

Отчёт ИДЗ-3

Автор: Лев Светличный

Группа: БПИ 225

Вариант 25

Условие

Разработать программу, которая определяет в ASCII-строке частоту встречаемости различных идентификаторов, являющихся словами, состоящими из букв и цифр, начинающихся с буквы. Разделителями являются все другие символы. Для тестирования можно использовать программы, написанные на различных языках программирования. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

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.

ASCII-строка — строка, содержащая символы таблицы кодировки 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.

Используемые опции компиляции

- Используется опция “popul dialog for syscalls 5, 6, 7, 8, 12” (но даже если эта опция отключена, ввод осуществляется корректно)

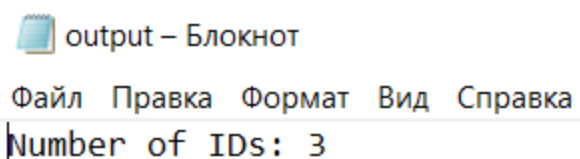
- Разрешены расширенные псевдоинструкции
- Для корректного запуска требуется, чтобы папка AoCS-IHW-3 Var 25 находилась в одном пространстве с исполняемым файлом RARS

Результаты тестовых прогонов

Для осуществления тестов используется три файла:

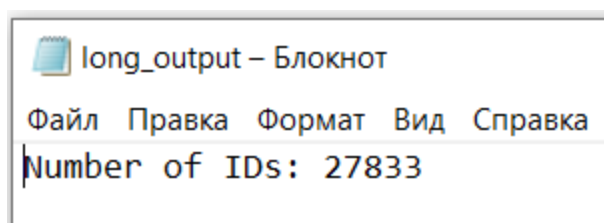
- input.txt - содержит текст "hello world 123" - 3 ID.

Результат теста (в output.txt):



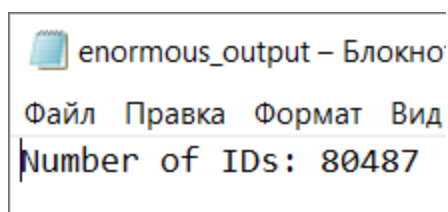
- long_input.txt - содержит текст, взятый с сайта [The Longest Text Ever](#); содержит 27833 ID. Весит 147 Кб

Результат теста (в long_output.txt):



- enormous_input.txt - содержит текст книги Harry Potter and the Sorcerer's Stone. Содержит 80487 ID. Весит 430 Кб

Результат теста (в enormous_output.txt):

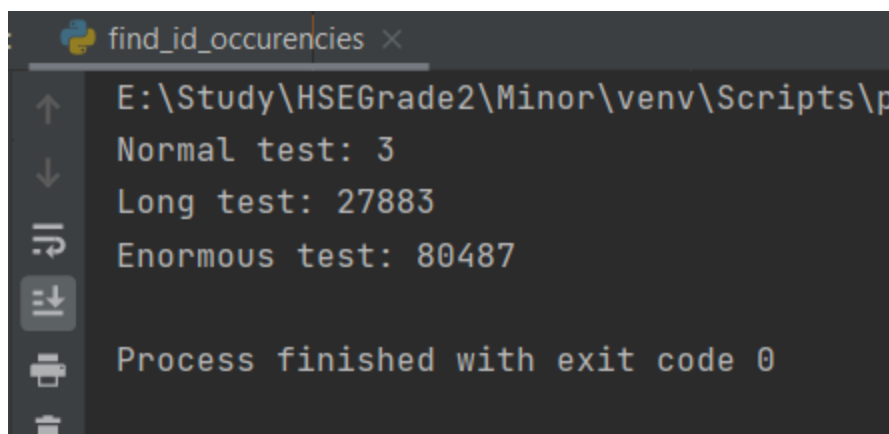


Общий результат:

```
Welcome to IHW-3's program: "Symbol string processing"!
Test on short data: All done
Test on long data: All done
Test on enormous data: All done

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

Также была создана аналогичная программа на языке программирования Python



```
find_id_occurencies x
E:\Study\HSEGrade2\Minor\venv\Scripts\p
Normal test: 3
Long test: 27883
Enormous test: 80487

Process finished with exit code 0
```

Соответствие требованиям

4-5 баллов

- Приведено решение программы на ассемблере. Программа из файла читает данные. Результаты записываются в другой файл.
- Все изменяемые параметры программы вводятся с консоли (в manual-режиме)
- В программе присутствуют комментарии, поясняющие выполняемые ей действия.
- Обработка данных, полученных из файла сформирована в виде отдельной подпрограммы find_id_frequencies.s
- В подкаталоге данных присутствуют файлы, используемые для тестирования (папки test_input и test_output).
- ~~Буфер для текста программы имеет фиксированный размер размером не менее 4096 байт, допускающий ввод без искажений только тексты,~~

~~ограниченные этим размером.~~ Это условие не выполнено, т.к. выполнено более высокоуровневое условие - буфер имеет размер 512 байт, и файл считывает из файла последовательно и без искажений.

- При чтении файла размером, превышающим размер буфера, не должно происходить падения программы. Программа должна корректно обработать введенный «урезанный» текст. (Программа прекрасно работает с буфером, как было сказано выше).
- Сформирован отчет с результатами тестовых прогонов и описанием используемых опций компиляции, проведенных модификаций ассемблерной программы.

6-7 баллов

- Внутри функций используются регистровые переменные.
- Для чтения текста из файла реализован буфер ограниченного размера, равного 512 байтам. При этом программа легко читает файлы размером до 10 килобайт.
- Реализован ввод исходных данных, их обработка, вывод результатов через соответствующие подпрограммы в MacroLib. Подпрограммы получают необходимые им данные через параметры в соответствии с принятым соглашением о передаче параметров (через %arg, ибо так легче).
- Возвращаемые из подпрограмм значения должны возвращаться через параметры в соответствии с общепринятыми соглашениями (через регистры а).

8 баллов

- В программу добавлена возможность дополнительного вывода результатов на консоль. Выводить или нет решает пользователь отвечая «Y» или «N» на соответствующий вопрос компьютерной программы. Данная возможность полезна при быстром отображении коротких данных. Вывод программы при этом должен полностью соответствовать выводу результатов в файл.
- 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.

9 баллов

- В программу добавлено 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.
- Реализована дополнительная тестовая программа, выполненная на языке Python для сверки корректности работы.

10 баллов

- Программа разбита на несколько единиц компиляции. При этом подпрограммы ввода–вывода составляют унифицированные модули, используемые повторно как в программе, осуществляющей ввод исходных данных, так и в программе, осуществляющей тестовое покрытие.
- Макросы выделены в отдельную автономную библиотеку MacroLib
- Отчет содержит полное описание программы