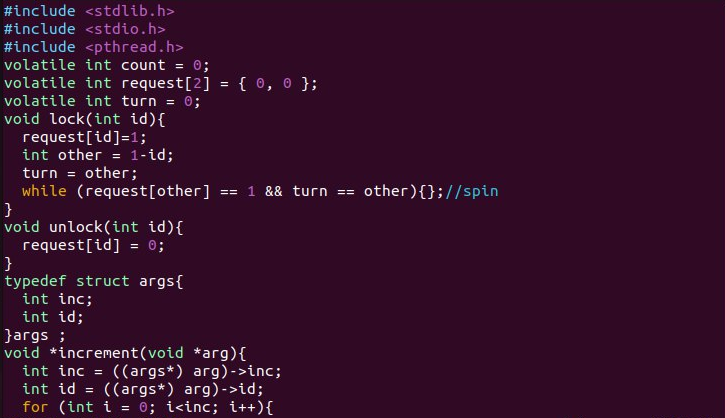
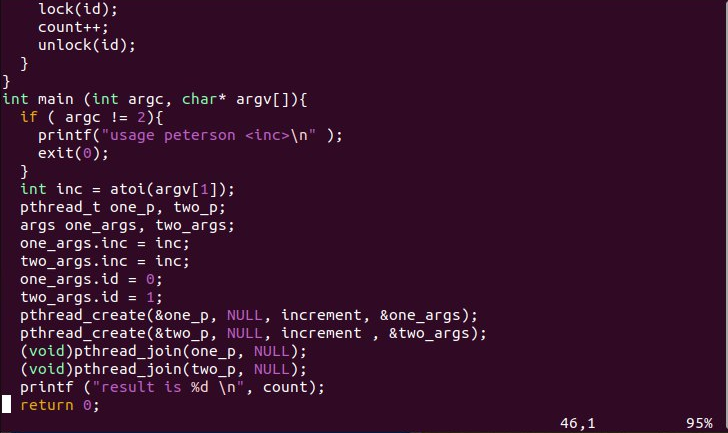
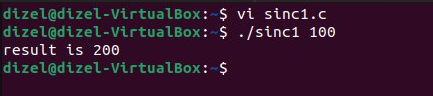
**Синев Денис Евгеньевич**

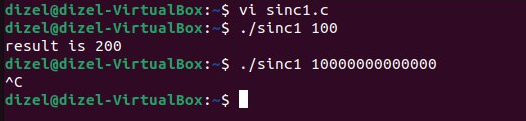
**Программа 1 из ПДФ**

****

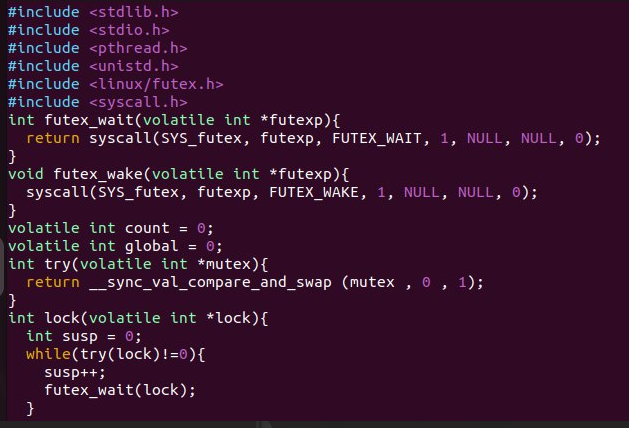
****

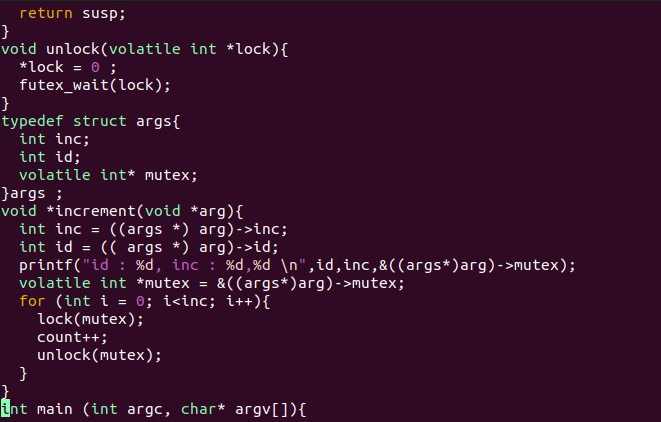
****

При запуске малых чисел программа отрабатывает мгновенно, а если вносить большие числа, то программа работает очень и очень долго.



**Программа 2 из ПДФ**

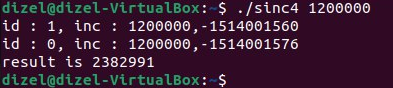




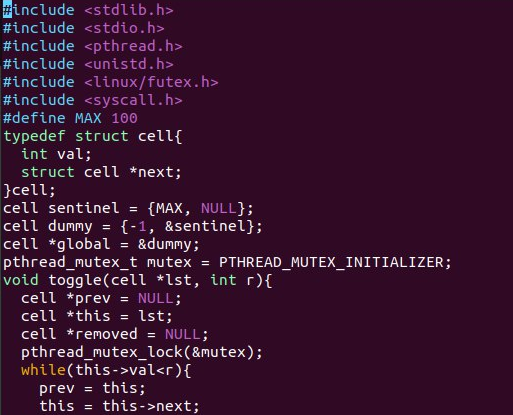


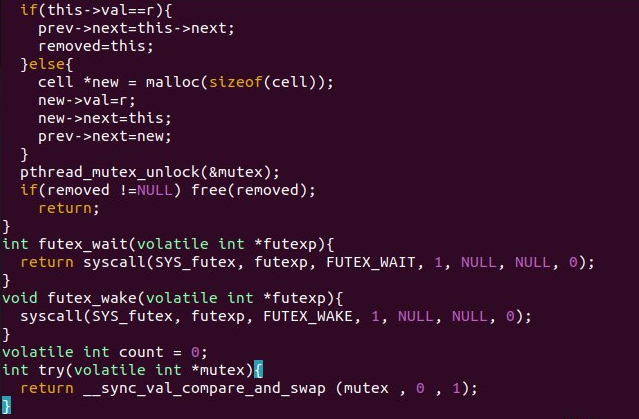
В данном примере функция increment увеличивает переменную count на значение inc с использованием mutex для обеспечения критической секции. Функции lock и unlock используются для захвата и освобождения mutex соответственно.

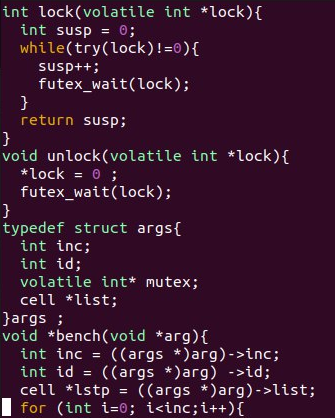
В функции main сначала проверяется правильность количества аргументов командной строки. Затем создаются две структуры args (one\_args и two\_args) с соответствующими значениями и указателем на общий мьютекс. Далее создаются два потока, каждому из которых передается соответствующая структура args. После завершения работы обоих потоков программа выводит результат увеличения переменной count.

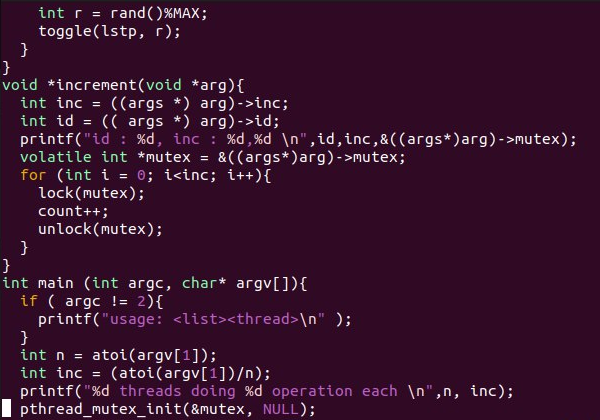
****

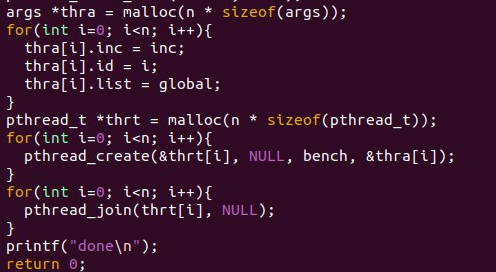
**Программа 3 из ПДФ**

****

****

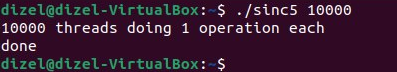
****

****

****

Эта программа демонстрирует работу с потоками в среде Linux, использование мьютексов для синхронизации доступа к общему ресурсу и применение примитивов синхронизации для эффективного управления доступом к критическим областям кода.

В функции main считывается количество потоков и операций из аргументов командной строки. Затем происходит инициализация мьютекса и создание массива структур args для каждого потока. Затем создаются и запускаются потоки с помощью функции pthread\_create.

****