|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«*Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по домашнему заданию № 1**

# Дисциплина: Машинно-зависимые языки и основы компиляции

**Название домашнего задания:** Обработка символьной информации.

Студент гр. ИУ6-42Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Д. Шатский**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ C.C.Данилюк**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Вариант 2.23**

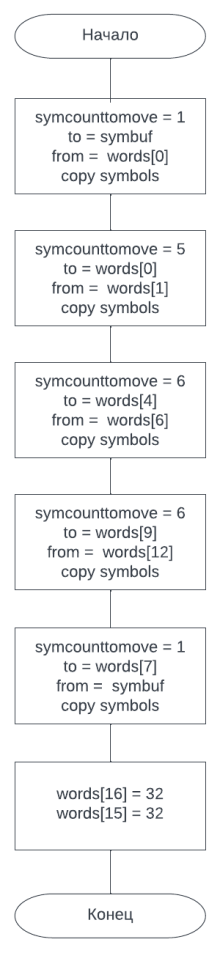
**Цель работы.** Изучение команд обработки цепочек и приемов   
обработки символьной информации.

**Задание.**

Дан текст 18 символов: 3 слова по 6 символов. Осуществить кольцевой сдвиг каждого слова: 1-го на 1 символ влево, 2-го на 2 символа, 3-го на 3 символа.

**Схема.**

Схему алгоритма можно увидеть на рисунке 1:



*Рисунок 1 – Схема алгоритма*

**Программа.**

section .data

ExitMsg db "Answer:",10

lenExit equ $-ExitMsg

EnterMsg db "Ender string:",10

lenEnter equ $-EnterMsg

wordsLenOut equ 19

wordsLenIn equ 19

section .bss

words resb 19

symbuf resb 1

section .text

global \_start

\_start:

mov rax, 1

mov rdi, 1

mov rsi, EnterMsg

mov rdx, lenEnter

syscall

; read

mov rax, 0

mov rdi, 0

mov rsi, words

mov rdx, wordsLenIn

syscall

mov RCX,1

lea RDI,[symbuf]

lea RSI,[words]

rep movsb

mov RCX,5

lea RDI,[words]

lea RSI,[words+1]

rep movsb

mov RCX,6

lea RDI,[words+4]

lea RSI,[words+6]

rep movsb

mov RCX,6

lea RDI,[words+9]

lea RSI,[words+12]

rep movsb

mov RCX,1

lea RDI,[words+17]

lea RSI,[symbuf]

rep movsb

mov BYTE[words+16],32

mov BYTE[words+15],32

mov rax, 1

mov rdi, 1

mov rsi, words

mov rdx, wordsLenOut

syscall

mov rax, 60; системная функция 60 (exit)

xor rdi, rdi; return code 0

syscall

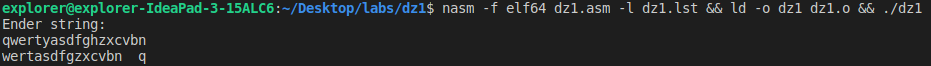
**Тесты.**

Результаты тестирования можно увидеть в таблице 1.

Таблица 1 – результаты тестирования

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
| aaaaaabbbbbbcccccc | aaaabbbbbcccccc a | aaaabbbbbcccccc a |
| qwertyasdfghzxcvbn | wertasdfgzxcvbn q | wertasdfgzxcvbn q |
| qwertyuiopasdfghjk | wertuiopadfghjk q | wertuiopadfghjk q |
| 123456123456123456 | 234512345123456 1 | 234512345123456 1 |
| 123456789123456789 | 234578912456789 1 | 234578912456789 1 |

Результат запуска программы на выполнение можно увидеть на рисунке 2.



*Рисунок 2 – Результат выполнения программы*

**Контрольные вопросы.**

**1. Дайте определение символьной строки.**

Символьная строка – это некоторая последовательность байт, содержащая некоторую последовательность кодов символов, в конце которой находится знак конца строки.

**2. Назовите основные команды обработки цепочек?**

lodsb, lodsw, lodsd - загрузка элемента из цепочки

stosb, stows, stosd - сохранение элемента в цепочке

movsb, movsw, movsd - пересылка цепочки

cmpsb, cmpsw, cmpsd - сравнение цепочек

scasb, scasw, scads - сканирование цепочки

**3. Какие операции выполняют строковые команды MOVS? Какие особенности характерны для этих команд?**

Строковые команды MOVS выполняют операцию пересылки элемента из цепочки-источника по адресу в DS:ESI/SI, в цепочку-приемник по адресу в ES:EDI/DI. После пересылки элементов, регистры ESI/SI и EDI/DI изменяются в соответствии со значением флага DF. Если перед командой написать префикс rep, то командой можно переслать несколько элементов, загрузив счётчик ECX перед исполнением команды.

**4. Какие операции выполняют строковые команды CMPS, SCAS? Какие особенности характерны для этих команд?**

Строковые команды CMPS выполняют сравнение элементов цепочек,

находящихся по адресам DS:ESI/SI и ES:EDI/DI. Данная команда выполняет

вычитание (источник - приемник) над элементами обеих цепочек. Результат

не записывается. Меняются флаги ZF, SF и OF. Префиксы повторения repe/repz, repne/repnz проверяют различные и совпадающие элементы.

Строковые команды SCAS выполняют сравнение элемента цепочки по

адресу ES:EDI/DI с значением, которое хранится в AL/AX/EAX. Принцип

работы аналогичен CMPS. Префикс повторения repne/repnz используется для

поиска элемента равного значению регистра-аккумулятора, а repe/repz для

нахождения отличного от значения регистра-аккумулятора. Адрес ES:EDI/DI

указывает на следующий после элемента, из-за которого был остановлен поиск. ESI/SI и EDI/DI изменяются в соответствии со значением флага DF.

**5. Как обеспечить циклическую обработку строк?**

Для циклической обработки строк используются префиксы повторения

rep, repe/repz, repne/repnz.

Префикс rep перед командой будет выполняться количество раз,

сколько внесено в регистр-счётчик ECX/CX.

Префиксы repe/repz и repne/repnz используют вместе с командами

CMPS и SCAS для сравнения строк и поиска элементов.

Все цепочечные команды, кроме обработки текущего элемента,

производят автоматическое продвижение к следующему элементу.

При этом изменяются индексные регистры ESI/SI и EDI/DI в соответствии с флагом DF.

**6. Какова роль флага DF во флажковом регистре при выполнении   
команд обработки строк?**

Цепочечные команды выполняют изменение индексных регистров.

Знак изменений определяется значением флага направления DF.

Если DF = 0, то значение индексных регистров будет увеличиваться, то

есть обработка в направлении возрастания адресов.

Если DF = 1, то наоборот значение индексных регистров будет уменьшаться, то есть обработка в направлении убывания адресов.

Команда CLD - очистить флаг направления. DF = 0

Команда STD - установить флаг направления. DF = 1

**7. Как правильно выбрать тестовые данные для проверки   
алгоритма обработки строки?**

Для проверки алгоритма обработки строки необходимо проверить корректность работы всех частей программы, поэтому тестовые данные должны содержать такие исходные данные, чтобы все части программы были протестированы.

**Вывод.** Былиизучены команды обработки цепочек и приемов   
обработки символьной информации.Программа работает корректно на заданных тестовых данных.