Отчет по лабораторной №14

Камалиева Лия Дамировна

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. 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. 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. 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

# 3 Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это про- грамма, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; – С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления програм- мой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей сов- мещает свойства оболочек С и Корна (разработка компании Free Software Foundation)

# 4 Выполнение лабораторной работы

Шаг 1.прописываю код по первому заданию и создаю файл semaphore.sh

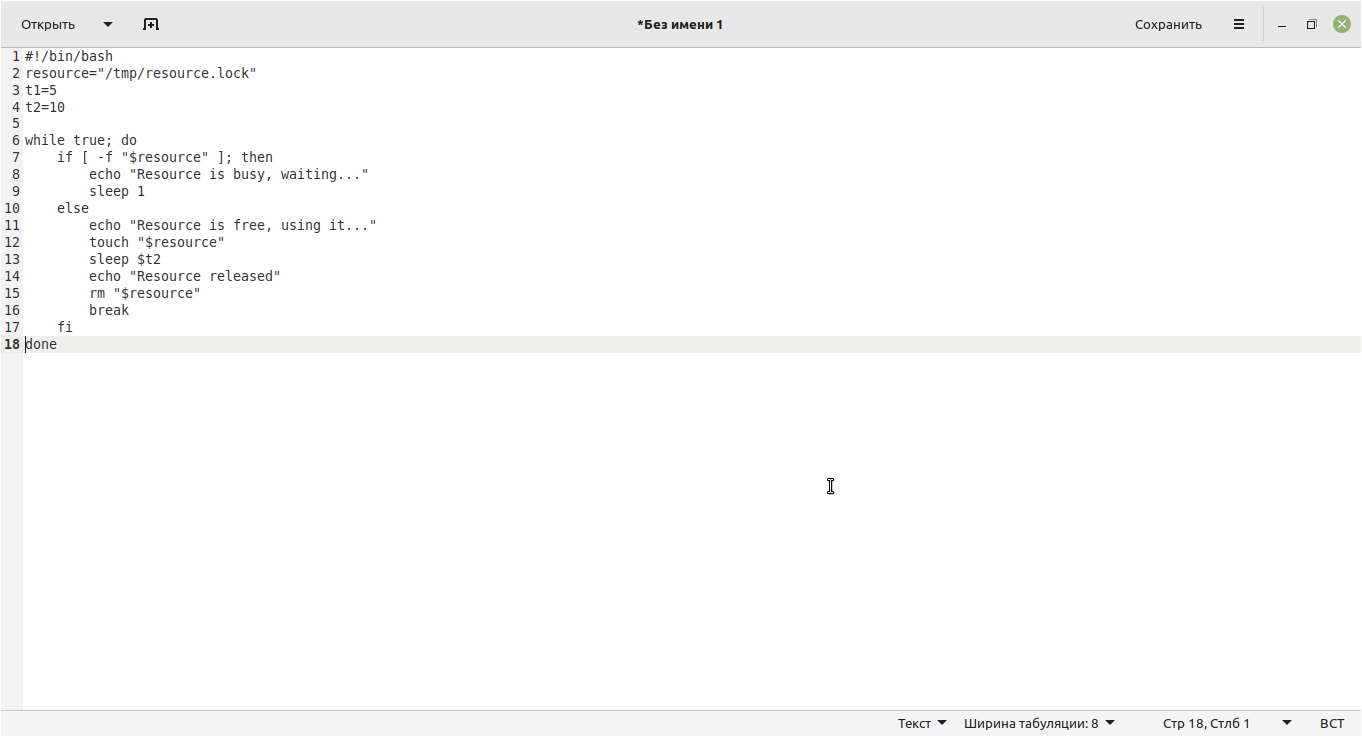


рис.1.1

Шаг 2. запускаю код

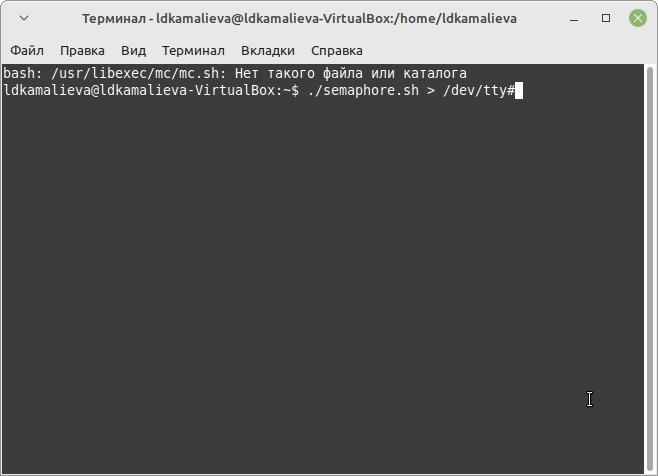


рис.1.2

Шаг 3. прописываю код по 2 заданию и сохраняю файл под именем man.sh

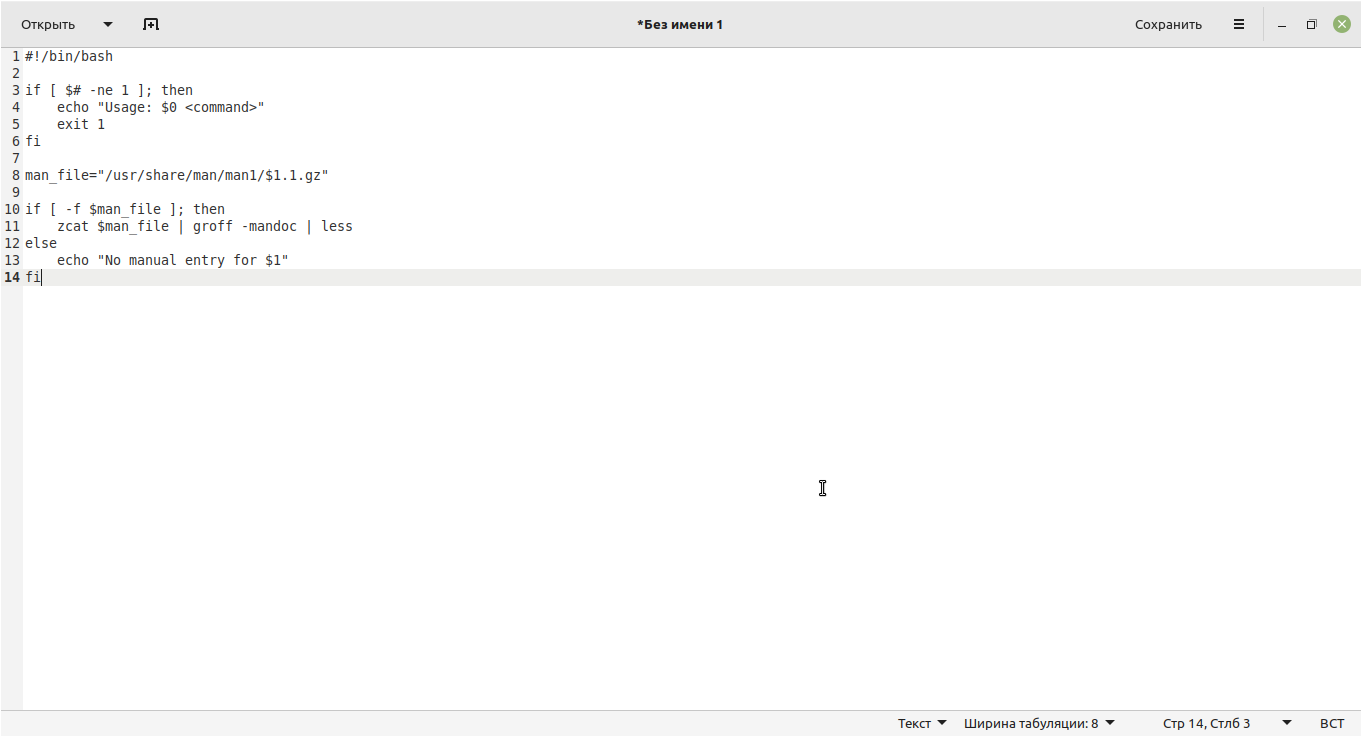


рис.1.3

Шаг 4. делаю код исполняемым

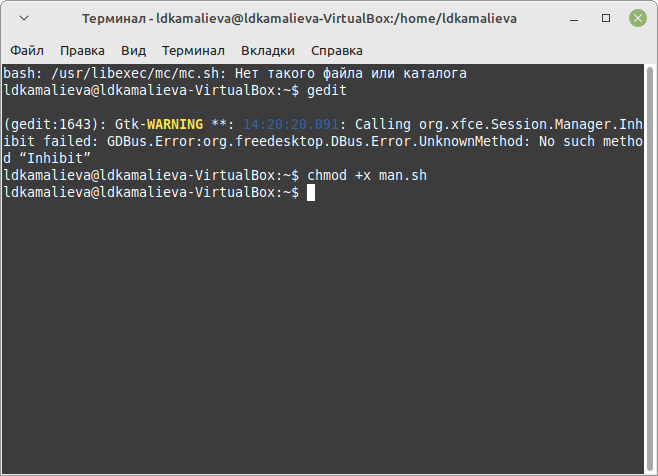


рис.1.4

Шаг 5. Этот скрипт будет пытаться найти файл справки для указанной команды и откроет его в less, если он найден. Если файла справки нет, он выдаст сообщение об отсутствии справки.

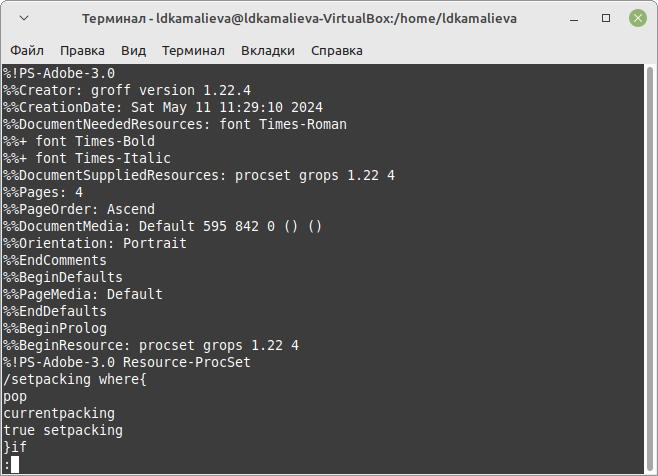


рис.1.5

Шаг 6. прописываю скрипт для задания с рандомными числами

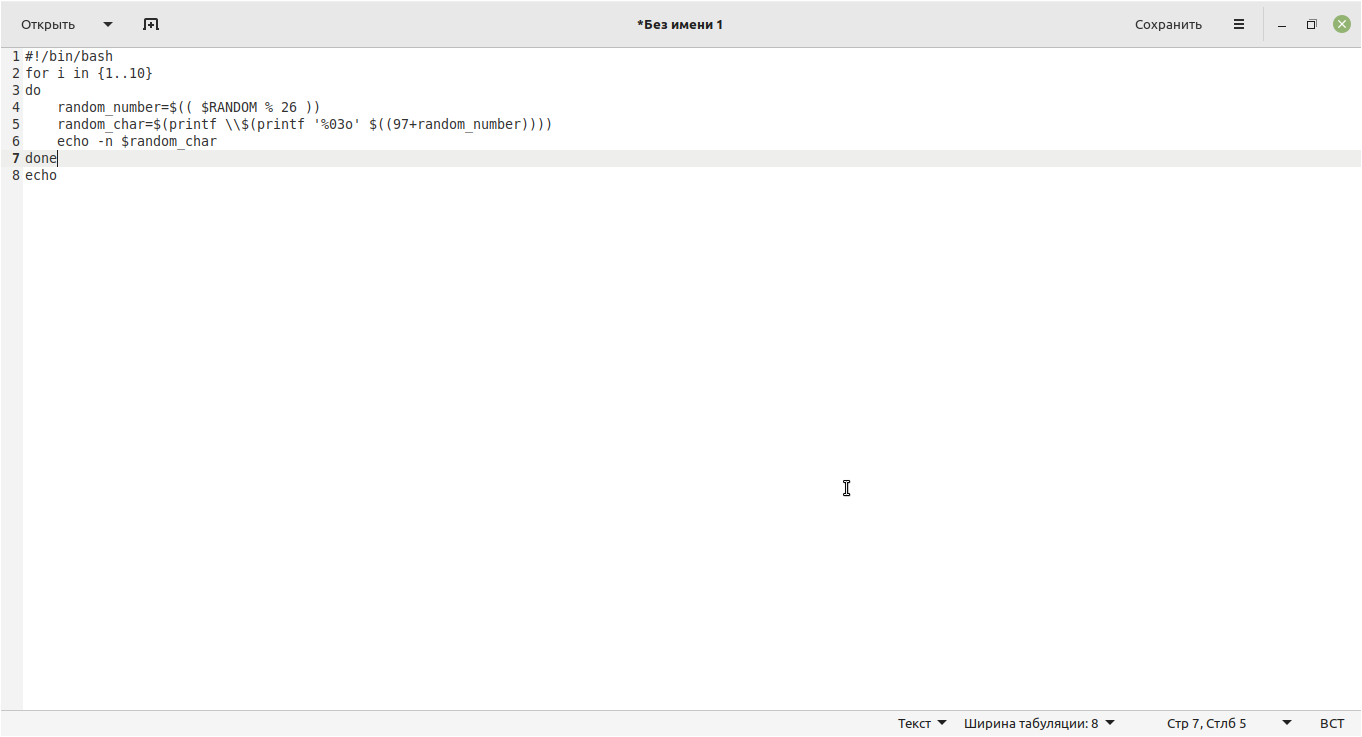


рис.1.6

Шаг 7. пишу команду, чтобы запустить код

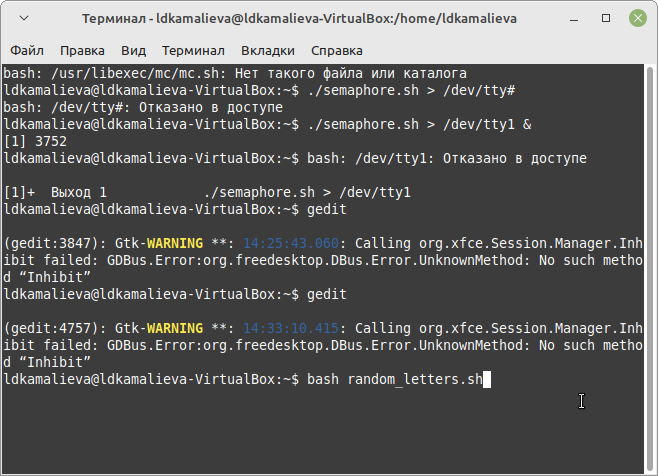


рис.1.7

Шаг 8. проверяю работу, всё работает

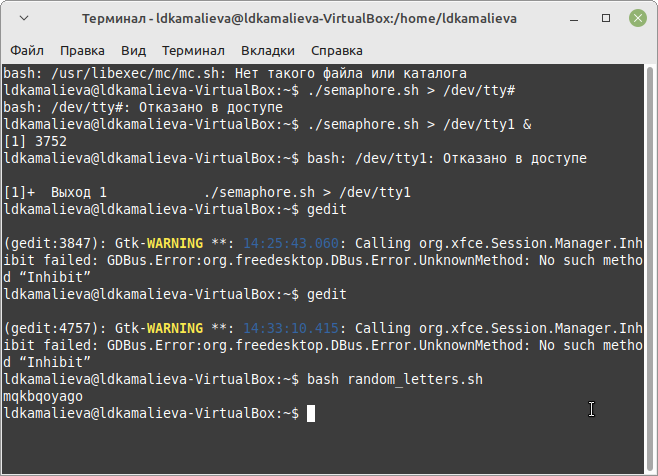


рис.1.8

## 4.1 Контрольные вопросы

1 Каково предназначение команды getopts? Команда getopts осуществляет синтаксический анализ командной строки, выделяя флаги, ииспользуется для объявления переменных. Синтаксис команды следующий: getopts option-string variable [arg. . . ] Флаги это опции командной строки, обычно помеченные знаком минус; Например, для команды ls флагом может являться -F. Строка опций option-string это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за символом, обозначающим этот флаг, должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Еслик оманда getopts может распознать аргумент, то она возвращает истину. Принято включать getopts в цикл while и анализировать введённые данные с помощью оператора case. Функция getopts включает две специальные переменные среды OPTARG и OPTIND. Если ожидается доплнительное значение,то OPTARG устанавливается в значение этого аргумента. Функция getopts также понимает переменные типа массив, следовательно, можно использовать её в функции не только для синтаксического анализа аргументов функций, но и для анализа введённых пользователем данных. 2 Какое отношение метасимволы имеют к генерации имён файлов? Приперечислении имён файлов текущего каталога можно использовать следу- ющие символы: 1 соответствует произвольной, в том числе и пустой строке; 2 ? соответствует любому одинарному символу; 3 [c1-c2] соответствует любому символу, лексикографически находящемуся между символами с1 и с2. Например, 1.1 echo выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls; 1.2. ls.c выведет все файлы с последними двумя символами, совпадающими с.c. 1.3. echoprog.? выведет все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog.. 1.4.[a-z] соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита. 3 Какие операторы управления действиями вы знаете? Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования bash предоставляет возможность использовать такие управляющие конструкции, как for, case, if иwhile. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании команд- ных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути, являются операторами языка программирования bash. Поэтому при описании языка программирования bash термин оператор будет использоваться наравне с термином команда. Команды ОСUNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использо- вания в командных файлах. Единственная функция этой команды заключается в выработке кода завершения. 4 Какие операторы используются для прерывания цикла? Два несложных способа позволяют вам прерывать циклы в оболочке bash. Ко- манда break завершает выполнение цикла, а команда continue завершает данную итерацию блока операторов. Команда break полезна для завершения цикла while в ситуациях, когда условие перестаёт быть правильным. Команда continue исполь- зуется в ситуациях, когда больше нет необходимости выполнять блок операторов,но вы можете захотеть продолжить проверять данный блок на других условных выражениях. 5 Для чего нужны команды false и true? Следующие две команды ОСUNIX используются только совместно с управля- ющими конструкциями языка программирования bash: это команда true,которая всегда возвращает код завершения, равный нулю(т.е.истина),и команда false,которая всегда возвращает код завершения,неравный нулю(т.е.ложь).Примеры бесконечных циклов:while true do echo hello andy done until false do echo hello mike done. 6 Что означает строка if test -f mans/i.$s, встреченная в командном файле? Строка if test-fmans/i.sпроверяет,существуетлифайлmans/i.s и является ли этот файл обычным файлом.Если данный файл является каталогом,то команда вернет нулевое значение (ложь). 7 Объясните различия между конструкциями while и until. Выполнение оператора цикла while сводится к тому,что сначала выполняет- ся последовательность команд(операторов), которую задаёт список-команд в строке, содержащей служебное слово while,а затем,если последняя выполненная команда из этой последовательности команд возвращает нулевой код заверше- ния(истина),выполняется последовательность команд(операторов),которую задаёт список-команд в строке,содержащей служебное слово do,после чего осуществляется безусловный переход на начало оператора цикла while.Выход из цикла будет осуществлён тогда,когда последняя выполненная команда из последовательности команд (операторов),которую задаёт список-команд в строке,содержащей служебное слово while, возвратит ненулевой код завершения(ложь). При замене в операторе цикла while служебного слова while на until условие,при выполнении которого осуществляется выход из цикла,меняется на противоположное.В остальном оператор цикла while и оператор цикла until идентичны.

# 5 Выводы

я ознакомилась с функциями emacs

# Список литературы