



Отчет по лабораторной работе № 8 по курсу Практикум на ЭВМ

Студент группы М8О-104Б-22 Алхимова Дарья Игоревна, № по списку 02

Контакты www, e-mail, icq, skype fl81m@yandex.ru

Работа выполнена: « 1 » января 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Стандартные утилиты Unix для обработки файлов.

2. **Цель работы:** Изучить основные команды обработки текстовых файлов ОС Unix.

3. **Задание (вариант №):** В среде ОС Unix опробовать ряд утилит и получить навыки работы с ними.

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор _____ Intel Core i5 _____ с ОП _____ 8 _____ Гб, НМД _____ 2097152 _____ Мб. Монитор _____
Другие устройства _____

5. **Программное обеспечение (лабораторное):**

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____

Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства _____ macOS _____, наименование _____ macOS Catalina _____ версия _____ 10.15.4 _____
интерпретатор команд _____ cmd _____ версия _____ 10.0.19044.2130 _____
Система программирования _____ версия _____
Редактор текстов _____ nano _____ версия _____ 2.0.6 _____
Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных на домашнем компьютере _____

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

Изучим основные команды обработки текстовых файлов ОС Unix (cmp, comm, wc, dd, diff, grep и др.). Исследуем их синтаксис, после чего реализуем команды на практике с помощью bash-терминала.

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

1. cmp - сравнивает два файла байт за байтом и сообщает место первого различия между ними.
2. comm - сравнивает два отсортированных файла и выводит три столбца: строки только из файла 1, строки только из файла 2 и строки, общие для обоих файлов.
3. wc - подсчитывает количество строк, слов и байтов в указанных файлах.
4. dd - копирует файл.
5. diff - сравнивает файлы и выводит различия между ними.
6. grep - ищет текст в файлах по заданному шаблону.
7. join - объединяет строки двух файлов на основе общего поля.
8. sort - сортирует строки в текстовых файлах.
9. tail - выведет последние строки файла.
10. tee - читает из стандартного ввода и записывает вывод в файл.
11. tr - транслирует или удаляет символы. Он может преобразовывать символы, удалять определенные символы и даже обрабатывать наборы символов.
12. uniq - отфильтровывает смежные повторяющиеся строки из входных данных.
13. od - выводит содержание файла в различных форматах (ASCII, десятичное/ шестнадцатеричное/восьмеричное представление).
14. sum - вычисляет и печатает контрольную сумму и количество блоков для файла.
15. cut - вырезает определенные разделы каждой строки из файла или стандартного ввода.
16. nroff - утилита печати и форматирования текста для терминала и принтера.
17. vi/vim - текстовый редактор, который используется для создания и изменения файлов.
18. mc - текстовый файловый менеджер, который позволяет пользователю управлять файлами и каталогами в двух панелях.
19. tar - создает архив файлов/директорий.

20. gzip - сжимает файлы с помощью QuickLZ.
21. ed - линейный текстовый редактор, редактирует файлы на месте, без необходимости загружать весь файл в память.
22. awk - читает текст построчно, разбивает каждую строку на поля и выполняет указанные операции.
23. sed - потоковый текстовый редактор, который производит базовые текстовые преобразования на входном потоке (файле или вводе с клавиатуры).
24. bzip2 - сжимает файлы с использованием алгоритма Берроуза-Уилера.
25. head - выводит первые строки входного файла или потока.
26. iconv - преобразует текст из одной кодировки символов в другую.
27. patch - анализирует файлы с патчами и применяет изменения к оригинальным файлам.
28. md5 - вычисляет и выводит 128-битный хеш-код (md5 хеш) входного файла.
29. du - оценивает и выводит использование дискового пространства файлов и каталогов.
30. file - которая определяет тип файла.
31. touch - создает новый пустой файл или меняет время последнего изменения уже существующего файла.
32. find - ищет файлы в директории, соответствующие заданным условиям.
33. xargs - преобразует ввод из стандартного потока в аргументы для команды.
34. df - отображает доступное и использованное дисковое пространство на файловых системах.
35. paste - объединяет строки из нескольких файлов.
36. cpp - обрабатывает исходный код на языке C, раскрывая макросы и включая указанные файлы.
37. indent - форматирует исходный код на языке C, делая его более читаемым.
38. split - разделяет файл на несколько файлов меньшего размера.
39. mktemp - создает уникальный временный файл для безопасного использования.

Создадим файлы 'touch file1.txt' 'touch file2.txt'.

Отредактируем содержимое первого файла 'vi file1.txt'. Для начала редактирования введем 'i', для завершения редактирования и сохранения изменений в файл 'ZZ'.

Узнаем вес файла, количество строк, слов и байтов в нем, его MD5-хеш, а также его контрольную сумму 'wc file1.txt' 'md5 file.txt' 'sum file1.txt'.

Результат запишем в файл result.txt 'wc file1.txt | tee result.txt'. Просмотрим первые (утилита head) и последние (утилита tail) 2 строки файла. 'head -n 2 file1.txt' 'tail -n 2 file1.txt'.

Напечатаем все неповторяющиеся строки из первого файла. 'uniq file1.txt'.

Используя утилиту grep найдем все строки, содержащие "string" в первом файле. 'grep "string" file1.txt'.

Напечатаем все неповторяющиеся строки из первого файла. 'uniq file1.txt'.

Преобразуем строчные буквы в заглавные и запишем результат во второй файл.

'tr 'a-z' 'A-Z' < file1.txt > file2.txt'

Сравним файлы сначала при помощи утилиты cmp 'cmp file1.txt file2.txt', а затем при помощи comm 'comm file1.txt file2.txt'.

Выведем первое поле каждой строки из первого файла 'awk '{print \$1}' file1.txt'.

Заменим все вхождения "string" на "table" в файле 'sed 's/string/table/g' file1.txt'.

Определим тип файла 'file file1.txt'.

Сожмем второй файл при помощи gzip 'gzip file2.txt'.

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```
[daraalhimova@MacBook-Air ~ % uniq file1.txt
first string
second sreing
third string
string
Hello!
[daraalhimova@MacBook-Air ~ % touch file2.txt
[daraalhimova@MacBook-Air ~ % wc file1.txt
      6      10      67 file1.txt
[daraalhimova@MacBook-Air ~ % sum file1.txt
54766 1 file1.txt
[daraalhimova@MacBook-Air ~ % head -n 2 file1.txt
first string
second sreing
[daraalhimova@MacBook-Air ~ % tail -n 2 file1.txt
string
Hello!
[daraalhimova@MacBook-Air ~ % awk '{print $1}' file1.txt
first
second
third
third
string
Hello!
[daraalhimova@MacBook-Air ~ % sed 's/string/table/g' file1.txt
first table
second sreing
third table
third table
table
Hello!
[daraalhimova@MacBook-Air ~ % tr 'a-z' 'A-Z' < file1.txt > file2.txt
[daraalhimova@MacBook-Air ~ % cat file2.txt
FIRST STRING
SECOND SREING
THIRD STRING
THIRD STRING
STRING
HELLO!
[daraalhimova@MacBook-Air ~ % cmp file1.txt file2.txt
file1.txt file2.txt differ: char 1, line 1
[daraalhimova@MacBook-Air ~ % comm file1.txt file2.txt
      FIRST STRING
      SECOND SREING
      THIRD STRING
      THIRD STRING
      STRING
      HELLO!
first string
second sreing
third string
third string
string
Hello!
[daraalhimova@MacBook-Air ~ % grep "string" file1.txt
first string
third string
third string
string
[daraalhimova@MacBook-Air ~ % md5 file1.txt
MD5 (file1.txt) = 7c0b5e9a7cedad851fc1bca37f7df2b9
[daraalhimova@MacBook-Air ~ % file file1.txt
file1.txt: ASCII text
```

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 ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы: _____

11. **Выводы:** В ходе выполнения лабораторной работы я изучила основные утилиты Unix и научилась применять их на практике для создания и редактирования файлов, вывода информации о них, копирования содержания одного файла в другой. Изученные команды будут полезны при работе в будущем, они позволяют автоматизировать рутинные задачи, управлять файлами и директориями, мониторить и управлять процессами и выполнять многие другие функции.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____