# Настройка csh. Специфика по сравнению с bash.

## Терминал и командная строка

Карпова Есения Алексеевна

## Содержание



# 1 Цель работы

Цель данного доклада заключается в изучении особенностей настройки оболочки C Shell (csh) и сравнении ее с оболочкой Bourne-Again Shell (bash).

# 2 Задания

1. Представить исторический контекст и основные особенности оболочек csh и bash
2. Рассмотреть специфику настройки csh
3. Проанализировать основные отличия в синтаксисе и функциональности между csh и bash.
4. Подчеркнуть преимущества и недостатки каждой из оболочек при выполнении различных задач.
5. Предложить рекомендации по выбору оболочки в зависимости от конкретных потребностей пользователя или сценария использования.

# 3 Основная часть

## 3.1 Исторический контекст и основные особенности оболочек csh и bash

### 3.1.1 C Shell (csh)

C Shell была разработана Биллом Джоем и впервые появилась в 1970-х годах. Она имеет синтаксис, аналогичный языку программирования C

Уникальные особенности:

- Разработана на основе синтаксиса языка программирования C
- Известна своими интерактивными возможностями, такими как автозавершение, буфер истории, инструменты навигации и редактирования командной строки
- Предлагает возможности настройки подсказок, интерактивных и неинтерактивных параметров оболочки
- Изначально из-за своего уникального синтаксиса столкнулся с ограничениями в совместимости с другими оболочками, такими как sh
- Несмотря на свои ограничения, csh повлиял на развитие современных функций оболочки, присутствующих в таких оболочках, как bash и zsh

### 3.1.2 Bourne-Again Shell (bash):

Это один из наиболее распространенных командных процессоров в Unix-подобных системах. Bash была заново изобретена Брайаном Фоксом в 1989 году. Она похожа на оболочку Bourne и включает в себя некоторые функции csh и Korn Shell (KSH), а также имеет свои уникальные функции

Уникальные особенности:

- Производная от оболочки Bourne (sh), разработанная Брайаном Фоксом в 1989 году
- Широко используется в системах Linux и известен своей обширной базой пользователей и универсальностью
- Поддерживает огромное количество пользователей и является оболочкой по умолчанию во многих дистрибутивах Linux
- Предлагает богатый набор функций, включая выполнение команд, сценариев и работу с файлами
- Обеспечивает надежный язык сценариев и хорошо настраивается с помощью различных конфигураций и плагинов

### 3.1.3 Сравнение

Таким образом, C Shell (csh) и Bourne-Again Shell (bash) представляют собой различные командные оболочки с уникальными особенностями, синтаксисом и функциональностью, что делает их подходящими для различных потребностей пользователей в работе с командами операционной системы: C Shell ориентирована на синтаксис языка программирования C, в то время как Bash, будучи обновленной версией оболочки Bourne, включает в себя множество функций, делая ее более мощной и широко используемой в сравнении с C Shell.

## 3.2 Специфика настройки csh

Специфика настройки csh (C Shell) включает в себя ряд конфигурационных файлов, которые диктуют поведение оболочки при запуске и выходе. В процессе запуска csh в качестве оболочки входа пользователя в систему в определенном порядке происходит обращение к следующим конфигурационным файлам:

```
/etc/csh.cshrc
/etc/csh.login,
~/.tcshrc (или ~/.cshrc)
~/.history
~/.login
~/.cshdirs.
```

Эти файлы определяют поведение оболочки и считываются в определенной последовательности

При выходе из сеанса оболочки для входа в систему сначала считывается файл ~/.logout пользователя, если он существует, а затем - общесистемный файл /etc/csh.logout. Кроме того, при запуске оболочки считываются два ключевых конфигурационных файла: /etc/csh.cshrc считывается каждый раз при запуске интерактивного экземпляра csh, а /etc/csh.login считывается только при запуске csh в качестве оболочки для входа в систему

Такой структурированный подход к работе с конфигурационными файлами в csh обеспечивает последовательное применение определенных настроек и поведения при запуске и завершении работы оболочки, предоставляя пользователям предсказуемую среду для решения задач администрирования Unix-систем.

Также оболочка csh имеет несколько особенностей, вот некоторые из них:

1. Настройка переменных окружения.

В оболочке csh можно настроить переменные окружения, которые влияют на работу системы. Например, переменная PATH определяет путь к исполняемым файлам, а переменная EDITOR определяет редактор по умолчанию.

2. Настройка автозагрузки.

В оболочке csh можно настроить автозагрузку команд и скриптов при входе в систему. Это делается с помощью файла .cshrc.

3. Настройка истории команд.

В оболочке csh можно настроить историю команд, чтобы иметь возможность повторно использовать ранее введенные команды. Это делается с помощью файла .history.

4. Настройка цветов.

В оболочке csh можно настроить цвета, чтобы сделать работу более удобной и наглядной. Например, можно задать разные цвета для команд, ошибок и вывода.

5. Настройка автозавершения.

В оболочке csh можно настроить автозавершение команд и переменных. Это делается с помощью файла .cshrc.

6. Настройка перенаправления вывода.

В оболочке csh можно настроить перенаправление вывода команд в файлы или устройства. Это делается с помощью символов >, >> и |.

7. Настройка скриптов.

В оболочке csh можно настроить скрипты, которые будут выполняться автоматически при определенных условиях. Это делается с помощью файлов .cshrc и .login.

8. Настройка профиля пользователя.

В оболочке csh можно настроить профиль пользователя, который определяет настройки и параметры для всех пользователей системы. Это делается с помощью файла /etc/csh.cshrc.

9. Настройка прав доступа.

В оболочке csh можно настроить права доступа к файлам и каталогам, чтобы защитить их от несанкционированного доступа. Это делается с помощью команды chmod.

10. Настройка синтаксиса.

В оболочке csh можно настроить синтаксис команд и скриптов, чтобы сделать их более удобными для использования. Это делается с помощью файла .cshrc.

## 3.3 Основные отличия в синтаксисе и функциональности между csh и bash

### 3.3.1 Синтаксис:

CSH (C-подобная оболочка):

- Синтаксис основан на языке программирования C
- Команды начинаются с символа ‘#’
- Интерактивный терминал

BASH (Bourne Again Shell):

- Синтаксис основан на оболочке Bourne
- Команды начинаются с символа ‘$’
- Неинтерактивный терминал

### 3.3.2 Функциональность:

CSH:

- Поддерживает управление заданиями.
- Менее популярен и используется реже, чем BASH.
- Интерпретирует внешние команды.

BASH:

- Более широко используется профессионалами.
- Быстрее, чем C shell.
- Включает уникальные и расширенные функции, такие как командное редактирование, расширенное глобирование и улучшенное управление сигналами.

## 3.4 Преимущества и недостатки каждой из оболочек при выполнении различных задач.

### 3.4.1 Bash

Преимущества:

- Мощный скриптовый язык: Bash обладает широким набором встроенных команд, структур управления и функций, что делает его мощным инструментом для автоматизации и администрирования систем
- Большое сообщество пользователей: Bash имеет большое сообщество пользователей и обширную поддержку, что облегчает получение помощи и решение проблем
- Широкая доступность: Bash широко распространен и является стандартной оболочкой на большинстве дистрибутивов Linux

Недостатки:

- Синтаксис: Некоторые пользователи могут считать синтаксис Bash менее интуитивным по сравнению с другими оболочками
- Не всегда подходит для сложных задач: Для некоторых продвинутых задач Bash может оказаться менее подходящим из-за ограничений в сравнении с другими оболочками

### 3.4.2 Csh:

Преимущества:

- Интерактивность: Csh предлагает интерактивное окружение с возможностью выполнения команд и настройки пользовательской среды
- Простота использования: Csh известен своей простотой и удобством в использовании, что делает его предпочтительным для некоторых разработчиков

Недостатки:

- Ограниченные возможности скриптования: Csh не обладает таким богатым набором функций для скриптования, как Bash, что может быть недостатком при выполнении сложных автоматизированных задач
- Меньшая популярность: Csh менее популярен среди профессиональных пользователей по сравнению с Bash

### 3.5 Рекомендации по выбору оболочки в зависимости от конкретных потребностей пользователя или сценария использования.

В итоге, если Bash предлагает мощные возможности автоматизации, универсальность в написании сценариев и широко распространен среди профессионалов, то Csh обеспечивает знакомый синтаксис для тех, кто имеет опыт программирования на C, но не имеет такой популярности и широких возможностей, как Bash.

В конечном счете выбор между Bash и Csh зависит от конкретных требований к решаемым задачам и знакомства пользователя с каждой оболочкой: если необходим мощный скриптовый язык и широкая поддержка, то предпочтительнее использовать Bash; если же ценится простота использования и интерактивность, то Csh может быть более удобным выбором.

## 4 Вывод

В ходе доклада были рассмотрены основные особенности настройки оболочек C Shell (csh) и Bourne-Again Shell (bash) с акцентом на специфику csh. Изучение и понимание особенностей настройки csh позволяет пользователям эффективно использовать эту оболочку, учитывая ее уникальные возможности и функциональность.

Данный доклад предоставил обзор ключевых аспектов настройки csh и подчеркнул важность выбора оболочки в зависимости от потребностей пользователя. Понимание особенностей csh и умение правильно настраивать ее поможет повысить эффективность работы в Unix-подобных системах и улучшить пользовательский опыт.

## 5 Список литературы


1. Wikipedia
2. Linux.org.ru
3. Bestprogrammer.ru
4. Habr
5. Eternalhost