Отчёт по лабораторной работе №12

Дисциплина: Операционные системы

Горпинич Елена Михайловна

Содержание

[Цель работы 1](#_Toc104284659)

[Выполнение лабораторной работы 1](#_Toc104284660)

[Вывод 8](#_Toc104284661)

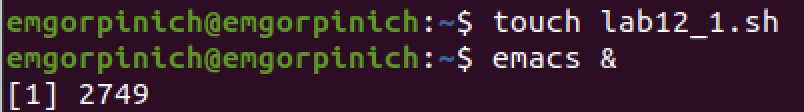
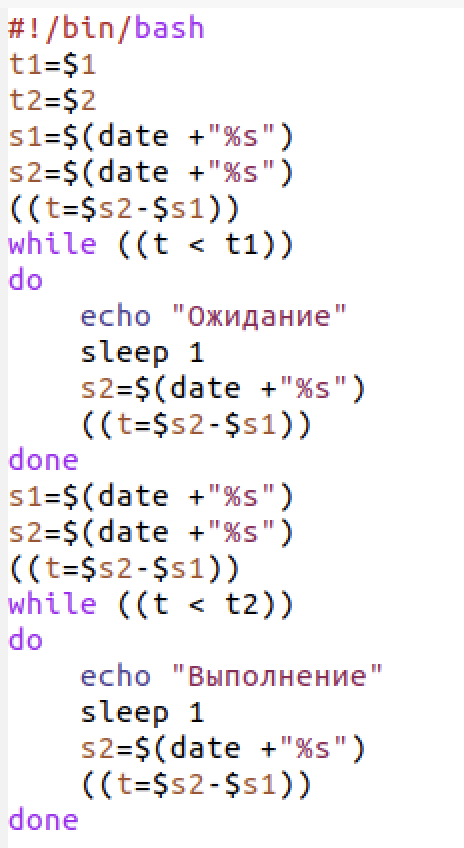
[Контрольные вопросы: 8](#_Toc104284662)

# Цель работы

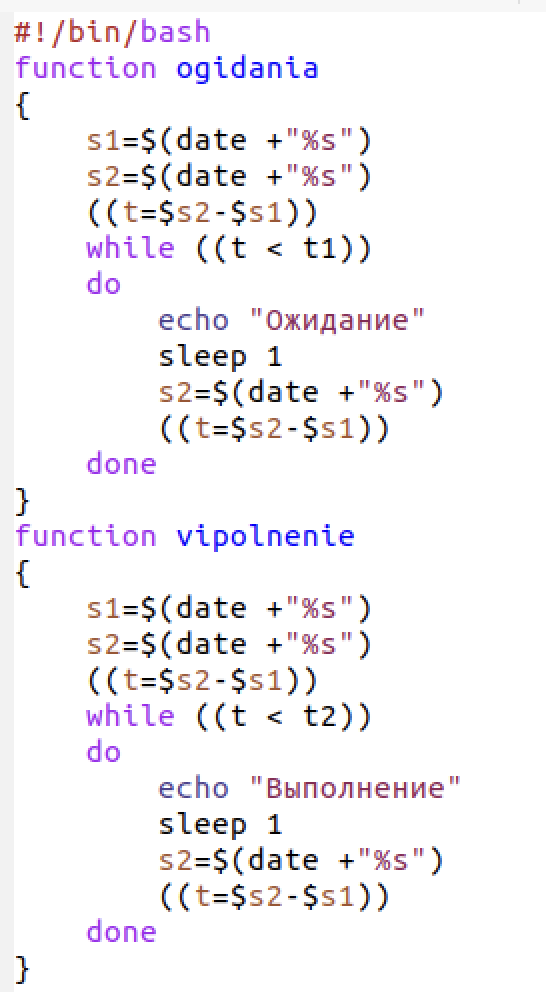
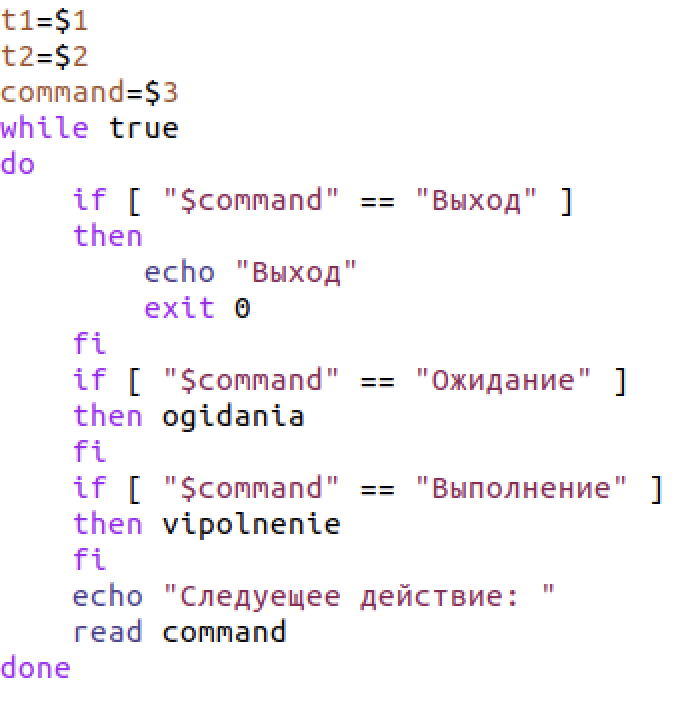
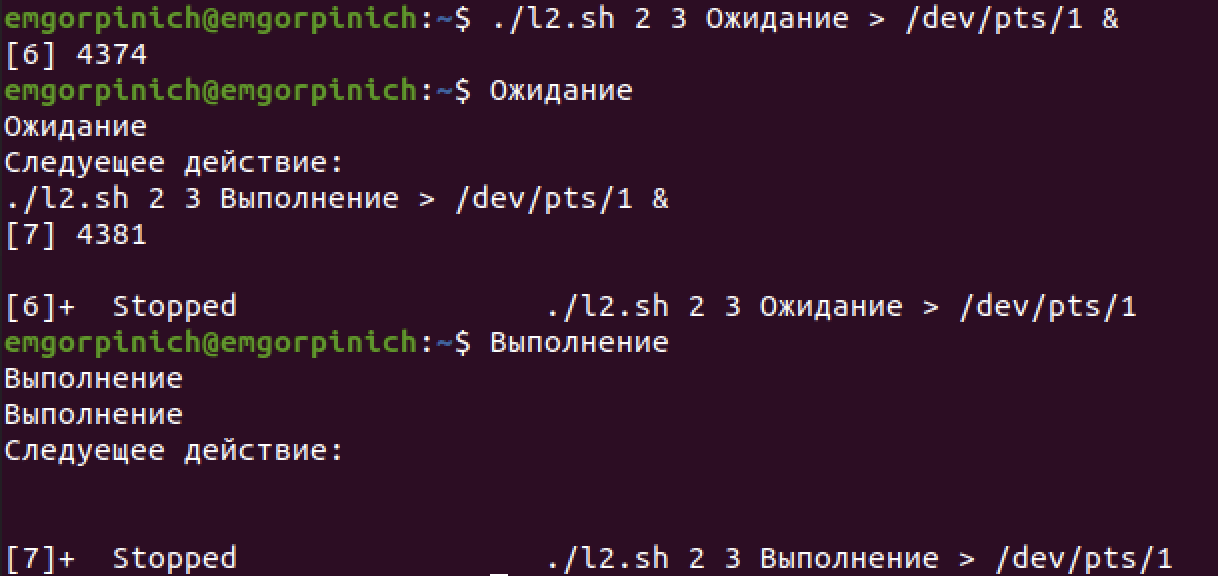
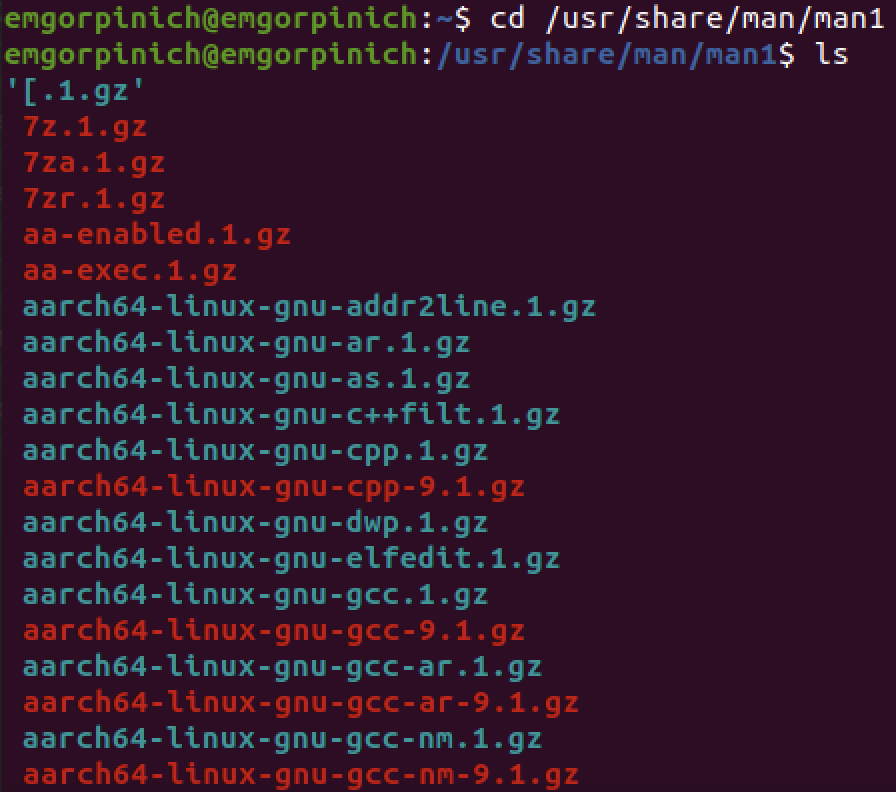
Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов. # Задание

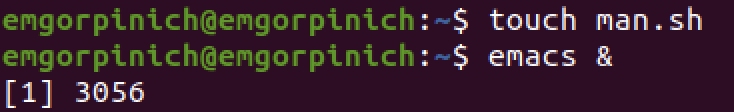
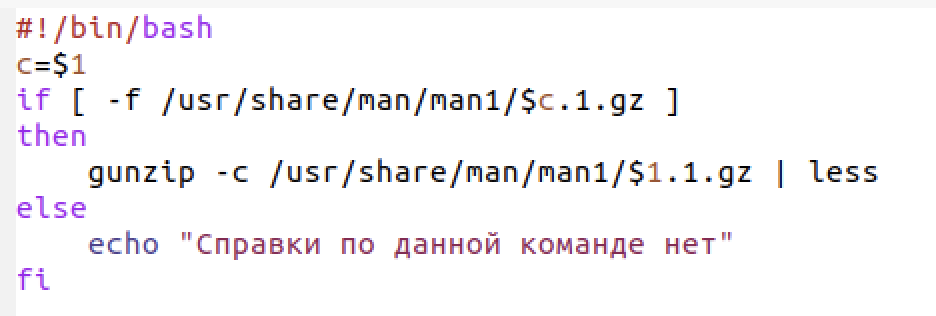
1. Ознакомиться с теоретическим материалом.
2. Сделать отчёт по лабораторной работе №12 в формате Markdown.
3. Изучить основы программирования в оболочке ОС UNIX/Linux.
4. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

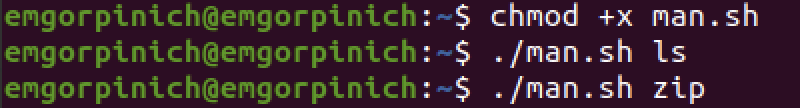
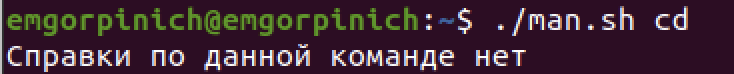
# Выполнение лабораторной работы

1. Написала командный файл, реализующий упрощённый механизм семафоров. Командный 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). Для данной задачи я создала файл и написала соответствующий скрипт.  

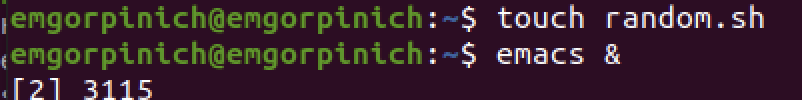
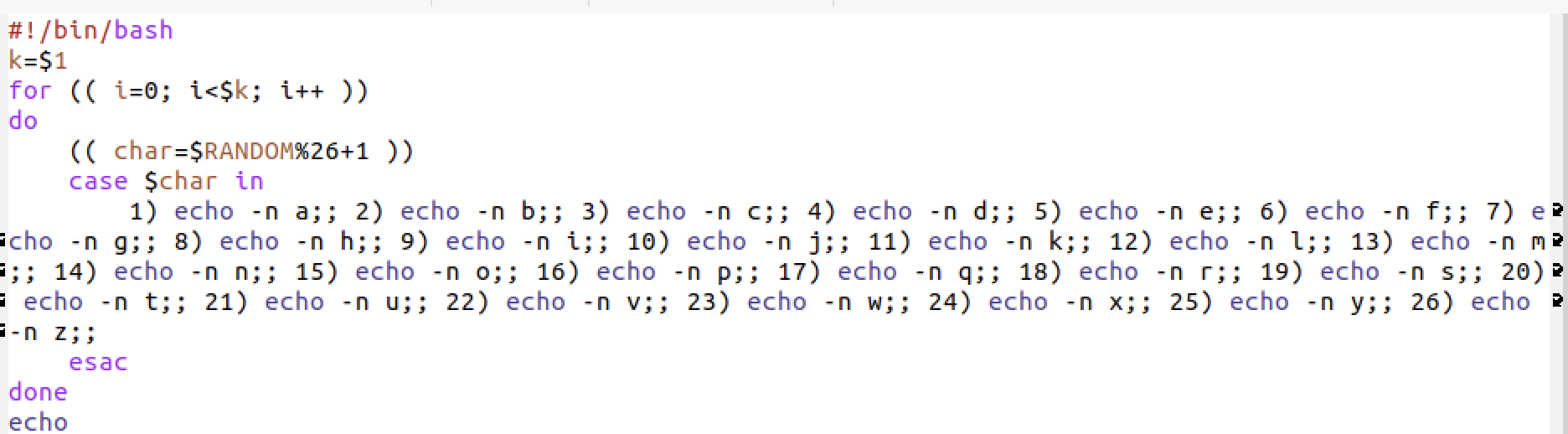
Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла. Скрипт работает корректно. 

После этого я изменила скрипт так, чтобы его можно было выполнять в нескольких терминалах и проверила его работу    2) Реализовала команду man с помощью командного файла. Изучила содержимое каталога /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 

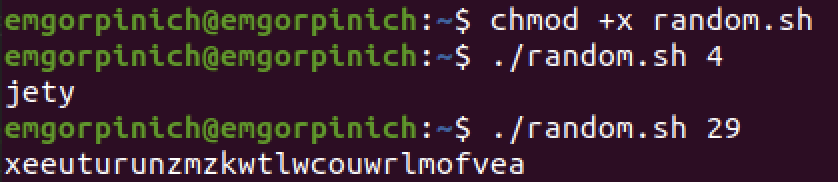
Для данной задачи я создала файл и написала соответствующий скрипт    
Далее я проверила работу написанного скрипта , предварительно добавив право на исполнение файла. Скрипт работает корректно.

1. Используя встроенную переменную $RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Для данной задачи я создала файл: random.sh и написала соответствующий скрипт

Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла. Скрипт работает корректно.



15

# Вывод

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Контрольные вопросы:

1. while [$1 != “exit”] В данной строчке допущены следующие ошибки: • не хватает пробелов после первой скобки [и перед второй скобкой ] • выражение $1 необходимо взять в “ ”, потому что эта переменная может содержать пробелы. Таким образом, правильный вариант должен выглядеть так: while [“$1”!= “exit”]
2. Чтобы объединить несколько строк в одну, можно воспользоваться несколь- кими способами: • Первый: VAR1=”Hello, “VAR2=” World” VAR3=“𝑉 𝐴𝑅1VAR2” echo “VAR1” Результат: Hello, World
3. Команда seq в Linux используется для генерации чисел от ПЕРВОГО до ПОСЛЕДНЕГО шага INCREMENT. Параметры: • seq LAST: если задан только один аргумент, он создает числа от 1 до LAST с шагом шага, равным 1. Если LAST меньше 1, значение is не выдает. • seq FIRST LAST: когда заданы два аргумента, он генерирует числа от FIRST до LAST с шагом 1, равным 1. Если LAST меньше FIRST, он не выдает никаких выходных данных. • seq FIRST INCREMENT LAST: когда заданы три аргумента, он генерирует числа от FIRST до LAST на шаге INCREMENT . Если LAST меньше, чем FIRST, он не производит вывод. • seq -f «FORMAT» FIRST INCREMENT LAST: эта команда используется для генерации последовательности в форматированном виде. FIRST и INCREMENT являются необязательными. • seq -s «STRING» ПЕРВЫЙ ВКЛЮЧЕНО: Эта команда используется для STRING для разделения чисел. По умолчанию это значение равно /n. FIRST и INCREMENT являются необязательными. • seq -w FIRST INCREMENT LAST:эта команда используется для выравнива- ния ширины путем заполнения начальными нулями. FIRST и INCREMENT являются необязательными.
4. Результатом данного выражения $((10/3))будет 3, потому что это целочис- ленное деление без остатка.
5. Отличия командной оболочки zsh от bash: • В zsh более быстрое автодополнение для cdс помощью Тab • В zsh существует калькулятор zcalc,способныйвыполнятьвычислениявнут- ри терминала • В zsh поддерживаются числа с плавающей запятой • В zsh поддерживаются структуры данных «хэш» •В zsh поддерживается раскрытие полного пути на основе не полных данных • В zsh поддерживается замена части пути •В zsh есть возможность отображать разделенный экран,такой же какразделенный экран vim
6. for((a=1; a<= LIMIT; a++)) синтаксис данной конструкции верен, потому что, используя двойные круглые скобки, можно не писать $ перед переменными ().
7. Преимущества скриптового языка bash: • Один из самых распространенных и ставится по умолчаниюв большинстве дистрибутивах Linux, MacOS • Удобное перенаправление ввода/вывода • Большое количество команд для работы с файловыми системами Linux • Можно писать собственные скрипты, упрощающие работу в Linux Недостатки скриптового языка bash: • Дополнительные библиотеки других языков позволяют выполнить больше действий • Bash не является языков общего назначения • Утилиты,привыполнениискрипта,запускаютсвоипроцессы,которые,в свою очередь, отражаются на быстроте выполнения этого скрипта • Скрипты,написанныенаbash,нельзязапуститьнадругихоперационных системах без дополнительных действий.