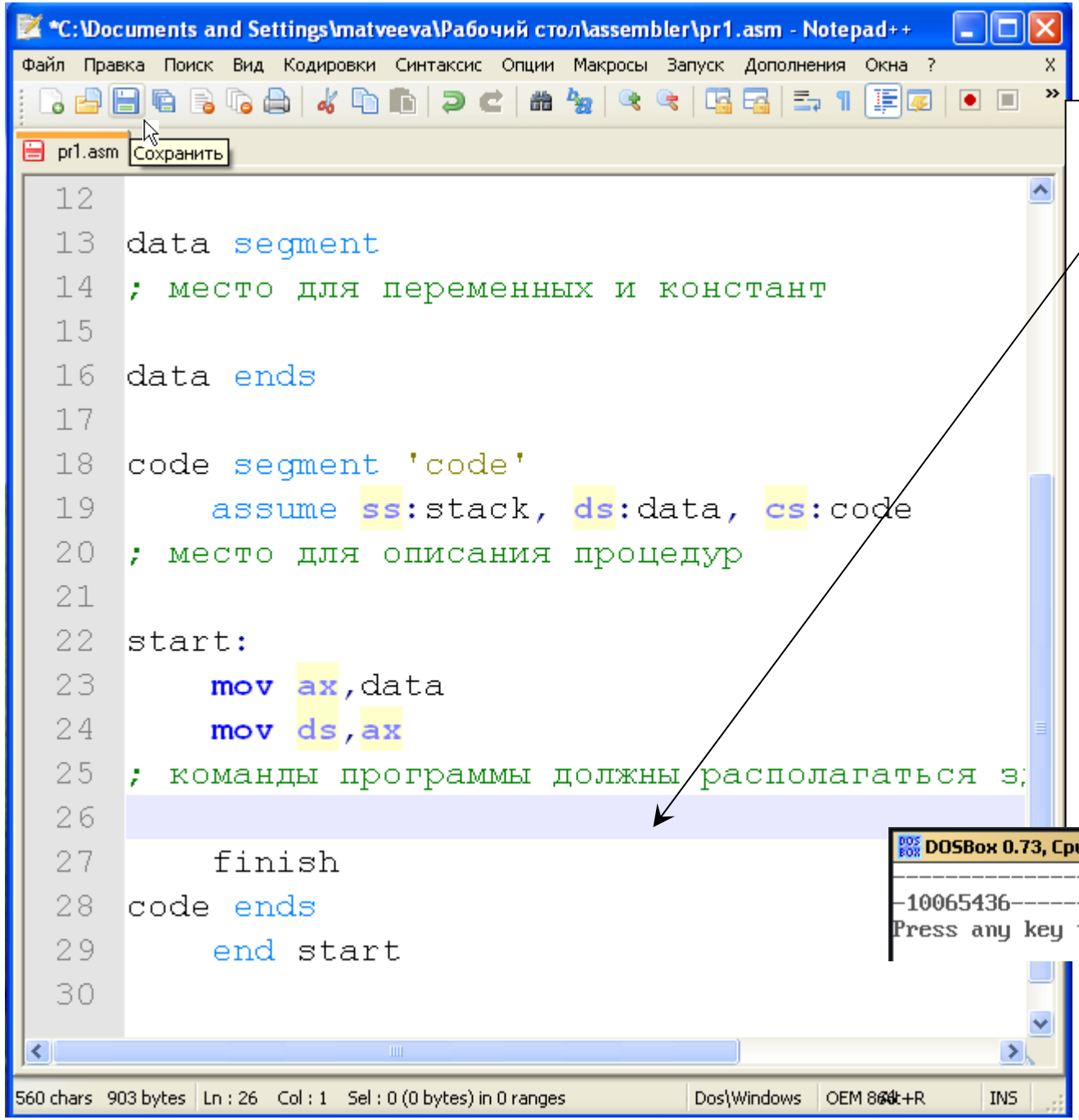


рис.15.



В примере pr1.asm тестируем макрокоманды вывода чисел.
В подсвеченной строке с первой позиции набираем

```
mov ax,-1000
outint ax
outword ax
```

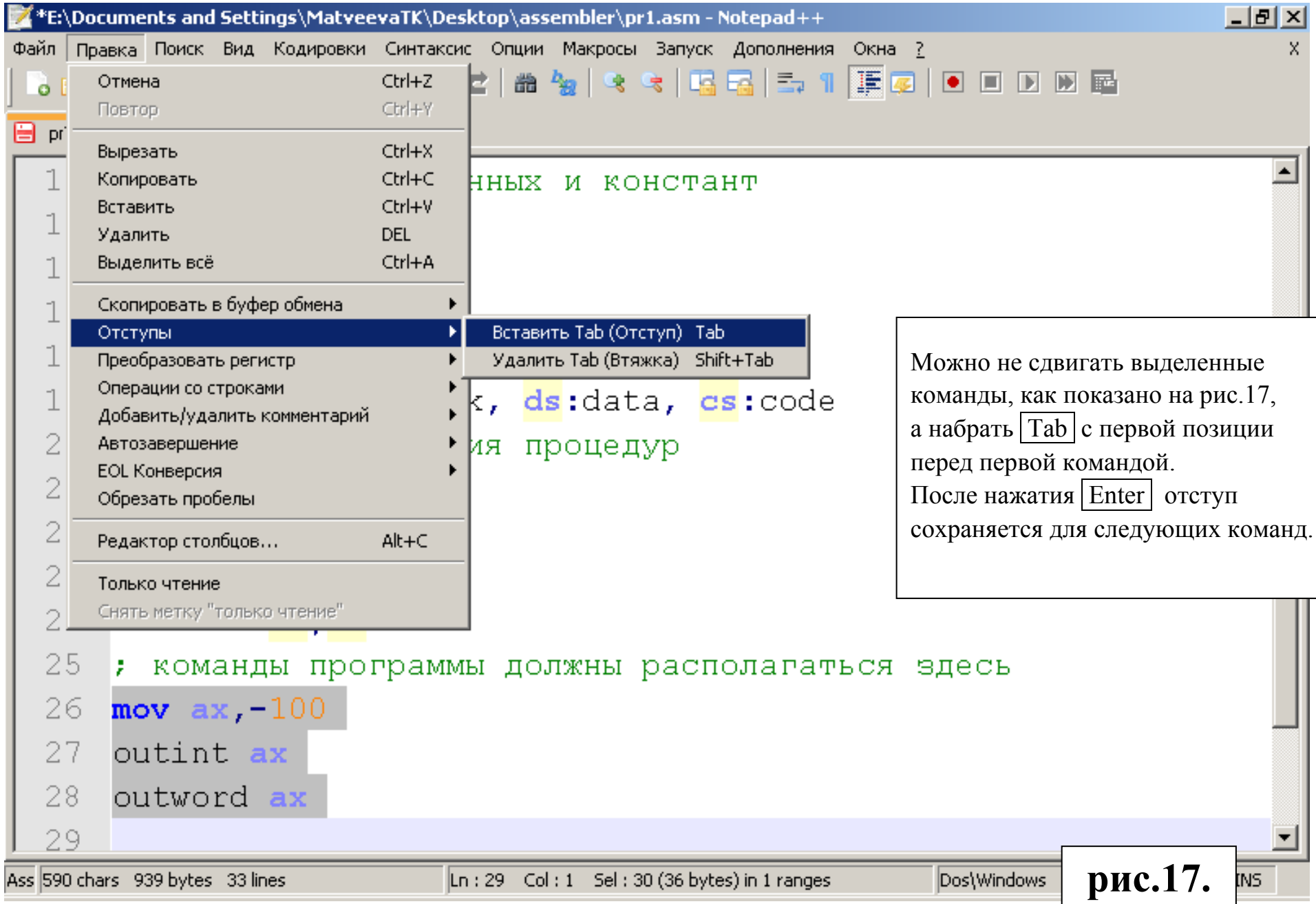
Можно сдвинуть выделенные команды на один отступ при помощи специального пункта меню см.рис.17.

Сохраняем сделанные изменения при помощи кнопки Сохранить .

Запускаем программу на выполнение клавишами Ctrl + F9 (см.рис.18).

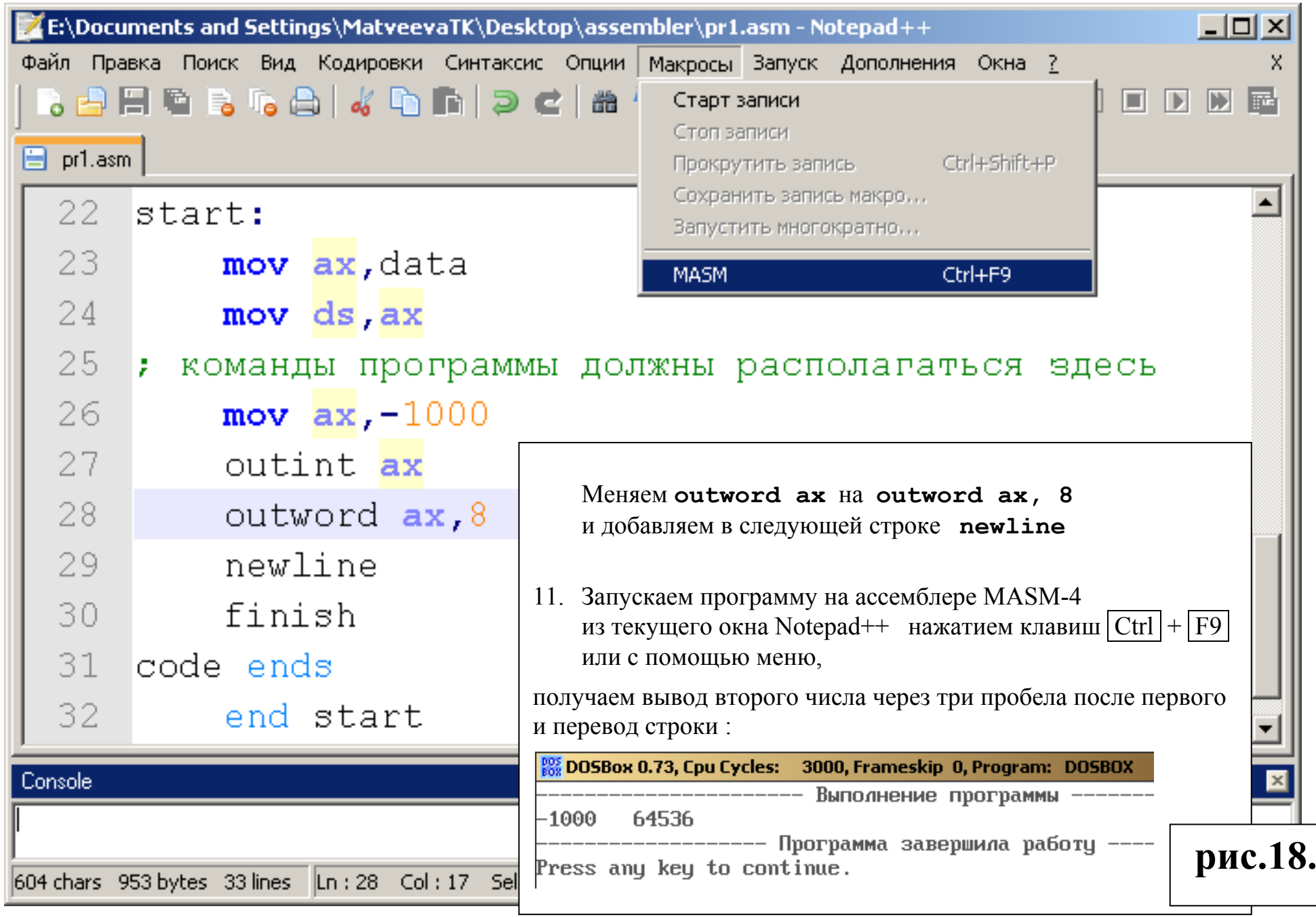
Получаем окно результатов:

рис.16.



Можно не сдвигать выделенные команды, как показано на рис.17, а набрать Tab с первой позиции перед первой командой.
После нажатия Enter отступ сохраняется для следующих команд.

рис.17.



Меняем **outword ax** на **outword ax, 8**
и добавляем в следующей строке **newline**

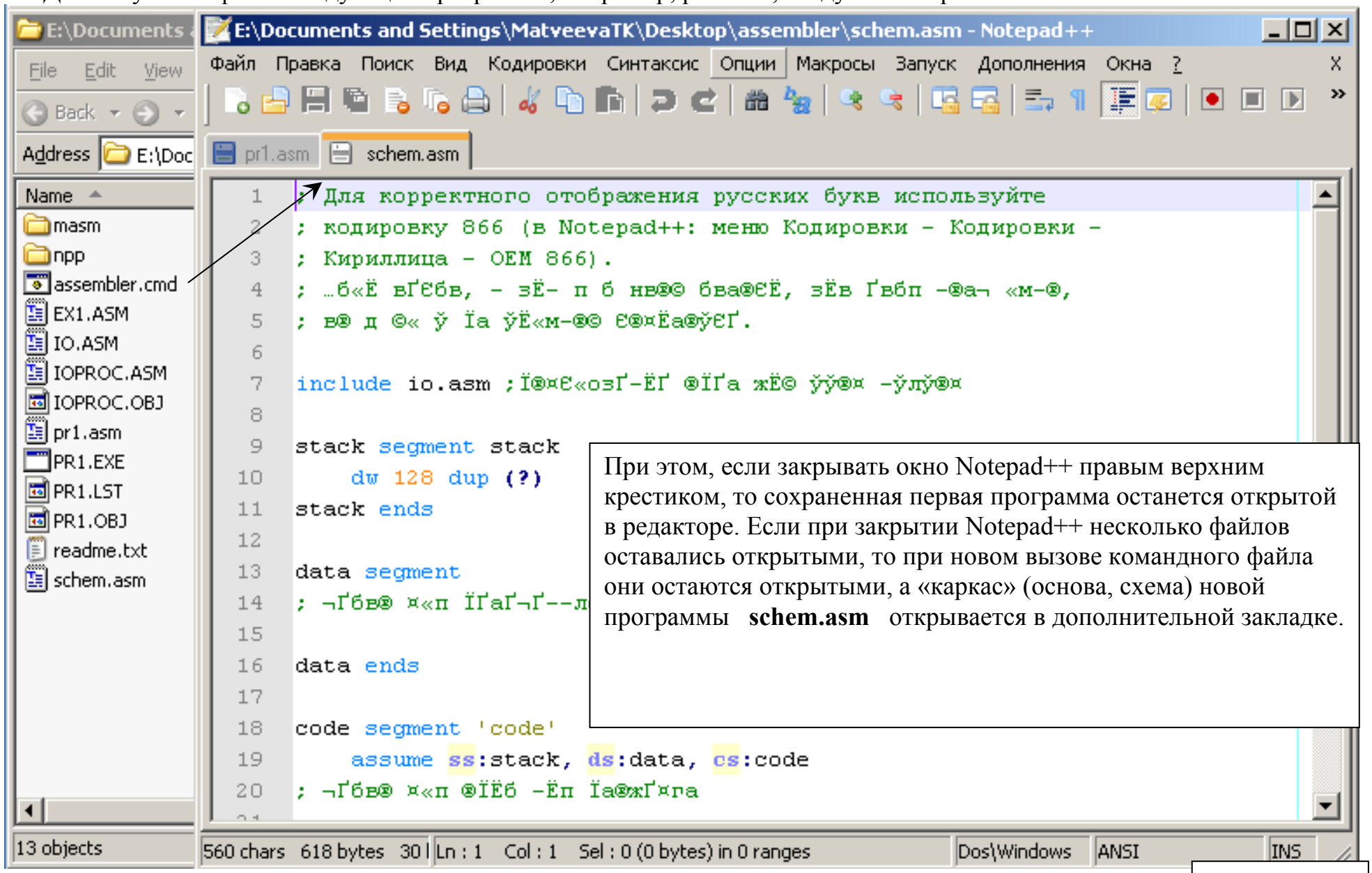
11. Запускаем программу на ассемблере MASM-4
из текущего окна Notepad++ нажатием клавиш Ctrl + F9
или с помощью меню,

получаем вывод второго числа через три пробела после первого и перевод строки :

```
DOSBox 0.73, Cpu Cycles:    3000, Frameskip  0, Program:  DOSBOX
---------------------- Выполнение программы -------
-1000   64536
------------------ Программа завершила работу ----
Press any key to continue.
```

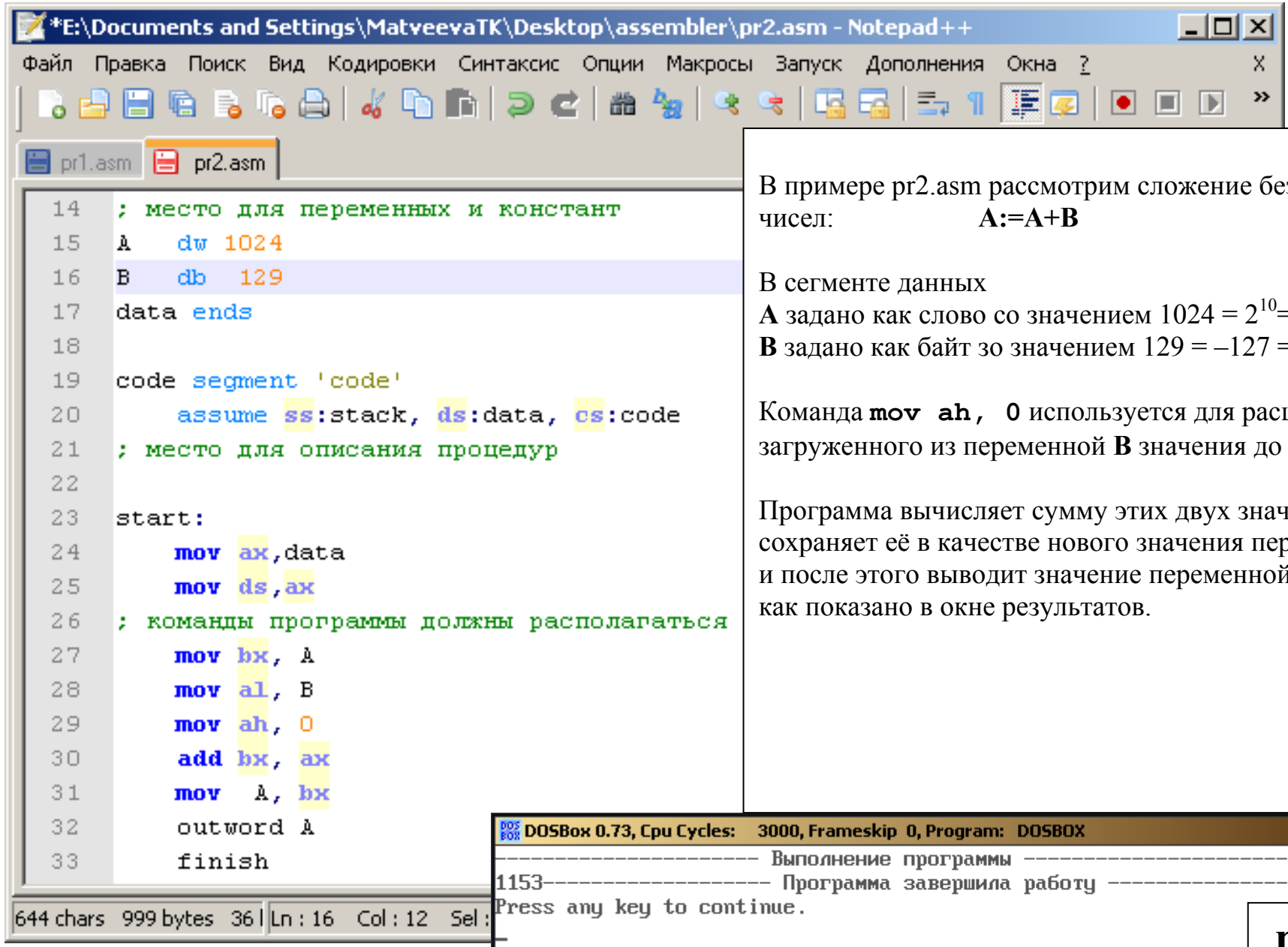
рис.18.

12. Для получения файла следующей программы, например, pr2.asm , следует повторить пп. 7 – 11.



При этом, если закрывать окно Notepad++ правым верхним крестиком, то сохраненная первая программа останется открытой в редакторе. Если при закрытии Notepad++ несколько файлов оставались открытыми, то при новом вызове командного файла они остаются открытыми, а «каркас» (основа, схема) новой программы **schem.asm** открывается в дополнительной закладке.

рис.19.



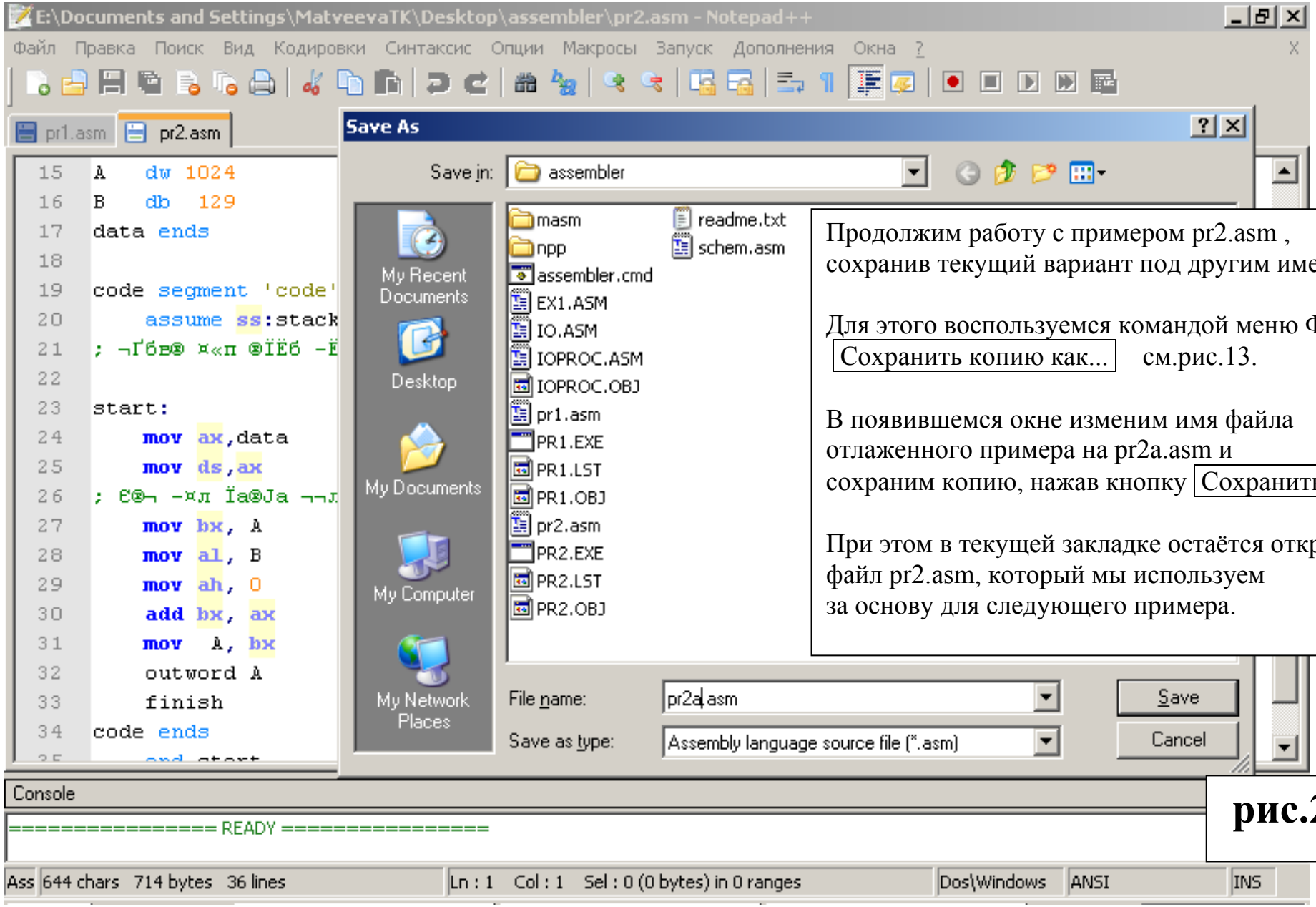
В примере pr2.asm рассмотрим сложение беззнаковых чисел: **A:=A+B**

В сегменте данных
A задано как слово со значением 1024 = 2^{10}= 0400h ,
B задано как байт зо значением 129 = –127 = 0FF81h .

Команда **mov ah, 0** используется для расширения загруженного из переменной **B** значения до размера слова.

Программа вычисляет сумму этих двух значений, сохраняет её в качестве нового значения переменной **A** и после этого выводит значение переменной **A** , как показано в окне результатов.

рис.20.



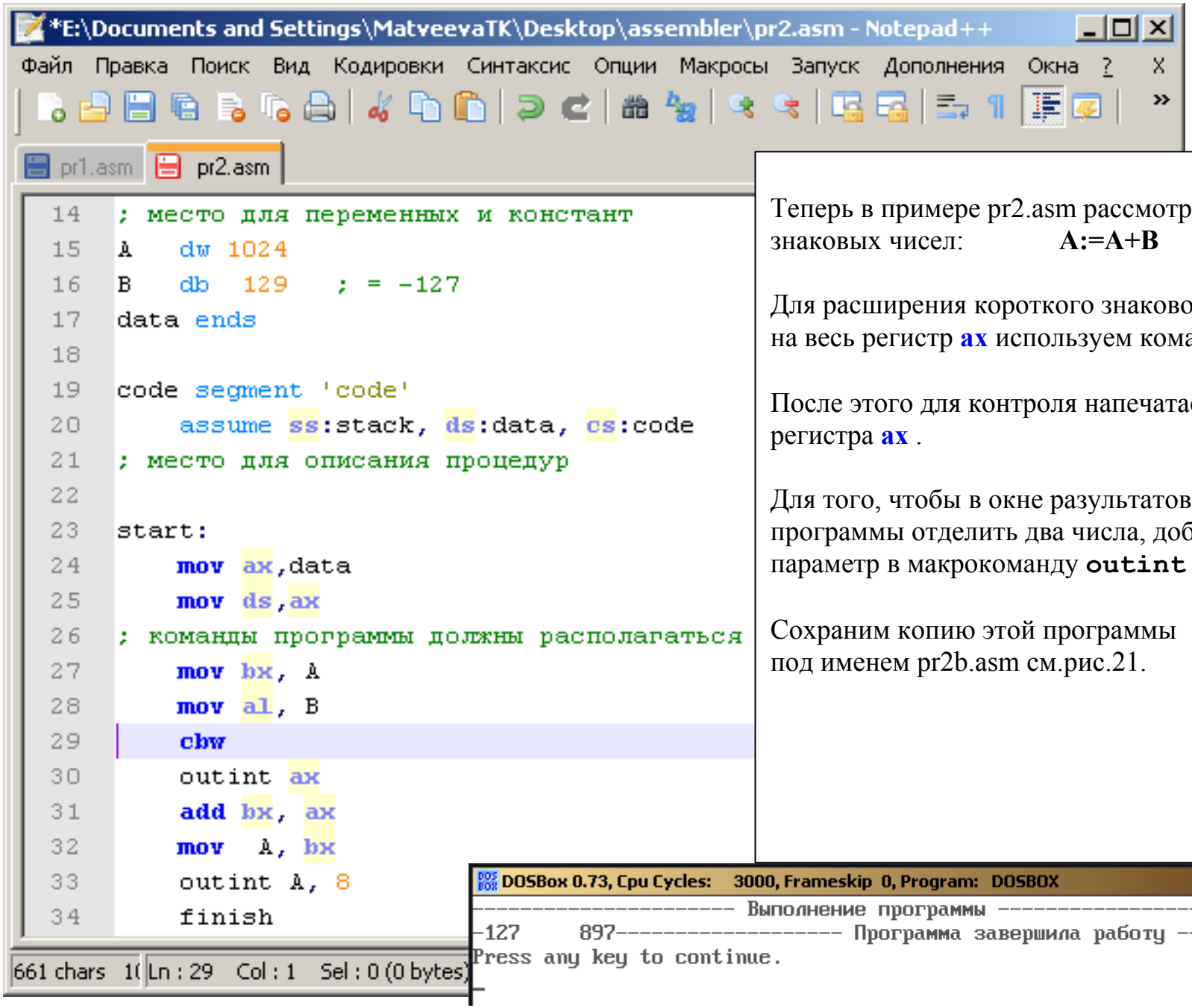
Продолжим работу с примером pr2.asm , сохранив текущий вариант под другим именем.

Для этого воспользуемся командой меню Файл Сохранить копию как... см.рис.13.

В появившемся окне изменим имя файла отлаженного примера на pr2a.asm и сохраним копию, нажав кнопку Сохранить .

При этом в текущей закладке остаётся открыт файл pr2.asm, который мы используем за основу для следующего примера.

рис.21.



Теперь в примере pr2.asm рассмотрим сложение знаковых чисел: **A:=A+B**

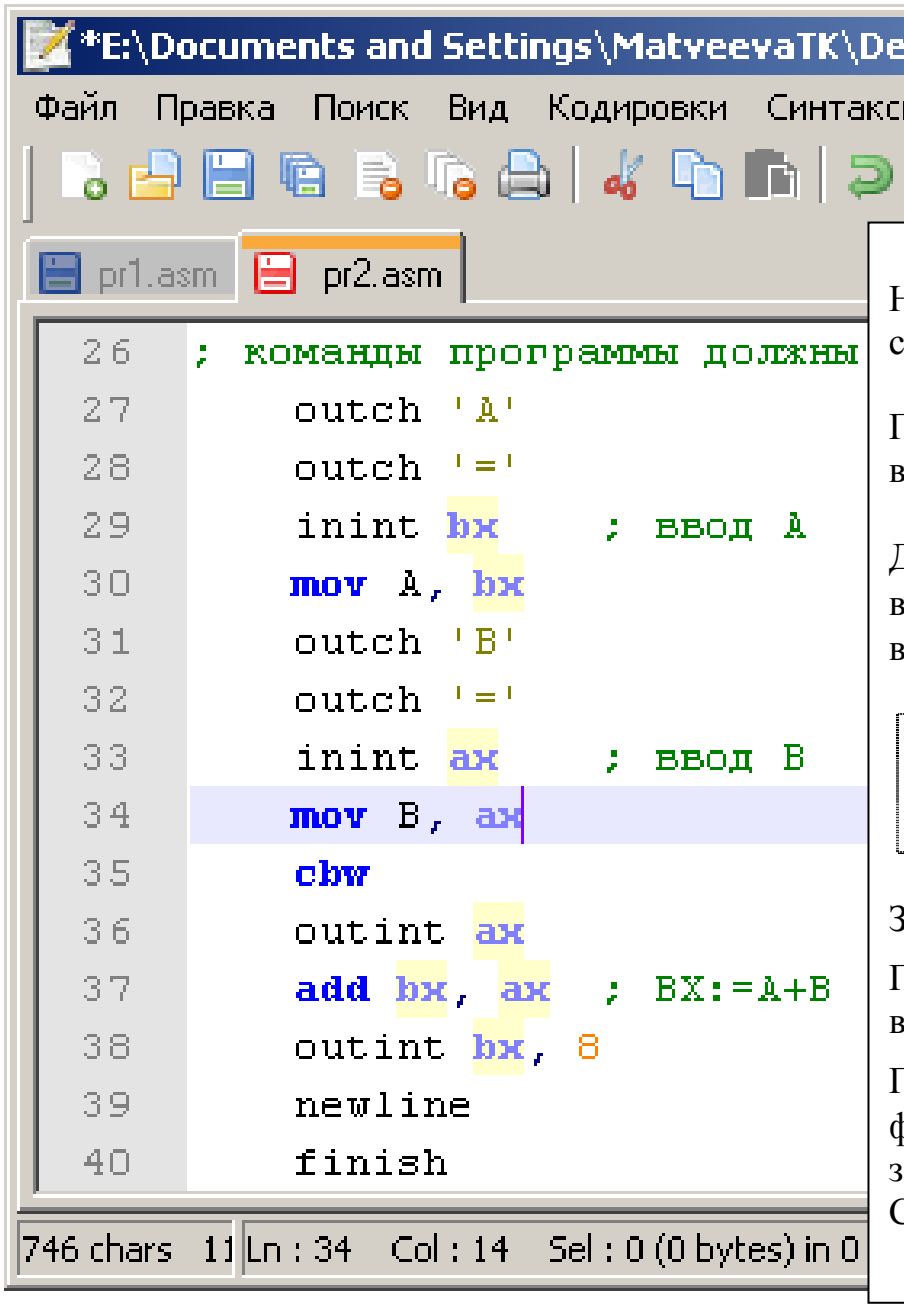
Для расширения короткого знакового числа из **al** на весь регистр **ax** используем команду **cbw** .

После этого для контроля напечатаем содержимое регистра **ax** .

Для того, чтобы в окне разультатов выполнения программы отделить два числа, добавим второй параметр в макрокоманду **outint ax, 8** .

Сохраним копию этой программы под именем pr2b.asm см.рис.21.

рис.22.



Наконец, в третьем варианте примера pr2.asm рассмотрим сложение вводимых пользователем знаковых чисел **bx:=A+B** .

Пусть нам надо сохранить введённые пользователем значения в переменных **A** и **B** , а результат сложения вывести на экран.

Добавим макрокоманды **outch** для вывода приглашения при ввода значений **A** и **B**, и макрокоманды **inint** для ввода чисел в регистры **bx** и **ax** соответственно.

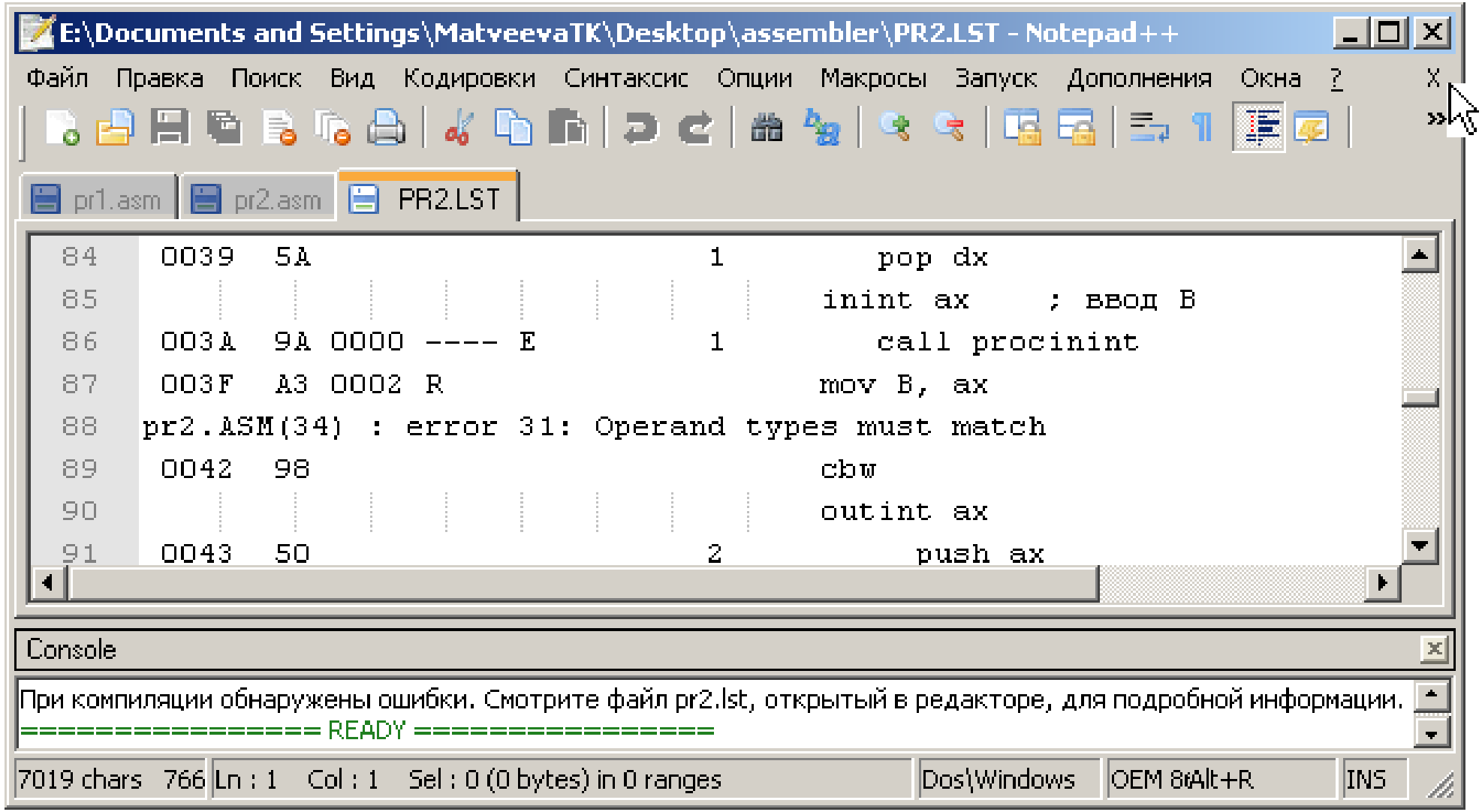
Заметим, что если бы мы вводили текстовые данные, то переключиться на русские буквы и обратно в окне DOSBox можно при помощи правого Ctrl .

Заменим порядок аргументов в командах пересылки.

Попробуем допустить ошибку, пытаясь поместить в переменную **B** типа байт значение регистра **ax** .

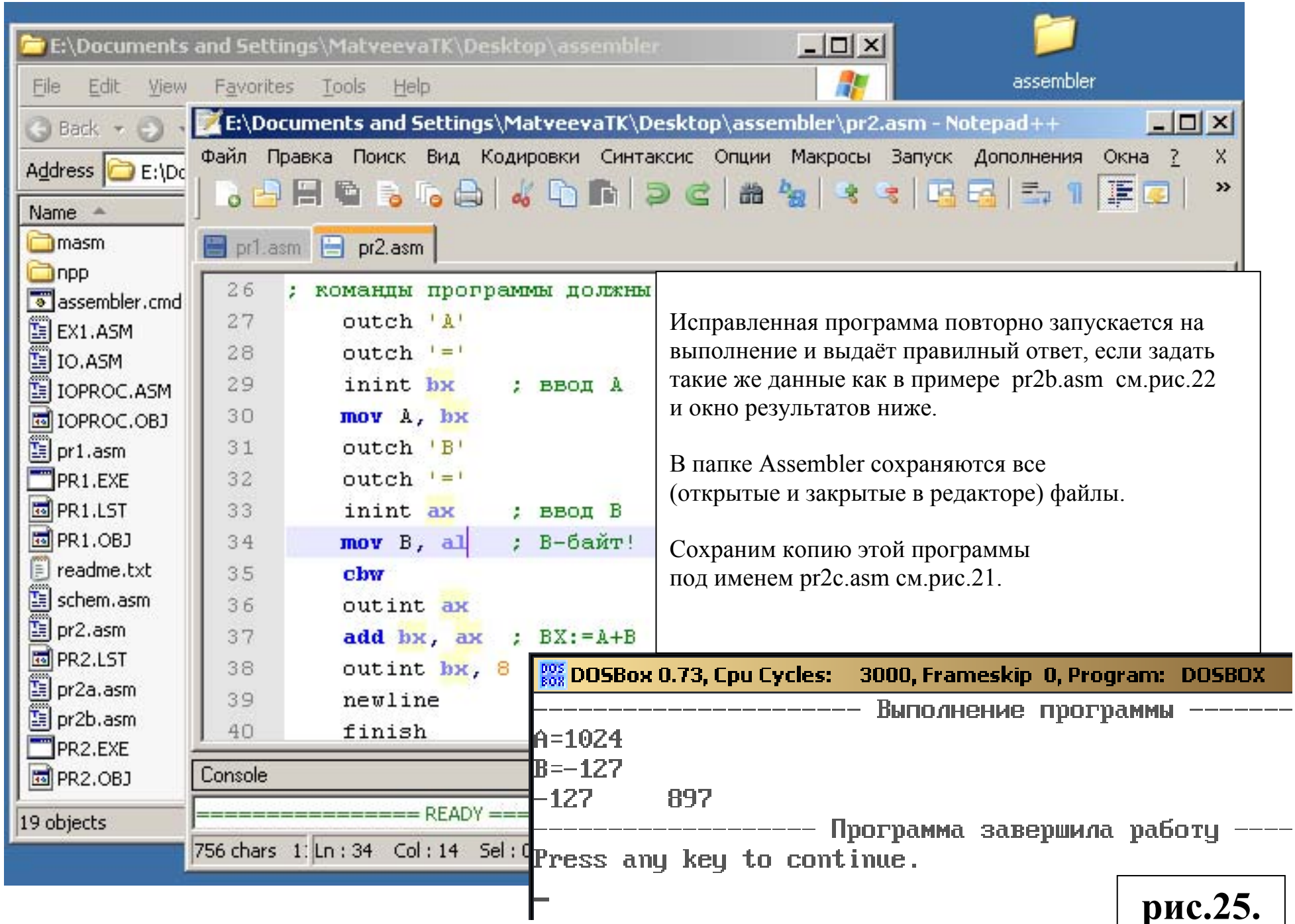
При запуске программы при помощи Ctrl + F9 компилятор формирует файл pr2.lst, который открывается в отдельной закладке, а сообщение о результате компиляции выводится в окно Console, отображаемое в нижней части окна Notepad++ см.рис.24.

рис.23.



Если при компиляции обнаружены ошибки, то в строке закладок появится одноименный файл с расширением **.lst**.
В нем ошибки могут быть найдены при помощи поиска ключевого слова error и изучены. В даном случае сообщение pr2.ASM(34) : error 31: Operand types must match может быть переведено как «тип операндов должен соответствовать».
Внимание! Изменения вносятся не в этот файл pr2.lst , а в файл программы pr2.asm .

рис.24.

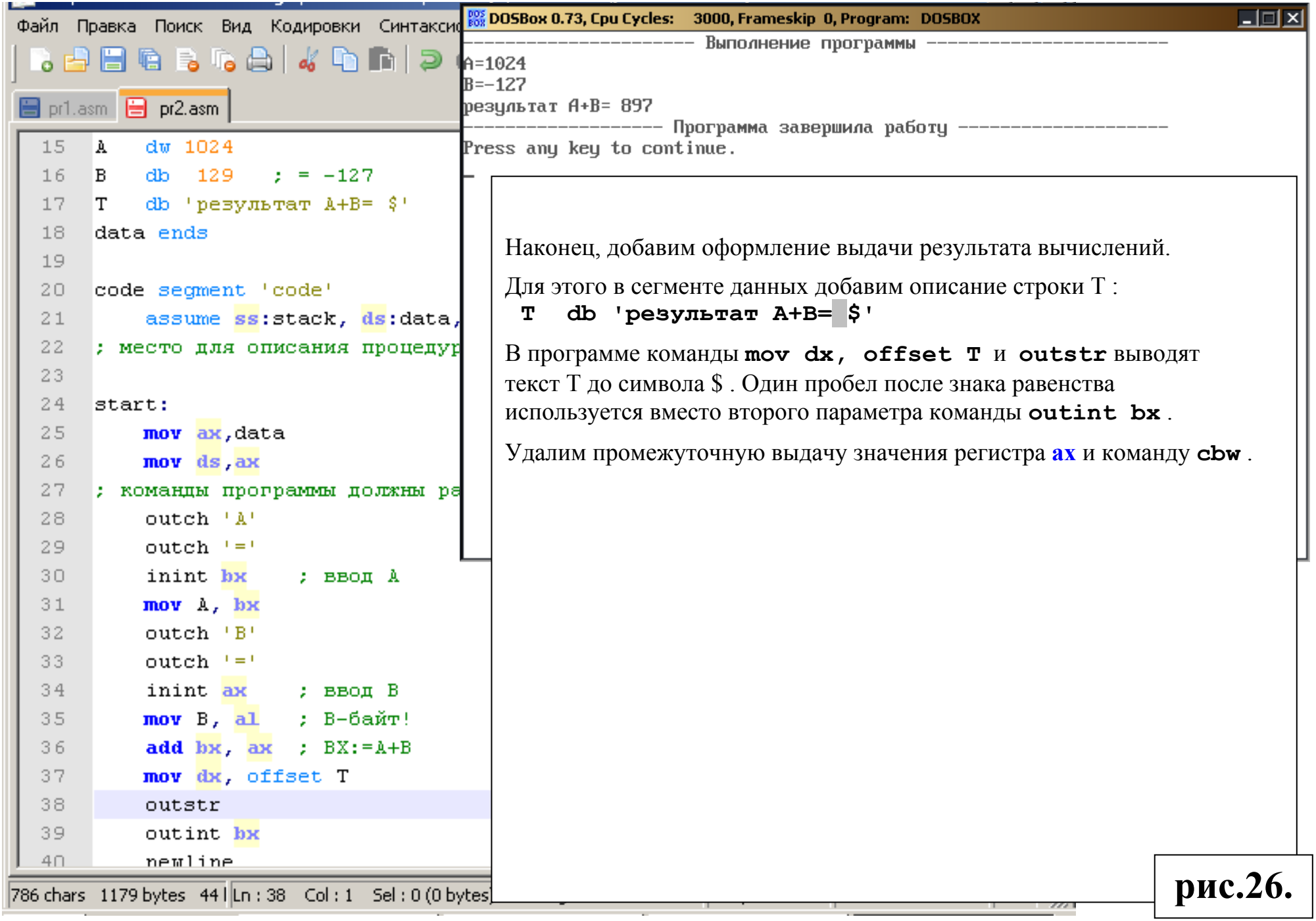


Исправленная программа повторно запускается на выполнение и выдаёт правилный ответ, если задать такие же данные как в примере pr2b.asm см.рис.22 и окно результатов ниже.

В папке Assembler сохраняются все (открытые и закрытые в редакторе) файлы.

Сохраним копию этой программы под именем pr2c.asm см.рис.21.

рис.25.



Наконец, добавим оформление выдачи результата вычислений.

Для этого в сегменте данных добавим описание строки T :

```
T   db 'результат A+B= $'
```

В программе команды **mov dx, offset T** и **outstr** выводят текст T до символа $. Один пробел после знака равенства используется вместо второго параметра команды **outint bx** .

Удалим промежуточную выдачу значения регистра **ax** и команду **cbw** .

рис.26.

Приложения

Приложение 1

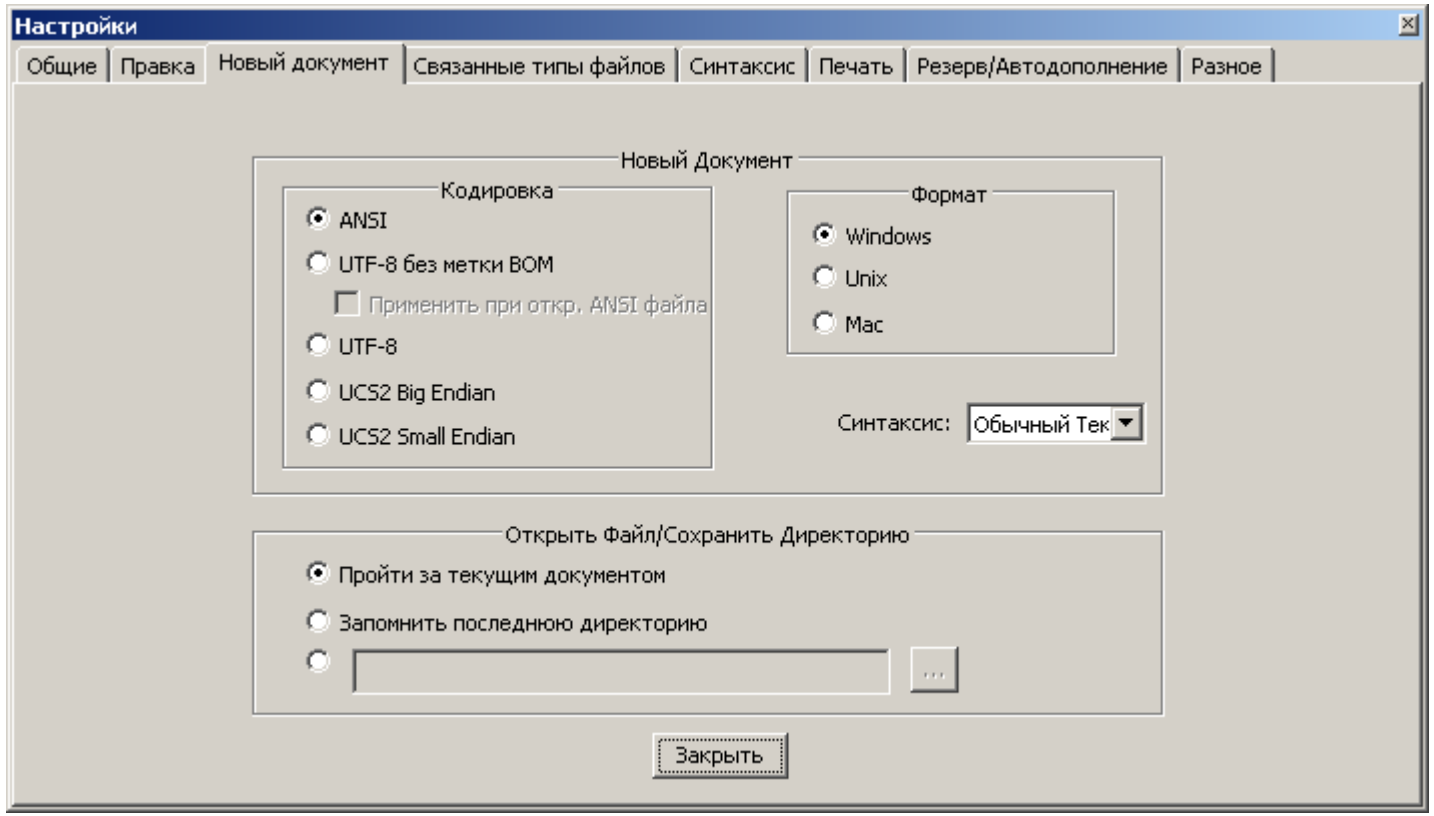


рис.27.

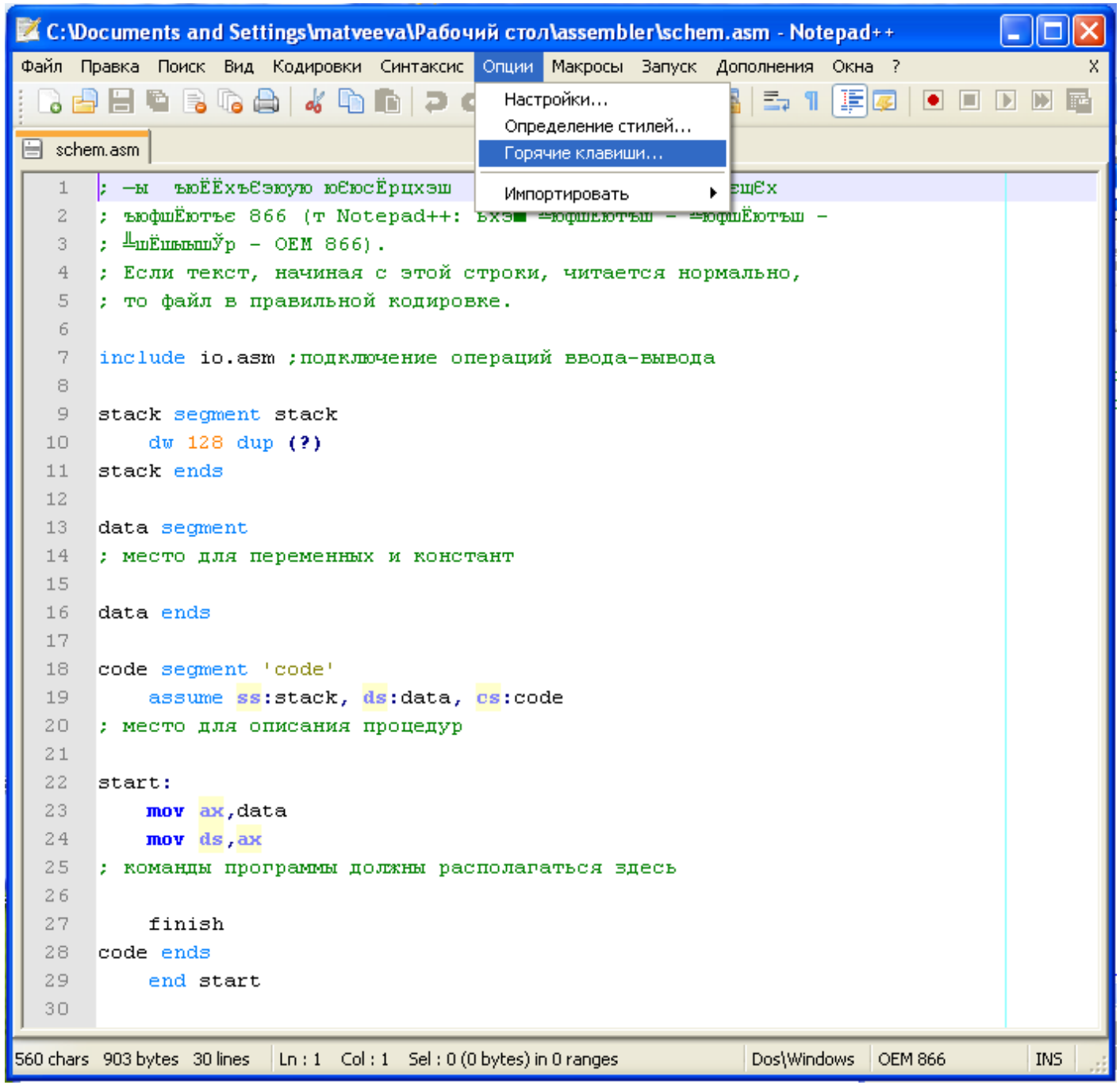

C:\Documents and Settings\matveeva\Рабочий стол\assembler\schem.asm - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск Дополнения Окна ?
Настройки...
Определение стилей...
Горячие клавиши...
Импортировать
schem.asm
; ъюфшЁютъе 866 (т Notepad++:
; ШшЁшьшшЎр - OEM 866).
; Если текст, начиная с этой строки, читается нормально,
; то файл в правильной кодировке.
include io.asm ;подключение операций ввода-вывода
stack segment stack
dw 128 dup (?)
stack ends
data segment
; место для переменных и констант
data ends
code segment 'code'
assume ss:stack, ds:data, cs:code
; место для описания процедур
start:
mov ax,data
mov ds,ax
; команды программы должны располагаться здесь
finish
code ends
end start
560 chars 903 bytes 30 lines Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges Dos\Windows OEM 866 INS

рис.28.

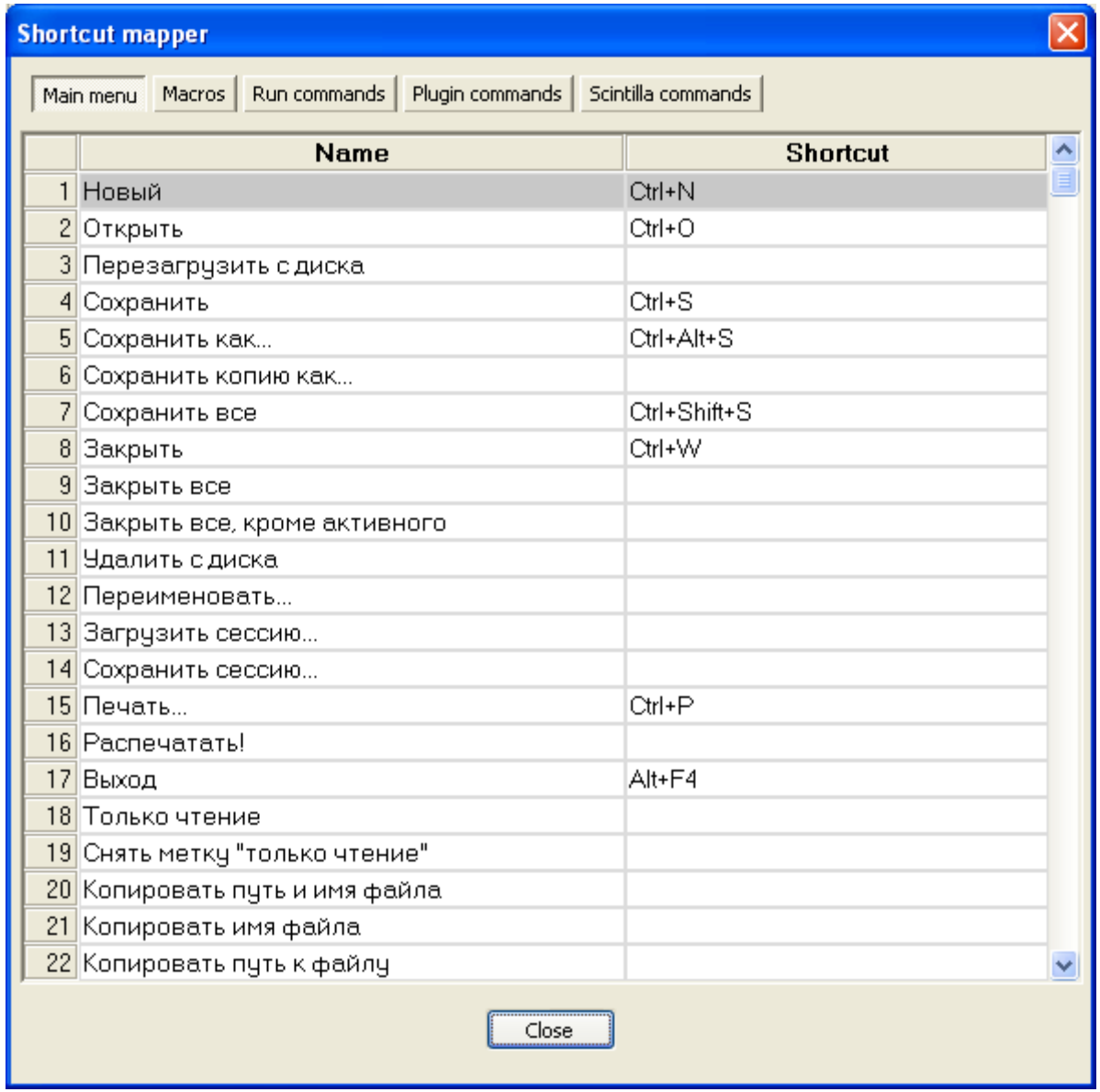

Shortcut mapper
Main menu
Macros
Run commands
Plugin commands
Scintilla commands
Name
Shortcut
1 Новый Ctrl+N
2 Открыть Ctrl+O
3 Перезагрузить с диска
4 Сохранить Ctrl+S
5 Сохранить как... Ctrl+Alt+S
6 Сохранить копию как...
7 Сохранить все Ctrl+Shift+S
8 Закрыть Ctrl+W
9 Закрыть все
10 Закрыть все, кроме активного
11 Удалить с диска
12 Переименовать...
13 Загрузить сессию...
14 Сохранить сессию...
15 Печать... Ctrl+P
16 Распечатать!
17 Выход Alt+F4
18 Только чтение
19 Снять метку "только чтение"
20 Копировать путь и имя файла
21 Копировать имя файла
22 Копировать путь к файлу
Close

рис.29.

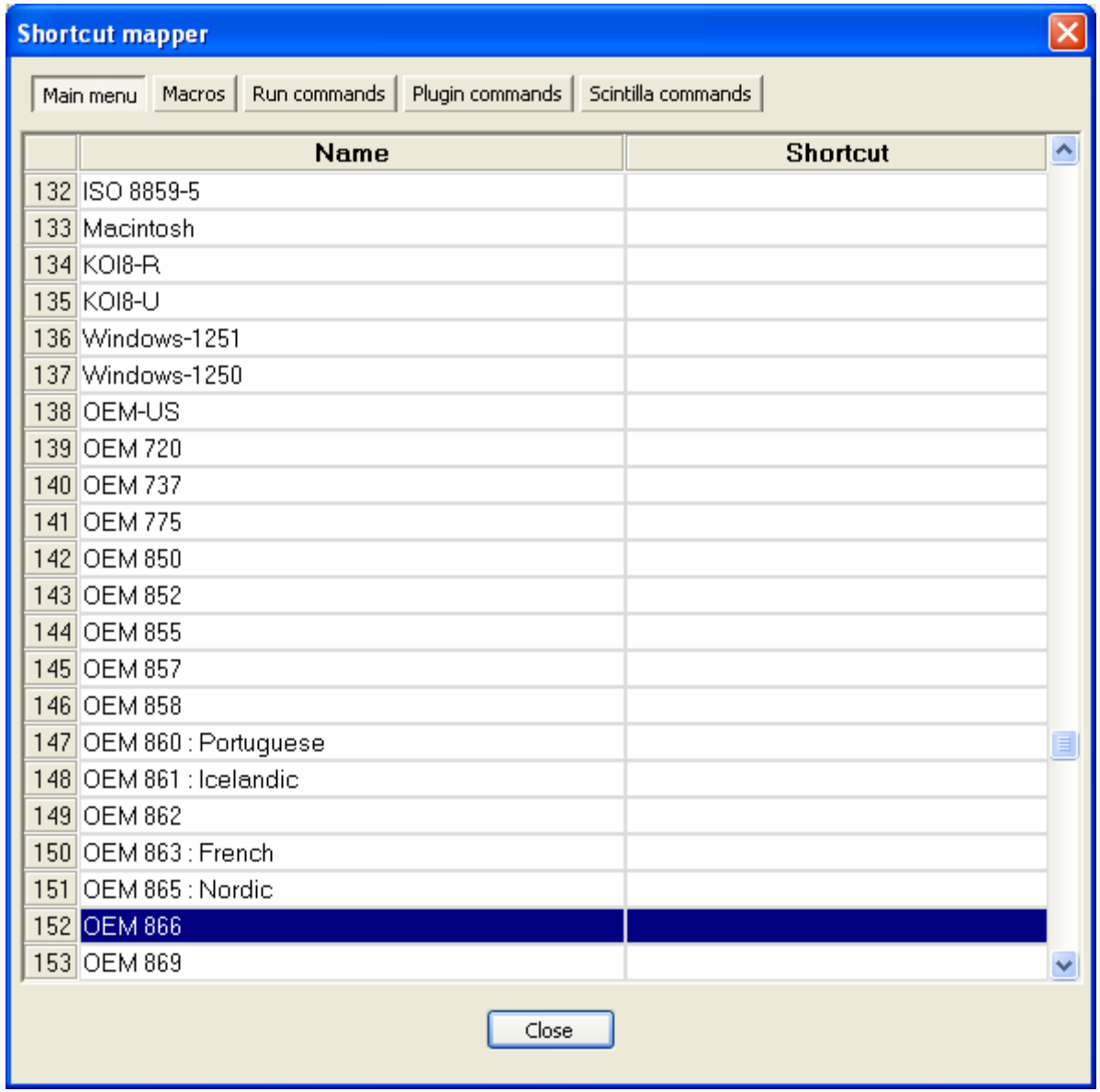

Shortcut mapper
Main menu
Macros
Run commands
Plugin commands
Scintilla commands
Name
Shortcut
132 ISO 8859-5
133 Macintosh
134 KOI8-R
135 KOI8-U
136 Windows-1251
137 Windows-1250
138 OEM-US
139 OEM 720
140 OEM 737
141 OEM 775
142 OEM 850
143 OEM 852
144 OEM 855
145 OEM 857
146 OEM 858
147 OEM 860 : Portuguese
148 OEM 861 : Icelandic
149 OEM 862
150 OEM 863 : French
151 OEM 865 : Nordic
152 OEM 866
153 OEM 869
Close

рис.30.

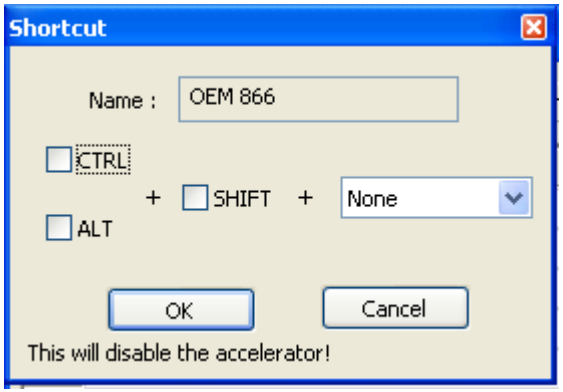

Shortcut
Name :
OEM 866
CTRL
+
SHIFT
+
None
ALT
OK
Cancel
This will disable the accelerator!

рис.31.

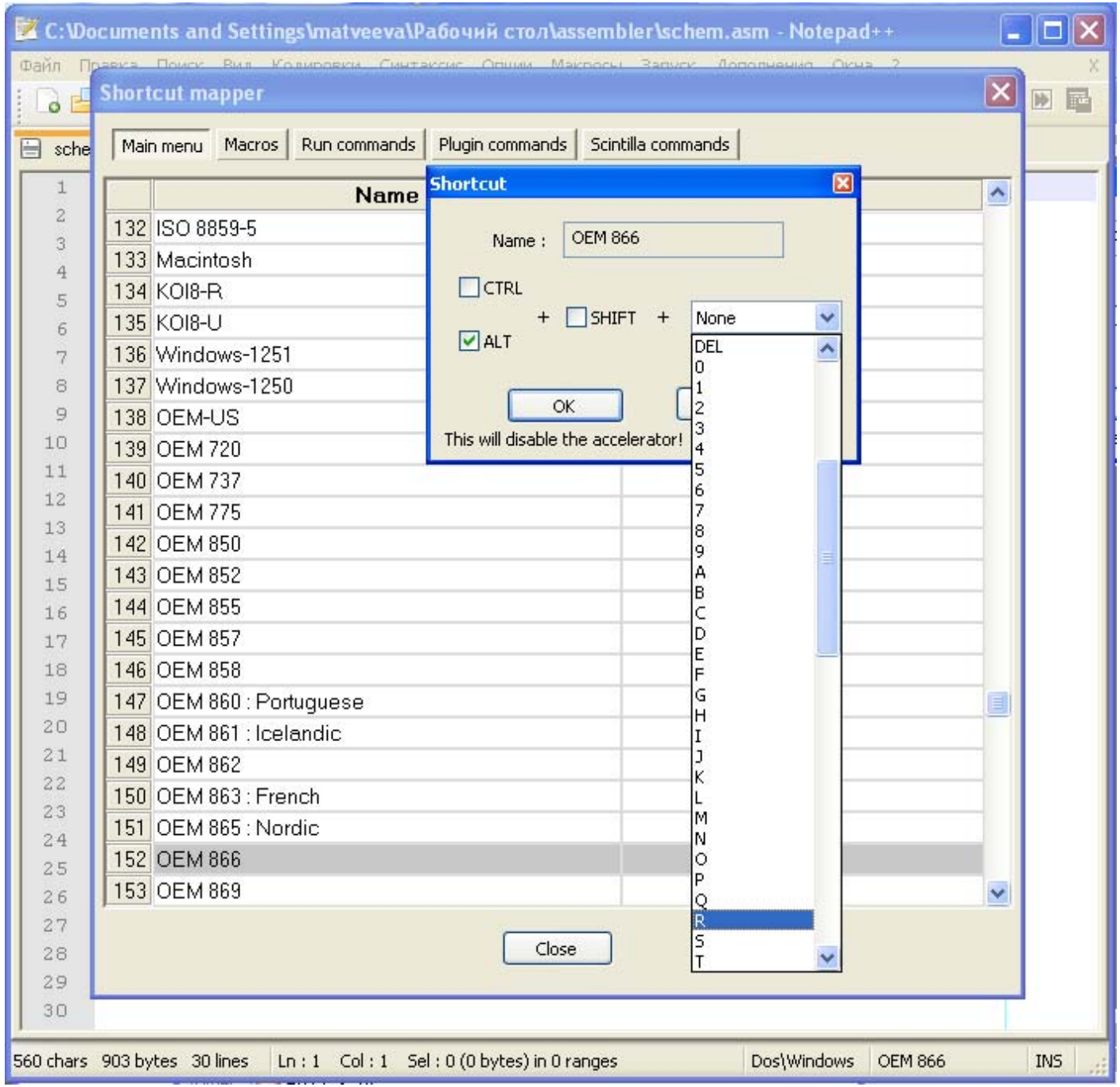

C:\Documents and Settings\matveeva\Рабочий стол\assembler\schem.asm - Notepad++
Shortcut mapper
Main menu
Macros
Run commands
Plugin commands
Scintilla commands
Name
132 ISO 8859-5
133 Macintosh
134 KOI8-R
135 KOI8-U
136 Windows-1251
137 Windows-1250
138 OEM-US
139 OEM 720
140 OEM 737
141 OEM 775
142 OEM 850
143 OEM 852
144 OEM 855
145 OEM 857
146 OEM 858
147 OEM 860 : Portuguese
148 OEM 861 : Icelandic
149 OEM 862
150 OEM 863 : French
151 OEM 865 : Nordic
152 OEM 866
153 OEM 869
Close
Shortcut
Name : OEM 866
CTRL
SHIFT
ALT
None
DEL
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
OK
This will disable the accelerator!
560 chars 903 bytes 30 lines Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges Dos\Windows OEM 866 INS

рис.32.

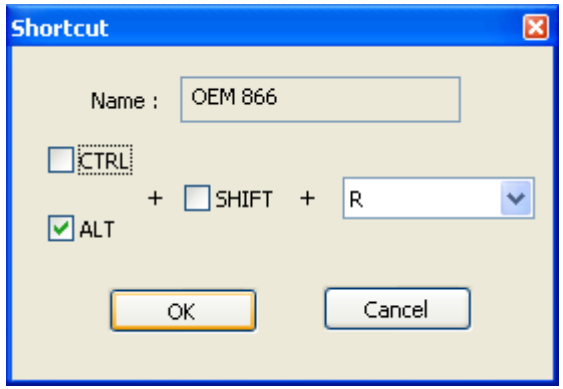

Shortcut
Name :
OEM 866
CTRL
+
SHIFT
+
R
ALT
OK
Cancel

рис.33.

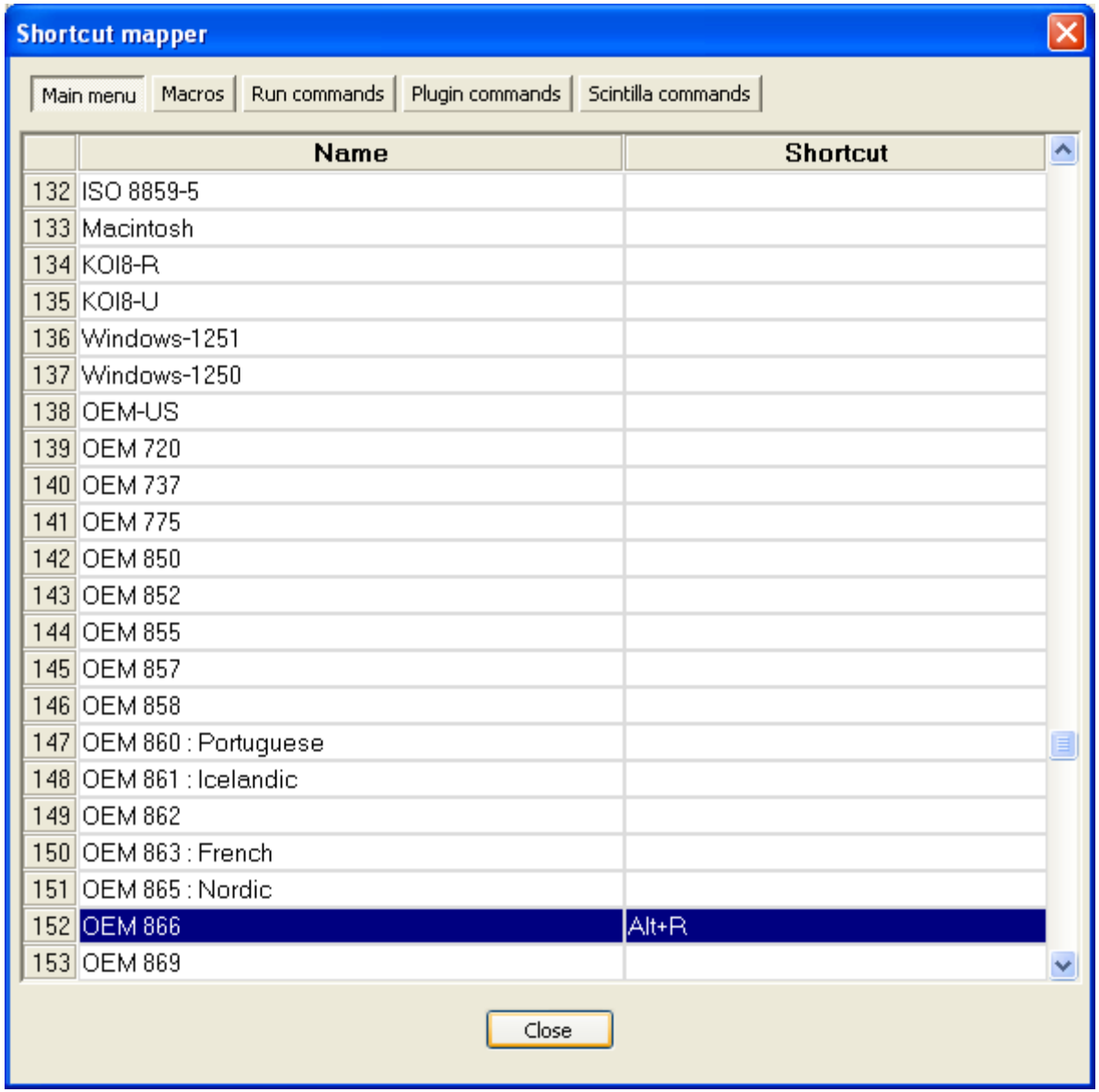

Shortcut mapper
Main menu
Macros
Run commands
Plugin commands
Scintilla commands
Name
Shortcut
132 ISO 8859-5
133 Macintosh
134 KOI8-R
135 KOI8-U
136 Windows-1251
137 Windows-1250
138 OEM-US
139 OEM 720
140 OEM 737
141 OEM 775
142 OEM 850
143 OEM 852
144 OEM 855
145 OEM 857
146 OEM 858
147 OEM 860 : Portuguese
148 OEM 861 : Icelandic
149 OEM 862
150 OEM 863 : French
151 OEM 865 : Nordic
152 OEM 866 Alt+R
153 OEM 869
Close

рис.34.

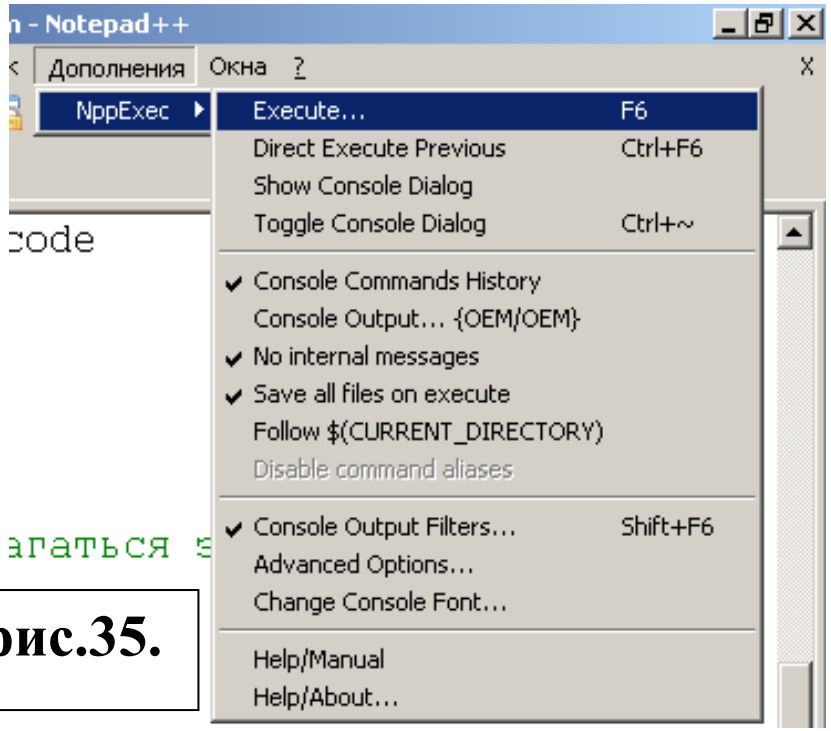


рис.35.

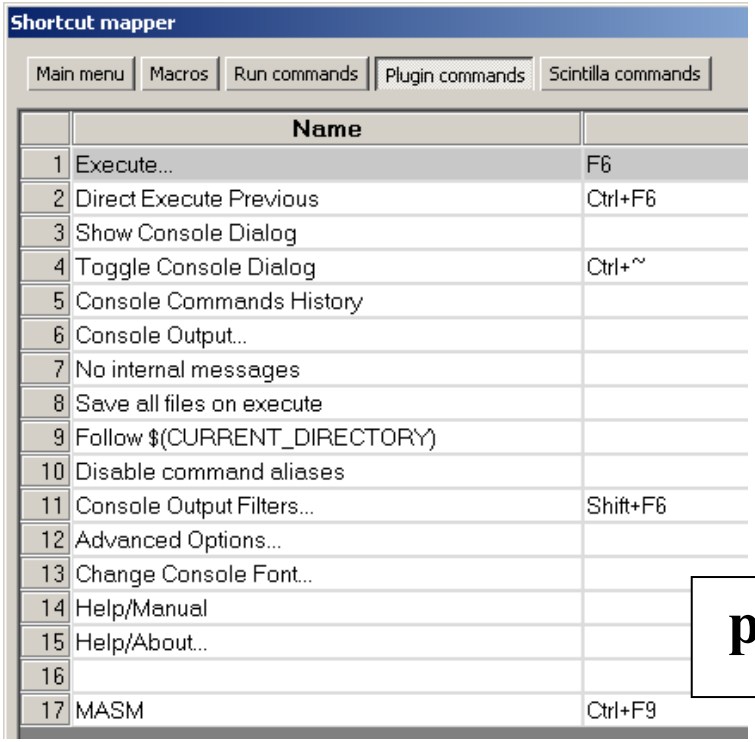


рис.36.

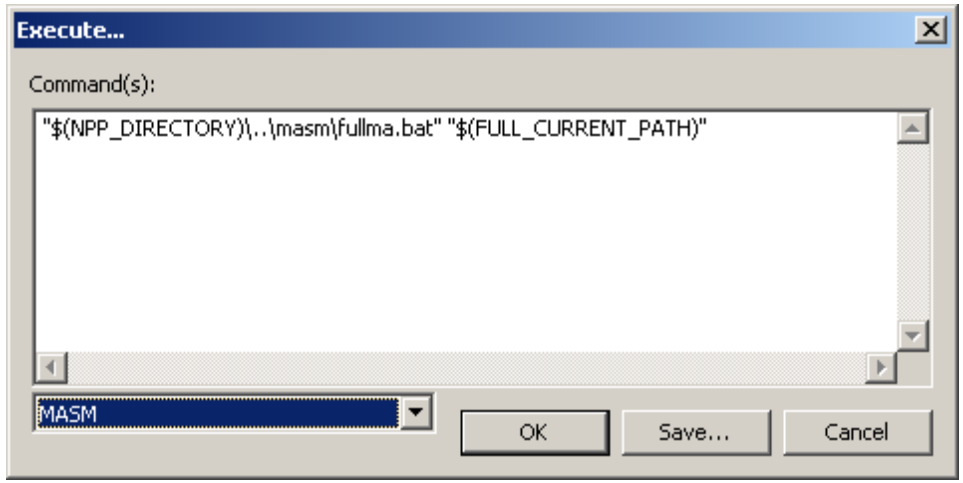


рис.37.

рис.38.

asmprog.bat

```
rem t:\masm4\masm.exe /nologo /c /Fou:\%FN%.obj /Flu:\%FN%.lst /W3 /X /Zm /Zi /It: u:\%FN%.asm
set FN=%1
u:
t:\masm4\masm.exe %FN%,%FN%,%FN%;
t:\masm4\link.exe %FN%+T:\masm4\ioproc.obj,%FN%;
exit
```

runprog.bat

```
@echo off
echo ---------- Выполнение программы ----------
%1
echo ---------- Программа завершила работу ----------
pause
exit
```

fullma.bat

```
@echo off
if --%~x1==--.asm goto ext_ok
echo Программа должна иметь расширение '.asm' (сейчас '%~x1').
set NAMEFAIL=true
:ext_ok
dir "%~dp1"\%~nx1 > nul 2>nul && goto name_ok
echo В имени файла программы не должно быть пробелов (сейчас '%~n1').
set NAMEFAIL=true
:name_ok
if not --%NAMEFAIL%==--true goto name_ext_ok
echo Исправьте имя файла и запустите компиляцию заново.
rem pause
exit
:name_ext_ok
subst t: /d >nul
set MP="%~dp0."
subst t: %MP%
subst u: /d >nul
set FP="%~dp1."
set FN=%~n1
subst u: %FP%

if exist u:\%FN%.exe del u:\%FN%.exe
call t:\dosbox\dosbox.exe -exit -c "t:\asmprog.bat %FN%" -conf t:\dosbox\dosbox.conf -noconsole
if not exist u:\%FN%.exe goto err

call t:\dosbox\dosbox.exe -c "t:\runprog.bat u:\%FN%.exe" -conf t:\dosbox\dosbox.conf -noconsole
goto fin
:err
if exist u:\%FN%.obj del u:\%FN%.obj
"%~dp0..\npp\notepad++" -n1 "%~dpn1.lst"
echo При компиляции обнаружены ошибки. Смотрите файл %~n1.lst, открытый в редакторе,
для подробной информации.
:fin
subst t: /d
subst u: /d
```

Заключение

Настоящая иллюстрированная пошаговая инструкция описывает возможную последовательность действий, при помощи которой в компьютерных классах факультета ВМК в среде Windows в 2014 году можно получить всё необходимое для написания, отладки и выполнения ассемблерных программ.

Рекомендуемый для использования комплект [1] для работы в среде Windows, содержит, кроме транслятора MASM версии 4.0 , редактор Notepad++, предназначенный для работы с програмным кодом, и виртуальную машину DOSBOX для безопасного исполнения машинного кода учебных программ.

В отличии от [2] данный материал описывает использование компилятора не из коммандной строки, а вызывая макрос текстового редактора Notepad++. Можно предположить, что для многих студентов, больщая часть данного материала окажется слишком детальным. Подробнее ознакомиться с возможностями редактора Notepad++ можно скачав контекстную помощь [3] , не входящую в состав [1] .

Системное описание языка и работы ассемблера можно найти в [4] и [5] .

С рекомендациями по поводу того, как принято оформлять код программ можно ознакомиться, например, в [6] п.3.1 главы 3 .

Ссылки и библиография.

[1] Архив с сопроводительными материалами и настроенными инструментальными средствами
http:// al.cs.msu.su / classes / assembler /

[5] Матвеева Т.К., Матвеев Ф.В. « Инструктивно-справочные материалы по практикуму на MASM’е. Часть 1 » Методическое пособие 2012.
http:// al.cs.msu.su / books /

[2] Контекстная помощь / описание Notepad++
http://notepad-plus.sourceforge.net/

[3] Пильщиков В.Н. Программитование на языке Ассемблера IBM PC. – Диалог-МИФИ, 1994.

[4] Баула В.Г. Введение в архитектуру ЭВМ и системы программирования: Учебно-методическое пособие. МАКС Пресс, 2007.
http:// arch.cs.msu.su / - Лекции по курсу «Архитектура ЭВМ».

[6] Столяров А.В. Оформление програмного кода: методическое пособие. МАКС Пресс, 2012.
http:// www.stolyarov.info /