Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Лабораторная работа №13

Козомазов Владимир Романович

10 мая 2025

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Козомазов Владимир Романович
- Студент факультета ФМЕН
- Российский университет дружбы народов

Вводная часть

Актуальность

1. Основы автоматизации в UNIX/Linux

- Командные процессоры (`sh`, `bash`, `zsh` и др.) — ключевой инструмент администрирования и разработки в UNIX-подобных системах.
- Скрипты с ветвлениями (`if-else`, `case`) и циклами (`for`, `while`) позволяют автоматизировать рутинные задачи (например, обработку файлов, мониторинг системы).

2. Широкое применение в DevOps и администрировании

- Современные DevOps-инструменты (Docker, Kubernetes, CI/CD) часто используют shell-скрипты для настройки окружения.
- Знание условий и циклов необходимо для написания надежных скриптов развертывания и обслуживания.

3. Переносимость и скорость

Объект и предмет исследования

Объект исследования

Объектом исследования являются:

- **Командные процессоры UNIX** (например, `sh`, `bash`, `zsh`, `ksh`).
- **Механизмы управления потоком выполнения** (ветвления и циклы) в shell-скриптах.
- **Синтаксис и семантика** конструкций ветвления (`if-else`, `case`) и циклов (`for`, `while`, `until`).

Предмет исследования

1. **Принципы работы ветвлений и циклов** в shell-скриптах:
   - Условные операторы (`if`, `elif`, `else`, `case`).
   - Циклические конструкции (`for`, `while`, `until`).
   - Управление выполнением (`break`, `continue`, `exit`).
2. **Особенности реализации** в разных командных оболочках (Bash vs POSIX `sh`).
3. **Практическое применение** в автоматизации задач:
   - Обработка файлов и данных

Цели и задачи

Цели исследования

1. **Изучить основы программирования** в командных процессорах UNIX (`sh`, `bash`, `zsh`).
2. **Разобрать синтаксис и принципы работы** ветвлений (`if-else`, `case`) и циклов (`for`, `while`, `until`).
3. **Научиться применять управляющие конструкции** для автоматизации задач в UNIX-системах.
4. **Оптимизировать и отлаживать** shell-скрипты с использованием ветвлений и циклов.

Задачи исследования

1. **Теоретические задачи:**
   - Изучить синтаксис условных операторов (`if`, `elif`, `else`, `case`) в shell-скриптах.
   - Разобрать работу циклов (`for`, `while`, `until`) и управляющих команд (`break`, `continue`, `exit`).
   - Сравнить особенности реализации конструкций в разных командных оболочках (Bash, POSIX `sh`, `zsh`).
2. **Практические задачи:**
   - Написать скрипты с использованием ветвлений для проверки условий (например, существования файла, прав доступа).
   - Реализовать циклы для обработки файлов, каталогов и текстовых данных (например, поиск, фильтрация, модификация).
   - Проанализировать эффективность разных подходов к организации циклов и условий.
3. **Прикладные задачи:**
   - Автоматизировать типовые задачи администрирования (очистка логов, мониторинг процессов, резервное копирование).
   - Интегрировать shell-скрипты с другими утилитами UNIX (`grep`, `sed`, `awk`, `find`).

Установка менеджера паролей pass

- Установка менеджера паролей `sudo dnf install pass pass-otp`, `sudo dnf install gopass`

Настройка

- Инициализация хранилища, командой `pass init <gpg-id>`
- Создание структуры git `pass git init`
- Синхронизация `pass git pull`, `pass git push`
- Проверка статуса синхронизации `pass git status`

Сохранение пароля

- Добавил новый пароль `pass insert [OPTIONAL DIR]/[FILENAME]`
- Отобразил пароль для указанного имени файла `pass [OPTIONAL DIR]/[FILENAME]`
- Заменил существующий пароль `pass generate --in-place FILENAME`

Дополнительное программное обеспечение

Установил дополнительное программное обеспечение: sudo dnf -y install

dunst

fontawesome-fonts

powerline-fonts

light

fuzzel

swaylock

kitty

waybar swaybg

wl-clipboard

mpv

grim

slurp

Дополнительное программное обеспечение

Установил шрифты: `sudo dnf copr enable peterwu/iosevka` `sudo dnf search iosevka` `sudo dnf install iosevka-fonts iosevka-aile-fonts iosevka-curly-fonts iosevka-slab-fonts iosevka-etoile-fonts iosevka-term-fonts`

Установка

- Установка бинарного файла `sh -c "$(wget -qO- chezmoi.io/get)"`

Создание собственного репозитория с помощью утилит

Создал репозиторий на основе шаблона

Подключение репозитория к своей системе

- Инициализировал chezmoi с моим репозиторием `chezmoi init git@github.com:<username>/dotfiles.git`
- Проверил внесённые изменения `chezmoi diff`

Ежедневные операции с chezmoi

- Включил функцию автоматической фиксации и отправке изменений в репозиторий: [git] autoCommit = true autoPush = true

Результаты

Вывод

Исследование позволяет:

Научиться писать эффективные скрипты для UNIX/Linux.

Автоматизировать рутинные задачи системного администрирования.

Глубже понять работу командного процессора и его возможности.

Эти результаты полезны для **администраторов, DevOps-инженеров и разработчиков**, работающих в UNIX-окружении.