3-й блок

Простейший вариант

Баженов Тимур

Содержание

Список иллюстраций

Список таблиц

# Итоговое занятие курса: Важные дополнения и полезные ресурсы по Linux и Bash

На завершающем занятии курса мы рассмотрим несколько важных тем, которые не были подробно охвачены в предыдущих уроках. Это позволит вам расширить понимание работы с Linux, повысить эффективность работы в терминале и научиться использовать продвинутые возможности Bash и сопутствующих инструментов.

## 1. Права доступа в Linux

### 1.1 Основные понятия

В Linux каждый файл и каталог имеют три категории прав доступа, определяющие, кто и как может с ними взаимодействовать:

* **Владелец (User)** — пользователь, который создал файл или назначен владельцем.
* **Группа (Group)** — группа пользователей, к которой относится владелец.
* **Прочие (Others)** — все остальные пользователи системы.

### 1.2 Типы прав

| Символ | Значение | Что позволяет делать |
| --- | --- | --- |
| r | Чтение (read) | Читать содержимое файла или список файлов в каталоге |
| w | Запись (write) | Изменять файл или создавать/удалять файлы в каталоге |
| x | Выполнение (execute) | Запускать файл как программу или переходить в каталог |

### 1.3 Просмотр прав доступа

Команда:

```bash ls -l filename Вывод:

-rwxr-xr– 1 user group 1234 May 24 15:30 filename rwx — права владельца (чтение, запись, выполнение) r-x — права группы (чтение и выполнение) r– — права остальных (только чтение) 1.4 Изменение прав доступа Команда chmod позволяет менять права доступа.

Примеры:

Добавить право на выполнение владельцу: chmod u+x filename Убрать право на запись у группы: chmod g-w filename Установить права доступа числовым способом (rwxr-xr– = 754): chmod 754 filename Разъяснение числового способа

Каждое право кодируется числом:

r = 4 w = 2 x = 1 Складывая их, получаем число для каждой категории пользователей:

Права Число rwx 7 (4+2+1) r-x 5 (4+0+1) r– 4 (4+0+0) 1.5 Смена владельца и группы Изменить владельца: sudo chown newowner filename Изменить группу: sudo chgrp newgroup filename Изменить владельца и группу одновременно: sudo chown newowner:newgroup filename 2. Новые команды терминала

2.1 Архивирование с помощью tar tar — одна из самых популярных утилит для создания архивов и их распаковки.

Создать архив из каталога: tar -cvf archive.tar folder/ Распаковать архив: tar -xvf archive.tar Создать архив с сжатием gzip: tar -czvf archive.tar.gz folder/ Распаковать gzip-архив: tar -xzvf archive.tar.gz 2.2 Загрузка и работа с интернетом — curl curl — инструмент для передачи данных по различным протоколам (HTTP, FTP и др.).

Загрузить содержимое страницы: curl https://example.com Скачать файл в текущую папку: curl -O https://example.com/file.zip Отправить POST-запрос с данными формы: curl -d “param1=value1&param2=value2” -X POST https://example.com/api Использовать заголовки HTTP: curl -H “Authorization: Bearer TOKEN” https://api.example.com/data 2.3 Мониторинг процессов — htop htop — интерактивный мониторинг процессов, более удобный, чем стандартный top.

Установка: sudo apt-get install htop Запуск: htop В htop можно сортировать процессы, завершать их, фильтровать и многое другое.

2.4 Автоматический повтор команд — watch Команда watch позволяет периодически выполнять любую команду и обновлять её вывод.

Пример — каждые 2 секунды показывать список файлов в каталоге: watch -n 2 ls -l Отслеживание использования памяти: watch free -m 3. Расширенные возможности известных команд

3.1 grep с регулярными выражениями grep умеет использовать сложные шаблоны для поиска текста.

Поиск слова в файле: grep “error” logfile.txt Игнорирование регистра: grep -i “error” logfile.txt Поиск с использованием регулярных выражений: grep -E “error|warning” logfile.txt Показать номер строки: grep -n “error” logfile.txt 3.2 find с выполнением команд find позволяет искать файлы и выполнять над ними действия.

Найти все .txt файлы и вывести их имена: find . -name “*.txt” Найти и удалить все .tmp файлы: find . -name ”*.tmp” -exec rm {} ; Найти файлы, изменённые за последние 7 дней: find . -mtime -7 3.3 sed — потоковый редактор текста Замена всех вхождений слова foo на bar в файле: sed ‘s/foo/bar/g’ input.txt > output.txt Замена на месте (без создания нового файла): sed -i ‘s/foo/bar/g’ input.txt Удаление строк, содержащих слово error: sed ‘/error/d’ input.txt 3.4 awk — мощный инструмент для обработки текстовых данных Вывод второго столбца из файла: awk ‘{print $2}’ file.txt Суммирование чисел из третьего столбца: awk ‘{sum += $3} END {print sum}’ file.txt Вывод строк, где значение в первом столбце больше 100: awk ‘$1 > 100’ file.txt 4. Дополнительные темы и советы

4.1 Bash функции Функции в Bash позволяют структурировать скрипты и переиспользовать код.

function greet() { echo “Привет, $1!” }

greet “Мир” 4.2 Запуск внешних программ из скриптов В Bash скриптах можно запускать любые внешние программы и обрабатывать их результаты:

output=$(ls -l)
echo "Список файлов:"
echo "$output” 4.3 Арифметические операции в Bash Bash умеет выполнять базовые арифметические операции:

a=5 b=3 sum=$((a + b)) echo “Сумма: $sum” Поддерживаются операции +, -, \*, /, % и др.