# Министерство образования и науки Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»

NSTU_Logo_blue

## Кафедра прикладной математики

### Лабораторная работа № 1 по дисциплине «Языки программирования и методы трансляции»

**Основные элементы языка ассемблера**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| сигма градиент синий1 | Факультет: | ПМИ |  |  |
| Группа: | ПМ-63 |  |  |
| Студент: | Шепрут И.И. |  |  |
| Вариант: | 8 |  |  |
| Преподаватели: | Еланцева И.Л.  Петров Р.В. |  |  |

Новосибирск

2018

1. Задание

Разработать программу на языке Ассемблера, решающую поставленную задачу:

1. Ввод с клавиатуры 2-х чисел в заданной системе счисления;
2. Выполнение арифметической операции над этими числами (в предположении, что размер чисел не вызывает переполнения регистров);
3. Вывод результата в заданной системе счисления.

Условия:

* Все промежуточные данные должны сохраняться в памяти.
* При выводе результата не использовать функцию wsprintfA.

1. Алгоритм

|  |
| --- |
| Ввести число  Ввести строку  Перевести все символы в нижний регистр  Проверить, являются ли все символы шестнадцатеричными цифрами  Если нет, то попросить пользователя ввести заного  Цикл по цифрам  Перевести текущую цифру из символа в число  Добавить в EAX  Умножить EAX на 16  Разделить EAX на 16  Результат: EAX  Вывести число (получает EAX)  Пока EAX не равно нулю  Разеделить EAX на 10  Поместить остаток в стек  Перебор всех данных в стеке  Взять из стека  Преобразовать число в символ цифры  Вывести в буфер в текущую позицию  Сместиться в буфере  Добавить символы конца строки  Вывести строку  Главная  Ввести число 1  Ввести число 2  Умножить число 1 на 2, поместить в EAX  Вывести число |

|  |  |
| --- | --- |
|  |  |
|  |  |
| 1. Текст программы  |  | | --- | | .386  .MODEL FLAT, STDCALL  EXTERN GetStdHandle@4: PROC  EXTERN WriteConsoleA@20: PROC  EXTERN CharToOemA@8: PROC  EXTERN ReadConsoleA@20: PROC  EXTERN ExitProcess@4: PROC  EXTERN lstrlenA@4: PROC  ;------------------------------------------------------------------------------  ;------------------------------------------------------------------------------  ;------------------------------------------------------------------------------  .DATA  enterStr1 DB "Введите первое число в 16 системе счисления: ", 0  enterStr1Len DD ?  enterStr2 DB "Введите второе число в 16 системе счисления: ", 0  enterStr2Len DD ?  writeStr DB "Результат умножения этих чисел в 10 системе счисления: ", 0  writeStrLen DD ?  errorStr DB "Вы ввели число неправильно, попробуйте снова: ", 0  errorStrLen DD ?  firstNumber DD ?  secondNumber DD ?  DIN DD ?  DOUT DD ?  BUF DB 16 dup (?)  readedLen DD ?  digitsCount DD ?  ;------------------------------------------------------------------------------  ;------------------------------------------------------------------------------  ;------------------------------------------------------------------------------  .CODE  ;------------------------------------------------------------------------------  ; Записывает в EAX введенное число в шестнадцатеричной системе счисления.  ; Если число введено неправильно, повторяет снова  EnterNumber PROC  START:  ; Ввод данных  PUSH 0  PUSH OFFSET readedLen  PUSH 16  PUSH OFFSET BUF  PUSH DIN  CALL ReadConsoleA@20  SUB readedLen, 2  CMP readedLen, 0  JE WRITE\_INCORRECT  ; Перевод в нижний регистр  MOV ECX, readedLen  MOV ESI, OFFSET BUF  XOR EAX, EAX  TOLOW:  MOV AL, [ESI]  CMP AL, 'A'  JB ELSE\_UPPER  CMP AL, 'Z'  JA ELSE\_UPPER  SUB AL, 'A'  ADD AL, 'a'  MOV [ESI], AL  ELSE\_UPPER:  INC ESI  LOOP TOLOW  ; Проверка на корректность  MOV ECX, readedLen  MOV ESI, OFFSET BUF  CHECK:  XOR EAX, EAX  XOR EBX, EBX  MOV AL, [ESI]  CMP AL, 'a'  JB NOT\_SYMBOL  CMP AL, 'f'  JA NOT\_SYMBOL  MOV BL, 1  NOT\_SYMBOL:  CMP AL, '0'  JB NOT\_NUMBER  CMP AL, '9'  JA NOT\_NUMBER  JMP ALL\_CORRECT  WRITE\_INCORRECT:  PUSH 0  PUSH OFFSET errorStrLen  PUSH errorStrLen  PUSH OFFSET errorStr  PUSH DOUT  CALL WriteConsoleA@20  JMP START  NOT\_NUMBER:  CMP BL, 1  JNE WRITE\_INCORRECT  ALL\_CORRECT:  INC ESI  LOOP CHECK  ; Перевод числа в регистр EAX  MOV ECX, readedLen  MOV ESI, OFFSET BUF  XOR EAX, EAX  TRANSLATION:  XOR EBX, EBX  MOV BL, [ESI]  CMP BL, 'a'  JB NOT\_SYMBOL1  CMP BL, 'f'  JA NOT\_SYMBOL1  SUB BL, 'a'  ADD BL, 10  ADD EAX, EBX  NOT\_SYMBOL1:  CMP BL, '0'  JB NOT\_NUMBER2  CMP BL, '9'  JA NOT\_NUMBER2  SUB BL, '0'  ADD EAX, EBX  NOT\_NUMBER2:  INC ESI  MOV EDX, 16  MUL EDX  LOOP TRANSLATION  MOV ECX, 16  DIV ECX  RET  EnterNumber ENDP  ;------------------------------------------------------------------------------  ; Выводит число из регистра EAX в 10 системе счисления  WriteNumber PROC  MOV digitsCount, 0  MOV ECX, 10  DECOMPOSE:  XOR EDX, EDX  DIV ECX  PUSH EDX  INC digitsCount  CMP EAX, 0  JNE DECOMPOSE  MOV ECX, digitsCount  MOV ESI, OFFSET BUF  WRITING:  POP EAX  ADD EAX, '0'  MOV [ESI], AL  INC ESI  LOOP WRITING  MOV AL, 13  MOV [ESI], AL  INC ESI  MOV AL, 10  MOV [ESI], AL  INC ESI  MOV AL, 0  MOV [ESI], AL  INC ESI  ADD digitsCount, 2  ; вывод полученной строки  PUSH 0  PUSH OFFSET digitsCount  PUSH digitsCount  PUSH OFFSET BUF  PUSH DOUT  CALL WriteConsoleA@20  RET  WriteNumber ENDP  ;------------------------------------------------------------------------------  MAIN PROC  ; Получим дескриптор ввода  PUSH -10  CALL GetStdHandle@4  MOV DIN, EAX  ; Получим дескриптор вывода  PUSH -11  CALL GetStdHandle@4  MOV DOUT, EAX  ;--------------------------------------------------------------------------  ; Перекодируем строку enterStr  MOV EAX, OFFSET enterStr1  PUSH EAX  PUSH EAX  CALL CharToOemA@8  MOV EAX, OFFSET enterStr2  PUSH EAX  PUSH EAX  CALL CharToOemA@8  ; Перекодируем строку writeStr  MOV EAX, OFFSET writeStr  PUSH EAX  PUSH EAX  CALL CharToOemA@8  ; Перекодируем строку errorStr  MOV EAX, OFFSET errorStr  PUSH EAX  PUSH EAX  CALL CharToOemA@8  ;--------------------------------------------------------------------------  ; Определим длину строки enterStr  PUSH OFFSET enterStr1  CALL lstrlenA@4  MOV enterStr1Len, EAX  PUSH OFFSET enterStr2  CALL lstrlenA@4  MOV enterStr2Len, EAX  ; Определим длину строки writeStr  PUSH OFFSET writeStr  CALL lstrlenA@4  MOV writeStrLen, EAX  ; Определим длину строки erorrStr  PUSH OFFSET errorStr  CALL lstrlenA@4  MOV errorStrLen, EAX  ;--------------------------------------------------------------------------  ;--------------------------------------------------------------------------  ;--------------------------------------------------------------------------  ; Выводим enterStr1  PUSH 0  PUSH OFFSET enterStr1Len  PUSH enterStr1Len  PUSH OFFSET enterStr1  PUSH DOUT  CALL WriteConsoleA@20  ; Ввод первого числа  CALL EnterNumber  MOV firstNumber, EAX  ;--------------------------------------------------------------------------  ; Выводим enterStr2  PUSH 0  PUSH OFFSET enterStr2Len  PUSH enterStr2Len  PUSH OFFSET enterStr2  PUSH DOUT  CALL WriteConsoleA@20  ; Ввод второго числа  CALL EnterNumber  MOV secondNumber, EAX  ;--------------------------------------------------------------------------  ; Умножаем два числа  MOV EAX, firstNumber  MOV EDX, secondNumber  MUL EDX  MOV firstNumber, EAX  ;--------------------------------------------------------------------------  ; Выводим строку о результатах вычислений  PUSH 0  PUSH OFFSET writeStrLen  PUSH writeStrLen  PUSH OFFSET writeStr  PUSH DOUT  CALL WriteConsoleA@20  ;--------------------------------------------------------------------------  ; Выводим число в десятичной системе счисления  MOV EAX, firstNumber  CALL WriteNumber  ;--------------------------------------------------------------------------  ;--------------------------------------------------------------------------  ;--------------------------------------------------------------------------  MOV ECX,0FFFFFFFFH  L1: LOOP L1  PUSH 0  CALL ExitProcess@4  MAIN ENDP  END MAIN |  1. Тестирование |  |

|  |  |  |
| --- | --- | --- |
| № | Результат работы программы | Пояснение |
| 1 |  | - |
| 2 |  | Тестирование перевода строки в нижний регистр.  ABCD16 = 4398110 |
| 3 |  | Тестирование ввода некорректных символов и пустой строки.  ABCD16 = 4398110 |
| 4 |  | DEAD16 = 5700510  111116 = 436910  57005 \* 4369 = 249054845 |
| 5 |  | FFFF16 = 6553510  65535 \* 65535 = 4294836225 |