Практическое задание №1

Самостоятельно выберите вариант задания, выполните его и сдайте в машзале. Можно сделать несколько вариантов задания.

Вариант 1. В этом варианте нужно написать две программы.

Программа 1 читает последовательность целых чисел со стандартного ввода и формирует из них односвязный список. Затем она сортирует этот список по неубыванию и печатает на стандартный вывод полученную последовательность.

Программа 2 получает имя текстового файла в командной строке. Программа должна реверсировать содержимое каждой строки в файле. Программе запрещается считывать файл в память целиком. Можно предполагать, что каждая строка – не длиннее 80 символов и ее можно записать в память.

Вариант 2. Сортировка абзацев - 1. Программа получает имя текстового файла в командной строке. Файл разделен пустыми строками на абзацы. Программа сортирует строки в файле внутри каждого абзаца в лексикографическом порядке, не меняя порядка абзацев. Размеры строк и количество абзацев могут быть произвольными. Программа может размещать в памяти содержимое одного (каждого) абзаца, но не более одного.

Вариант 3. Сортировка абзацев - 2. Смотри предыдущий вариант. Программа не может размещать в памяти содержимое абзаца. Вместо этого программа должна хранить каждый абзац в памяти в виде списка элементов. Каждый элемент по порядку соответствует своей строке абзаца. Если строка «короткая», элемент хранит строку в памяти целиком. Иначе элемент – это пара чисел: смещение от начала файла и длина строки – строка целиком может не уместиться в памяти. Строка называется «короткой», если ее длина меньше некоторой константы. Константа задается в тексте программы.

Вариант 4. Производная. Программа получает на стандартный ввод последовательность выражений. Для каждого выражения она печатает на стандартный вывод его производную или сообщение об ошибке (например, из-за ошибки в синтаксисе выражения). Язык

выражений придумайте самостоятельно. Он должен включать арифметику, тригонометрические функции, переменные, числа, скобки. Можно предполагать, что каждое выражение можно разместить в памяти. Должны поддерживаться аргументы командной строки, при помощи которых можно задать ввод и вывод с файлами вместо стандартных потоков. Предусмотрите аргумент командной строки, при помощи которого можно ознакомиться с языком выражений.

Вариант 5. Исследование менеджера памяти. Под «менеджером памяти» подразумевается библиотека, реализующая функции malloc, calloc, realloc, free. Экспериментальным путем определить, какие алгоритмы и структуры данных использует менеджер памяти на вашей платформе. Исходный код менеджера памяти вам недоступен. Обосновать корректность и полноту выбора экспериментов. Сделать выводы о наиболее эффективном использовании менеджера памяти на вашей платформе. Определить, как используется менеджер памяти в функциях работы с файлами из stdio.h. Проведенное исследование, его анализ и результаты оформить в виде отчёта по заданию.

Требования к программам (не обязательные, но желательные):

1. Следовать единому стилю оформления программы. Например, этому [1].
2. Потенциально переиспользуемые части программы оформлять в виде библиотек.
3. Автоматизировать действия над программой (в частности, сборку программы при помощи утилиты make, CMake и т.п.). Есть пособие [2].
4. Тестировать свою программу и измерять качество тестов, определяя покрытие кода и требований, даваемое тестами. Тем самым, научиться самостоятельно тщательно проверять свою программу. Инструменты gcov, lcov.
5. Проверять программу на наличие ошибок, используя различные анализаторы (valgrind, санитайзеры, фаззеры). Тем самым, научиться самостоятельно тщательно проверять свою программу. Про valgrind есть пособие [2].

[1] https://ejudge.ru/study/3sem/style.shtml

[2] http://jaffar.cs.msu.ru/mash/os/2019-2020/%cc%e5%f2%ee%e4%e8%f7%e5%f1%ea%e8%e5%20%ec%e0%f2%e5%f0%e8%e0%eb%fb/%c8%ed%f1%f2%f0%f3%ec%e5%ed%f2%e0%eb%fc%ed%fb%e5%20%f1%f0%e5%e4%f1%f2%e2%e0%20%f0%e0%e7%f0%e0%e1%ee%f2%ea%e8%20%cf%ce%20%e2%20%ce%d1%20UNIX.pdf