Лабораторная работа № 14

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Мальянц Виктория Кареновна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Задание № 1. Написать командный файл, реализующий упрощённый механизм семафоров. Ко- мандный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (> /dev/tty#, где # — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.
2. Задание № 2. Реализовать команду man с помощью командного файла. Изучите содержимое ката- лога /usr/share/man/man1. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1.
3. Задание № 3. Используя встроенную переменную $RANDOM, напишите командный файл, генерирую- щий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.
4. Контрольные вопросы

# 3 Выполнение лабораторной работы

## 3.1 Задание № 1

Создаю файл lab14-1.sh и открываю его (рис. 1).

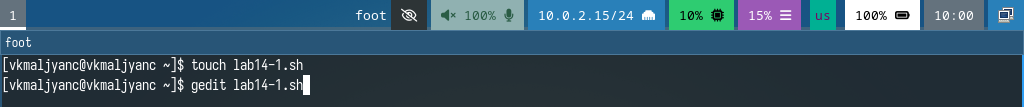


Рис. 1: Создание файла lab14-1.sh и открытие его

Ввожу код в файл lab14-1.sh (рис. 2).

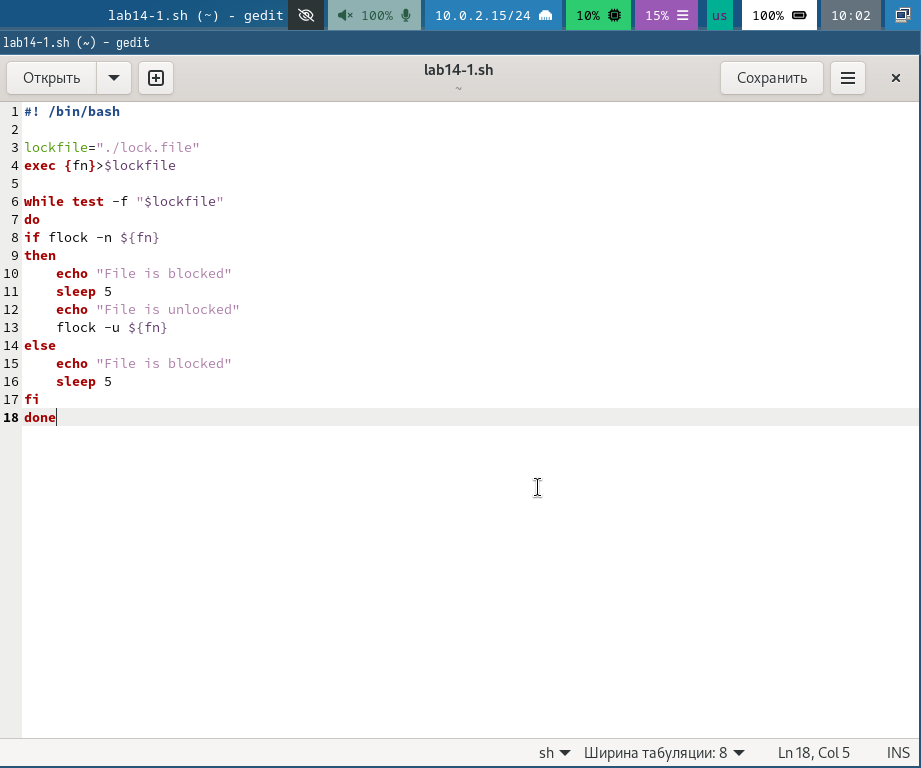


Рис. 2: Редактирование файла

Даю право на исполнение файла lab14-1.sh и запускаю его. Убеждаюсь в том, что программа работает корректно (рис. 3).



Рис. 3: Право на исполнение файла lab14-1.sh и запуск этого файла

## 3.2 Задание № 2

Просматриваю содержимое /usr/share/man/man1 (рис. 4).

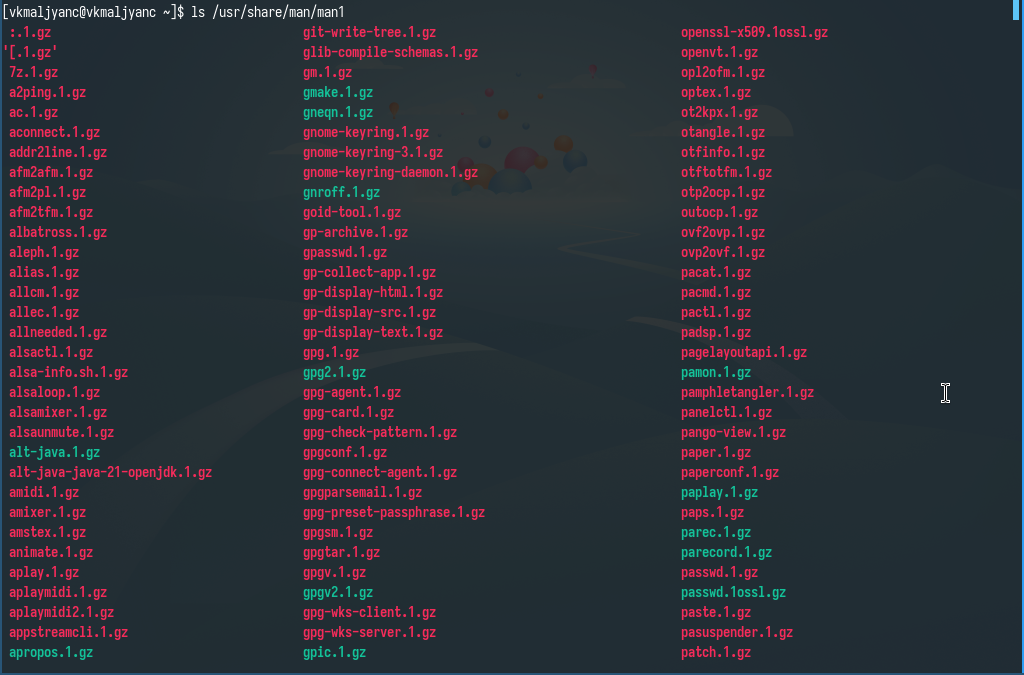


Рис. 4: Просмотр содержимого /usr/share/man/man1

Создаю файл lab14-2.sh и открываю его (рис. 5).

Создание файла lab14-2.sh и открытие его

Рис. 5: Создание файла lab14-2.sh и открытие его

Ввожу код в файл lab14-2.sh (рис. 6).

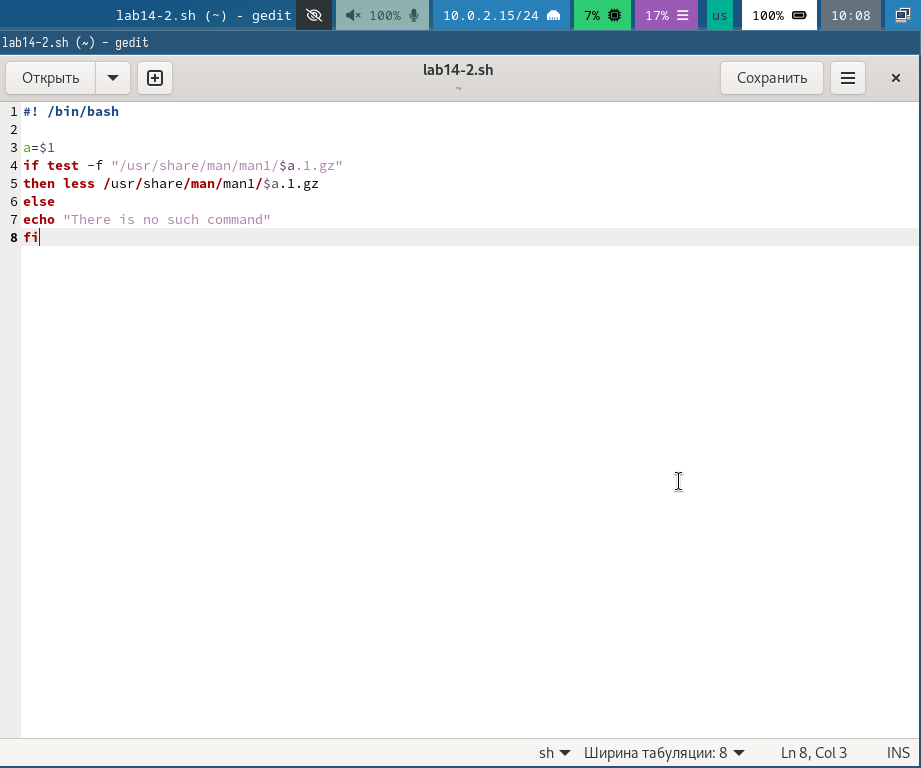


Рис. 6: Редактирование файла

Даю право на исполнение файла lab14-2.sh и запускаю его (рис. 7).

Право на исполнение файла lab14-2.sh и запуск этого файла

Рис. 7: Право на исполнение файла lab14-2.sh и запуск этого файла

Убеждаюсь в том, что программа работает корректно (рис. 8).

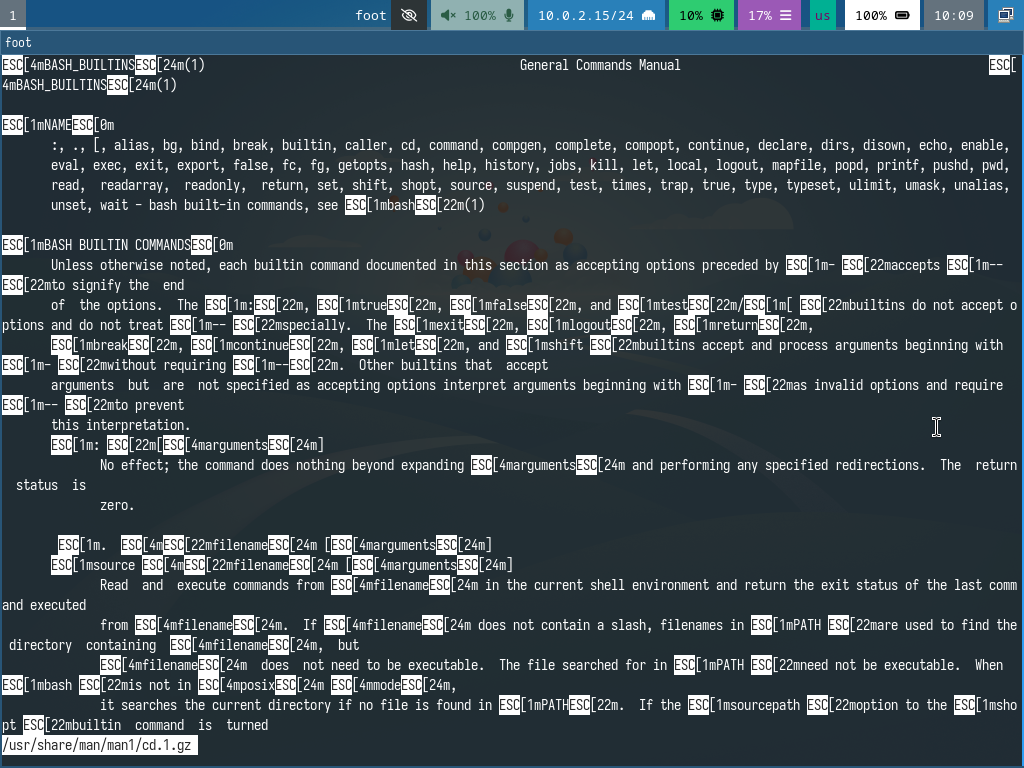


Рис. 8: Справка

## 3.3 Задание № 3

Создаю файл lab14-3.sh и открываю его (рис. 9).

Создание файла lab14-3.sh и открытие его

Рис. 9: Создание файла lab14-3.sh и открытие его

Ввожу код в файл lab14-3.sh (рис. 10).

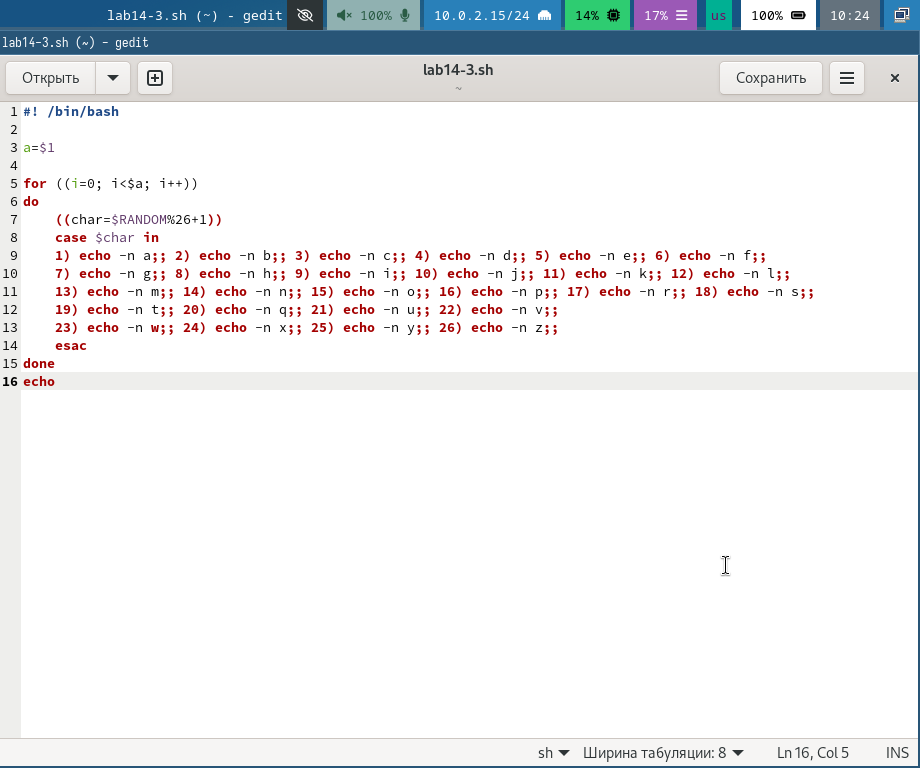


Рис. 10: Редактирование файла

Даю право на исполнение файла lab14-3.sh и запускаю его. Убеждаюсь в том, что программа работает корректно (рис. 11) [1].

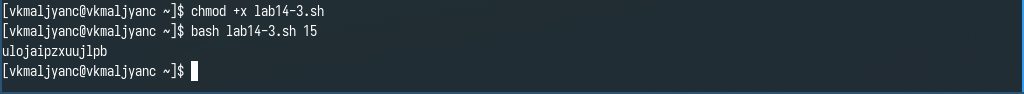


Рис. 11: Право на исполнение файла lab14-3.sh и запуск этого файла

# 4 Выводы

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 5 Контрольные вопросы

1. Пробел должен быть между [ и условием.
2. В bash можно объединить строки, используя оператор + или просто ставя строки рядом.
3. Утилита seq используется для генерации последовательностей чисел. Иные способы: использование цикла for, использование printf.
4. Результат будет 3, так как в bash происходит целочисленное деление.
5. Отличия командной оболочки zsh от bash: расширенные возможности автозаполнения, темы и плагины, маски, глобальные алиасы.
6. Верен.
7. Примущества: простота использования, интеграция с системными утилитами, низкий уровень. Недостатки: ограниченная функциональность, отсутствие строгой типизации, медлительность (по сравнению с компилируемыми языками, такими как C или Go).

# Список литературы

1. [Лабораторная работа № 14](https://esystem.rudn.ru/course/view.php?id=113).