Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.04 – «Программная инженерия»

Дисциплина: «Администрирование информационных систем»

Профиль: «Разработка программно-информационных систем»

Семестр 5

ОТЧЕТ

по лабораторной работе №3

Тема: «Обработка текстовых данных»

Выполнила: студент группы РИС-22-1б

Поважный В.Е. \_\_\_\_\_\_\_\_\_\_\_

Проверил: ст. преподаватель кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2024

**Цель работы**

Познакомиться с инструментами работы с текстовыми данными доступными в ОС Linux. Получить представление о пользовательском окружении.

**Задание**

1. Отсортируйте вывод команды ls -l по дате изменения содержимого по месяцам.
2. Из файла passwd\_example получите имена всех пользователей, содержащихся в первом поле каждой строки, и поместите отсортированный в обратном порядке результат в файл cut\_result.
3. С помощью текстового редактора (gedit, nano) измените имена нескольких пользователей в этом файле и сохраните результат в новый файл cut\_result2.
4. Сравните содержимое файлов с именами пользователей с помощью программы diff.
5. Добавьте к содержимому файла cut\_result содержимое файла cut\_result2.
6. С помощью команды uniq избавьтесь от дубликатов в файле cut\_result.
7. С помощью одной команды получите домашний каталог пользователя user из файла passwd\_example.
8. Выведите на экран значения всех переменных среды окружения. Проанализируйте полученные результаты и объясните значения известных вам переменных окружения.
9. Определите тип используемого терминала.
10. Измените содержимое переменной PS1 так, чтобы в приглашении в качестве скобок использовались символы «< >».
11. Выполните команду PS1="\[\033[0;41m\]\$\[\033[0m\] ".

**Теоретическая часть**

В операционных системах текстовые файлы используются для хранения данных различного предназначения. В качестве примеров можно выделить файлы конфигурации, системные журналы, файлы с исходным кодом программ и др. Для работы с этими данными разработано большое количество утилит.

Для объединения содержимого нескольких файлов и вывод его в стандартный канал вывода или в файл, используется команда **cat**. Опция **-n** производит нумерацию строк при выводе.

# $ cat file1 file2 > file3

Команда **less** позволяет организовать постраничную работу с большим набором данных.

# $ less file1

Утилита **split** выполняет разделение исходного файла. Например на файлы содержащие определенное количество байт (опция -b), строк (опция - l).

Утилита **sort** выполняет сортировку поступающих ей на вход данных. Использование дополнительных опций позволяет провести сортировку по одному из полей сгруппированных данных.

# $ ls -l ~ | sort -n -k 5

Приведенная выше команда производит числовую (опция **-n**) сортировку полученных от команды **ls** данных по пятому столбцу (опция **-k**).

Утилита **cut** выполняет фильтрацию текста по столбцам. В качестве опций команда принимает номер поля (-**f**), разделитель (**-d**) и др.

# $ cut -d : -f 1 file1

Приведенная выше команда из множества столбцов разделенных символом «**:**» в файле **file1** выбирает первый.

Утилита **uniq** удаляет соседние повторяющиеся строки в файле. Опции команды позволяют также найти неуникальные строки и подсчитать количество вхождений каждой строки. Утилиту **uniq** часто используют совместно с утилитой **sort**.

# $ cat filename | sort | uniq

Команды **head** и **tail** используются для отображения выбранного числа строк в начале или в конце файла. По умолчанию число строк равно 10. Изменить количество выводимых строк можно с помощью опции **-n**.

# $ tail -n 3 file1

Команда **wc** используется для подсчета строк, слов, байт и символов в файле. С помощью опций команде можно передать какую конкретно информацию необходимо получить из файла.

Утилита **diff** производит построчное сравнение файлов переданных в качестве аргументов.

# $ diff file1 file2

Помимо утилит командной строки пользователям Linix доступно большое количество текстовых редакторов. Существуют как консольные (**vi**, **nano**, **mcedit**) редакторы, так и редакторы, используемые в графическом режиме (**gedit**, **kate**). Функциональность некоторых из них можно расширить установкой дополнительных модулей-плагинов.

Таблица 1. Примеры команд

|  |  |
| --- | --- |
| **Команда** | **Описание** |
| cat | Вывод объединенного содержимого файла(-ов). |
| less | Постраничный вывод файла |
| split | Деление файла на части |
| sort | Сортировка и/или слияние файлов |
| cut | Выборка отдельных полей из строк файла |
| uniq | Выдача информации о повторяющихся строках файла |
| head | Вывод начальных строк файла |
| tail | Вывод конечных строк файла |
| diff | Выявление различий между файлами |
| wc | Подсчет количества символов, слов и строк в файле |

# Среда окружения

При работе с командной оболочкой определен набор переменных, описывающих текущий сеанс взаимодействия пользователя с системой, называемый *окружением* (environment)*.*

Переменные окружения доступны одновременно нескольким процессам.

Загрузка переменных окружения из конфигурационных файлов происходит при запуске командного интерпретатора. В дополнение к переменным окружения, оболочка так же хранит псевдонимы и функции оболочки.

Список всех установленных переменных можно получить, используя команды **env** или **set** без опций и аргументов.

Установка новых и изменение значения существующих переменных среды окружения осуществляется путем *экспортирования* (помещения в среду):

# $ export переменная=значение

Узнать значение конкретной переменной можно также при помощи команды:

# $ echo $переменная

Чтобы удалить переменную, используется команда **unset**.

Таблица 2. Некоторые стандартные переменные среды окружения

|  |  |
| --- | --- |
| **Имя** | **Значение** |
| UID | Содержит числовой идентификатор текущего пользователя. Инициализируется при запуске  оболочки. |
| HOME | Домашний каталог текущего пользователя. |
| PATH | Список каталогов, разделённых двоеточием, в которых командная оболочка выполняет поиск файла, в случае если в командене задан его путь. |
| PS1 | Формат строки-приглашения |
| PWD | Текущий каталог. |
| TERM | Тип используемого терминала. |
| HOSTNAME | Сетевое имя компьютера. |

**Ход работы**

Для начала выведем все файлы, которые будут отсортированы по дате последней модификации, при помощи команды ls –l с добавлением утилиты sort.

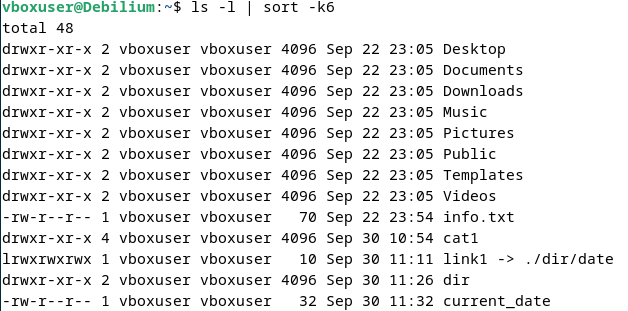


Рис. 1 – отсортированные файлы по дате модификации.

Для выполнения второго задания нужно создать файл passwd\_example, в который запишем всех пользователей. Сделаем это, при помощи команды getent passwd.



Рис. 2 – Создание файла с пользователями.

Далее уже выполним само задание, используя команды cut и sort.

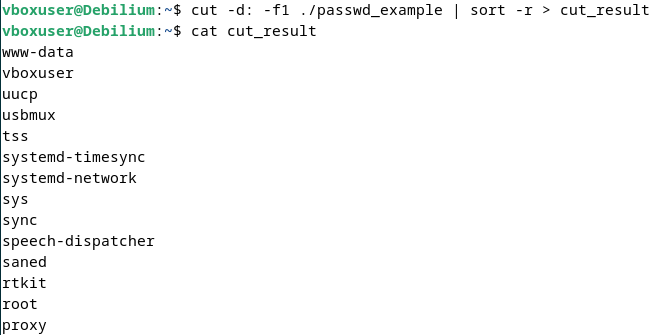


Рис. 3 – разбиение и сортировка файла.

В третьем задания открываем файл cut\_result при помощи текстового редактора (я решил использовать nano). Добавим к нескольким пользователям 1 спереди и сохраним в файл cut\_result2.

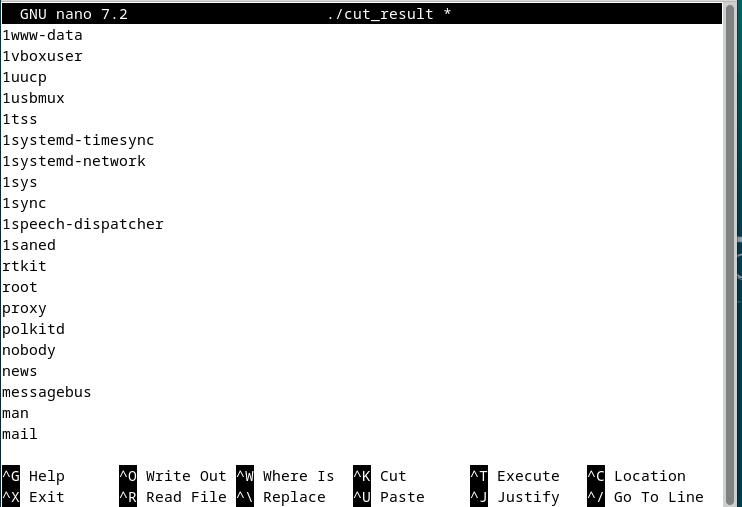


Рис. 4 - текстовый редактор nano

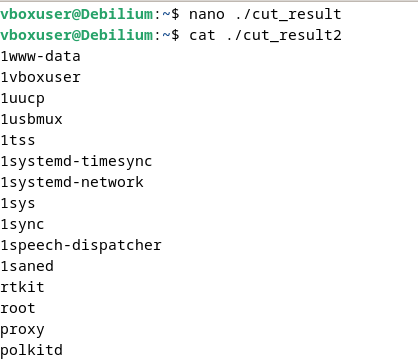


Рис. 5 – результат работы.

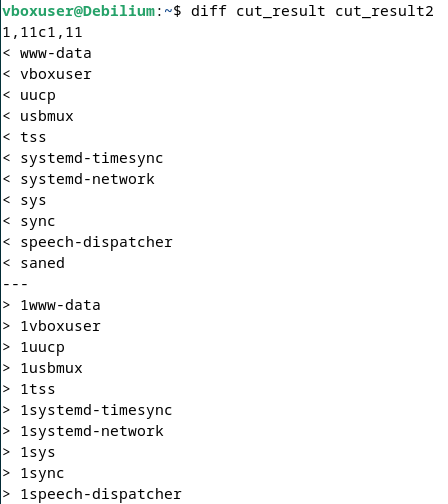


Рис. 6 – сравнение двух файлов через diff

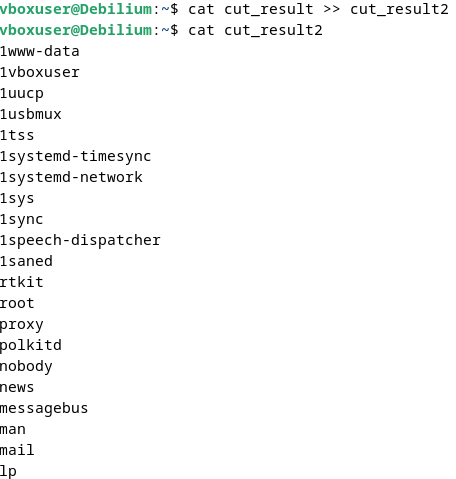


Рис. 7 – добавление содержимого одного файла в другое

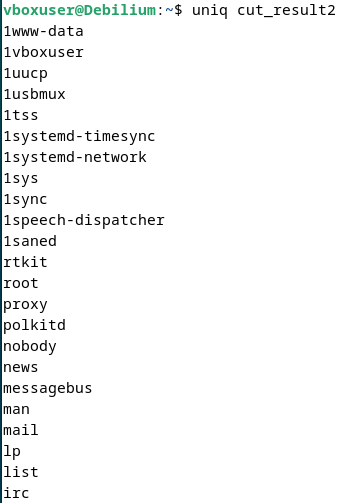


Рис. 8 – удаление дубликатов.



Рис. 9 – домашний каталог пользователя.

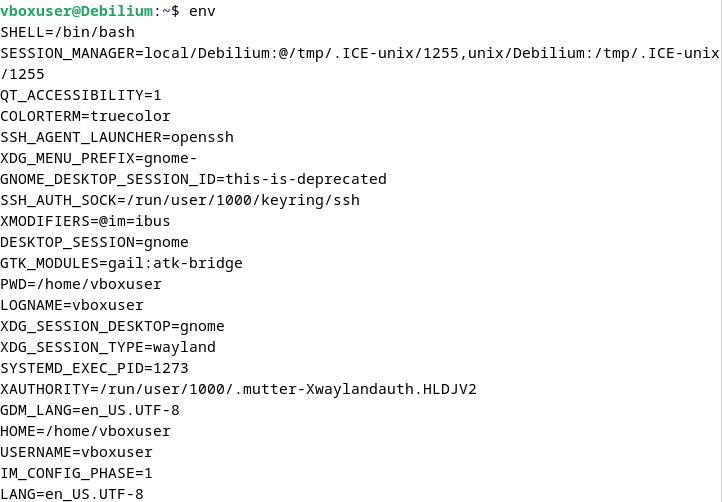
****

Рис. 10 – вывод всех переменных среды окружения.



Рис. 11 – тип терминала.

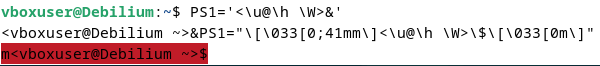
****

Рис. 12 – последнее задание.

**Ответы на контрольные вопросы**

1. Какие утилиты для работы с текстом вы знаете?

Вот некоторые из утилит для работы с текстом в Unix-подобных системах:

- cat — выводит содержимое файлов.

- grep — поиск текста по шаблону.

- awk — язык для обработки текстовых данных.

- sed — утилита для строчной обработки и трансформации текста.

- cut — удаляет (вырезает) определенные поля из строк.

- paste — объединяет строки из разных файлов.

- sort — сортирует строки текста.

- uniq — находит и убирает дубликаты строк.

- tr — заменяет или удаляет символы.

- wc — подсчитывает строки, слова и символы в текстовых файлах.

- head и tail — выводят начало и конец файлов, соответственно.

- nl — нумерует строки.

- tee — считывает со стандартного ввода и записывает как в файл, так и на стандартный вывод.

1. Какая опция используется для изменения порядка сортировки sort?

Для изменения порядка сортировки утилиты sort используется опция -r (reverse), которая сортирует строки в обратном порядке.

1. Что такое переменные окружения?

Переменные окружения — это динамичные значения, которые могут влиять на поведение процессов, выполняемых в операционной системе. Они часто используются для хранения настроек, таких как местоположение домашнего каталога, предпочитаемые текстовые редакторы, настройки локализации и другие параметры, которые могут быть использованы приложениями и скриптами.

1. Как задать значение переменной окружения и как вывести его на экран?

Задать значение переменной окружения:

export MY\_VAR="значение"

Вывести значение переменной на экран:

echo $MY\_VAR

1. Как изменить значения переменных окружения, удалить переменную?

Изменить значение переменной:

export MY\_VAR="новое\_значение"

Удалить переменную:

unset MY\_VAR