**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ**

**И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

(СПбГУТ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ (**ИТПИ**)

КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (ПИ И ВТ)

**Проект**

по дисциплине «Операционные системы и сети»

Тема «Сервис для операционной системы Linux»

Выполнили:

Студенты группы ИКПИ-23

Даненко Д. А.

Харлова А. А.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял:

Преподаватель

Дагаев А. В.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024 г.

# 

**Оглавление**

[Введение 1](#_Toc27137)

[Постановка задачи 1](#_Toc17390)

[Описание внутренней логики 1](#_Toc11384)

[6.1. Ожидание команд 3](#_Toc15827)

[6.2. Обработка команд 3](#_Toc24333)

[6.3. Очистка файла команд 3](#_Toc6388)

[6.4. Бесконечный цикл 3](#_Toc31275)

[Заключение 5](#_Toc12974)

# **Введение**

В современных операционных системах управление пользователями и системными ресурсами является важной задачей для обеспечения безопасности и стабильности работы. Автоматизация таких процессов позволяет администраторам эффективно управлять системой, минимизировать человеческий фактор и упростить выполнение повседневных задач. В данной работе представлен проект сервиса, который позволяет автоматизировать управление пользователями и очистку системных ресурсов.

# **Постановка задачи**

В рамках данного проекта необходимо разработать сервис для управления пользователями и системными ресурсами в операционной системе *Linux*. Сервис должен предоставлять следующие функции:

* Добавление и удаление пользователей.
* Изменение прав доступа пользователей.
* Очистка кэша пользователей.
* Остановка приложений, использующих большой объём трафика.

Сервис будет реализован на языке *Bash* и интегрирован с системой инициализации *systemd* в *Linux*.

# **Описание внутренней логики**

Сервис реализован в виде *Bash*-скрипта, который выполняет следующие функции:

1. Добавление пользователя (*add\_user*)

Описание:

Создаёт нового пользователя в системе.

Логика:

* Принимает имя пользователя (*username*) в качестве аргумента.
* Использует команду *sudo* *useradd* *-m* для создания пользователя с домашней директорией.
* Записывает результат операции в лог-файл */var/log/user\_management.log*.

*Пример*:

add\_user "new\_user"

1. Удаление пользователя (*remove\_user*)

Описание:

Удаляет существующего пользователя.

Логика:

* Принимает имя пользователя (*username*) в качестве аргумента.
* Использует команду *sudo userdel -r* для удаления пользователя вместе с его домашней директорией.
* Записывает результат операции в лог-файл */var/log/user\_management.log*.

Пример:

*remove\_user "old\_user"*

1. Изменение прав доступа (*change\_permissions*)

Описание:

Изменяет права доступа пользователя, добавляя его в указанную группу.

Логика:

* Принимает имя пользователя (*username*) и группу (*permissions*) в качестве аргументов.
* Использует команду *sudo usermod -aG* для добавления пользователя в указанную группу.
* Записывает результат операции в лог-файл */var/log/user\_management.log*.

Пример:

*change\_permissions "new\_user" "sudo"*

1. Очистка кэша (*clear\_cache*)

Описание:

Очищает кэш пользователя, удаляя содержимое директории *~/.cache/*.

Логика:

* Принимает имя пользователя (*username*) в качестве аргумента.
* Определяет домашнюю директорию пользователя с помощью команды *eval echo "~$username"*.
* Проверяет существование директории *~/.cache/*.
* Использует команду *sudo rm -rf* для удаления содержимого директории.
* Записывает результат операции в лог-файл */var/log/user\_management.log*.

Пример:

*clear\_cache "new\_user"*

1. Остановка приложения по трафику (*stop\_app\_by\_traffic*)

Описание:

Останавливает приложение, использующее большой объём трафика.

Логика:

* Принимает пороговое значение трафика (*threshold*) в качестве аргумента.
* Использует команду *sudo netstat -tunp* для анализа сетевого трафика.
* Фильтрует приложения, использующие трафик выше указанного порога.
* Останавливает приложение с помощью команды *sudo kill -9*.
* Записывает результат операции в лог-файл */var/log/user\_management.log*.

Пример:

*stop\_app\_by\_traffic "1000"*

1. Главный цикл демона (*main*)

Главный цикл обеспечивает работу демона в режиме ожидания и обработки команд.

6.1. Ожидание команд

Использует утилиту *inotifywait* для отслеживания изменений в файле */tmp/user\_management\_commands.txt*. Когда файл изменяется, демон начинает обработку команд.

6.2. Обработка команд

* Читает команды из файла построчно.
* Каждая команда состоит из имени функции и аргументов, разделённых пробелами.
* Вызывает соответствующую функцию в зависимости от команды.

6.3. Очистка файла команд

После обработки всех команд файл */tmp/user\_management\_commands.txt* очищается с помощью команды *> /tmp/user\_management\_commands.txt*.

6.4. Бесконечный цикл

Демон работает в бесконечном цикле, постоянно ожидая новых команд.

1. Логирование

Все операции записываются в лог-файл */var/log/user\_management.log*. Логирование позволяет отслеживать выполнение команд и диагностировать возможные ошибки.

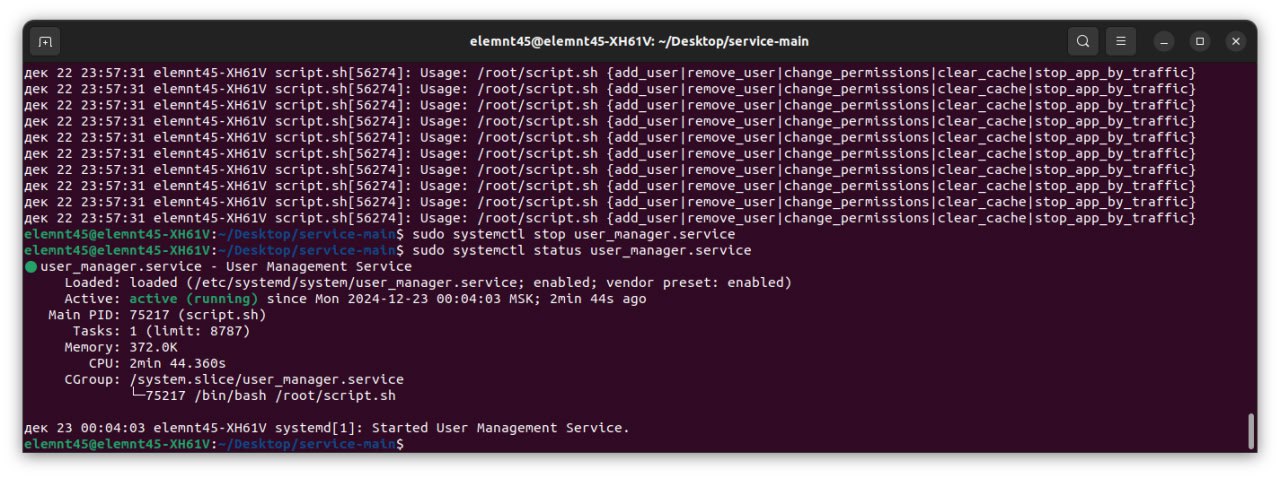
**Пример работы программы**

Тестирование разработанного сервиса для управления пользователями и системными ресурсами в *Linux* является важным этапом для обеспечения его корректной работы, стабильности и безопасности. Тестирование сервиса позволит убедиться в его готовности к использованию в реальных условиях и обеспечит надежную работу системы управления пользователями и ресурсами.

В ходе тестирования была проведена проверка работы отдельных функций и команд, реализованных в скрипте *Bash*, взаимодействия сервиса с системными инструментами и другими компонентами *Linux*, а также интеграции с *systemd* (запуск, остановка, перезапуск сервиса), с системными командами (например, *useradd*, *deluser*, *chmod*, *kill*).

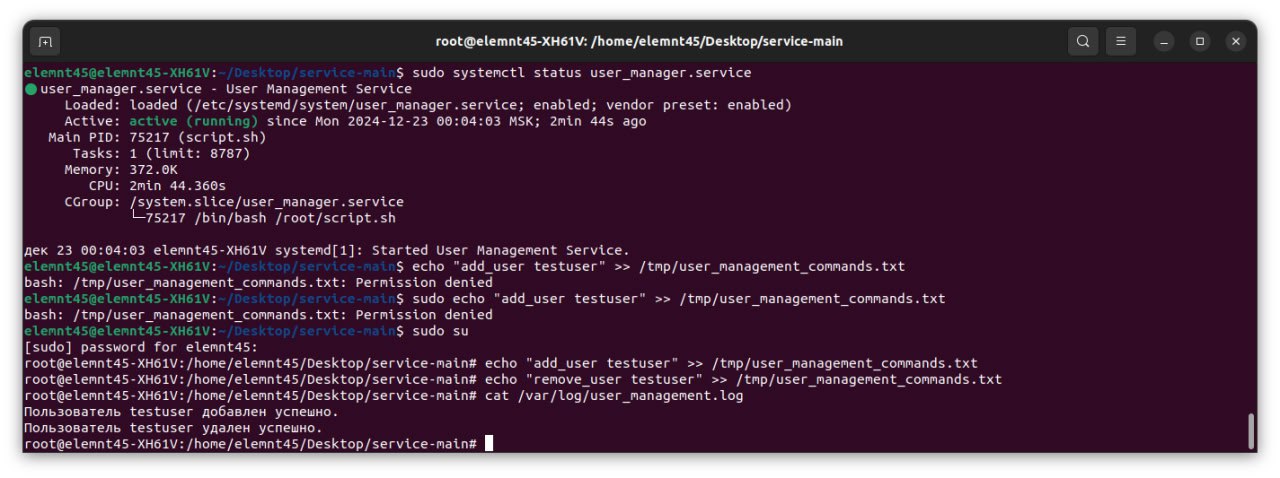
Проверка работы сервиса в реальной системной среде представлена на рисунках 1-3.

На рисунке 1 представлен статус работы сервиса, интегрированного с системой инициализации systemd. Вывод команды systemctl status отображает текущее состояние сервиса, включая информацию о том, активен ли он, запущен ли процесс, а также последние логи, связанные с его работой. Это позволяет администратору оперативно отслеживать состояние сервиса и устранять возможные проблемы.

****

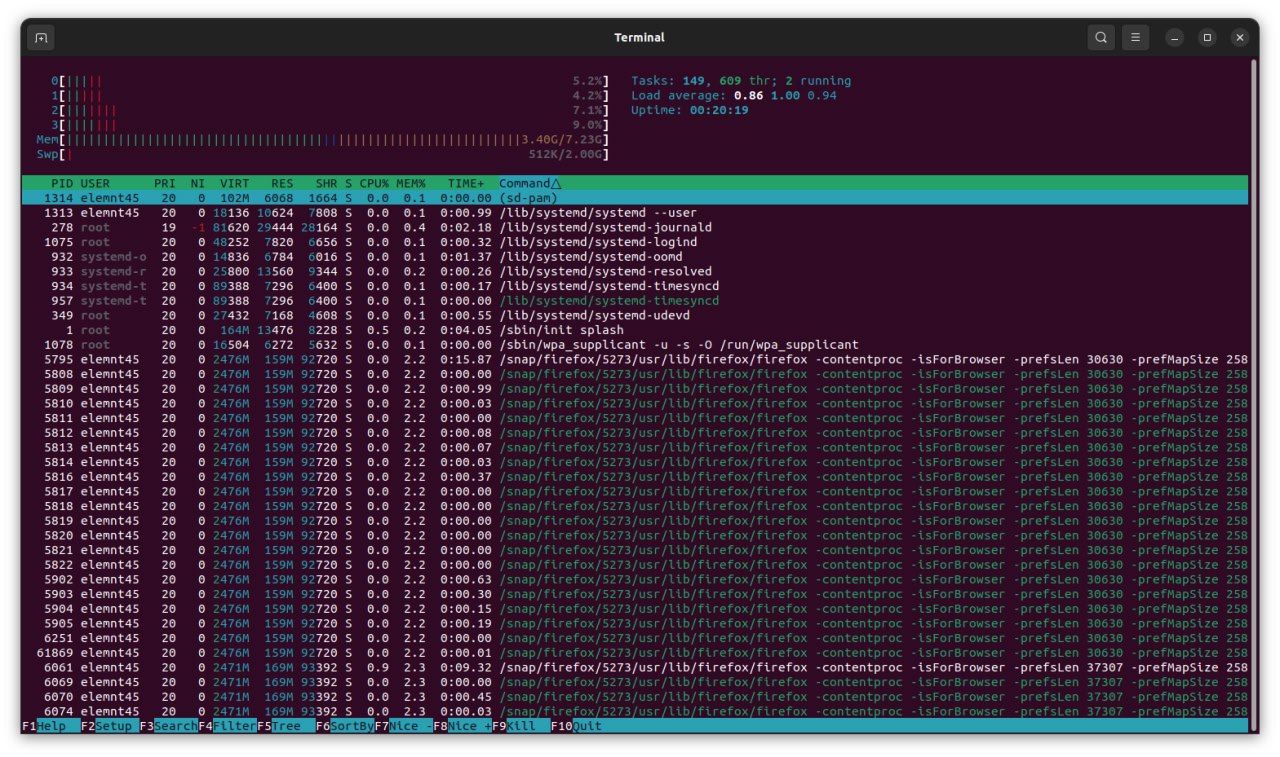
*Рисунок 1 - Статус сервиса в systemd*

На рисунке 2 показан пример выполнения команд, предоставляемых сервисом для управления пользователями и системными ресурсами. Вывод команды, выполненной через терминал, демонстрирует, как сервис обрабатывает запросы на добавление пользователя, изменение прав доступа или очистку кэша. В данном случае показан процесс добавления нового пользователя с указанием имени и пароля, а также автоматическое создание домашнего каталога.

****

*Рисунок 2 - Пример выполнения команд, предоставляемых сервисом*

На рисунке 3 представлено отображение сервиса в трее процессов операционной системы. Сервис, работающий в фоновом режиме, отображается в списке процессов, что позволяет администратору контролировать его работу, проверять потребление ресурсов (например, CPU и памяти) и при необходимости останавливать или перезапускать его. Это важный инструмент для мониторинга стабильности и производительности сервиса.

****

*Рисунок 3 - Отображение сервиса в трее процессов*

# **Заключение**

В рамках данного проекта был разработан сервис для управления пользователями и системными ресурсами в операционной системе *Linux*. Сервис предоставляет функции добавления и удаления пользователей, изменения прав доступа, очистки кэша и остановки приложений по трафику. Реализация на языке *Bash* и интеграция с *systemd* позволили создать удобный и эффективный инструмент для автоматизации администрирования.

В будущем можно расширить функциональность сервиса, добавив поддержку дополнительных операций, таких как управление группами пользователей или мониторинг системных событий.