Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: «КСиС»

Тема: «Макроопределения»

Вариант №6

Выполнил:

Студент 2 курса

Группы ПО-7

Комиссаров А.Е.

Проверил:

Савицкий Ю. В.

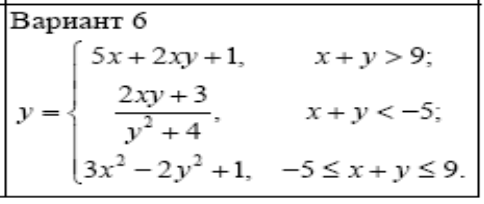
2022

**Цель**: изучить приемы разработки макроопределений и использования их в программах.

**Задание:**

Написать макроопределение, реализующее функцию заданного преподавателем варианта

работы: Удалить все пробелы из строки символов



**Код программы:**

section .data *;Текстовые сообщения*

entermsg db 'Enter your message: '

enlen equ $ - entermsg

resultmsg db 0xa, 'Final result: '

reslen equ $ - resultmsg

myString db 'This is my input string' *;Строка, которую*

strlen equ $ - myString *; будем обрабатывать*

*;==========================================*

section .bss

buff resb strlen *;Зарезервируем буферную строку*

section .text

global \_start *;Точка входа*

*;======*====================================

*;Макрос на вывод сообщения %1 длиной в %2*

%macro printMsg 2

mov ebx, 1

mov eax, 4

mov ecx, %1

mov edx, %2

int 0x80

%endmacro

*;Конец макроопределения*

*;==========================================*

*;Начало макроопределения, удаляющего все пробелы из строки*

%macro deleteSpaces 2

lea eax, [%1] *;Заносим адрес %1 в регистр eax.*

mov edi, buff *;Заносим буферную строку в edi.*

mov esi, eax *;Заносим eax в esi и с ней будем работать.*

mov ecx, %2 *;Приравниваем ecx (счётчик цикла loop) к длине строки,*

*; чтобы loop выполнился такое кол-во раз,*

*; как и длина строки.*

jmp check *;Прыжок в check.*

check: mov al, ' ' *;Заносим в al символ пробела, чтобы сравнить.*

cmp [esi], al *;Сравниваем символ строки с пробелом.*

jne copy *;Если они не равны, прыгаем в copy.*

je nxt *;Если равны, переходим на следующий символ (прыгаем в nxt).*

nxt: inc esi *;Переход на следующий символ исходной строки.*

loop check *;Начинаем проверку следующего символа (прыжок в check).*

jmp ext *;Если прыжок в check не произошёл, это значит что мы*

*; дошли до конца строки (регистр ecx стал равен нулю),*

*; значит выходим из нашего макроопределения.*

copy: mov al, [esi] *;Переносим из исходной строки символ в регистр al*

mov [edi], al *;Переносим из регистра al символ в буферную строку.*

inc edi *;Переходим на следующий символ буферной строки.*

jmp nxt *;Прыжок в nxt.*

ext: *;выход из макроопределения*

%endmacro

*;Таким образом мы скопировали каждый символ,*

*; который не является пробелом, в буферную строку.*

*;закончили макроопределение*

*;==========================================*

*;начало выполнения программы*

\_start:

printMsg entermsg, enlen *;"Enter your message:"*

printMsg myString, strlen *;"This is my input string"*

deleteSpaces myString, strlen *;собственно обработка строки*

printMsg resultmsg, reslen *;"Final result: "*

printMsg buff, strlen *;"Thisismyinputstring"*

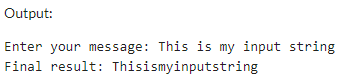
;завершение работы программы

mov ebx,0 *;код выхода*

mov eax,1 *;(sys\_exit)*

int 80h *;вызов ядра*

**Результат работы программы:**

****

Вывод: я изучил приёмы разработки макроопределений и использования их в программах.