## Создание и запуск собственного сервиса

### Цель работы

Познакомиться с механизмом управления службами и автозапуском в Linux, создать и запустить собственноручно написанную программу как сервис операционной системы.

### Задания для выполнения

1. Создайте скрипт на языке Python, который выполняет какую-нибудь операцию в фоне.
2. Зарегистрируйте этот скрипт как системную службу (демона), которая стартует при запуске системы с помощью инициализатора systemd.

### Методические указания

На большинстве современных дистрибутивов Linux за автозапуск системных служб отвечает специальная программа systemd. Она постепенно заменяет более старую систему инициализации операционной системы init. В настоящее время init считается устаревшей системой и постепенно ожидается, что все основные дистрибутивы перейдут на systemd. Эта система инициализации предоставляет больше возможностей и гибче в настройке.

#### Создание программы

Для начала создадим простой скрипт. Он использует сокеты для создания простейшего эхо сервера на порту 9988. Вы можете модифицировать этот скрипт на свое усмотрение.

#!/usr/bin/python3

import socket

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.bind(("localhost", 9988))

s.listen(1)

while True:

conn, addr = s.accept()

data = conn.recv(1024)

conn.send(data)

conn.close()

Для того, чтобы система могла запустить нашу программу при старте нужно указать команду запуска, включающую полный путь к файлу. Вы можете скопировать его в одну из системных папок (например, /usr/bin/) или вызывать его из домашней директории. В любом случае, нужно запомнить команду запуска нашей программы.

Обратите внимание на шебанг в начале скрипта. Он позволяет запускать его непосредственно командой ./server.py, без указания пути к интерпретатору. В принципе, вы можете обойтись без шебанга и включить в команду запуска и путь к интерпретатору Python.

#### Настройка автозапуска

Чтобы создать сервис, вы должны создать соответствующий конфигурационный файл. Он содержит все необходимые настройки для запуска сервиса в виде текста в определенном формате. Он должен называться <servicename>.service, где <servicename> - это системное имя вашего демона. Этот файл должен располагаться в одном из определенных директорий, которые сканируются программой systemd. Полный список таких директорий вы можете найти в [документации](https://www.freedesktop.org/software/systemd/man/systemd.unit.html#User%20Unit%20Search%20Path). Некоторые из них являются системными и для доступа к ним нужны права суперпользователя. Некоторые - специфичны для текущего пользователя. В данном примере мы будем использовать папку /etc/systemd/system/.

Давайте создадим текстовый файл /etc/systemd/system/dummy.service. В него запишем такие директивы:

[Unit]

Description=Dummy Service

[Service]

Type=simple

ExecStart=/home/user/dummy.py

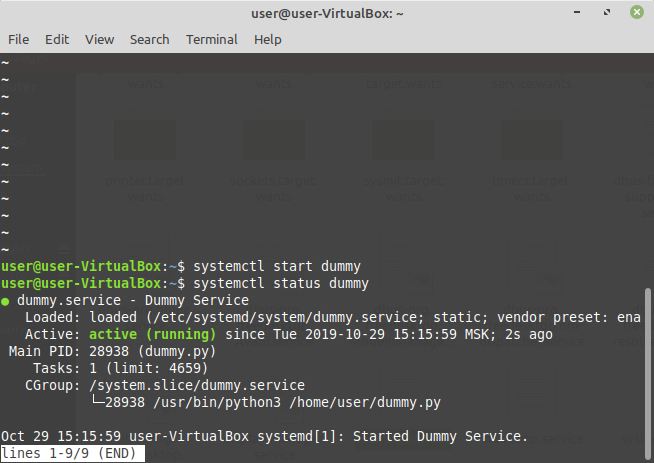
Файл юнитов организован, как и многие конфигурационные файлы Linux в виде разделов. Главный раздел называется Unit. Подробное описание всех разделов и директив можете почитать, например, [здесь](https://linux-notes.org/pishem-systemd-unit-fajl/). После этого вы можете управлять этим сервером используя команды systemctl.

#### Команды управления демоном

Теперь вы можете использовать команду systemctl, чтобы управлять своим сервисом. Для начала введите

systemctl status dummy

Эта команда покажет информацию о вашем сервисе. Главное, что здесь можно узнать, это то, запущен ли он в данный момент. Запустить сервис можно командой systemctl start dummy:



Кроме этих двух могут быть полезными еще другие команды из семейства systemctl:

enable - включение автозапуска сервиса при старте системы;

disable - отключение автозапуска;

restart - перезапуск сервиса: сервис будет остановлен и заново запущен;

stop - остановка сервиса;

reload - обновление конфигурации сервиса без перезапуска.

#### Дополнительные настройки

При помощи директив в файле юнита можно довольно гибко настроить поведение сервиса при запуске. Например, можно указать, что данный сервис должен запускаться только после указанных. Это может быть полезно, если ваш демон зависит от каких-то других. Например, если при старте ваша программа подключается к базе данных, она должна быть запущена только после демона СУБД. Для этого используются директивы After, Requires, Wants.

Можно настроить запуск сервиса с привилегиями определенного пользователя при помощи директив User и Group и задать рабочий каталог для программы сервиса (WorkingDirectory).

При помощи директивы Restart можно настроить перезапуск процесса при его завершении. Здесь же можно указать, в каких случаях нужно заново запускать программу, также настроить интервал ожидания.

### Контрольные вопросы

1. Чем процесс-демон отличается от обычного процесса?
2. Какие основные программы управляют автозапуском систем Linux?
3. Какие действия можно проделать с помощью команды systemctl?
4. Чем отличается включение и запуск службы?
5. Как зарегистрировать программу как службу операционной системы?

### Дополнительные задания

1. Настройте автозапуск вашего демона таким образом, чтобы он автоматически перезапускался через 2 секунды после завершения процесса вследствие, например, программной ошибки.
2. Создайте скрипт установки вашей программы, который автоматически копирует ее в системный каталог и настраивает автовыполнение как службы Linux.
3. (\*) Создайте скрипт установки, который автоматически распознает программу управления службами (systemd, init, upstart) и регистрирует программу соответственно.