МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Системное программирование»

Лабораторная работа №3

**«Технология разработки программ в среде Linux»**

по дисциплине

Операционные системы

Выполнил: студент гр. БЭИ2202

Кулешов А. С.

Вариант 16

Проверил: Королькова Т.В.

Москва, 2024 г

1. В текстовом редакторе (например, vim, вызов: vim hello.c)

напишите программу hello.c, выводящую на экран фразу “HELLO

Linux”. скомпилируйте полученную программу компилятором gcc.

Запустите полученный файл hello на выполнение.

2. Напишите скрипт, компилирующий и запускающий программу (имя

исходного файла и исполняемого файла результата задаётся пользователем в

качестве аргументов командной строки). В случае ошибок при компиляции

выведите на консоль сообщение об ошибках и не запускайте программу на

выполнение. Проверьте работу разработанного скрипта на программе

hello.c.

3. Используя функцию system(), напишите

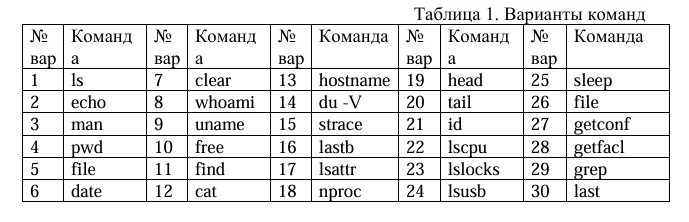
программу,

выполняющую команду в соответствии с вариантом задания из таблицы 1,

дополнив команду при необходимости параметрами. Продемонстрируйте

отладку программы с помощью gdb применив команды установки точек

останова, пошагового выполнения, просмотра значения переменной.



**ВЫПОЛНЕНИЕ**

Сделаю простую программу на ЯП C, после чего скомпилирую и запущу её при помощи gcc, затем сделаю bash скрипт для компиляции и запуска скрипта. Сделаю скрипт с запуском системной команды lastb(в соответствии с индивидуальным вариантом). Использую gdb для отладки полученной программы: создам точку остановы, выведу адрес функции system.

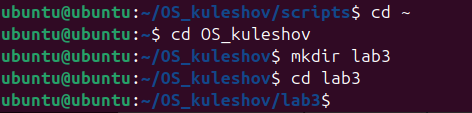
****

Рисунок 1 – Создание папки lab3



Рисунок 2 – Код hello.c

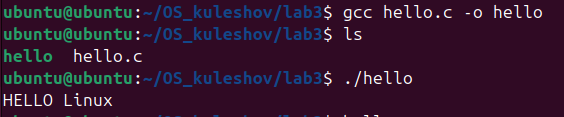


Рисунок 3 – Запуск hello linux

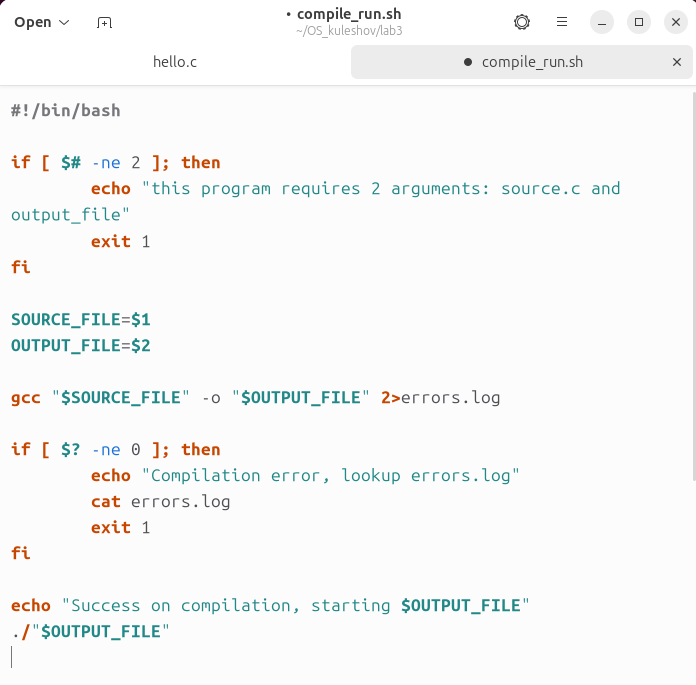


Рисунок 4 – создание скрипта для компиляции и запуска

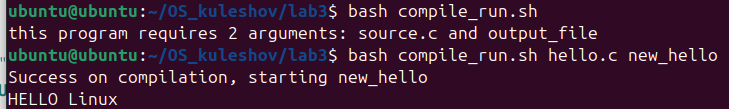


Рисунок 5 – Проверка скрипта (отсутствие аргументов и успех)

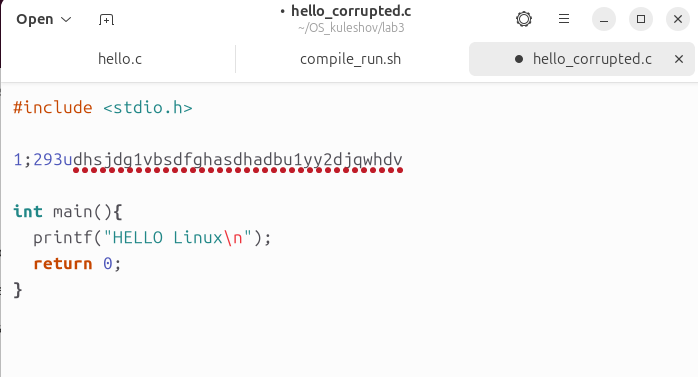


Рисунок 6 – Создание некорректного кода

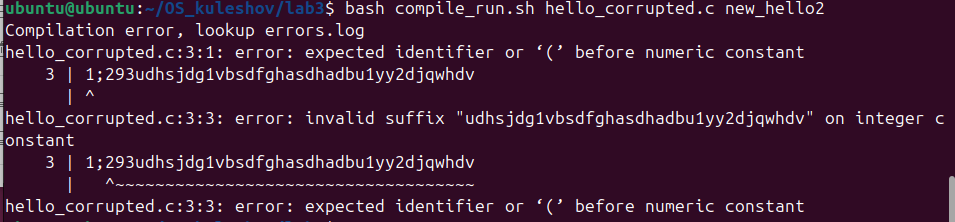


Рисунок 7 – Проверка скрипта (ошибка компиляции)

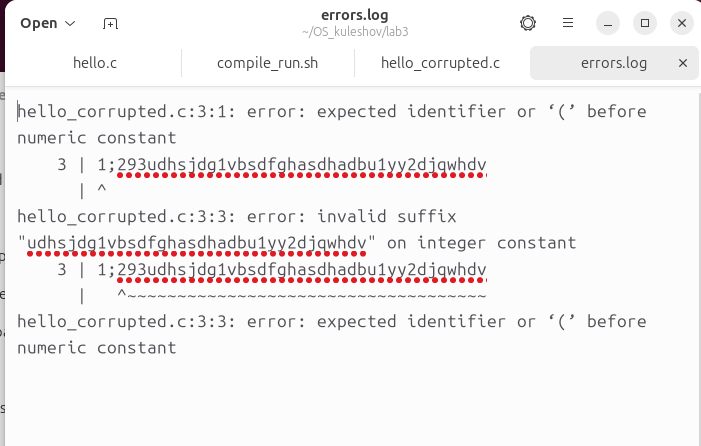


Рисунок 8 – Просмотр логов



Рисунок 9 – Создание кода в соответствии с вариантом

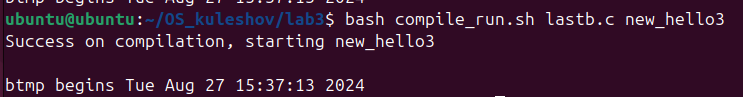


Рисунок 10 – Запуск кода



Рисунок 11 – Компиляция кода с отладкой

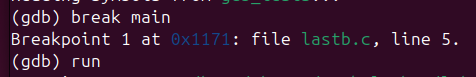


Рисунок 12 – Создание точки остановы

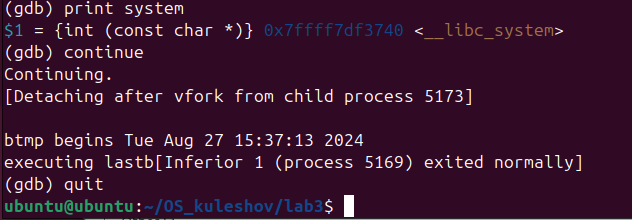


Рисунок 13 – Вывод адреса функции system, корректный запуск кода