Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 3

Тема «Знакомство с lc-3»

по дисциплине «Языки программирования и методы трансляции»

Выполнил: студент группы ИСТ-22-1б Петраков М.В.

Проверил: доцент каф. ВММБ Батин С.Е.

Пермь, 2023

[Задание 1 3](#_Toc151294367)

[1.1. Постановка задания 3](#_Toc151294368)

[1.2. Решение задачи, код программы 3](#_Toc151294369)

[1.3. Тестирование работы программы 3](#_Toc151294370)

[Задание 2 5](#_Toc151294371)

[2.1. Постановка задания 5](#_Toc151294372)

[2.2. Решение задачи, код программы 5](#_Toc151294373)

[2.3. Тестирование работы программы 6](#_Toc151294374)

[Задание 3 8](#_Toc151294375)

[3.1. Постановка задания 8](#_Toc151294376)

[3.2. Решение задачи, код программы 8](#_Toc151294377)

[3.3. Тестирование работы программы 9](#_Toc151294378)

[Задание 4 11](#_Toc151294379)

[4.1. Постановка задания 11](#_Toc151294380)

[4.2. Решение задачи, код программы 11](#_Toc151294381)

[4.3. Тестирование работы программы 13](#_Toc151294382)

[Задание 5 14](#_Toc151294383)

[5.1. Постановка задания 14](#_Toc151294384)

[5.2. Решение задачи, код программы 14](#_Toc151294385)

[5.3. Тестирование работы программы 16](#_Toc151294386)

Задание 1

1.1. Постановка задания

Написать и запустить программу, выводящую строку Hello World.

1.2. Решение задачи, код программы

.ORIG x3000

  LEA R0, HELLOWORLD

  PUTS

  HALT

HELLOWORLD .STRINGZ "Hello, World!"

.END

1.3. Тестирование работы программы



Рис.1 Успешная работа программы hello\_world.asm(obj)

Задание 2

2.1. Постановка задания

Написать и запустить программу, принимающую строку пользователя и выводящую ее в терминал.

2.2. Решение задачи, код программы

.ORIG x3000

    LEA R1, STORESTRING

    LEA R0, USERINPUT

    PUTS

LOOP

   GETC

   OUT

   STR R0, R1, #0

   ADD R1, R1, #1

   ADD R0, R0, #-10

   BRz OUTSIDE

   BRnzp LOOP

OUTSIDE

   LEA R0, OUTPUTTEXT

   PUTS

   LEA R0, STORESTRING

   PUTS

   HALT

USERINPUT .STRINGZ "\nPlease enter a text: "

OUTPUTTEXT .STRINGZ "Text you have typed is: "

STORESTRING .BLKW #999

.END

2.3. Тестирование работы программы

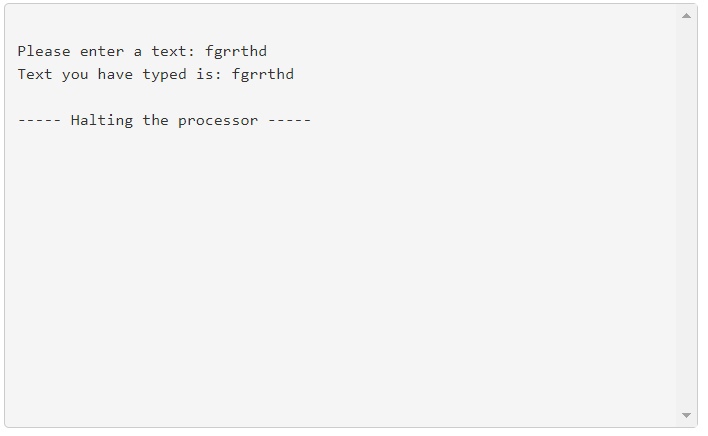


Рис.2 Успешная работа программы PutS

Задание 3

3.1. Постановка задания

Написать и запустить программу, реализующую сортировку заданного пользователем целочисленного массива из 10 целых чисел.

3.2. Решение задачи, код программы

.ORIG x3000

; Вывод первой части заспроса

LEA R0, FIRSTPROMPT

PUTS

; Вывод разделителя

LEA R0, SPACE

PUTS

; Вывод примера ввода

LEA R0, SAMPLE

PUTS

; Инициализация регистров и счетчиков

LD R3, POINTER

LD R6, IOCOUNTER

; Цикл ввода

INPUT

    IN ;поглотить символ

    AND R2, R2, #0     ; Очистка регистра R2

    AND R5, R5, #0     ; Очстка регистра R5

    LD R5, PLACEHUNDRED ; Загрузка сотни

    LD R2, HEXN48    ; Загрузка 16^4-48

    ADD R0, R0, R2 ;переместить значение

    ADD R2, R0, #0 ;записать значение в R2

    AND R0, R0, #0 ; Очистить поток

FIRST\_NUM

    ADD R0, R0, R2 ;перезаписать значение в R0

    ADD R5, R5, #-1 ;уменьшить количество на 1

    BRp FIRST\_NUM

    ADD R1, R0, #0 ; сохранить в R1

    AND R0, R0, #0 ; очистка

    IN ;снова ввод, повторение

    AND R2, R2, #0

    AND R5, R5, #0

    LD R5, PLACETEN

    LD R2, HEXN48

    ADD R0, R0, R2

    ADD R2, R0, #0

    AND R0, R0, #0

SECOND\_NUM

    ADD R0, R0, R2

    ADD R5, R5, #-1

    BRp SECOND\_NUM

    ADD R4, R0, #0

    AND R0, R0, #0

    IN

    AND R2, R2, #0

    LD R2, HEXN48

    ADD R0, R0, R2

    AND R2, R2, #0

    ADD R2, R1, R4

    ADD R2, R0, R2

    STR R2, R3, #0

    ADD R3, R3, #1

    ADD R6, R6, #-1

    BRp INPUT

    JSR BUBBLESORT

    JSR PRODUCTLOOP

    HALT

; Подпрограмма сортировки пузырьком

BUBBLESORT

    AND R3, R3, #0

    LD R3, POINTER

    AND R4, R4, #0

    LD R4, IOCOUNTER

    AND R5, R5, #0

    LD R5, IOCOUNTER

OUT\_LOOP

    ADD R4, R4, #-1

    BRz SORTED

    ADD R5, R4, #0

    LD R3, POINTER

IN\_LOOP

    LDR R0, R3, #0

    LDR R1, R3, #1

    AND R2, R2, #0

    NOT R2, R1

    ADD R2, R2, #1

    ADD R2, R0, R2

    BRn AUTOSWAP

    STR R1, R3, #0

    STR R0, R3, #1

AUTOSWAP

    ADD R3, R3, #1

    ADD R5, R5, #-1

    BRp IN\_LOOP

    BRzp OUT\_LOOP

SORTED

    RET

; Подпрограмма для вывода результата

PRODUCTLOOP

    LEA R0, PROMPTEXECUTE

    PUTS

    LD R3, POINTER

    LD R6, IOCOUNTER

RESULTLOOP

    AND R1, R1, #0

    AND R2, R2, #0

    AND R4, R4, #0

    AND R5, R5, #0

    AND R0, R0, #0

    LD R0, SPACE

    OUT

    AND R0, R0, #0

    LDR R0, R3, #0

    LD R2, PLACEHUNDRED

    NOT R2, R2

    ADD R2, R2, #1

MINUS01

    ADD R1, R1, #1

    ADD R0, R0, R2

    BRzp MINUS01

REMAINDER01

    AND R2, R2, #0

    LD R2, PLACEHUNDRED

    ADD R0, R0, R2

    ADD R1, R1, #-1

    STI R1, FIRSTNUM

    AND R2, R2, #0

    LD R2, PLACETEN

    NOT R2, R2

    ADD R2, R2, #1

MINUS02

    ADD R4, R4, #1

    ADD R0, R0, R2

    BRzp MINUS02

REMAINDER2

    AND R2, R2, #0

    LD R2, PLACETEN

    ADD R5, R0, R2

    STI R5, THIRDNUM

    ADD R4, R4, #-1

    STI R4, SECONDNUM

    AND R0, R0, #0

    LDI R0, FIRSTNUM

    AND R2, R2, #0

    LD R2, HEX48

    ADD R0, R0, R2

    OUT

    AND R0, R0, #0

    LDI R0, SECONDNUM

    AND R2, R2, #0

    LD R2, HEX48

    ADD R0, R0, R2

    OUT

    AND R0, R0, #0

    LDI R0, THIRDNUM

    AND R2, R2, #0

    LD R2, HEX48

    ADD R0, R0, R2

    OUT

    ADD R3, R3, #1

    ADD R6, R6, #-1

    BRp RESULTLOOP

    HALT

; Строковые константы и инициализация данных

FIRSTPROMPT .STRINGZ "Enter 10numbers in range 000 - 999 with 3 digits."

SAMPLE .STRINGZ "Example: 15 as 015, 5 as 005"

PROMPTEXECUTE .STRINGZ "Number in ascending order: "

SPACE .STRINGZ "\n"

HEXN48 .FILL xFFD0

HEX48 .FILL x0030

PLACEHUNDRED .FILL x0064

PLACETEN .FILL x000A

POINTER .FILL x4000

IOCOUNTER .FILL #10

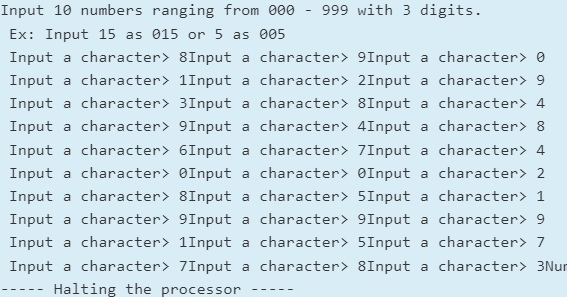
FIRSTNUM .FILL x400A

SECONDNUM .FILL x400B

THIRDNUM .FILL x400C

.END

3.3. Тестирование работы программы



https://sun9-1.userapi.com/impg/H1CtiD6u6za0-piWsYDsR-dvaEAsgJgEMWgsTg/Qzj2ocLuBPc.jpg?size=612x22&quality=96&sign=9dbd6fc71733c082ad8592ff183d6c2d&type=album

Рис.3 Работа программы mass

Задание 4

4.1.Постановка задания

Написать и запустить программу, реализующую сортировку заданной пользователем строки из 10 цифр

4.2. Решение задачи, код программы

.ORIG x3000

    LEA R0, PROMPT       ; Загрузить адрес строки приглашения в R0

    PUTS                 ; Вывести строку приглашения

    LEA R1, NUMBERS      ; Загрузить адрес массива чисел в R1

    LEA R4, ORIGINAL     ; Загрузить адрес массива для сохранения исходных чисел

    LD R2, COUNT         ; Загрузить счетчик в R2

READ\_NUMBERS

    GETC                 ; Считать символ с клавиатуры в R0

    OUT                  ; Вывести считанный символ

    STR R0, R4, #0       ; Сохранить исходный символ

    ADD R4, R4, #1       ; Перейти к следующему месту в массиве исходных чисел

    LD R3, NEG48         ; Загрузить -48 в R3

    ADD R0, R0, R3       ; Преобразовать ASCII-код в число

    STR R0, R1, #0       ; Сохранить число в массив

    ADD R1, R1, #1       ; Перейти к следующему месту в массиве

    ADD R2, R2, #-1      ; Уменьшить счетчик

    BRp READ\_NUMBERS     ; Повторять, пока счетчик не исчерпан

    LEA R1, NUMBERS      ; Адрес начала массива

    LD R2, COUNT         ; Счетчик чисел

    ADD R2, R2, #-1      ; Уменьшить счетчик, так как сортируем пары чисел

SORT\_LOOP

    LD R4, COUNT         ; Счетчик для внутреннего цикла

    ADD R4, R4, #-1

    LEA R3, NUMBERS      ; Сброс R3 к началу массива

INNER\_LOOP

    LDR R0, R3, #0       ; Загрузить текущее число

    LDR R1, R3, #1       ; Загрузить следующее число

    NOT R5, R1

    ADD R5, R5, #1

    ADD R5, R0, R5       ; Сравнить числа

    BRn SKIP\_SWAP

    STR R1, R3, #0       ; Меняем местами

    STR R0, R3, #1

SKIP\_SWAP

    ADD R3, R3, #1       ; Перейти к следующему числу

    ADD R4, R4, #-1      ; Уменьшить внутренний счетчик

    BRp INNER\_LOOP       ; Повторять внутренний цикл

    ADD R2, R2, #-1      ; Уменьшить внешний счетчик

    BRp SORT\_LOOP        ; Повторять внешний цикл

    ; Вывод новой строки для разделения введенной и отсортированной строк

    LD R0, NEWLINE

    OUT

    ; Вывод исходной введенной строки

    LEA R1, ORIGINAL

    LD R2, COUNT

PRINT\_ORIGINAL

    LDR R0, R1, #0       ; Загрузить символ из исходной строки

    OUT                  ; Вывести символ

    ADD R1, R1, #1       ; Перейти к следующему символу

    ADD R2, R2, #-1      ; Уменьшить счетчик

    BRp PRINT\_ORIGINAL   ; Повторять до вывода всех символов

    ; Вывод новой строки для разделения исходной и отсортированной строк

    LD R0, NEWLINE

    OUT

    ; Вывод отсортированных чисел

    LEA R1, NUMBERS

    LD R2, COUNT

PRINT\_NUMBERS

    LDR R0, R1, #0       ; Загрузить число из массива

    LD R3, POS48         ; Загрузить 48 в R3

    ADD R0, R0, R3       ; Преобразовать число в ASCII-код

    OUT                  ; Вывести число

    ADD R1, R1, #1       ; Перейти к следующему числу

    ADD R2, R2, #-1      ; Уменьшить счетчик

    BRp PRINT\_NUMBERS    ; Повторять до вывода всех чисел

    HALT                 ; Остановка программы

PROMPT .STRINGZ "Enter 10 numbers: "

NUMBERS .BLKW 10        ; Массив для хранения чисел

ORIGINAL .BLKW 10       ; Массив для хранения исходных символов

COUNT .FILL #10         ; Счетчик чисел для ввода

NEG48 .FILL #-48        ; Значение -48 для преобразования ASCII в число

POS48 .FILL #48         ; Значение 48 для преобразования числа в ASCII

NEWLINE .FILL x000A     ; Значение ASCII для перевода строки

.END

4.3. Тестирование работы программы

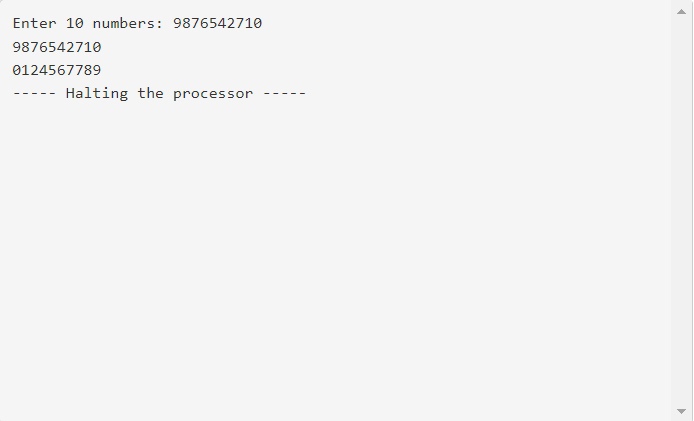


Рис. 4 Успешная работа программы number\_sort