Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Расширенное программирование

Панченко Денис Дмитриевич

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

Напишем командный файл, реализующий упрощённый механизм семафоров (рис. [1](#fig:001) - [3](#fig:003)).

Figure 1: Создание файла

Figure 1: Создание файла

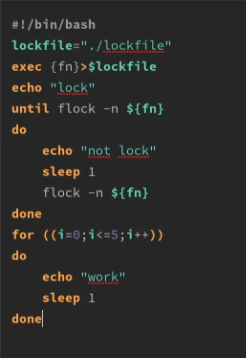


Figure 2: Программа

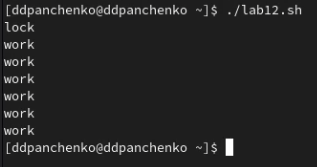


Figure 3: Запуск и вывод

Реализуем команду man с помощью командного файла (рис. [4](#fig:004) - [7](#fig:007)).

Figure 4: Создание файла

Figure 4: Создание файла

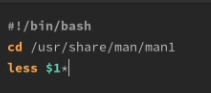


Figure 5: Программа

Figure 6: Запуск

Figure 6: Запуск

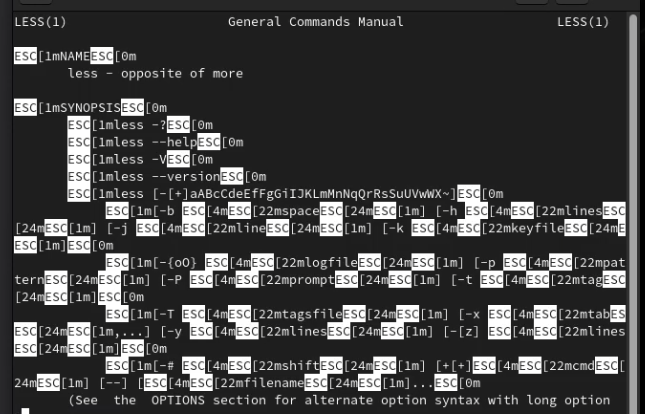


Figure 7: Вывод

Используя встроенную переменную $RANDOM, напишем командный файл, генерирующий случайную последовательность букв латинского алфавита (рис. [8](#fig:008) - [10](#fig:010)).

Figure 8: Создание файла

Figure 8: Создание файла



Figure 9: Программа

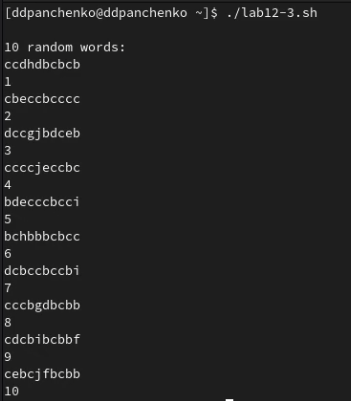


Figure 10: Запуск и вывод

# 3 Вывод

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 4 Контрольные вопросы

1. Синтаксическая ошибка в данной строке заключается в том, что в bash используется двойное квадратное скобочное выражение для условных операторов, а не круглые скобки. Правильно будет написать так: while [ $1 != “exit” ].
2. Несколько строк можно объединить с помощью оператора конкатенации строк -, например: str1=“Hello”; str2=“world”; result=str2; echo $result. В данном примере результатом будет строка “Hello-world”.
3. Утилита seq используется для генерации последовательностей чисел. Пример использования: seq 1 10. Эта команда выведет последовательность чисел от 1 до 10. Альтернативными способами генерации последовательностей являются использование циклов for или while с использованием оператора счетчика, например: for i in {1..10}; do echo $i; done.
4. Выражение $((10/3)) даст результат 3, так как в bash при делении целых чисел используется целочисленное деление без округления.
5. Основные отличия командной оболочки zsh от bash:

* zsh имеет более продвинутый и удобный интерфейс командной строки;
* zsh имеет множество дополнительных функций и возможностей, таких как автодополнение путей и команд, подсветка синтаксиса и др.;
* zsh имеет более продвинутую систему настройки и использования алиасов и функций.

1. Данный синтаксис верен. Он используется для цикла for с заданным начальным значением, конечным значением и шагом счетчика.
2. Bash - это язык скриптовой обработки командной строки, который удобен для быстрого и простого выполнения повседневных задач в системе Unix/Linux. Он имеет простой синтаксис и понятную логику, что делает его легко доступным для новичков. Однако, в отличие от других языков программирования, bash не имеет широких возможностей для создания сложных алгоритмов и программ. Также, bash может быть медленнее в выполнении сложных задач по сравнению с более производительными языками программирования, такими как Python или Java.