отчёта по лабораторной работе 13

Ндри Ив Алла Ролан

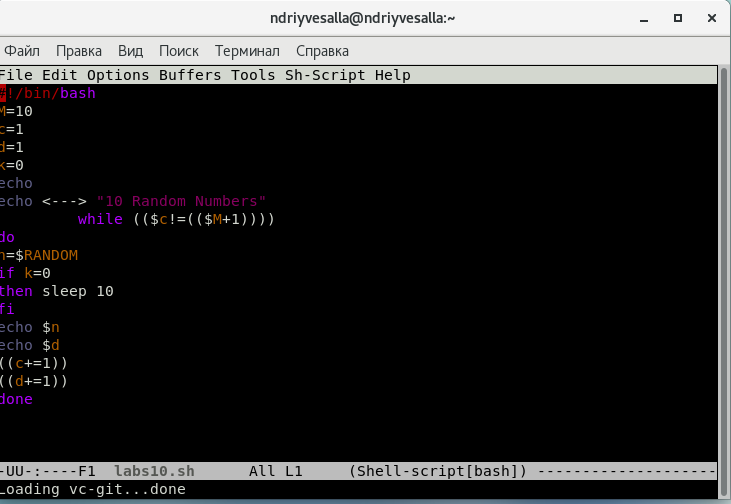
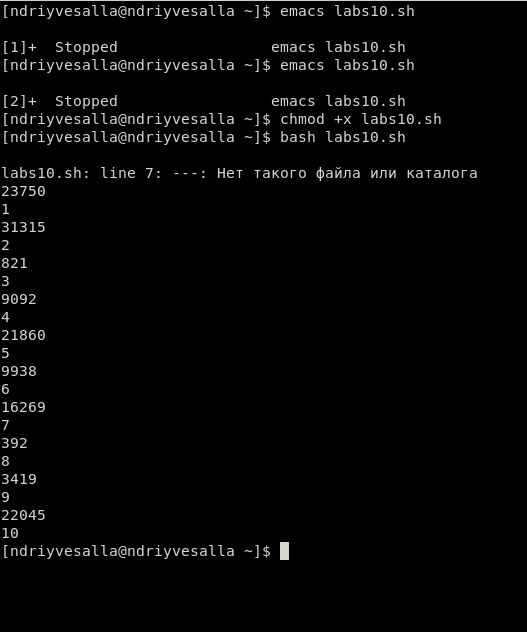
Содержание

## Цель работы

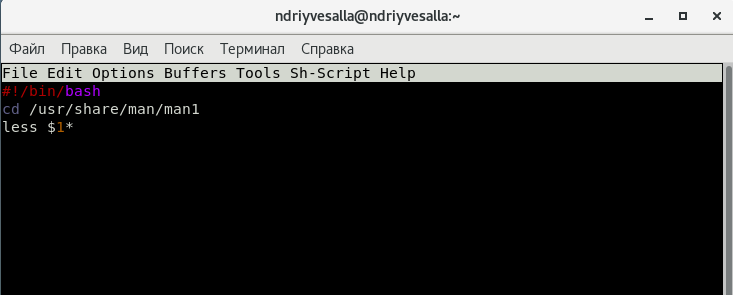
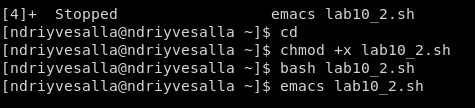
Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

## Выполнение работы.

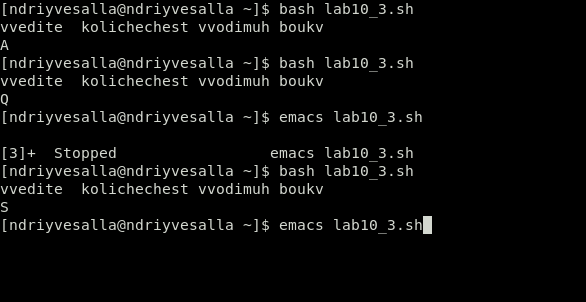
1 1 Осуществили вход в систему, создали текстовый документ, затем перешли в него. Написали командный файл, реализующий упрощённый механизм семафоров. Командный файл в течение некоторого времени t1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустили командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой, в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработали программу так, чтобы имелась возможность взаимодействия трёх и более процессов.  
(рис. **¿fig:001?**) (рис. **¿fig:002?**)

 { # fig:001 width=70%}  { # fig:002 width=70%}

2 C помощью командного файла реализуем команды man. Изучим содержимое каталога usr/share/man/man1. Каждый архив можно открыть при помощи команды less. Командный файл получает в виде аргументы командной строки название команды и и виде результат выдает справку. (рис. **¿fig:003?**) (рис. **¿fig:004?**)

 { # fig:003 width=70%}  { # fig:004 width=70%}

3.3 Используя встроенную переменную $RANDOM напишем командный файл, генерирующий случайную последовательность цифр. (рис. **¿fig:005?**) (рис. **¿fig:006?**)

 { # fig:005 width=70%}  { # fig:006 width=70%}

## Вывод

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

## Ответы на контрольные вопросы

1. Нужно поставить пробелы после [ и до ].
2. С помощью знака >. 3 )seq — Утилита, выводязая последовательность целых чисел с шагом, заданным пользователем. Так же можно реализовать с помощью утилиты jot.
3. 3
4. 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.
5. Верно.
6. Bash является языком программирования очень высокого уровня. При должном навыке программиста, он может работать быстрее, чем Си. Однако его сложностью является то, что нужно следовать строгим правилам синтаксиса при написании программы.