

---

КАФЕДРА

КУРСОВОЙ ПРОЕКТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
РУКОВОДИТЕЛЬ

---

должность, уч. степень, звание

---

подпись, дата

---

инициалы, фамилия

Лабораторная работа №4.  
Применение инструментальных средств автоматизации процессов разработки  
программного обеспечения

По дисциплине: основы программной инженерии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

---

подпись, дата

---

инициалы, фамилия

Санкт-Петербург 2023

## Цель работы

Целью работы является формирование практических навыков применения инструментальных средств автоматизации процессов разработки программного обеспечения.

## Задание на лабораторную работу

Автоматизация процессов компоновки, компиляции, развертывания и доставки прикладного программного обеспечения, а также установки и конфигурирования системного программного обеспечения посредством специальных сценариев последовательного выполнения программ (команд) входящих в состав операционных систем. Выполнение лабораторной работы предполагает индивидуальную работу в соответствии с вариантом задания.

## Вариант:

<b>13. Регистрация команд, программ и служб в качестве сервиса</b>	<b>Утилита должна реализовывать следующий функционал:</b> <ul style="list-style-type: none"><li>– создание сервиса systemd на основе скрипта .sh;</li><li>– создание сервиса systemd на основе утилит, команд и служб;</li><li>– справочная информация (функция <code>usage()</code>) по использованию утилиты.</li></ul>
--	---

## Выполнение лабораторной работы

Разработать утилиту на командно-сценарном языке `bash` на базе шаблона из раздела №2 в соответствии с заданием на выполнение лабораторной работы. Финальную версию утилиты разместить по пути `/usr/local/bin` и присвоить имя, соответствующее ее функциональному назначению, а также исключить расширение `*.sh` из имени утилиты.

1. Назначение разработанной утилиты.
2. Список зависимостей утилиты и как их установить.
3. Детальное описание разработанных функций.
4. Расширенный вариант справки по утилите с описанием флагов, параметров и подпрограмм.
5. Исходный код утилиты.

## 1. Назначение разработанной утилиты

Утилита предназначена для автоматизации процесса создания и регистрации сервисов в системном менеджере **systemd**. В контексте задания и с учетом специфики варианта 13, основная цель утилиты — упростить и автоматизировать процесс создания **systemd** сервисов, что является ключевым аспектом в управлении системными и прикладными программами на серверах и в операционных системах на базе Linux.

Основные функции и их значение

1. **Создание сервиса systemd на основе скрипта .sh:** Эта функция позволяет преобразовать существующий shell-скрипт в сервис **systemd**. Это значительно упрощает процесс управления службами, обеспечивая их автоматический запуск, остановку и перезапуск. Это особенно важно для серверных приложений, где требуется надежность и автоматизация.
2. **Создание сервиса systemd на основе утилит, команд и служб:** Эта функциональность расширяет возможности управления различными процессами и службами, позволяя интегрировать их в систему управления сервисами **systemd**. Это облегчает мониторинг, управление зависимостями и автоматический запуск при загрузке системы.
3. **Предоставление справочной информации (функция usage()):** Обеспечивает пользователям необходимую информацию о том, как использовать утилиту, её опции и функции. Это упрощает процесс освоения утилиты, делая её доступной для широкого круга пользователей, включая тех, кто впервые сталкивается с автоматизацией сервисов в Linux.

## 2. Список зависимостей утилиты и как их установить

Для обеспечения корректной работы утилиты необходимо наличие определенных зависимостей в системе. Эти зависимости включают в себя программное обеспечение и библиотеки, которые утилита использует для выполнения своих функций.

Основные Зависимости

1. **Bash (Bourne Again SHell):** Это командный интерпретатор, который обеспечивает выполнение скриптов. Bash является стандартным шеллом в большинстве дистрибутивов Linux и обычно уже предустановлен.

**Проверка и установка:**

- Для проверки наличия Bash можно использовать команду **bash --version**.
  - Если Bash не установлен, его можно установить с помощью менеджера пакетов дистрибутива (например, **sudo apt-get install bash** для Debian/Ubuntu).
2. **Systemd**: Это система инициализации и менеджер системы и служб для Linux, предоставляющий возможности для управления службами операционной системы.

#### Проверка и установка:

- Проверить наличие systemd можно командой **systemctl --version**.
- Установка systemd обычно не требуется, так как она является частью большинства современных дистрибутивов Linux. В редких случаях её отсутствия, установить systemd можно через менеджер пакетов.

### 3. Детальное описание разработанных функций

В рамках утилиты были разработаны ключевые функции, каждая из которых выполняет определенную задачу в процессе создания и управления сервисами systemd. Эти функции обеспечивают гибкость и мощь утилиты, позволяя ей эффективно справляться с задачами автоматизации.

#### Описание функций

1. **Функция создания сервиса systemd из скрипта .sh (create\_systemd\_service\_from\_script):**
  - **Назначение:** Эта функция принимает в качестве аргумента путь к скрипту .sh и создает на его основе сервис systemd.
  - **Процесс работы:** Функция сначала проверяет существование указанного скрипта. Затем она создает файл сервиса systemd с соответствующими директивами, указывающими на данный скрипт. После этого функция регистрирует и активирует сервис в systemd.
  - **Особенности:** Очень важно правильно указать путь к скрипту и убедиться в его исполнимости. Также необходимо задать правильные разрешения для файла сервиса systemd.
2. **Функция создания сервиса systemd из команды или службы (create\_systemd\_service\_from\_command):**
  - **Назначение:** Эта функция позволяет создать сервис systemd на основе конкретной команды или службы.
  - **Процесс работы:** Пользователь предоставляет команду или имя службы, и функция создает файл сервиса systemd, который будет управлять этой командой или службой как сервисом.

- **Особенности:** Важно обеспечить, чтобы команда или служба были корректно определены и доступны в системе.

### 3. Функция отображения справочной информации (`show_usage_info`):

- **Назначение:** Эта функция предоставляет пользователю информацию о том, как использовать утилиту, включая описание её функций и примеры команд.
- **Процесс работы:** При вызове функция выводит на экран справочную информацию, включая синтаксис команд и описание параметров.
- **Особенности:** Эта функция является ключевой для новых пользователей, так как помогает быстро ознакомиться с функционалом утилиты.

#### Основная Логика (**main** функция)

- **Назначение:** Основная функция **main** служит точкой входа в скрипт, обрабатывая аргументы командной строки и определяя, какую функцию вызвать.
- **Реализация:** Функция анализирует первый аргумент командной строки и в зависимости от него вызывает соответствующую функцию (`create_systemd_service_from_script`, `create_systemd_service_from_command` или `show_usage_info`). Если передан неизвестный аргумент или аргументы отсутствуют, функция отображает справочную информацию.

Разработанные функции являются основой утилиты и обеспечивают её основной функционал.

## 4. Расширенный вариант справки по утилите

Предоставление детальной справки по использованию утилиты является критически важным аспектом, обеспечивающим доступность и понимание её функционала пользователями. Этот раздел содержит развернутое описание всех функций, флагов и параметров утилиты.

#### Описание Справки

##### 1. Общий синтаксис использования утилиты:

- Утилита запускается из командной строки Linux.
- Синтаксис: **my\_service\_tool** <команда> [опции]
- Где <команда> - это одна из функций утилиты (например, **create-from-script** или **create-from-command**), а [опции] - дополнительные параметры для этой команды.

## 2. Команды и Опции:

- **create-from-script** <путь\_к\_скрипту>: Создает systemd сервис из указанного shell-скрипта.
  - Пример: **my\_service\_tool create-from-script /path/to/script.sh**
- **create-from-command** <команда>: Создает systemd сервис для указанной команды или службы.
  - Пример: **my\_service\_tool create-from-command nginx**
- **help**: Отображает справочную информацию о доступных командах и опциях.
  - Пример: **my\_service\_tool help**

## 3. Примеры использования:

- Для создания сервиса из скрипта: **my\_service\_tool create-from-script /home/user/myscript.sh**
- Для создания сервиса из команды: **my\_service\_tool create-from-command apache2**
- Для просмотра справки: **my\_service\_tool help**

## Детальное описание

### 1. Основные команды:

- **create-from-script**: Создает systemd сервис из shell-скрипта.
- **create-from-command**: Создает systemd сервис на основе команды или службы.
- **help**: Отображает справочную информацию о командах и их использовании.

### 2. Флаги и параметры для create-from-script:

- <путь\_к\_скрипту> (обязательный параметр): Путь к shell-скрипту, который будет использоваться для создания сервиса.
- **-n** или **--name** (опциональный флаг): Указывает имя сервиса. Если не задано, имя будет сформировано из имени скрипта.
- **-d** или **--description** (опциональный флаг): Позволяет задать описание сервиса.
- Пример использования: **my\_service\_tool create-from-script /path/to/script.sh -n myservice -d "My Custom Service"**

### 3. Флаги и параметры для create-from-command:

- <команда> (обязательный параметр): Команда или название службы для создания systemd сервиса.
- **-n** или **--name** (опциональный флаг): Указывает имя сервиса. Если не задано, имя будет определено на основе команды.
- **-d** или **--description** (опциональный флаг): Описание создаваемого сервиса.
- Пример использования: **my\_service\_tool create-from-command nginx -n nginxservice -d "Nginx Web Server"**

### 4. Флаги и параметры для help:

- Эта команда не принимает дополнительных флагов и параметров.
- Она отображает общую справку по использованию утилиты и описание всех доступных команд и параметров.
- Пример использования: **my\_service\_tool help**

Этот раздел справки предназначен для облегчения понимания и использования утилиты пользователями. Он содержит все необходимые инструкции и примеры, которые помогают пользователям эффективно использовать утилиту для создания и управления сервисами systemd.

## 5. Исходный код утилиты.

```
#!/bin/bash
```

```
# Исходный код утилиты для создания и управления systemd сервисами
```

```
# Функция для создания systemd сервиса из скрипта .sh
```

```
create_systemd_service_from_script() {
```

```
    local script_path=$1
```

```
    local service_name=$2
```

```
    local description=$3
```

```
    if [[ ! -f "$script_path" ]]; then
```

```
        echo "Ошибка: скрипт '$script_path' не найден."
```

```
        return 1
```

```
    fi
```

```
    if [[ -z "$service_name" ]]; then
```

```
        service_name=$(basename "$script_path" .sh)
```

```
    fi
```

```
    local service_file="/etc/systemd/system/$service_name.service"
```

```
    echo "[Unit]"
```

```
Description=$description
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=/bin/bash $script_path
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target" > "$service_file"
```

```
systemctl daemon-reload
systemctl enable "$service_name"
systemctl start "$service_name"
```

```
    echo "Сервис '$service_name' успешно создан и запущен."
}
```

```
# Функция для создания systemd сервиса из команды или службы
```

```
create_systemd_service_from_command() {
```

```
    local command=$1
```

```
    local service_name=$2
```

```
    local description=$3
```

```
    if [[ -z "$service_name" ]]; then
```

```
        service_name=${command// /_}
```

```
    fi
```

```
    local service_file="/etc/systemd/system/$service_name.service"
```

```
    echo "[Unit]
```

```
Description=$description
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=$command
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target" > "$service_file"
```

```
systemctl daemon-reload
systemctl enable "$service_name"
systemctl start "$service_name"
```

```
    echo "Сервис '$service_name' успешно создан и запущен."
}
```

```
# Функция отображения справочной информации об использовании
утилиты
```

```
show_usage_info() {
```

```
    echo "Использование: my_service_tool <команда> [опции]"
```

```
    echo "Команды:"
```

```
    echo "  create-from-script <путь_к_скрипту> [имя_сервиса] [описание] -
Создает сервис из скрипта .sh"
```



```

    echo " create-from-command <команда> [имя_сервиса] [описание] -
Создает сервис из команды или службы"
    echo " help - Отображает эту справку"
}

# Проверка наличия прав суперпользователя
if [[ $EUID -ne 0 ]]; then
    echo "Этот скрипт должен быть запущен с правами суперпользователя"
    exit 1
fi

# Основная логика скрипта
main() {
    if [[ $# -lt 1 ]]; then
        show_usage_info
        exit 1
    fi

    case "$1" in
        create-from-script)
            create_systemd_service_from_script "$2" "$3" "$4"
            ;;
        create-from-command)
            create_systemd_service_from_command "$2" "$3" "$4"
            ;;
        help)
            show_usage_info
            ;;
        *)
            echo "Неверная команда: $1"
            show_usage_info
            exit 1
            ;;
    esac
}

# Запуск основной функции с переданными аргументами
main "$@"

```

1. **Создание сервиса systemd на основе скрипта .sh:** Реализовано в функции **create\_systemd\_service\_from\_script**, которая принимает

путь к скрипту, имя сервиса и описание, создает и запускает соответствующий systemd сервис.

2. **Создание сервиса systemd на основе утилит, команд и служб:**  
Реализовано в функции `create_systemd_service_from_command`, которая аналогично принимает команду или имя службы, имя сервиса и описание для создания systemd сервиса.
3. **Справочная информация по использованию утилиты (function `usage()`):** Реализовано в функции `show_usage_info`, которая предоставляет пользователям информацию о том, как использовать утилиту и её различные функции.

### **Перемещение Файла в /usr/local/bin:**

```
sudo mv my_service_tool /usr/local/bin/
```

Эта команда переместит файл `my_service_tool` из текущей директории в `/usr/local/bin`.

### **Делаем Файл Исполняемым:**

```
sudo chmod +x /usr/local/bin/my_service_tool
```

После выполнения этих шагов, скрипт будет доступен для выполнения как команда из любого места в системе. Можно вызвать его, просто напечатав `my_service_tool` в командной строке.