

Архитектура вычислительных систем.

ИДЗ 2. Отчёт.

Вариант 22.

Работа на 10 баллов.

Глебов Павел. БПИ228.

26 ноября 2023 г.

1 Условие задачи

Разработать программу, вычисляющую число вхождений различных знаков препинания в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

2 Тесты, которые проходит программа

Программа проходит 4 теста, покрывающие все возможные случаи. Содержание тестов:

- 1.
2. "!Hello.Words"
3. ":D ;) !?"
4. "OneDriveCloudMail"

3 Алгоритм решения

Моя программа делает два массива - со знаками препинания и булевскими значениями. Первый массив нам нужен для проверки символа на знак препинания, а второй для записи, если знак препинания встретился. После считаем не нулевые элементы второго массива и получаем количество уникальных знаков препинания строки.

4 Дополнительная информация по коду ассемблера и условию

1. В большинстве макросов используется сохранение на стек с помощью `push` и `pop`. Также я добавлял сохранение регистров на стек как локальные переменные, не смотря на то, что нам хватает регистров.
2. Требование реализовать макросы ввода и вывода массива(и не только их) противоречит требованию реализовать подпрограммы для тех же нужд, поэтому я выбрал реализацию макросов, поскольку это условие на более высокую оценку.
3. Реализованные макросы поддерживают многократное использование программы для тестирования на различных массивах через вывод в консоль запроса Y/N. Каждый макрос не привязан к конкретному регистру или метке, что обеспечивает унификацию, то есть позволяет использовать макросы многократно. Так, макросы с чтением и записью используются в программе многократно, что и подтверждает разбиение программы на унифицированные модули (критерий на 10 баллов).
4. Автоматизированное тестирование программы реализовано через программу на языке C++, а тестовые данные поступают из файлов, что соответствует выполнению критерия, где данные не должны вводиться вручную. Данные тесты покрывают все возможные исходы, и программа выдает на них верный ответ, что доказывает корректную работоспособность программы.
5. Все участки кода по возможности прокомментированы, чтобы было легко проследить логику решения. Также добавлены комментарии перед макросами, которые поясняют работу определенного макроса.
6. Макрос чтения текста реализован в соответствии с требованием через буфер ограниченного размера, равного 512 байтам, и чтением файла до 10 килобайт.
7. Так как в файл можно записывать только строки, был реализован макрос для преобразования intового значения в строковой тип.

5 Результаты тестирующих файлов.

Для тестирования программы на ассемблере я использую разработанную программу на C++ (`testCheck.cpp`), запускающую ассемблерный код из файла `main.asm` для данных из тестовых файлов.

Результаты тестирования:

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

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

-----
```