

ACS. Individual Homework 3

ACS

Александр Васюков | БПИ235 (239)

Задание

ASCII-строка – строка, содержащая символы таблицы кодировки ASCII. (https://ru.wikipedia.org/wiki/ASCII). Размер строки может быть достаточно большим, чтобы вмещать многостраничные тексты, например, главы из книг, если задача связана с использованием файлов или строк, порождаемых генератором случайных чисел. Тексты при этом могут не нести смыслового содержания. Для обработки в программе предлагается использовать данные, содержащие символы только из первой половины таблицы (коды в диапазоне 0–12710), что связано с использованием кодировки UTF-8 в ОС Linux в качестве основной. Символы, содержащие коды выше 12710, должны отсутствовать во входных данных кроме оговоренных специально случаев. Разработанная программа должна читать обрабатываемый текст из файла и загружать полученные результаты также в файл. Ввод имен входного и выходного файлов должен осуществляться с использованием консоли. Аналогичным образом осуществляет ввод остальных параметров, необходимых для выполнения программы.

Вариант 14

Разработать программу, вычисляющую отдельно число прописных и строчных букв в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

Решение претендует на 10 баллов

Файлы хранятся на GitHub по ссылке: https://github.com/vasyukov1/HSE-FCS-SE-2-year/tree/main/ACS/Homework/IHW_3 Есть 6 файлов:

- 1. `main.asm` - взаимодействие с пользователем. Предлагается выбор автотестов или ввод пути к файлу.
- 2. `program.asm` - программа, которая читает файл, считает количество строчных и заглавных букв, выводит результат.
- 3. `file_work.asm` - работа с файлом. Содержит подпрограммы, которые:
 - 1. Читает название файла
 - 2. Читает текст в файле
 - 3. Записывает результат в файл
 - 4. Предлагает записать результат в файл
- 4. `worker.asm` - решение задачи. Содержит функции, которые:
 - 1. Определяет тип буквы (заглавная/строчная)
 - 2. Считает количество заглавных и строчных букв
 - 3. Составляет текст для ответа о количестве букв
- 5. `macrolib.m` - библиотека макросов. Содержит:
 - 1. Печать целого числа
 - 2. Печать строки
 - 3. Печать символа
 - 4. Печать новой строки (\n)
 - 5. Чтение числа из консоли
 - 6. Запись строки в буфер
 - 7. Открытие файла
 - 8. Чтение файла
 - 9. Чтение дескриптора файла в буфер
 - 10. Закрытие файла
 - 11. Выделение памяти
 - 12. Завершение программы
 - 13. Сохранение регистра на стеке
 - 14. Возвращение регистра из стека
- 6. `autotests.asm` - программа автоматических тестов. Проверка программы на файлах:
 - 1. `1984.txt` - книга "1984" Дж. Оруэлл
 - 2. `Never Gonna Give You Up.txt` - песня "Never Gonna Give You Up" Рика Эстли
 - 3. `Holy Bible.txt` - книга "Библия"

Также есть директория `texts`, содержащая тексты для тестов.

Пример работы программы

```
Enter 'Y' for start autotest, or another key to open your file ==> Y

Do you want to print output into console? (Y/N) ==> Y
number of lowercase letters: 7882
number of uppercase letters: 0274

Do you want to print output into console? (Y/N) ==> Y
number of lowercase letters: 1386
number of uppercase letters: 0065

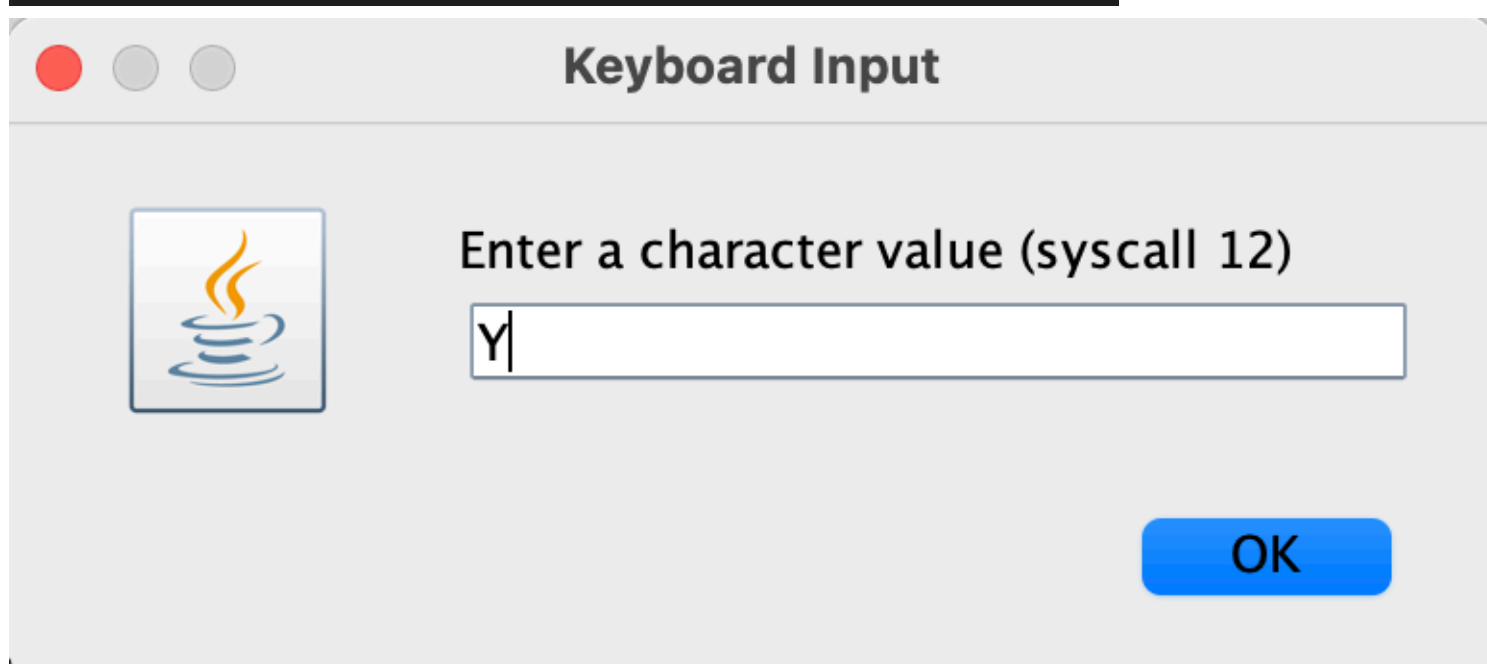
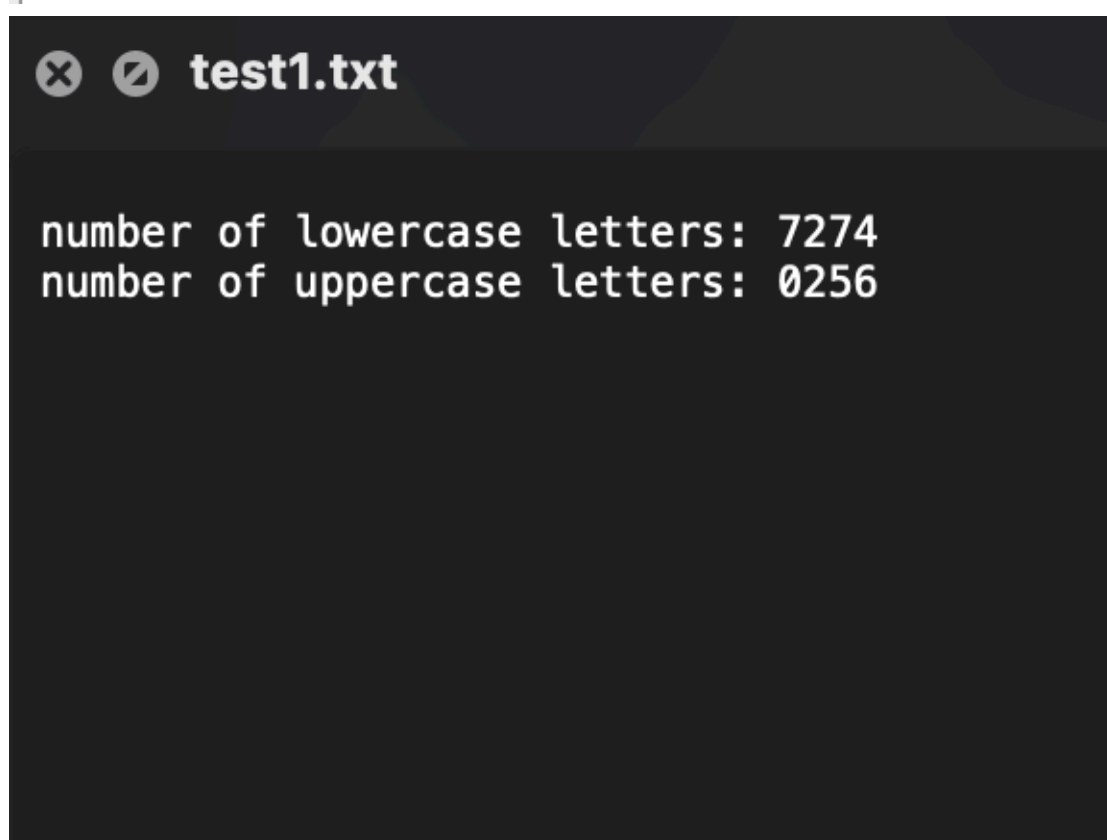
Do you want to print output into console? (Y/N) ==> N
Input path to file for writing: /Users/alexvasyukov/Desktop/1/texts/test1.txt

-- program is finished running (0) --

Enter 'Y' for start autotest, or another key to open your file ==> N
Input path to file for reading: /Users/alexvasyukov/Desktop/1/texts/1984.txt

Do you want to print output into console? (Y/N) ==> Y
number of lowercase letters: 7882
number of uppercase letters: 0274

-- program is finished running (0) --
```



Выполненные условия:

На 4-5

- 1. Приведено решение программы на ассемблере. Программа из файла читает данные. Результаты записываются в другой файл.

2. Все изменяемые параметры программы вводятся с консоли.
3. В программе присутствуют комментарии, поясняющие выполняемые ей действия.
4. Обработка данных, полученных из файла сформирована в виде отдельной подпрограммы.
5. В подкаталоге данных присутствуют файлы, используемые для тестирования.
6. При чтении файла размером, превышающим размер буфера, не происходит падения программы. Программа корректно обрабатывает введенный «урезанный» текст.
7. Сформирован отчет с результатами тестовых прогонов.

На 6-7

1. Внутри функций используются регистровые и локальные переменные.
2. Для чтения текста из файла реализован буфер ограниченного размера, равного 512 байтам. Программа читает файлы размером до 10 килобайт.
3. Реализован ввод исходных данных, их обработку, вывод результатов через соответствующие подпрограммы. Подпрограммы получают необходимые им данные через параметры в соответствии с принятым соглашением о передаче параметров.
4. Возвращаемые из подпрограмм значения возвращаются через параметры в соответствии с общепринятыми соглашениями.
5. Расширен отчет.

На 8

1. Добавлена в программу возможность дополнительного вывода результатов на консоль.
2. Реализована дополнительная тестовая программа, которая осуществляет многократный вызов процедур, обеспечивающих ввод файлов, их обработку и вывод для различных исходных данных, расположенных в каталоге с исходными тестовыми данными.
3. Расширен отчет.

На 9

1. Добавлено в программу использование макросов для реализации ввода, вывода, и обработки данных. Макросы поддерживают повторное использование с различными массивами и другими параметрами. Внутри макросов расположены вызовы соответствующих подпрограмм.
2. Реализована дополнительная тестовая программа, которая вызывает выполняемые подпрограммы через макросы, реализуя ту же функциональность, что и предыдущая тестовая программа.
3. Расширен отчет.

На 10

1. Программа разбита на несколько единиц компиляции.
2. Макросы выделены в отдельную автономную библиотеку.
3. Используются дополнительные графические диалоговые окна для ввода и отображения диалогов, предоставляемые симулятором RARS.
4. Расширен отчет.

Show Labels Window (symbol table)

Program arguments provided to program

✓Popup dialog for input syscalls (5,6,7,8,12)

✓Addresses displayed in hexadecimal

✓Values displayed in hexadecimal

Assemble file upon opening

Assemble all files in directory

Assemble all files currently open

Assembler warnings are considered errors

✓Initialize Program Counter to global 'main' if defined

✓Derive current working directory

✓Permit extended (pseudo) instructions and formats

✓Self-modifying code

64 bit

Editor...

Highlighting...

Exception Handler...

Memory Configuration...