Министерство высшего образования и науки

Российской Федерации

Национальный исследовательский университет «МИЭТ»

Дисциплина: Операционные системы

Отчет по лабораторной работе №6

Выполнил:

Студент группы П-32

Селезнева В.В.

Проверил:

Фомин Р.А.

Москва, 2020

***Задание 1***. Запишите назначение основных опций компилятора gcc (-c, -S, -E, -o). Воспользуйтесь командой $man gcc или $info gcc

* **Опция –с**

Компилировать или ассемблировать исходные файлы, но не линковать (Линковка это процесс компоновки различных кусков кода и данных вместе, в результате чего получается один исполняемый файл). Стадия линковки просто не выполняется. Конечный вывод происходит в форме объектного файла для каждого исходного файла.

* **Опция -S**

Остановиться после собственно компиляции; не ассемблировать. Вывод производится в форме файла с ассемблерным кодом для каждого не ассемблерного входного файла.

* **Опция –Е**

Остановиться после стадии препроцессирования; не запускать собственно компилятор. Вывод делается в форме препроцессированного исходного кода, который посылается на стандартный вывод.

Входные файлы, которые не требуют препроцессирования игнорируются.

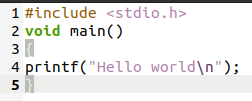
* **Опция -о файл**

Поместить вывод в файл 'файл'. Эта опция применяется вне зависимости от вида порождаемого файла, есть ли это выполнимый файл, объектный файл, ассемблерный файл или препроцессированный C код.

Поскольку указывается только один выходной файл, нет смысла использовать '-o' при компиляции более чем одного входного файла, если вы не порождаете на выходе выполнимый файл.

***Задание 2.***

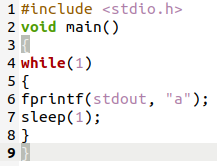
***2.1.*** Напишите и выполните программу Hello, выводящую строку "Hello, world".

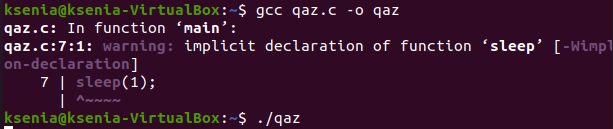




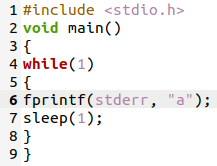
***2.2*** Определите главное отличие stdout от stderr.

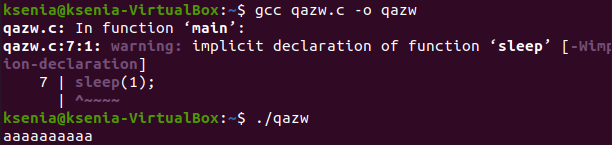
## ***Stdout*** - Стандартный поток вывода данных для программ. Например, когда мы пишем команду ls, то список папок и файлов она выводит именно в этот поток, который отображается у нас в консоли.



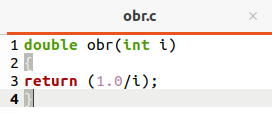
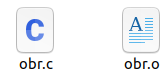


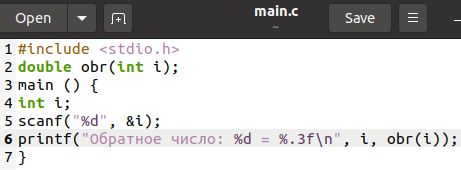
***Stderr*** - Поток вывода ошибок. Если программа не смогла сделать все как надо — она пишет именно в этот поток. Например, когда rm пытается удалить несуществующий файл.

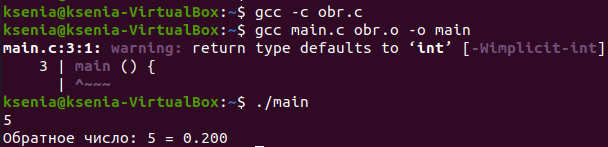




***2.3. (Необязательное задание)*** Напишите программу obr, состоящую из двух модулей: модуль obr.c содержит функцию double obr (int i), возвращающую число, обратное числу i; модуль main.c содержит функцию main(), которая запрашивает целое число i и выводит obr(i).

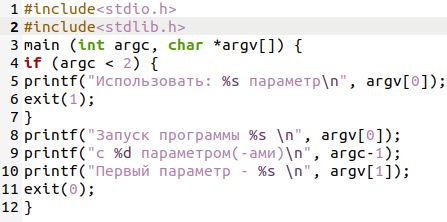


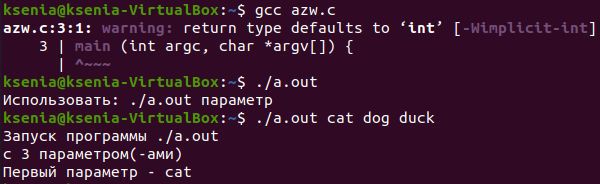




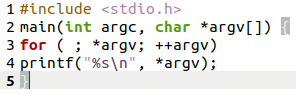
***Задание 4.*** Запишите в отчет текст примеров и результаты их тестирования с параметрами.

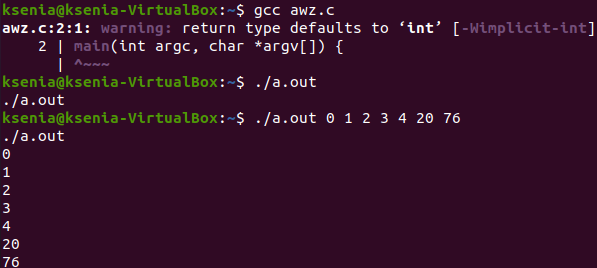
***4.1*** Печать параметров командной строки



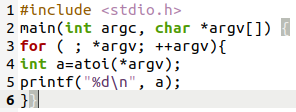


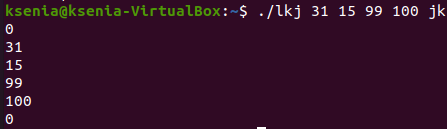
***4.2*** Печать произвольного числа параметров





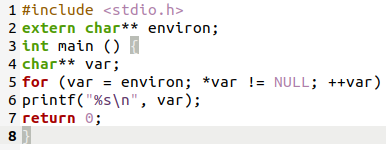
***4.3*** Измените предыдущий пример, задав параметры целого типа. Для преобразования строки в целое число используйте функцию **atoi**.

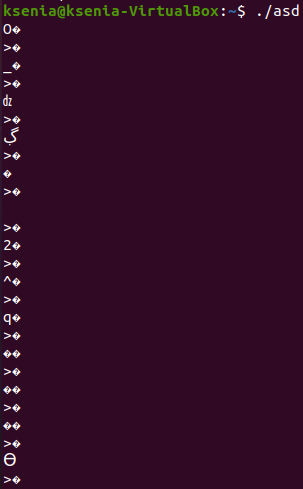




***Задание 5.*** Выполните пример, запишите в отчет его текст и результаты тестирования.

Печать переменных окружения

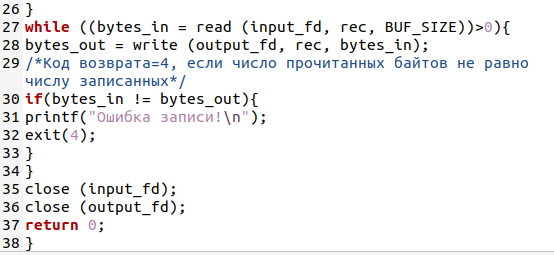




***Задание 6.***

***6.1.*** Модифицируйте приведенную ниже программу copyfile.c копирования файла, добавив в нее комментарии, проверку количества аргументов командной строки (при неверном количестве параметров установите код возврата=1) и обработку ошибок после системных вызовов open (коды возврата 2 и 3), write (код возврата 4).





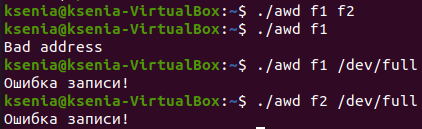
Файл 1



Файл 2



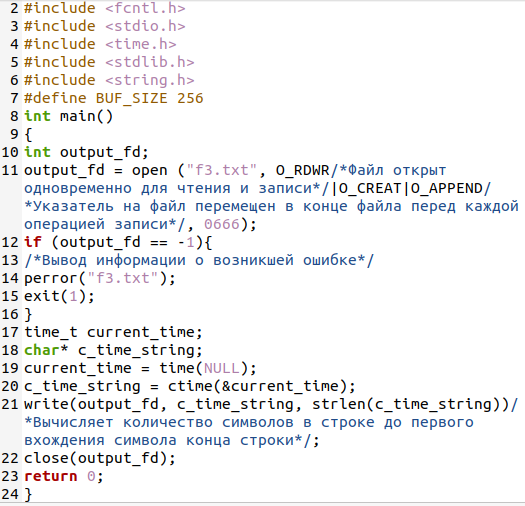
Различные выходы



Файл 2 после перезапуска после компиляции



6.2 Напишите и выполните программу, которая: а) при помощи функции open создает файл file1 с разрешением чтения/записи для всех пользователей; б) записывает в конец файла file1 текущую дату и время.





Созданный файл 3

