Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Медникова Екатерина Михайловна

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 Выполнение лабораторной работы

1. Написала командный файл, реализующий упрощённый механизм семафоров.

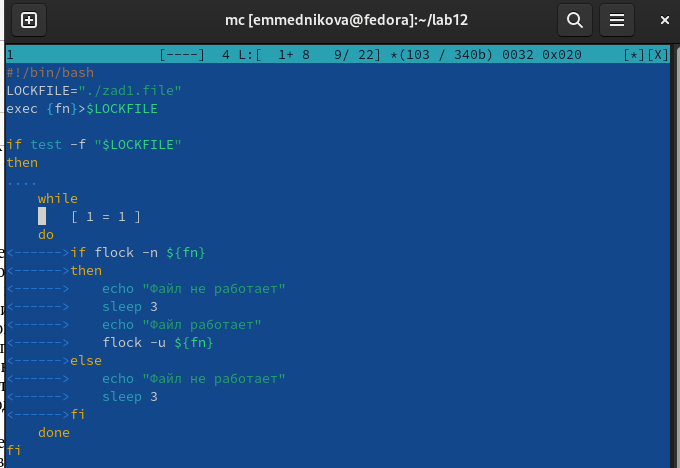


Figure 1: Задание 1



Figure 2: Результат задания 1

1. Реализовала команду man с помощью командного файла. Изучила содержимое каталога /usr/share/man/man1.

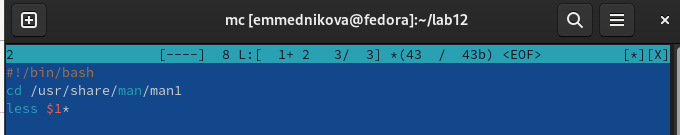


Figure 3: Задание 2



Figure 4: Результат задания 2

1. Написала командный файл, генерирующий случайную последовательность букв латинского алфавита.

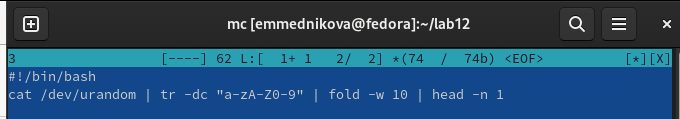


Figure 5: Задание 3

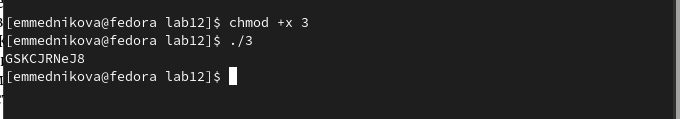


Figure 6: Результат задания 3

# 3 Выводы

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

# 4 Контрольные вопросы

1. *Найдите синтаксическую ошибку в следующей строке:* while [$1 != “exit”]

Ответ: Ошибка заключается в $1. Также нет пробелов между квадратными скобками.

1. *Как объединить (конкатенация) несколько строк в одну?*

Ответ: Добавление переменных или литеральных строк к переменной с помощью оператора +=

1. *Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash?*

Ответ: Утилит seq выдаёт последовательность чисел. Реализовать можно с помощью команды for n in {1..5} do done.

1. *Какой результат даст вычисление выражения $((10/3))?*

Ответ: 3

1. *Укажите кратко основные отличия командной оболочки zsh от bash.*

Ответ: Оболочка zsh предлагает встроенную коррекцию орфографии, улучшенное завершение командной строки, загружаемые модули, которые выступают в качестве плагинов для оболочки, глобальные псевдонимы, которые позволяют использовать псевдонимы имен файлов или чего-либо еще в командной строке вместо просто команд, и больше поддержки тем. Это похоже на bash, но с множеством дополнительных возможностей, дополнительных функций и настраиваемых параметров.

1. *Проверьте, верен ли синтаксис данной конструкции:* for ((a=1; a <= LIMIT; a++))

Ответ: Да, верен.

1. *Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?*

Bash — это командный язык, а не язык программирования общего назначения. Поэтому с усложнением кода он становится более запутанным и менее читаемым. Кроме того, Bash всё и всегда воспринимает как команду, потому что это командный язык. У Bash нет стандартного API, однако он поставляется с простыми встроенными функциями (например, со встроенной тестовой обработкой). Однако часто придется создавать процессы для обработки данных.