Отчет по лабораторной работе №10

Операционные системы

Фадин В.В.

Содержание

# 1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 2 Выполнение лабораторной работы

## 2.1 Задача 1

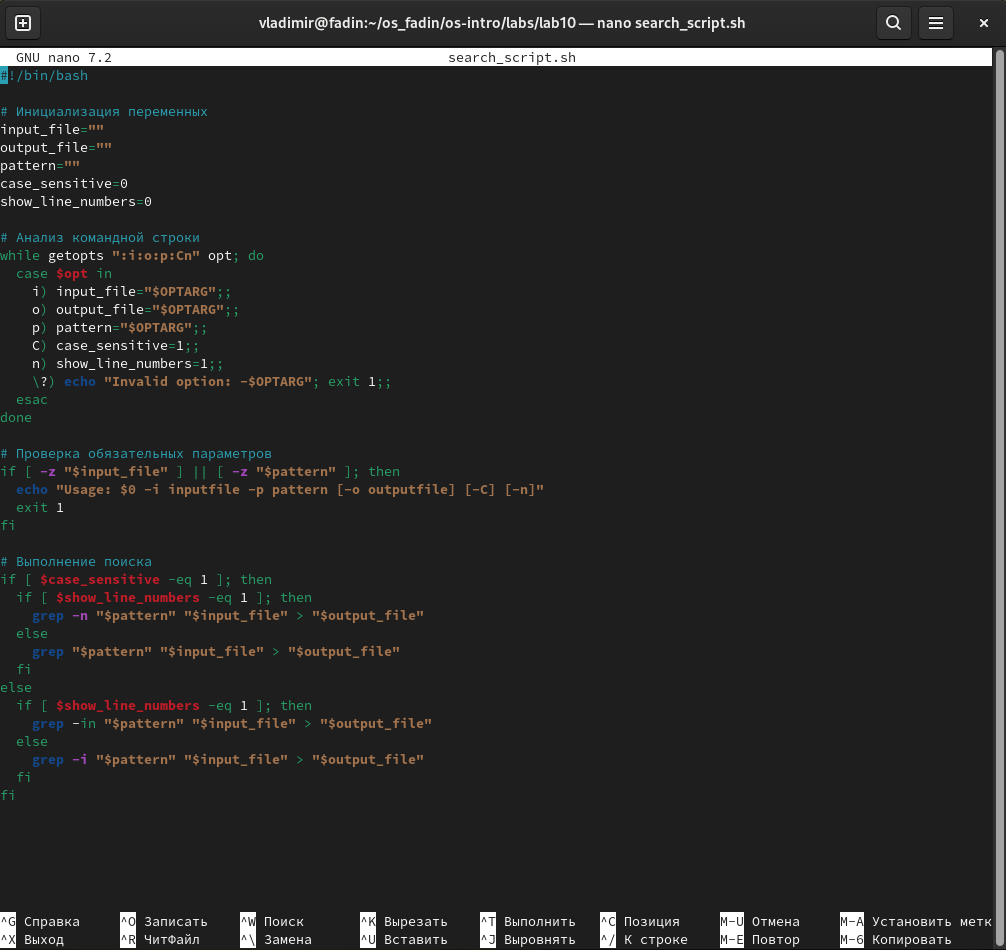


Рис. 1: Скрипт поиска

В этом скрипте мы используем getopts для анализа командной строки и извлечения значений параметров. Затем мы проверяем, были ли указаны обязательные параметры -i и -p. Если они указаны, мы выполняем поиск с помощью grep, учитывая параметры -C и -n. Результат поиска выводится в файл, указанный параметром -o, если он был указан.

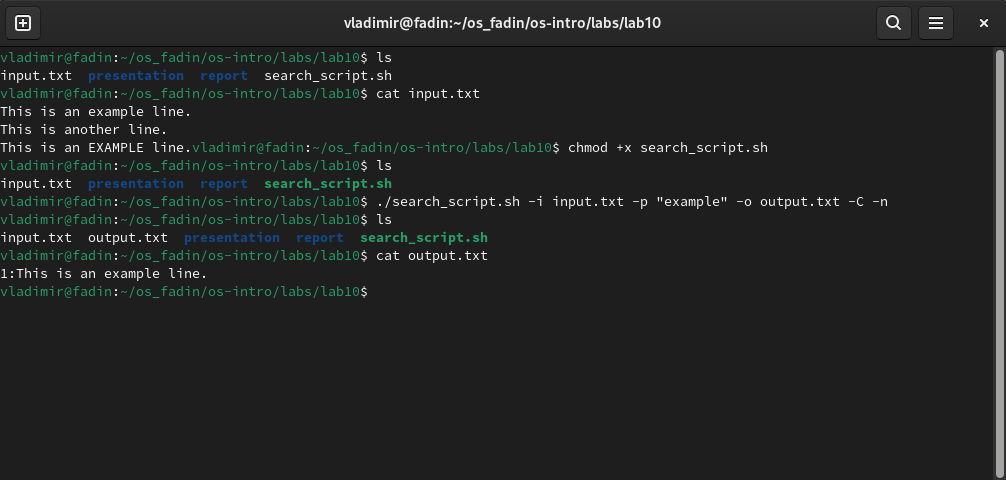


Рис. 2: Запуск скрипта поиска

## 2.2 Задача 2

Вот пример программы на языке Си, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю:

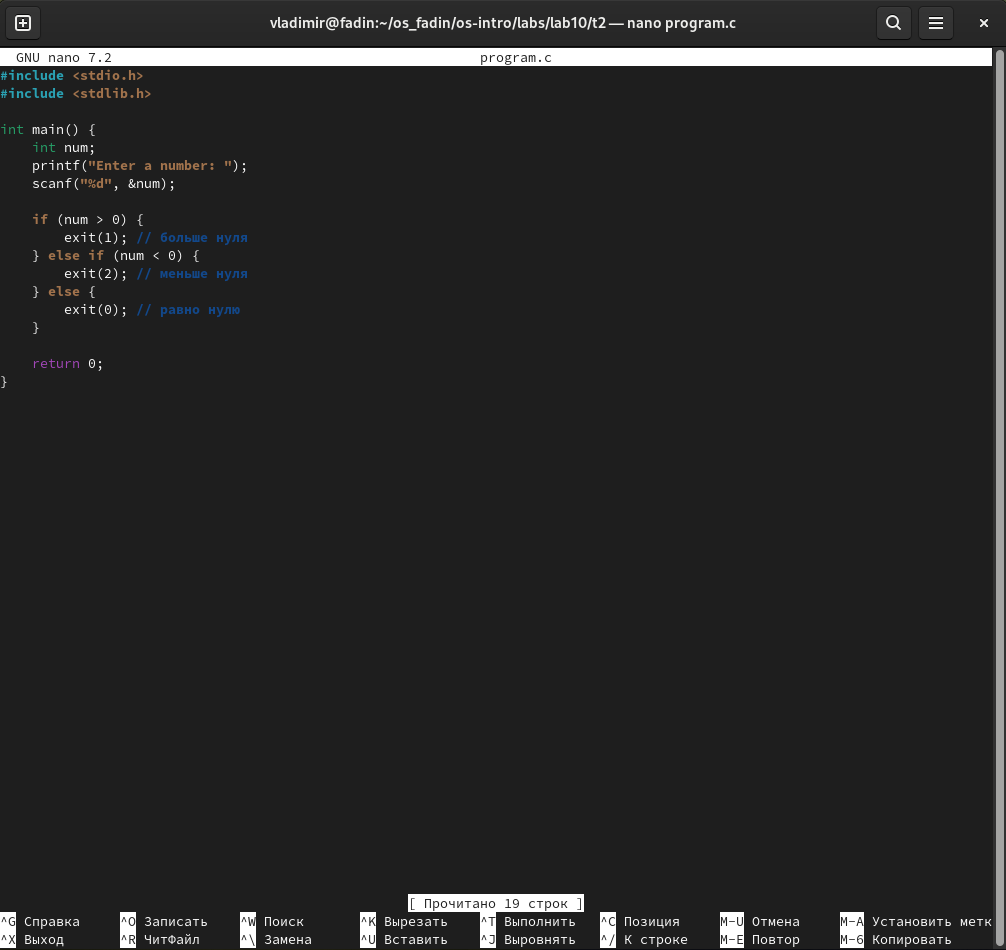


Рис. 3: Создание программы на языке Си

Далее компилируем программу:

gcc program.c -o program

Теперь создадим командный файл, который будет вызывать эту программу и анализировать код завершения:

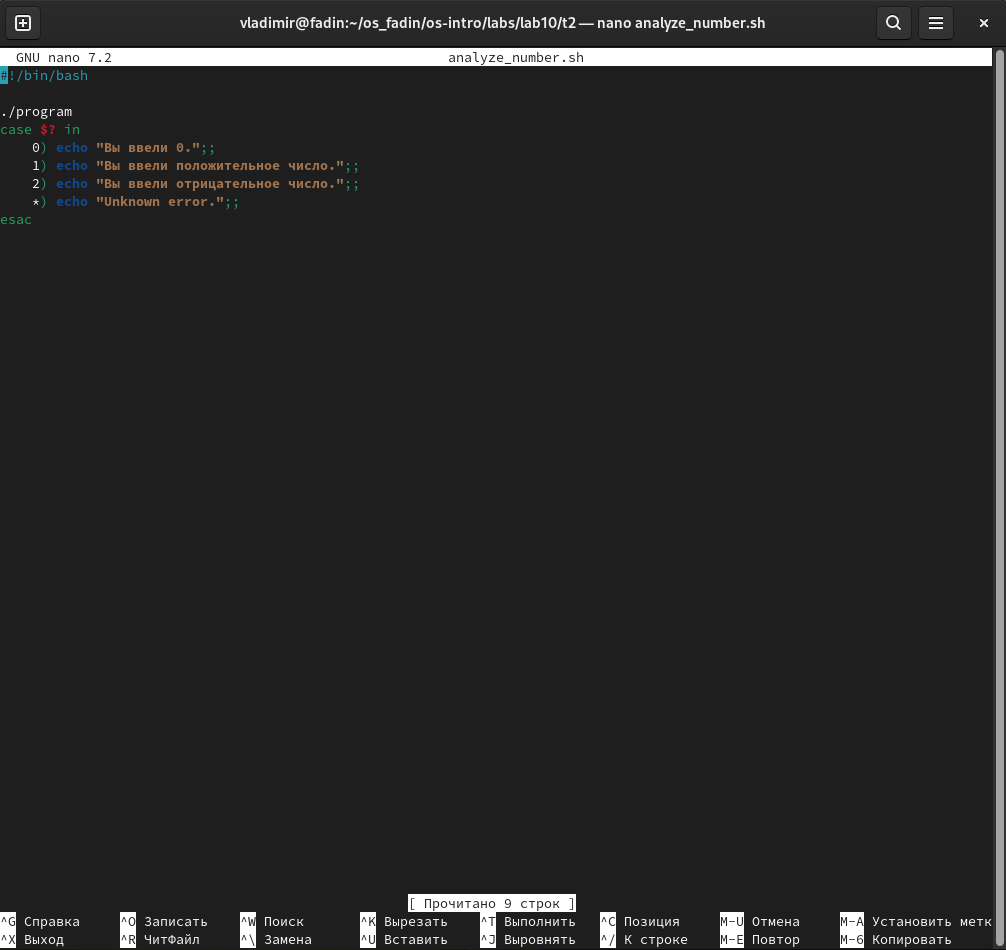


Рис. 4: Скрипт анализа кода

Сохраняем командный файл, например, как analyze\_number.sh, и делаем его исполняемым:

chmod +x analyze\_number.sh

Теперь мы можем запустить командный файл:

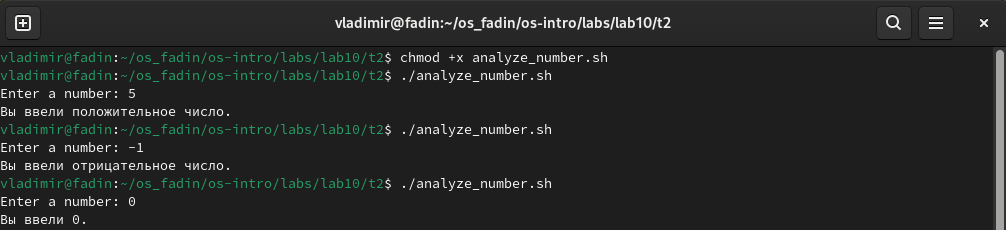


Рис. 5: Запуск анализа кода

Программа будет запрашивать ввод числа, и после ввода она будет завершаться с кодом, соответствующим типу введенного числа. Командный файл будет анализировать код завершения и выводить соответствующее сообщение.

## 2.3 Задача 3

Вот пример командного файла, который создает указанное число файлов и может удалять их:



Рис. 6: Скрипт создания и удаления файлов

Командный файл принимает один аргумент - количество файлов, которые нужно создать. Он использует цикл for с командой seq для создания файлов с именами от 1.tmp до <number\_of\_files>.tmp.

Если аргументом является строка "delete", то командный файл удаляет все созданные файлы, используя цикл for с командой rm -f.

Пример использования:

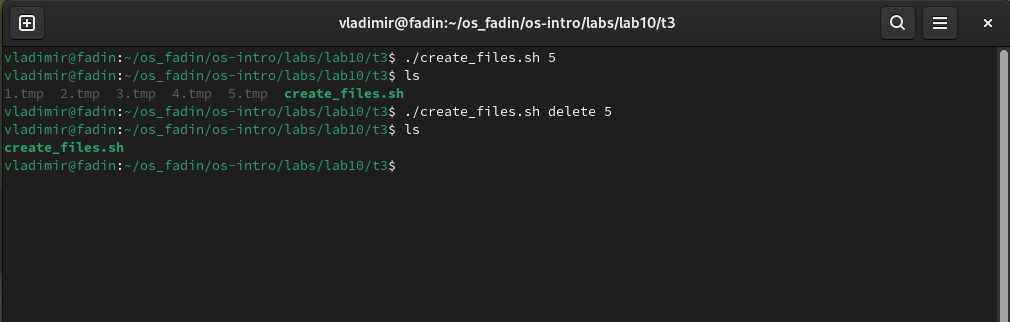


Рис. 7: Выполнение скрипта создания и удаления файлов

В первом примере командный файл создает 5 файлов с именами 1.tmp, 2.tmp, …, 5.tmp. Во втором примере он удаляет эти файлы, если они существуют.

## 2.4 Задача 4

Вот пример командного файла, который запаковывает в архив все файлы в указанной директории:

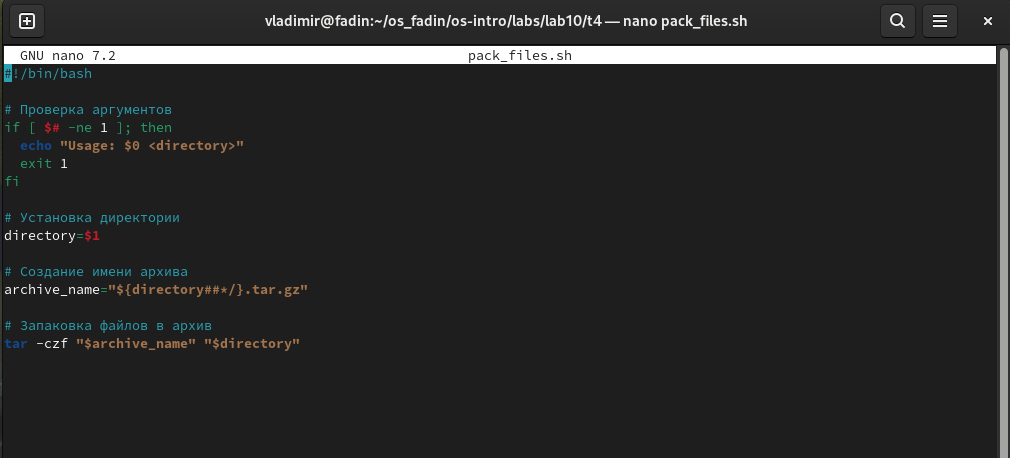


Рис. 8: Скрипт архиватора

Командный файл принимает один аргумент - путь к директории, которую нужно запаковать.

Чтобы модифицировать его, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад, мы можем использовать команду find:



Рис. 9: Модификация скрипта архиватора

В этом модифицированном командном файле мы используем команду find для поиска файлов в указанной директории, которые были изменены менее недели тому назад (-mtime -7). Затем мы используем опцию -exec для запуска команды tar с каждым найденным файлом, добавляя его в архив.

Пример использования:

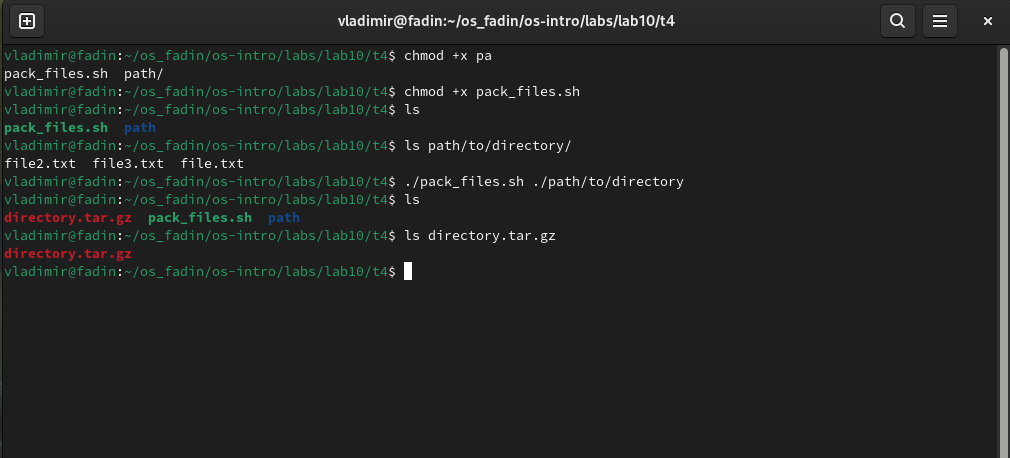


Рис. 10: Модификация скрипта архиватора

Командный файл создаст архив с именем directory.tar.gz в текущей директории, содержащий все файлы в указанной директории, которые были изменены менее недели тому назад.

# 3 Выводы

Изучили основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 4 Ответы на контрольные вопросы

1. Команда getopts предназначена для парсинга параметров командной строки в shell-скриптах. Она позволяет извлекать и обрабатывать опции, передаваемые в скрипт, и их аргументы.
2. Метасимволы (\*, ?, [, ], (, ), {, }, ~, .) используются в Linux для генерации имён файлов, известной как globbing. Они позволяют указать шаблоны для поиска файлов, что может быть полезно при выполнении различных операций с файлами, таких как поиск, удаление, копирование и т.д.
3. Операторы управления действиями в Linux включают в себя:
   * && (логическое И) - выполняет вторую команду только если первая команда выполнена успешно.
   * || (логическое ИЛИ) - выполняет вторую команду только если первая команда не выполнена успешно.
   * ; (точка с запятой) - разделяет команды, которые выполняются последовательно.
   * & (амперсанд) - запускает команду в фоне.
   * | (вертикальная черта) - перенаправляет вывод одной команды на вход другой.
4. Операторы, используемые для прерывания цикла, включают в себя:
   * break - прерывает выполнение цикла и продолжает выполнение скрипта после цикла.
   * continue - прерывает текущую итерацию цикла и продолжает выполнение с следующей итерации.
5. Команды false и true используются для возвращения определенного статуса выполнения. true всегда возвращает 0 (успешное выполнение), а false всегда возвращает 1 (неуспешное выполнение). Они могут быть полезны в скриптах для управления потоком выполнения в зависимости от результатов предыдущих операций.
6. Строка if test -f man$s/$i.$s встреченная в командном файле, проверяет, существует ли файл с именем $i.$s в директории man$s. Если файл существует, то условие if будет истинным, иначе - ложным.
7. Конструкции while и until используются для создания циклов в shell-скриптах.
   * while - цикл будет продолжаться, пока условие является истинным.
   * until - цикл будет продолжаться, пока условие является ложным.