

# Программирование в командном процессоре ОС UNIX. Расширенное программирование

Лабораторная работа №14

---

Козомазов Владимир Романович

12 мая 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

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



## Вводная часть

---

Тема расширенного программирования в UNIX-shell сохраняет высокую актуальность в 2024 году, несмотря на развитие современных языков и инструментов. Рассмотрим ключевые аспекты ее востребованности:

# 1. Фундаментальная значимость для IT-инфраструктур

- **Поддержка legacy-систем:**

Многие критически важные системы (банковские, промышленные, телеком) продолжают работать на UNIX-платформах, требуя глубокого знания shell-скриптинга для их обслуживания.

- **Базовый слой автоматизации:**

Даже в облачных средах (AWS, GCP) shell-скрипты остаются основой для начальной настройки инстансов через user-data и cloud-init.

## 2. Ограничения и пути их преодоления

- **Проблема:** Сложность отладки многофайловых проектов.  
**Решение:** Интеграция с VS Code (расширение Bash Debug).
- **Проблема:** Низкая производительность в числовых расчетах.  
**Решение:** Вынос критичных участков в C/Rust через **FFI**.

Объектом исследования являются:

### 1. Командные процессоры UNIX/Linux:

- Расширенные возможности Bash (версии 5.0+), Zsh, KornShell
- POSIX-совместимые оболочки (dash, ash)

### 2. Продвинутое механизмы shell-программирования:

- Работа с ассоциативными массивами
- Регулярные выражения в условиях
- Параллельное выполнение команд

### 3. Интеграционные возможности:

- Взаимодействие с API (curl, jq)
- Связь с другими языками (Python, Perl) через pipe



Предмет включает изучение:

## 1. Углубленные языковые конструкции

- **Функции:**
  - Рекурсивные вызовы
  - Возврат значений через stdout
  - Локальные переменные (local)
- **Строковые операции:**
  - Подстановки `${var%.*}`
  - Регулярные выражения в `[[ =~ ]]`
- **Арифметика:**
  - Числовые массивы
  - Побитовые операции

1. Освоить продвинутые техники написания shell-скриптов.
2. Научиться эффективно использовать встроенные возможности командных процессоров (Bash, sh, ksh, zsh и др.).
3. Изучить взаимодействие с системными вызовами и утилитами UNIX.
4. Развить навыки автоматизации задач администрирования и обработки данных.

### Материалы

- **“Advanced Bash-Scripting Guide”** – Mendel Cooper  
(Подробное руководство по Bash, включая сложные примеры.)
- **“Unix Shell Programming”** – Stephen G. Kochan, Patrick Wood  
(Классика по shell-программированию.)
- **“Learning the bash Shell”** – Cameron Newham  
(Практическое руководство по Bash для продвинутых пользователей.)

## Методы

- Объяснение синтаксиса, особенностей shell.
- Разбор сложных конструкций (`trap`, `eval`, `exec`, подстановки).
- Анализ готовых скриптов (системные утилиты, автоматизация).
- **Написание скриптов** (от простых до сложных, например: бэкап, мониторинг, парсинг логов).
- Разбор ошибок (отладка с `set -x`, `bash -n`, `shellcheck`).
- Оптимизация (сравнение `grep/awk/sed`, использование `xargs` и параллельных процессов).

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

- 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`

- Добавил новый пароль `pass insert [OPTIONAL DIR]/[FILENAME]`
- Отобразил пароль для указанного имени файла `pass [OPTIONAL DIR]/[FILENAME]`
- Заменял 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`



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

Создал репозиторий на основе шаблона

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

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

## Результаты

---

В ходе выполнения лабораторной работы №14 были получены не только теоретические знания, но и **практические навыки**, позволяющие эффективно работать в UNIX-средах, сокращая время на рутинные операции за счёт автоматизации. Владение расширенным shell-программированием делает специалиста более универсальным и востребованным в IT-индустрии.