

Архитектура вычислительных систем
ИДЗ 3. Отчёт
Вариант 23
Работа на 10 баллов

Фролов-Буканов Виктор Дмитриевич БПИ-228

25 ноября 2023

1 Условие задачи

Разработать программу, которая ищет в ASCII-строке заданную подстроку и возвращает **индекс первого символа первого вхождения подстроки в строке**. Подстрока вводится как параметр. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

2 Описание метода решения задачи

Для решения задачи я реализовал макробибблиотеку вместо реализации подпрограмм, так как это более удобно и так как это является требованием на более высокую оценку. Макросы чтения из файла и записи в файл реализованы отдельно, равно как и макрос с алгоритмом непосредственного решения задачи и другие вспомогательные макросы

3 Настройки RARS

<input type="checkbox"/>	Show Labels Window (symbol table)
<input type="checkbox"/>	Program arguments provided to program
<input checked="" type="checkbox"/>	Popup dialog for input syscalls (5,6,7,8,12)
<input checked="" type="checkbox"/>	Addresses displayed in hexadecimal
<input checked="" type="checkbox"/>	Values displayed in hexadecimal
<hr/>	
<input type="checkbox"/>	Assemble file upon opening
<input checked="" type="checkbox"/>	Assemble all files in directory
<input type="checkbox"/>	Assemble all files currently open
<input type="checkbox"/>	Assembler warnings are considered errors
<input checked="" type="checkbox"/>	Initialize Program Counter to global 'main' if defined
<input checked="" type="checkbox"/>	Derive current working directory
<hr/>	
<input checked="" type="checkbox"/>	Permit extended (pseudo) instructions and formats
<input type="checkbox"/>	Self-modifying code
<input type="checkbox"/>	64 bit
<hr/>	
Editor...	
Highlighting...	
Exception Handler...	
Memory Configuration...	

4 Описание тестовых прогонов с представлением информации о результатах тестирования

Все тесты лежат в файле tests.txt Содержимое файла:

```
file1.txt
ple
ans1.txt
Y
file2.txt
upala
ans2.txt
N
file3.txt
t
ans3.txt
Y
file4.txt
gg
ans4.txt
Y
file5.txt
789
ans5.txt
N
file6.txt
k
ans6.txt
Y
file7.txt
blue
ans7.txt
N
file8.txt
-1
ans8.txt
Y
file9.txt
red
ans9.txt
N
file10.txt
federer
ans10.txt
N
```

Дополнительная тестовая программа *main.cpp* сначала создаёт файлы `file1.txt` ... `file10.txt`, которые будут хранить строки для 10 тестов. Это **неизменяемый** параметр программы. Далее программа создаёт отдельные тестовые файлы `test1.txt` ... `test10.txt`, на которых, собственно, и будет тестироваться программа. Важно ещё то, что программа дополнительно к названиям файлов с расширением

добавляет полные пути до этих файлов. Если этого не сделать, то при дальнейшем тестировании RARS почему-то не будет видеть файлы, несмотря на то, что у меня стоит галочка "Derive current working directory". При запуске программы на своем компьютере пути к файлам нужно поменять в соответствии с тем, какие они стоят у вас. Содержимое файлов test1.txt ...test10.txt является уже **изменяемыми** параметрами программы, которые формально вводятся пользователем с консоли симулятора RARS, но далее в программе пользователь ничего непосредственно вводить не будет, так как будет использоваться оператор переадресации ввода "<", который вместо ручного ввода введет в RARS данные из каждого тестового файла test1.txt ...test10.txt. Собственно, именно это и делает дальше программа. Она вызывает команду консоли виндовс для запуска написанной ассемблерной программы, делая переадресацию ввода на каждый конкретный файл. В результате работы программы получим такой вывод (он ниже), но нам важно то, что будут сгенерированы файлы с ответом в нужной нам директории. В следующей секции приведем несколько скриншотов, демонстрирующих корректную работу программы. Все ответы на тесты и сами тесты лежат на гитхабе, так что можно будет их дополнительно просмотреть и проверить корректность

```
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
3
-----
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-----
```

```
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-1
-----
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-1
-----
```

```
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-----
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
7
-----
```

```
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-----
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-1
-----
```

```

RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-----
RARS 1.6 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

Input path to file for reading: Input substring: Input path to the finite file: Would you like the program to additionally output the answer to the console? Y/N?
-----

```

Повторюсь, что нам важно то, что программа сгенерировала нужные нам файлы, а данный вывод, по сути, никакой смысловой нагрузки не несет, кроме того, что все 10 тестов были успешно запущены

5 Содержимое некоторых тестовых файлов и ответов, которые сгенерировала на них написанная ассемблерная программа


Для примера рассмотрим 4 файла: 1, 5, 7, 10. Ниже будут скриншоты исходной строки (из файла, откуда читает программа) и скриншоты ответа (из файла, который создает программа и куда записывает посчитанное значение). Для тестов 1, 5, 7, 10 вводились соответствующие подстроки: ple, 789, blue, federer



file1.txt – Блокнот


Файл Правка Формат Вид Справка

sample data

 ans1.txt – Блокнот


Файл Правка Формат Вид Справка

3

 file5.txt – Блокнот


Файл Правка Формат Вид Справка

123456789

 ans5.txt – Блокнот

Файл Правка Формат Вид Справка

6

 file7.txt – Блокнот

Файл Правка Формат Вид С

3b1ue1brown



ans7.txt – Блокнот

Файл Правка Формат Вид Справка

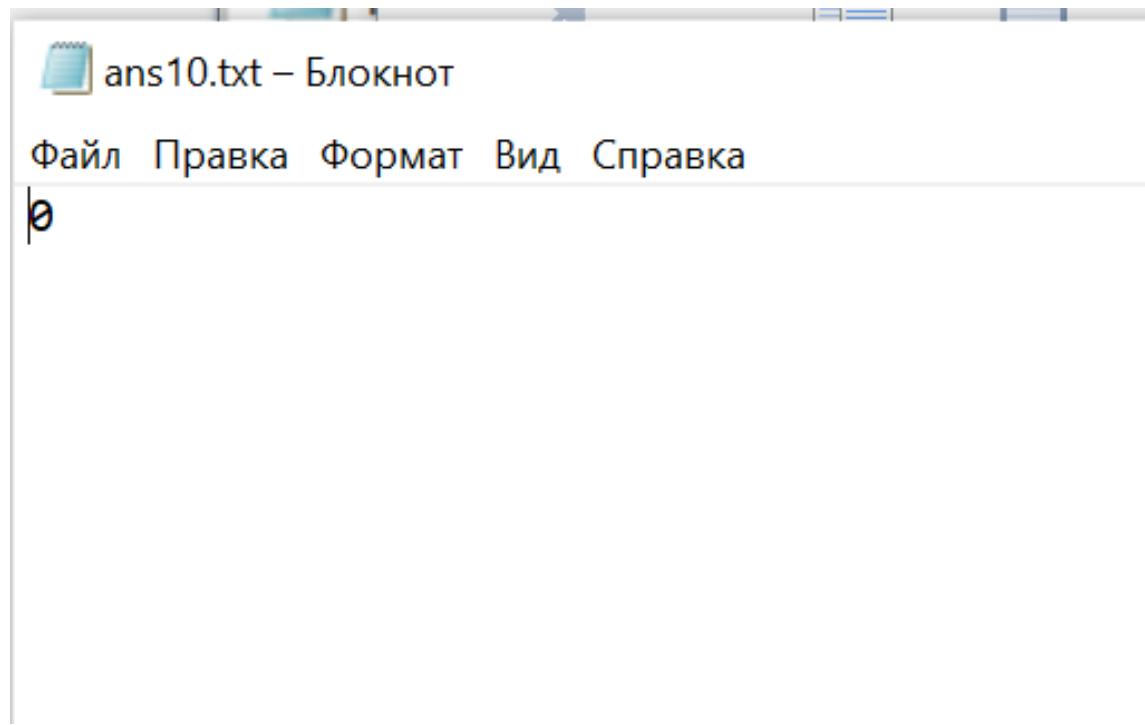
1



file10.txt – Блокнот

Файл Правка Формат Вид Справка

federer



Учитывая то, что строки нумеруются с нуля, программа выдает корректный ответ

6 Дополнительная информация, подтверждающая выполнение задания в соответствии требованиям на предполагаемую оценку

1. Все комментарии, а также текстовые сообщения, выводимые пользователю в качестве подсказок, написаны на английском языке во избежание конфликтов с кодировками, а также для соответствия предоставленным требованиям (кроме, может быть, комментариев в макросах ввода и вывода файла)
2. В каждом макросе, где уместно, используется сохранение локальных переменных (целочисленных) на стек с использованием макросов **push** и **pop** из макробιβлиотеки с семинаров
3. Так как требование реализовать макросы для ввода и вывода (и не только их) противоречит требованию реализовать подпрограммы для тех же нужд, то я выбрал реализацию через макросы, так как это требование на более высокую оценку
4. Перед каждым макросом написан комментарий, поясняющий его функциональность. Неочевидные шаги в ходе реализации макросов я также помечал комментариями. В основной программе я не оставлял комментарии около вызова макросов, так как по названию вполне понятно, за что отвечает каждый из них

5. Учитывая замечание от самих авторов RARS относительно их макроассемблера ("Макросредства ассемблера не входят ни в какой стандарт и остаются на усмотрение авторов ассемблера"), я не придерживался каких-то строгих конвенций при реализации макросов. Единственное, чтобы не запутаться в ходе реализации **собственных** макросов, передаю я регистр, непосредственно значение, метку или название другого макроса, я использовал следующий подход:

- если передаётся регистр, то я сопровождал его постфиксом `_reg`
- если передаётся непосредственное значение, то я сопровождал его постфиксом `_imm`
- если передаётся метка, то я сопровождал её постфиксом `_label`

Такой подход упрощает чтение сигнатуры макроса и защищает от ошибок в реализации, при этом отмечу, что код некоторых макросов взят с семинарских занятий, поэтому там не гарантируется, что будет соблюдаться такой "кодстайл"

6. Реализованные макросы поддерживают многократное использование с различными наборами исходных данных, включая возможность подключения различных исходных данных (имеется в виду подключение из тестовых файлов). Каждый макрос не привязан к конкретному регистру или метке, что обеспечивает унификацию, то есть возможность многократного использования одних и тех же конструкций. Так, макрос с выводом строки, переданной по непосредственному значению, используется дважды, а все остальные макросы хоть и используются единожды, но они являются унифицированными модулями в силу своей реализации, что и подтверждает выполнение одного из критерия на 10 баллов (разбиение программы на **унифицированные модули**)
7. Автоматизированное тестирование программы реализовано через код на C++ (файл **main.cpp**), а тестовые данные поступают из файла **tests.txt**, что позволяет осуществить прогон программы, осуществляющей решение поставленной задачи для различных тестовых данных (вместо их ввода). Данные тесты составляют полный набор тестового покрытия, и программа выдаёт на них корректный ответ, что позволяет утвердить о корректности реализации ассемблерной программы
8. В тестовой программе **main.cpp** при запуске на локальной машине нужно поменять пути к файлам в соответствии с тем, как они расположены на вашем компьютере
9. Так как в RARS записывать в файл (впрочем, как и считывать из него) можно только строки, то, учитывая то, что ответом на каждый тест является число, я реализовал макрос `int_to_string`, который конвертирует переданное по регистру число в строку и сохраняет её по переданной метке. Далее уже строка, сохраненная по этой метке выводится в файл и в консоль (если пользователь ответ Y)

10. Макрос чтения текста реализован в соответствии с одним из требований на 6-7 баллов ("Для чтения текста из файла реализовать буфер ограниченного размера, равного 512 байтам. При этом программа должна читать файлы размером до 10 килобайт")
11. Строки подразумеваются нуль-индексированными
12. Все макросы лежат в отдельной макробιβлиотеке
13. В программе реализован функционал дополнительного вывода в консоль с помощью ответа пользователя Y или N
14. Галочка в настройках ("Popup dialogue for input syscalls (5, 6, 7, 8, 12)") делает ввод информации в консоль через специальное диалоговое окно, что подтверждает выполнение соответствующего критерия на 10 баллов ("Использовать дополнительные графические диалоговые окна для ввода и отображения диалогов, предоставляемые симулятором RARS")