РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №13

дисциплина: Операционные системы

Студент: Вахадж Сараджуддин Группа: НПИбд-02-20

МОСКВА

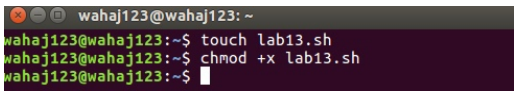
2021 г

Цель работы:

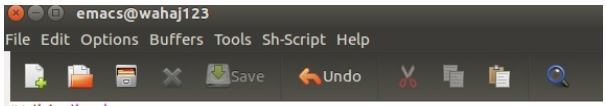
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Ход работы:

1. Авторизовался в системе, создал текстовый документ, затем зашел в него.



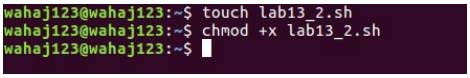
2. Мы Написали командный файл, реализующий упрощённый механизм семафоров. Ко-мандный файл должен в течение некоторого времени1дожидаться освобожде-ния ресурса, выдавая об этом сообщение, а дождавшись его освобождения, ис-пользовать его в течение некоторого времени2<>t1, также выдавая информа-цию о том, что ресурс используется соответствующим командным файлом.



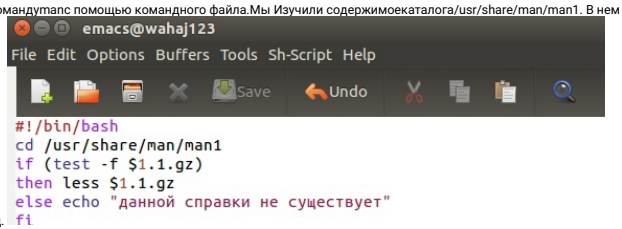

```
#!/bin/bash
x="./x"
exec {fn}>$x
echo "блокировка"
until flock -n ${fn}
do
    echo "не блокировка"
    sleep 1
    flock -n ${fn}
done
for((i = 0; i <= 5; i++))
do
    echo "работает"
    sleep 1
done
```

3. МЫ Запустили командный файл в одном виртуальном терминале в фоновомрежиме, перенаправив его вывод в другой (> /dev/tty#, где#— номер тер-минала куда перенаправляется вывод), в котором также запущен этот файл, ноне фоновом, а в привилегированном режиме. Доработать программу так, чтобыимелась возможность взаимодействия трёх и более процессов.

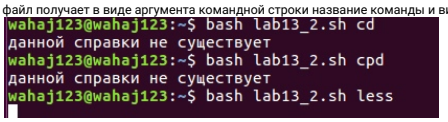
4. Мы Создали текстовый файл с расширением sh. после командой chtod разрешил выполнения файла.



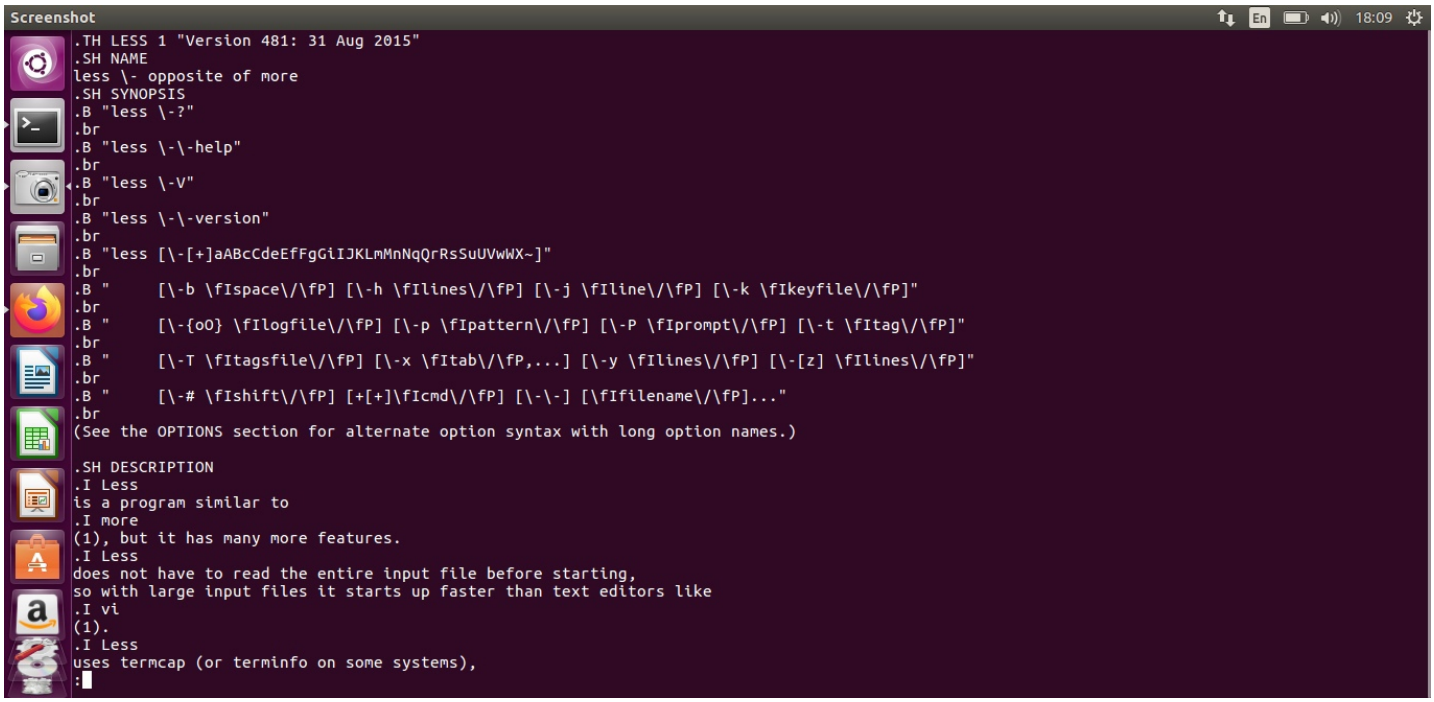
5. Мы Реализовали командуmanc помощью командного файла.Мы Изучили содержимоекаталога/usr/share/man/man1. В нем находятся архивы текстовых файлов,содержащих справку по большинству установленных в системе программ и команд.



6. Мы Запустили командный файл. Командный файл получает в виде аргумента командной строки название команды и виде результата выдавает справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет каталоге man1.



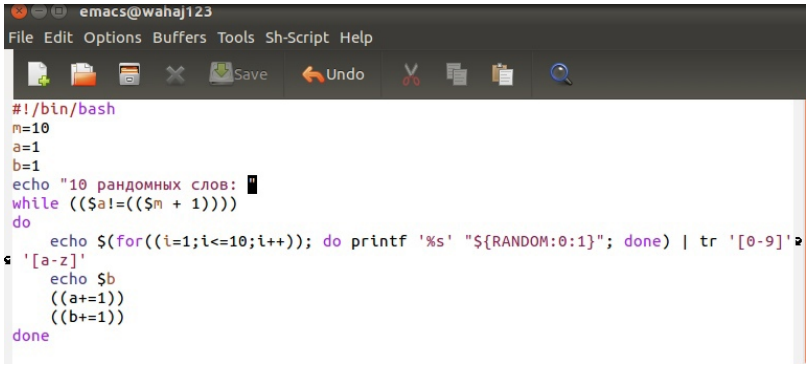
7. Каждый архив можно открыть командойlesssразу же просмотрев содер-жимое справки.



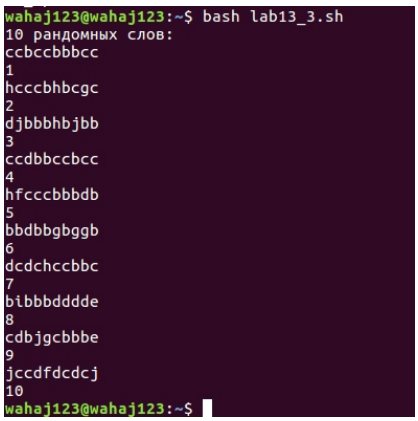
8. Мы Создали текстовый файл с расширением .sh, после командой chmod разрешил выполнения файла.

```
wahaj123@wahaj123:~$ touch lab13_3.sh
wahaj123@wahaj123:~$ chmod +x lab13_3.sh
wahaj123@wahaj123:~$ emacs
```

9. Используя встроенную переменную$RANDOM, напишите, командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите,что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.



10. мы Запустили командный файл. Как видим, вывел рандомные 10 слов, состоящих из рандомных букв латинского алфавита.



Вывод:

Мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Контрольные вопросы:

1. Найдите синтаксическую ошибку в следующей строке while [$1 != "exit"])$1 следует внести в кавычки(«»)
2. Как объединить (конкатенация) несколько строк в одну? С помощью знака >,|
3. Найдите информацию об утилите seq. Какими иными способами можно реализовать ее функционал при программировании на bash? Эта утилита выводит последовательность целых чисел с заданным шагом. Также можно реализовать с помощью утилиты jot.
4. Какой результат даст вычисление выражения $((10/3))? Результат: 3.
5. Укажите кратко основные отличия командной оболочки zsh от bash. В zsh можно настроить отдельные сочетания клавиш так, как вам нравится. Использование истории команд в zsh ничем особенным не отличается от bash. Zsh очень удобен для повседневной работы и делает добрую половину рутины за вас. Но стоит обратить внимание на различия между этими двумя оболочками. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1, чего совершенно невозможно понять. Так, если вы используете shell для повседневной работы, исключающей написание скриптов, используйте zsh. Если вам часто приходится писать свои скрипты, только bash! Впрочем, можно комбинировать. Как установить zsh в качестве оболочки по-умолчанию для отдельного пользователя:o
6. Проверьте, верен ли синтаксис данной конструкции for ((a=1; a <= LIMIT; a++)) Синтаксис верен.
7. Сравните язык bash с языками программирования, которые вы знайте. Какие преимущества у bash по сравнению с ними? Какие недостатки?
8. Скорость работы программ на ассемблере может быть более 50% медленнее, чем программ на си/си++, скомпилированных с максимальной оптимизаций;
9. Скорость работы виртуальной ява-машины с байт-кодом часто превосходит скорость аппаратуры с кодами, получаемыми трансляторами с языков высокого уровня. Ява-машина уступает по скорости только ассемблеру и лучшим оптимизирующим трансляторам;
10. Скорость компиляции и исполнения программ на яваскрипт в популярных браузерах лишь в 2-3 раза уступает лучшим трансляторам и превосходит даже некоторые качественные компиляторы, безусловно намного (более чем в 10 раз) обгоняя большинство трансляторов других языков сценариев и подобных им по скорости исполнения программ;
11. Скорость кодов, генерируемых компилятором языка си фирмы Intel, оказалась заметно меньшей, чем компилятора GNU и иногда LLVM;
12. Скорость ассемблерных кодов x86-64 может меньше, чем аналогичных кодов x86, примерно на 10%;
13. Оптимизация кодов лучше работает на процессоре Intel;
14. Скорость исполнения на процессоре Intel была почти всегда выше, за исключением языков лисп, эрланг, аук (gawk, mawk) и бэш. Разница в скорости по бэш скорее всего вызвана разными настройками окружения на тестируемых

системах, а не собственно транслятором или железом. Преимущество Intel особенно заметно на 32-разрядных кодах;

15. Стек большинства тестируемых языков, в частности, ява и яваскрипт, поддерживают только очень ограниченное число рекурсивных вызовов. Некоторые трансляторы (gcc, icc, ...) позволяют увеличить размер стека изменением переменных среды исполнения или параметром;
16. В рассматриваемых версиях gawk, php, perl, bash реализован динамический стек, позволяющий использовать всю память компьютера. Но perl и, особенно, bash используют стек настолько экстенсивно, что 8-16 ГБ не хватает для расчета ack(5,2,3).