

pipe and redirection



ПЛАН ЗАНЯТИЯ

1. Повторение
2. Вопросы по повторению
3. Основной блок
4. Вопросы по основному блоку
5. Задание для закрепления
6. Практическая работа
7. Оставшиеся вопросы

Повторение

- Команды линукса
- Терминалы
- Новые команды
- Как пользоваться командами
- Корневой раздел
- Работа с терминалом
- Флаги и аргументы
- Полный и относительные пути
- Перемещение и переименование
- История



Введение

- Перенаправление
- Запись и дозапись в файл
- Символы `>` и `>>`.



Потоки данных

В Linux существует три стандартных потока ввода/вывода данных.

Первый поток

Стандартный поток ввода (standard input).
В системе это — поток №0.
Этот поток представляет собой некую информацию, передаваемую в терминал, в частности — инструкции, переданные в оболочку для выполнения. Обычно данные в этот поток попадают в ходе ввода их пользователем с клавиатуры.

Второй поток

Стандартный поток вывода (standard output), ему присвоен №1.
Это поток данных, которые оболочка выводит после выполнения каких-то действий. Обычно эти данные попадают в то же окно терминала, где была написана команда, вызвавшая их появление.

Третий поток

Стандартный поток ошибок (standard error), он имеет дескриптор №2.
Этот поток похож на стандартный поток вывода, так как обычно то, что в него попадает, оказывается на экране терминала. Он, по своей сути, отличается от стандартного вывода, как результат, этими потоками, при желании, можно управлять отдельно.

*Номера потоков называют дескрипторами.
Дескриптор — ключевое слово, характеризующее блок информации.

Потоки данных

- Перенаправление ввода/вывода означает работу с потоками и перенаправление данных туда, куда нужно программисту.
- Делается это с использованием символов `>` и `<` в различных комбинациях, применение которых зависит от того, куда, в итоге, должны попасть перенаправляемые данные.

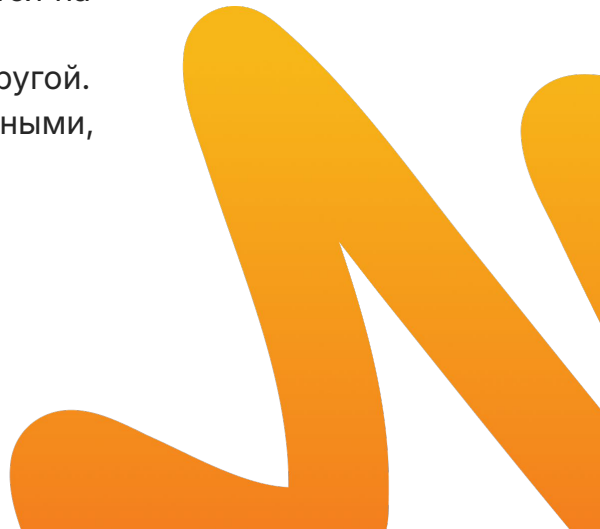


Piping and redirection

- Перенаправление `>` используется гораздо чаще (чем `<` - это вычитка из файла).

Что важно знать про `pipe`?

- Пайпинг (**знак `|`**)
- При использования `pipe` результат выполнения одной команды подается на вход другой.
- Это и есть перенаправление результата команды для выполнения другой. Это не запись в файл, это возможность продолжить работу с данными, которые получены при помощи предыдущей команды.



Команда date

Как работает команда **date**?

1. Программист отправляет текст на сервер.
2. На нем есть оболочка, в которой выполняются команды. Эти оболочки есть разные. Оболочка sh/ ash/ bash. Все зависит от дистрибутивов.
3. ОС получает команду, что нужно выполнить date.
4. ОС находит программу date и запускает ее.
5. Результат исполнения через стандартный вывод отправляется пользователю

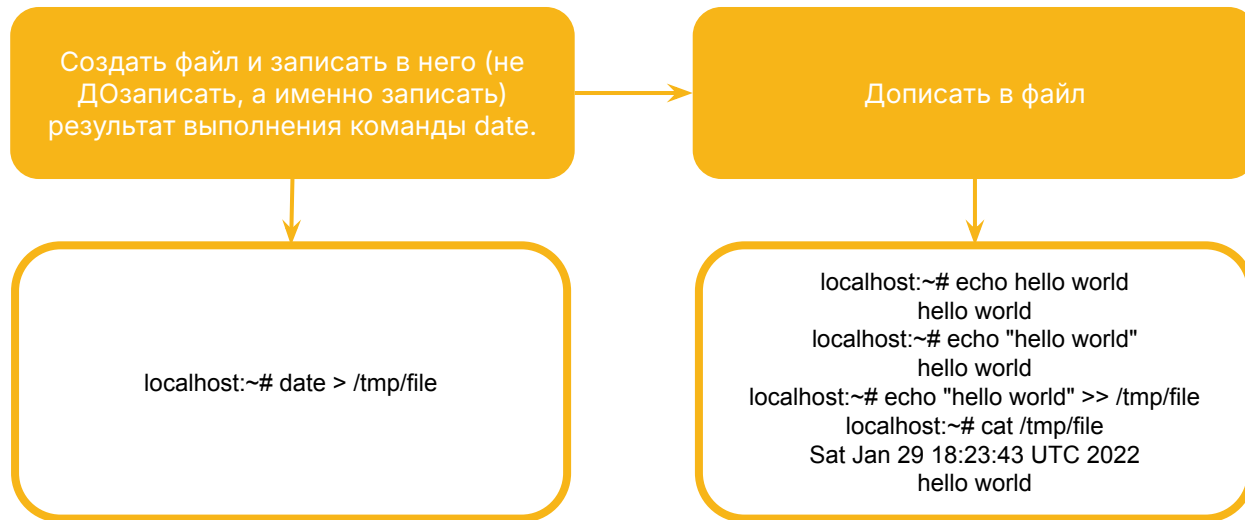


Перенаправление

> - записать в файл.

Можно записать в файл и то, что видим в файле и результат выполнения команды.

Пример:



Двойное перенаправление

Если записать в файл, и не использовать двойное перенаправление, то можно стереть все данные в файле, перезаписав его.

Всегда следите, когда дозаписываете.

>

Save output to a file.

>>

Append output to a file.



Экспресс-опрос

- **Вопрос 1.**

Чем отличается запись от дозаписи?

- **Вопрос 2.**

Какие бывают потоки ввода-вывода данных?



Tail and head

Команда	Пояснение
cat /etc/group	Выводит на экран все, что есть в этом файле
cat /etc/group wc -l	Посчитает количество строк в этом файле. Тут воспользуемся pipe, который перенаправит результат вывода всех данных в файле на новую команду, которая посчитает количество всех строк.
cat /etc/group wc	И результат мы передаем другой команде через pipe Где wc -l считает количество строк
wc	Покажет количество строк, слов и количество символов word count количество строчек. И с ключом -l оно просто покажет количество строк.

Tail and head

Команда	Пояснение
cat /etc/group head	Выведет на экран первые 10 строк. По умолчанию - 10 строк, количество можно менять через ключ -n
cat /etc/group tail	tail также по умолчанию показывает 10 строк, но из конца файла
history tail -3	Покажет последние три команды из истории.
History tail -3 >> /tmp/file	Дозаписали в файл эти три строки из истории, используя >>

Перенаправление можно использовать и в обратном направлении, но используют первый вариант.
wc -l < /etc/group

Grep

- Grep - это программа поиска и фильтрации.
Например, мы можем показать строки в `/etc/group`, в которых встречается слово `root`
`cat /etc/group | grep root`
- По умолчанию `grep` показывает и частичное совпадение слов и символов.
Для вывода определенных слов, мы можем использовать флаг `-w`, который будет искать конкретные слова.
Это будет не часть слова, а конкретное слово (слово целиком)

Grep -w

`cat /etc/group | grep -w user2`



Grep

Мы можем сортировать и фильтровать большое количество раз.



У tail есть интересный ключ -f, который поможет отслеживать в реальном времени текст в файле.

grep -v исключить строки из вывода.

df (disk full)

df - Команда, показывающая насколько используется дисковой пространство

Можно использовать с ключом `df -h` где `h` - human - человекочитаемые единицы

С помощью других команд и `pipe` можно отфильтровать результат.

Это можно сделать через `tail`, но это долгое и не оптимальное решение ,бывают случаи, когда нужно сделать быстрее.

```
df -h
```

```
df -h | head -2
```

Тут мы показываем две строки текста

```
df -h | head -1
```

Тут мы показываем только первую строку

df

Если добавить tail -1, то получим нужный результат, но это будет не совсем правильный алгоритм. Давайте поэкспериментируем:

```
df -h | head -2 | tail -1
```

Тут мы показываем все строчки, в которых есть символ /

```
df -h | grep /
```

Почему лучше использовать grep?

- Вариант с ним отработает практически на любой системе потому что мы ищем инфо о занятом месте в корневом разделе файловой системы.
- При grep меняется логика фильтрации. Он показывает сразу нужную строку.

AWK

- AWK - утилита для обработки данных, иногда ее называют языком программирования в Linux.
- Отсортируем строки, где будет только / и в этом нам поможет ключ -w (word)
- **df -h | grep -w /**
- Отсечем все лишнее, чтобы оставить % занятого места на корневом разделе.
- С этим нам поможет awk.
- **awk** - это утилита/язык для извлечения данных.
- Одним из самых простых и популярных способов использования awk является выбор столбца из текстового файла или из вывода другой команды.
- awk помогает относиться к тексту, как к таблице
- Отфильтруем так, чтобы он показал нам пятый столбец.

df -h | awk '{print \$5}'

Нам показало столбец.



sed

Sed - stream editor.

SED (Stream Editor) - это простая, но мощная утилита, которая анализирует текст и плавно преобразует его.

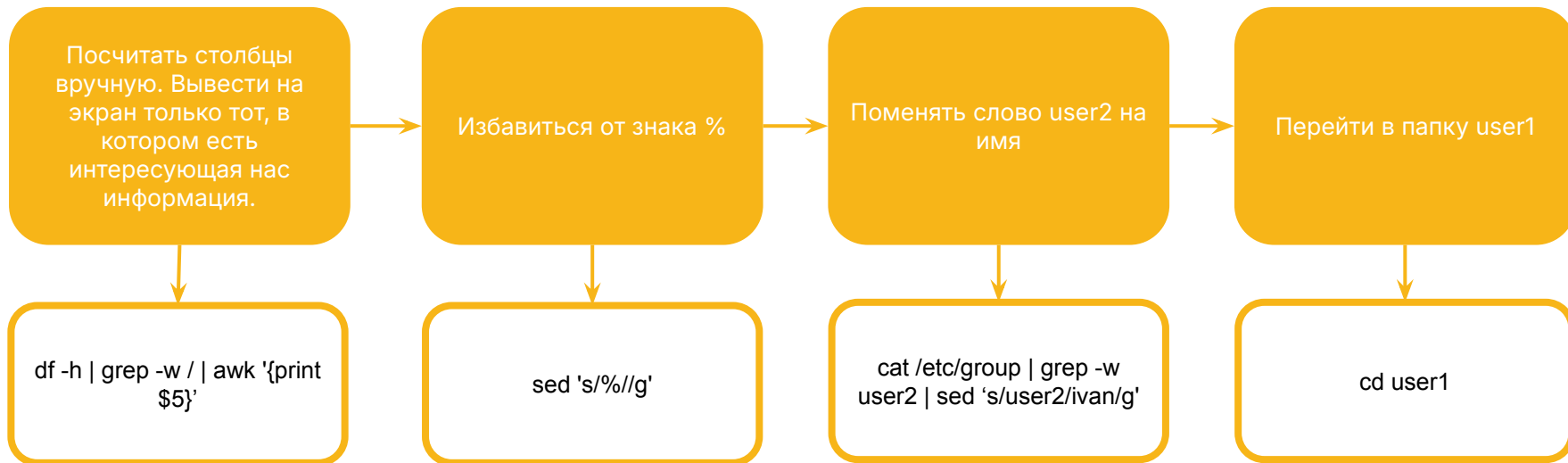
SED можно использовать по-разному, например:

- Подстановка текста,
- Выборочная печать текстовых файлов,
- Редактирование текстовых файлов на месте,
- Не интерактивное редактирование текстовых файлов и многое другое.



sed

Как показать 47%?



*каждый столбец считается до пробела справа.

Экспресс-опрос

- **Вопрос 1.**

В чем отличие AWK и sed?

- **Вопрос 2.**

Использование sed для замены символов.



ЗАДАНИЕ

Найдите **НЕ**верное утверждение:
Ответ напишите в чат.

1. Третий поток - это поток данных, которые оболочка выводит после выполнения каких-то действий.
2. Grep - это функция поиска по символам.
3. awk - Это простая, но мощная утилита, которая анализирует текст и преобразует его.
4. df - Команда, показывающая насколько занят диск.

Практическое задание

Условие задания:

1. Создайте файл с названием myfile.txt по пути /root/test3/
2. Запишите в этот файл вывод команды df -h
3. Допишите в файл 9 первых строк из списка всех объектов корня
4. Допишите в файл количество символов из трех последних строк файла /etc/group



Реализация задания

```
mkdir /root/test3  
df -h > /root/test3/myfile.txt  
ls -l / | head -9 >> /root/test3/myfile.txt  
cat /etc/group | tail -3 | wc -m
```


Домашнее задание

1. Создать текстовый файл `/home/user/start/1.txt` , создав перед этим необходимую директорию.
2. Добавьте слово `Start` в текстовый документ, созданный ранее.
3. Допisać в конец документа текущую дату, используя `date` (запуск командой `date`).
4. Скопировать получившийся документ в `/opt/task/start.txt` (Предварительно создаем необходимую директорию)
5. В документ `/opt/task/start.txt` дописать последние 4 строки из списка `VCESX` (в том числе скрытых) файлов, лежащих в `/` (корне).



Полезные ссылки

- [Команда awk – примеры использования в Linux и Unix / Хабр](#)
- [Команда SED в Linux/Unix с примерами / Хабр](#)