

Отчет по лабораторной работе № 2 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Любарский Иван Владимирович, № по списку 8

Контакты ivanred289@gmail.com, @rMeDGranD

Работа выполнена: «21» сентября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Операционная среда ОС UNIX

2. **Цель работы:** Изучение и освоение программного обеспечения ОС UNIX и приобретение навыков, необходимых для выполнения курсовых и лабораторных работ в среде UNIX.

3. **Задание:** Приобретение основных навыков работы в ОС UNIX

4. Оборудование:

Процессор *Intel Core i5-4210U @ 4x 1.7GH* с ОП 15873 Мб, НМД 512 Гб. Монитор 1600x900

5. Программное обеспечение:

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04.1 LTS

интерпретатор команд: *bash* версия 4.4.19.

Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Для получения основных навыков владения ОС Unix потребуется попробовать в терминале команды: перемещения между директориями, создания директорий и файлов, удаление директорий и файлов, чтения файлов, запись в файлы, копирование файлов, переименование файлов, поиск файлов, определения местоположения в файловой системе, вывода файлов директории.

Доп. идея заключается в том, что нужно отслеживать действия пользователя в терминале Ubuntu относительно незаметно для него. Для этого необходимо написать скрипт *bash*, после запуска которого будет создан скрытый файл, в который будут записываться все команды пользователя в терминале в последующем.

Нужно создать наиболее близкий к визуальному оформлению стандартного терминала вывод скрипта(имя пользователя, компьютера и путь к текущему местоположению в файловой системе) при помощи специальных кодом для команды вывода `echo.{0,1}m, {30..50}m`.

Также нужно реализовать выполнимость команд, поступающих на ввод в скрипт. Для этого понадобится создать локальную переменную. Пользовательский ввод будет перенаправляться в нее. После при помощи команды вывода `echo` исполняем данные инструкции.

Остается только записать имеющиеся данные. Для незаметной реализации этого надо создать скрытый текстовый файл, в который будет поступать информация в виде строк из уже имеющейся локальной переменной вместе с данными о времени выполнения команды.

Скрипт будет функционировать(при помощи цикла) до момента написания в терминале команды `exit`(при помощи условного оператора), что будет интерпретироваться им как сигнал к завершению отслеживания.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

План работы:

1. Определения состава текущей директории через команду `ls`.
2. Определение текущего местоположения в файловой системе через команду `pwd`.
3. Создание новой директории через команду `mkdir`.
4. Переход в новую директорию через команду `cd`.
5. Создание нового файла через команду `touch`.
6. Запись строки в новый файл через команду `cat`.
7. Редактирование нового файла через редактор `nano`.

8. Копирование нового файла через команду cp.
9. Переименование скопированного файла через команду mv.
10. Удаление скопированного файла через команду rm.
11. Перемещение в домашнюю директорию.
12. Поиск файла по названию через команду find.

Алгоритм выполнения скрипта:

1. Создание скрытого файла.
2. Вывод строки, похожей на строку ожидания ввода команды для терминала.
3. Сохранение пользовательского ввода в локальную переменную.
4. Запись в скрытый файл информацию о вводе.
5. Выполнение команды пользователя.
6. Сравнить, не является ли пользовательской командой exit.
- 6.1. Завершение скрипта.
7. Вернуться к пункту 2.

Входные данные	Выходные данные (в созданном файле)	Описание тестируемого случая
mkdir n	Cp 21 сен 2022 18:30:52 MSK:/home/moze/GitRepos/lab1 - mkdir n	Запись в файл данных о создании новой директории пользователем
ls	Cp 21 сен 2022 18:30:53 MSK:/home/moze/GitRepos/lab1 - ls	Запись в файл данных о просмотре файлов в данной директории
touch testFile.txt	Cp 21 сен 2022 18:31:03 MSK:/home/moze/GitRepos/lab1 - touch testFile.txt	Запись в файл данных о создании нового файла
rm testFile.txt	Cp 21 сен 2022 18:31:19 MSK:/home/moze/GitRepos/lab1 - rm testFile.txt	Запись в файл данных об удалении файла

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
moze@Moze:~$ ls
Games  OlimpC++ Видео  Загрузки  Музыка  'Рабочий стол'
GitRepos snap  Документы  Изображения  Общедоступные  Шаблоны
moze@Moze:~$ pwd
/home/moze
moze@Moze:~$ mkdir test
moze@Moze:~$ cd test/
moze@Moze:~/test$ touch testfile.txt
moze@Moze:~/test$ cat > testfile.txt
This is a line
moze@Moze:~/test$ nano testfile.txt
moze@Moze:~/test$ cp testfile.txt secondfile.txt
moze@Moze:~/test$ mv secondfile.txt da.txt
moze@Moze:~/test$ rm da.txt
moze@Moze:~/test$ cd ~
moze@Moze:~$ find -name testfile.txt
./test/testfile.txt
```

Код скрипта:

```
touch .log.txt
spyfunc(){
echo -n "\033[32m\033[1m$(whoami)@$(hostname)"'\033[37m:"'\033[34m$(pwd)"'\033[>
echo $(date):$(pwd) - $ACT >> .log.txt
END=$( $ACT)
if [ "$END" = "" ]; then
echo -n ""
else
echo $END
fi
if [ "$ACT" = "exit" ]; then
exit
fi
}
while [ 1 ]
do
spyfunc
done
```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	21.09.2022	18:56	Вывод лишней пустой строки при выполнении команд не подразумевающих вывода	Решено путем добавления дополнительной локальной переменной и ее проверкой на наличие пустой строки	Большинство системных команд работает исправно

10. Замечания автора по существу работы

11. Выводы

Данная лабораторная работа помогает разобраться в основах управления файловой системой Unix путем разбора основных команд. Благодаря этому даже человек, первый раз запустивший ОС основанную на Unix, сможет разобраться с азами управления системой и с принципом работы данного ПО.

Подпись студента _____