

Домашнее задание по курсу «Архитектура вычислительных систем»

Семинар 1

Скрыпина Дарья Кирилловна

БПИ 198

Результат работы консольного приложения HelloWorld.asm



```
flat assembler 1.73.25
File Edit Search Run Options Help
; Источник - https://board.flatassembler.net/topic.php?p=150882
format pe console

include 'win32ax.inc'

entry main

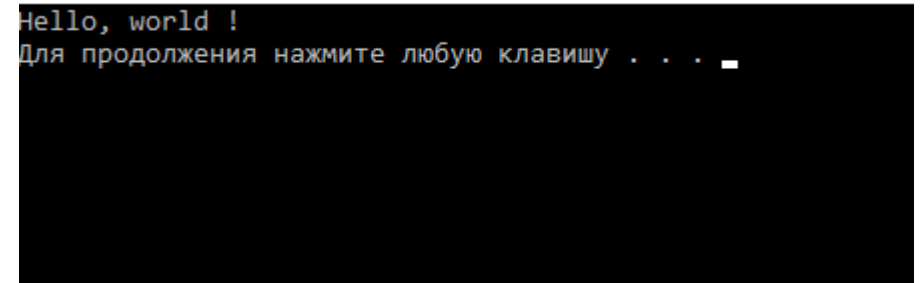
section '.data' data readable writeable
strHello db 'Hello, world !',13,10,0
strPause db 'pause',0

section '.text' code executable readable
main:
    cinvoke printf,strHello
    cinvoke system,strPause

    push 0
    call [ExitProcess]

section '.import' import data readable
library kernel32,'kernel32.dll',\
    msvcrt,'msvcrt.dll'

import kernel32,\
    ExitProcess,'ExitProcess'
import msvcrt,\
    printf,'printf',\
    system,'system'
```



```
Hello, world !
Для продолжения нажмите любую клавишу . . .
```

Программа выводит в
консольное окно надпись
«Hello, world!»

Результат работы консольного приложения Rus_HW.asm

```
flat assembler 1.73.25
File Edit Search Run Options Help
; Источник - http://datadump.ru/fasm-console-application/
format PE console 4.0 ; Консольное приложение.
entry start ; Точка входа
include 'win32a.inc' ; Делаем стандартное включение описателей.
include 'ENCODING\WIN1251.INC' ; Подключить кодировку WIN1251

;--- секция кода ---
section '.text' code readable executable

start:
    invoke SetConsoleOutputCP, 1251
    invoke SetConsoleCP, 1251
    invoke GetStdHandle, STD_OUTPUT_HANDLE
    mov [stdout], eax
    invoke GetStdHandle, STD_INPUT_HANDLE
    mov [stdin], eax
    invoke WriteConsole, [stdout], cMsg, 12, NULL, NULL
    invoke ReadConsole, [stdin], lpBuffer, 1, lpCharsRead, NULL

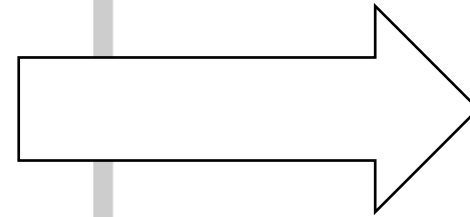
exit:
    invoke ExitProcess, 0

;--- секция данных ---
section '.data' data readable writeable

cMsg db 'Привет, мир!' ; Текстовая строка.
lpBuffer db 10 dup (0)
lpCharsRead dd ? ; Количество фактически считанных символов
stdin dd ?
stdout dd ?

;--- секция (таблица) импорта ---
section '.idata' import data readable writeable

<
Rus_HW.ASM
1,1
```



```
Привет, мир!_
```

Вывод в консоль сообщения «Привет, мир!». Подключение нужной кодировки для корректного отображения символов кириллицы

Результат работы консольного приложения MovingBytes.asm

```
flat assembler 1.73.25
File Edit Search Run Options Help
; Источник - https://www.youtube.com/watch?v=kd1gm4m4Tpw
format PE

entry start

include 'win32a.inc'

section '.data' data readable writable

    resStr db 'Result: %d', 0

    A dw ?
    B dw ?
    C dw ?

    NULL = 0

section '.code' code readable executable

start:
    mov ecx, 110b ;6 в десятичной системе счисления
    shl ecx, 1 ;побитовый сдвиг влево на заданное значение

    push ecx
    push resStr
    call [printf]

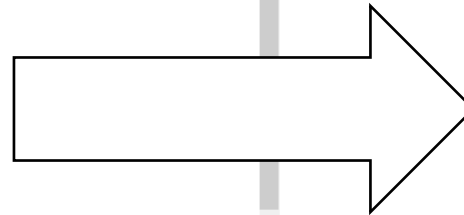
    call [getch]

    push NULL
    call [ExitProcess]

section '.idata' import data readable

    library kernel, 'kernel32.dll', \

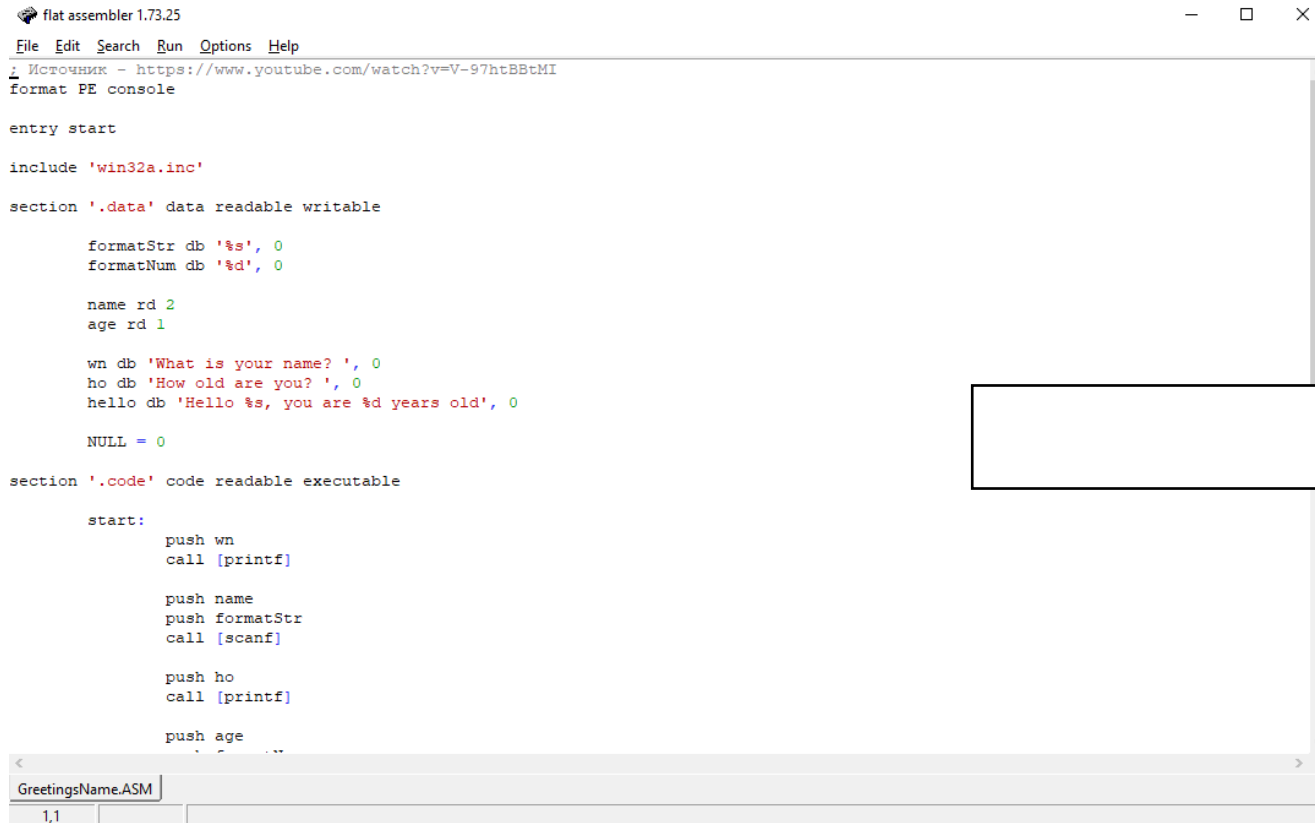
MovingBytes.ASM
1,1
```



```
Result: 12
```

Побитовый сдвиг заданного числа
на заданное значение

Результат работы консольного приложения GreetingsName.asm



```
flat assembler 1.73.25
File Edit Search Run Options Help
Источник - https://www.youtube.com/watch?v=V-97htBBtMI
format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

    formatStr db '%s', 0
    formatNum db '%d', 0

    name rd 2
    age rd 1

    wn db 'What is your name? ', 0
    ho db 'How old are you? ', 0
    hello db 'Hello %s, you are %d years old', 0

    NULL = 0

section '.code' code readable executable

start:
    push wn
    call [printf]

    push name
    push formatStr
    call [scanf]

    push ho
    call [printf]

    push age
```

```
What is your name? Daria
How old are you? 19
Hello Daria, you are 19 years old_
```

Считывание пользовательских
данных (имени и возраста) и
вывод их на экран консоли

Результат работы консольного приложения Calc.asm

```
flat assembler 1.73.25
File Edit Search Run Options Help
; Источник - https://www.youtube.com/watch?v=ttS0tccligpw
format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable

    strA db 'Enter A: ', 0
    strB db 'Enter B: ', 0
    strOp db 'Enter operation: ', 0

    resStr db 'Result: %d', 0
    resMod db '/%d', 0

    spaceStr db ' %d', 0
    emptyStr db '%d', 0

    infinity db 'infinity', 0
    point db ',', 0

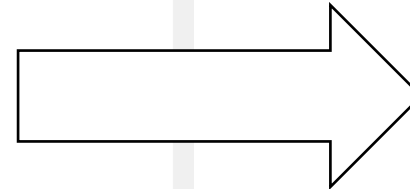
    A dd ?
    B dd ?
    C dd ?

    NULL = 0

section '.code' code readable executable

    Start:
        push strA
        call [printf]

        push A
```



Операция '+'

```
Enter A: 2
Enter B: 3
Enter operation: Result: 5
```

Операция '-'

```
Enter A: 12
Enter B: 5
Enter operation: Result: 7_
```

Операция '*'

```
Enter A: 6
Enter B: 6
Enter operation: Result: 36_
```

Операция '/'

```
Enter A: 27
Enter B: 3
Enter operation: Result: 9,000_
```

Операция '%'

```
Enter A: 27
Enter B: 5
Enter operation: Result: 5 2/5
```

Калькулятор, считывающий оператор (+, -, *, /, %) и два
целочисленных операнда с консоли и выводящий
результат операции