

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования  
«Гомельский государственный университет имени Франциска Скорины»  
Факультет физики и информационных технологий  
Кафедра общей физики

**Отчёт по лабораторной работе №5**  
**«Управление службами»**

Выполнил:  
студент группы КИ-22: Аксёнов И. С.

Проверил: Грищенко.В.В

Гомель 2025

**Цель работы:** Изучить основные принципы работы с процессами в операционных системах Windows и Linux.

## **Краткие сведения из теории**

### **Процессы и потоки операционных систем**

В современных многозадачных операционных системах существуют фоновые программы, которые не требуют взаимодействия с пользователем. Такие программы называют службами. Служба (от англ. service) – программа, запускаемая операционной системой и работающая в фоновом режиме без прямого взаимодействия с пользователем. Состав имеющихся в наличии и запущенных служб зависит от версии операционной системы и установленных в ней приложений.

В операционной системе Windows служба – это процесс, который запускается на машине в фоновом режиме для выполнения определенных действий в ответ на запросы пользователей. В качестве примера приведем службу World Wide Web (WWW), работающую на серверах, на которых установлен пакет Internet Information Services (IIS). Служба WWW работает в фоновом режиме на сервере, ожидая получения HTTP-запросов от web-браузеров. При получении такого запроса служба WWW отвечает на него, посылая запрошенный файл или выполняя определенное действие.

В UNIX-мире такого понятия, как служба, нет, а вместо него есть демоны (от англ. daemon). Название службы "демон" носит отнюдь не религиозный характер. Одной из версий такого названия является акронимом выражения "Disk And Execution MONitor".

В большинстве Linux-подобных операционных систем службы именуют именно демонами, однако в ОС Ubuntu вместо демонов "прижилось" название "службы".

### **Службы в ОС Ubuntu**

Для того чтобы централизованно управлять службами операционной системы, нужна система инициализации, которая будет запускать службы в нужный момент и следить за тем, чтобы они работали так, как это было задумано по определению.

В первых версиях ОС Ubuntu Linux управление запуском служб производилось с помощью команд Терминала, в котором требовалось выполнить команду запуска нужной службы. По мере развития концепции ОС Ubuntu Linux на смену устаревшей модели запуска служб пришла система инициализации systemd.

Система инициализации systemd – это программа инициализации служб операционной системы Linux. Другими словами, systemd является первичной службой, которая инициализируется ядром операционной системы и после этого запускает подчиненные ей службы операционной системы.

После запуска той или иной службы система инициализации `systemd` собирает весь вывод служебной информации запущенной службы и сохраняет в определенный лог-файл. В случае если служба аварийно завершила свою работу, система инициализации `systemd` будет пытаться ее перезапустить. В системе инициализации `systemd` есть такое понятие, как юниты. Юниты (от англ. `unit` – единица) – это файлы конфигурации, хранящие информацию о службе, устройстве или сокете. Юниты могут в своей работе взаимодействовать друг с другом. Приведем небольшой список юнитов, которые задействуются в работе службой `systemd`:

- `service` – программа, которая управляет работой других служб;
- `socket` – сокет для взаимодействия процессов (определенных службой);
- `device` – конфигурационный файл, который содержит инструкции для обработки списка устройств;
- `mount` – файл, содержащий информацию о точке монтирования файловой системы;
- `automount` – файл, содержащий информацию о списке устройств, которые должны быть смонтированы автоматически;
- `target` – файл, указывающий на группу служб, которые требуется выполнить этой единице;
- `snapshot` – файл, указывающий на другие единицы и восстанавливающий состояние ранее запущенных служб.

Юнит `service` как раз относится к службам в ОС Ubuntu Linux.

Для управления службами система инициализации `systemd` использует инструмент `systemctl`.

`Systemctl` – инструмент системы инициализации `systemd`, предназначенный для управления службами в ОС Ubuntu Linux. Данный инструмент не только разрешает управлять запуском, остановкой и перезагрузкой служб, но и имеет возможность проверки ее состояния, а также позволяет анализировать эффективность загрузки службы.

Синтаксис программы `systemctl` следующий:

`systemctl [опции] {команда}`

Таблица 1 – Описание опций команды systemctl

Сокращенный вариант	Полный вариант	Описание
-a	--all	Позволяет отобразить все запущенные службы, включая остановленные
-q	--quiet	Позволяет вывести минимальную информацию по службам
-t	--type=тип	Позволяет отобразить службы в соответствии с заданным типом
-f	--full	Позволяет отобразить полное имя единицы
	--runtime	Позволяет не сохранять вносимые изменения после перезагрузки системы
	--no-pager	Позволяет отключить постраничный вывод информации
	--state=состояние	Позволяет отсортировать выводимый список служб по заданному состоянию: LOAD – отображение только загруженных служб; SUB – отображение состояния активности единицы; ACTIVE – отображение активных служб
-s	--signal=сигнал	Позволяет отправить службе заданный сигнал

Полный набор параметров можно узнать, выполнив команду `man systemctl`.

Значения, которые принимает параметр [команда] команды `systemctl`, приведены в таблицу 2. Полный набор которых вы можете узнать, выполнив команду справки `man systemctl`.

Таблица 2 – Описание параметров [команда] программы systemctl

Команда	Описание
list-units	Позволяет вывести список активных юнитов
list-sockets	Позволяет вывести список загруженных сокетов, упорядоченных по адресу
start NAME	Позволяет запустить службу, имя которой передано в аргументе NAME
stop NAME	Команда позволяет остановить службу, имя которой передано в аргументе NAME
reload NAME	Позволяет обновить конфигурации службы, имя которой передано в аргументе name
restart NAME	Позволяет перезапустить службу, имя которой передано в аргументе NAME

Команда	Описание
try-restart NAME	Позволяет перезапустить службу, имя которой передано в аргументе NAME, при условии, что указанная служба активна в данный момент времени
reload-or-restart NAME	Позволяет обновить конфигурацию службы, а затем выполнить ее перезапуск. В случае если обновление конфигурации не предусмотрено, выполняется только перезапуск службы
isolate NAME	Позволяет запустить службу, имя которой передано в аргументе NAME, а все остальные остановить
kill NAME	Позволяет отправить службе сигнал завершения работы, имя которой передано в аргументе name
is-active PATTERN	Позволяет проверить, запущена ли указанная служба, имя которой передано в аргументе PATTERN
is-failed PATTERN	Позволяет проверить, не завершилась ли аварийно служба, имя которой передано в аргументе PATTERN
status [ PATTERN] PID]	Позволяет отобразить состояние службы, имя которой передано в аргументе PATTERN либо идентификатор которой задан в аргументе PID, и ее вывод
show (PATTERN] JOB]	Позволяет отобразить параметры управления службой, имя которой передано в аргументе PATTERN, либо отобразить параметры работающих служб
reset-failed [PATTERN)	Команда позволяет перезапустить службы, завершившиеся с ошибкой, имя которых передано в аргументе PATTERN
list-dependencies [NAME)	Позволяет отобразить зависимости службы, имя которой передано в аргументе name
list-unit-files (PATTERN]	Позволяет отобразить все файлы, установленные указанной службой
enable NAME	Позволяет активировать запуск службы, имя которой передано в аргументе NAME
disable NAME	Позволяет отключить запуск службы, имя которой передано в аргументе NAME
reenable NAME	Позволяет отключить запуск службы, а затем снова включить
is-enabled NAME	Позволяет отобразить, находится ли указанная служба в автозапуске или нет

Получить список запущенных служб:  
systemctl list-units --type=service

Данная конструкция позволит вывести список служб в удобном для восприятия виде.

```

dkotad@vbu16:~$ systemctl list-units --type=service
UNIT                                LOAD    ACTIVE SUB    JOB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
acpid.service                       loaded active running ACPI event daemon
apparmor.service                   loaded active exited LSB: AppArmor initia
apport.service                     loaded active exited LSB: automatic crash
apt-daily-upgrade.service           loaded inactive dead    start Daily apt upgrade an
apt-daily.service                  loaded activating start start Daily apt download a
avahi-daemon.service               loaded active running Avahi mDNS/DNS-SD St
colord.service                     loaded active running Manage, Install and
console-setup.service              loaded active exited Set console font and
cron.service                       loaded active running Regular background p
cups-browsed.service               loaded active running Make remote CUPS pri
cups.service                       loaded active running CUPS Scheduler
dbus.service                       loaded active running D-Bus System Message
fwupd.service                     loaded active running Firmware update daem
getty@tty1.service                 loaded active running Getty on tty1
grub-common.service                loaded active exited LSB: Record successf
irqbalance.service                loaded active exited LSB: daemon to balan

```

В полученном результате имеется 5 колонок:

- UNIT – имя службы (единицы);
- LOAD – состояние службы. Статус loaded означает, что конфигурационный файл службы был успешно запущен;
- ACTIVE – активность службы. Статус active означает, что служба была успешно запущена;
- SUB – текущее состояние службы. Статус running означает, что в данный момент времени служба выполняется, а статус exited свидетельствует о том, что служба успешно завершила свою работу;
- DESCRIPTION – краткое описание службы.

Если вам требуется отобразить список только запущенных служб, то для этого предназначена следующая конструкция:

```
systemctl list-units --type service --state=running
```

Аналогично производится получение списка служб, чья работа завершилась аварийно:

```
systemctl list-units --type=service --state=failed
```

Получать список служб мы научились, теперь рассмотрим, как производится запуск требуемой службы.

В качестве примера запустим службу cups.service:

```
sudo systemctl start cups.service
```

В ответ на эту команду Терминал запросит у вас пароль и запустит службу cups.service. В случае успешного запуска Терминал не выведет дополнительных сообщений.

Останавливается служба аналогичным образом:

```
sudo systemctl stop cups.service
```

Если вы сомневаетесь в том, запустилась или остановилась требуемая служба, вы в любой момент времени можете просмотреть ее текущее состояние:

```
sudo systemctl status cups.service
```

В ответ на это программа systemctl отобразит не только текущее состояние службы, но также и ее месторасположение, время запуска и другую полезную информацию, которая может пригодиться в случае решения проблем с запуском службы.

Службы ОС Windows могут стартовать вместе с запуском операционной системы, а также запускаться в процессе работы операционной системы, т. е. вручную.

В ОС Ubuntu Linux схожая концепция запуска служб. Таким образом, службы могут иметь 4 состояния:

- `enablEd` – данный статус означает, что служба находится в автозагрузке и будет загружена при старте операционной системы;
- `disablEd` – данный статус означает, что служба не находится в автозагрузке и она не будет запущена при старте операционной системы, однако она может быть запущена вручную;
- `maskEd` – данный статус означает, что служба не находится в автозагрузке и не будет запущена при старте операционной системы, однако запустить вручную ее не удастся;
- `static` – данный статус означает, что служба находится в автозагрузке и будет загружена при старте операционной системы, однако данную службу невозможно отключить.

В качестве примера получим список служб, которые имеют состояние `maskEd`:

```
systemctl list-unit-files --state=maskEd
```

Данная команда позволит получить имена служб, запуск которых невозможен.

Представим ситуацию, когда вам потребовалось внести в автозагрузку вашу службу `my.program.service`. В этом случае вам поможет следующая команда:

```
sudo systemctl enable my.program.service
```

Данная команда поместит в автозагрузку требуемую службу, и она будет автоматически запускаться при старте операционной системы.

Для отключения службы используйте следующую команду:

```
sudo systemctl disable my.program.service
```

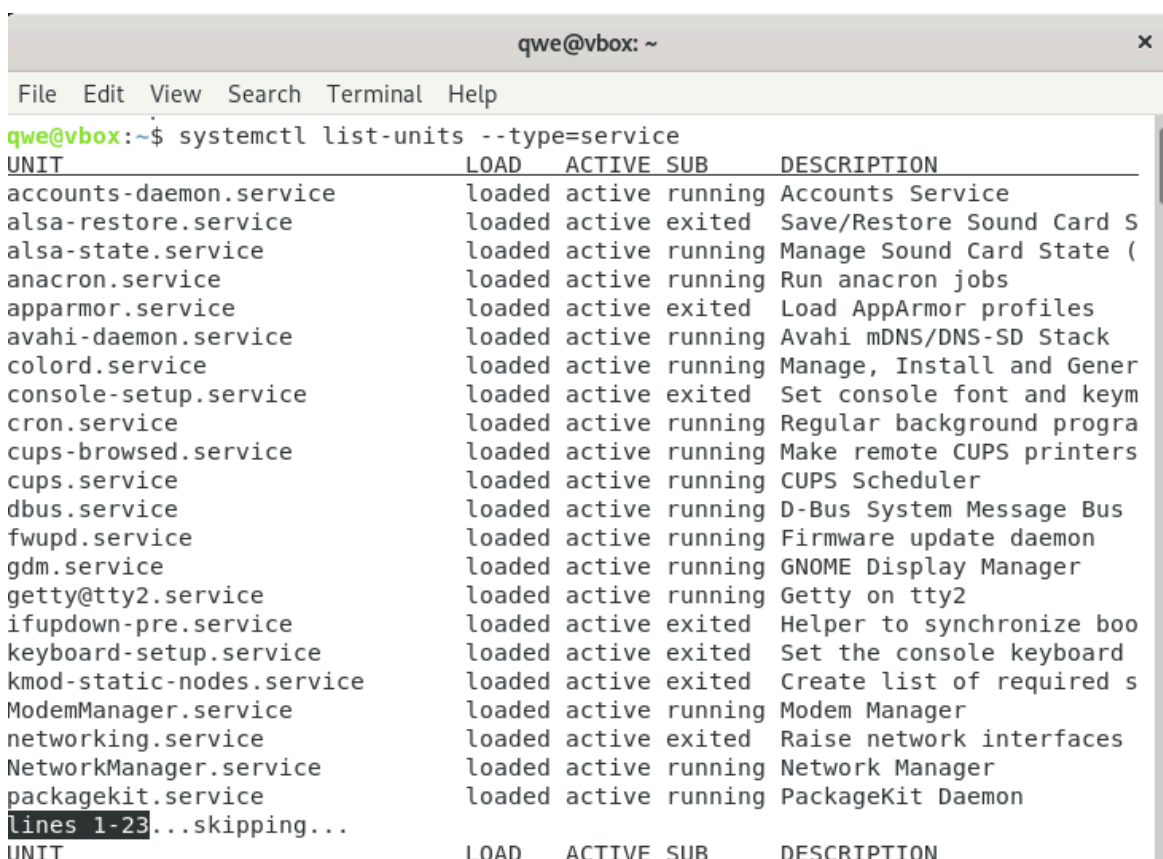
Разумеется, для выполнения предыдущих команд требуется существование самой службы, иначе Терминал сообщит о том, что данный юнит не был найден и поэтому его запуск невозможен.

Проверить успешность запуска службы можно, выполнив команду:

```
sudo systemctl is-enablEd my.program.service
```

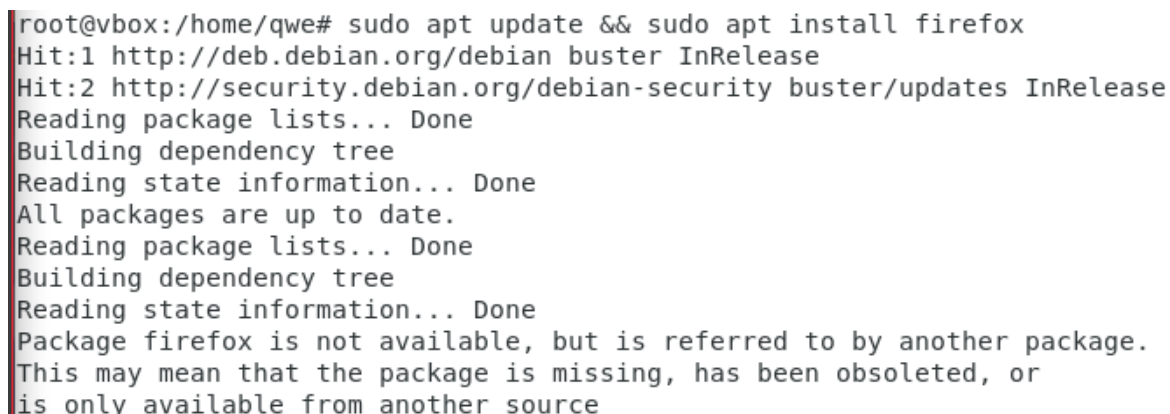
Данная команда выведет одну-единственную строку состояния службы: `enablEd`, `disablEd`, `maskEd` или `static`.

## Ход работы



```
qwe@vbox: ~  
File Edit View Search Terminal Help  
qwe@vbox:~$ systemctl list-units --type=service  
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION  
accounts-daemon.service             loaded active running Accounts Service  
alsa-restore.service                loaded active exited Save/Restore Sound Card S  
alsa-state.service                  loaded active running Manage Sound Card State (  
anacron.service                     loaded active running Run anacron jobs  
apparmor.service                    loaded active exited Load AppArmor profiles  
avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack  
colord.service                       loaded active running Manage, Install and Gener  
console-setup.service               loaded active exited Set console font and keym  
cron.service                        loaded active running Regular background progra  
cups-browsed.service                loaded active running Make remote CUPS printers  
cups.service                         loaded active running CUPS Scheduler  
dbus.service                        loaded active running D-Bus System Message Bus  
fwupd.service                       loaded active running Firmware update daemon  
gdm.service                         loaded active running GNOME Display Manager  
getty@tty2.service                  loaded active running Getty on tty2  
ifupdown-pre.service                loaded active exited Helper to synchronize boo  
keyboard-setup.service              loaded active exited Set the console keyboard  
kmod-static-nodes.service            loaded active exited Create list of required s  
ModemManager.service                loaded active running Modem Manager  
networking.service                  loaded active exited Raise network interfaces  
NetworkManager.service              loaded active running Network Manager  
packagekit.service                  loaded active running PackageKit Daemon  
lines 1-23...skipping...  
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
```

Рисунок 1 - Вывод списка всех служб



```
root@vbox:/home/qwe# sudo apt update && sudo apt install firefox  
Hit:1 http://deb.debian.org/debian buster InRelease  
Hit:2 http://security.debian.org/debian-security buster/updates InRelease  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
All packages are up to date.  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package firefox is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source
```

Рисунок 2 - Установка браузера FireFox

### Обнаружение службы автообновления Firefox:

В Linux: Firefox обновляется через пакетный менеджер (apt, dnf), поэтому специфическая служба автообновления отсутствует.



```
qwe@vbox: ~  
File Edit View Search Terminal Help  
root@vbox:/home/qwe# sudo apt remove firefox  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package 'firefox' is not installed, so not removed  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@vbox:/home/qwe# █
```

Рисунок 3 - Остановка службы автообновления Firefox

```
qwe@vbox: ~  
File Edit View Search Terminal Help  
GNU nano 3.2 /etc/systemd/system/sample_app.service  
[Unit]  
Description=Simple App Service  
After=network.target  
  
[Service]  
ExecStart=/usr/bin/python3 /opt/sample_app/app.py  
Restart=always  
User=nobody  
WorkingDirectory=/opt/sample_app  
PIDFile=/run/sample_app.pid  
  
[Install]  
WantedBy=multi-user.target  
  
[ Read 14 lines ]  
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text    ^J Justify    ^C Cur Pos  
^X Exit        ^R Read File  ^\ Replace    ^U Uncut Text  ^T To Spell   ^_ Go To Line
```

Рисунок 4 - Настройка sample\_app как службы

```
qwe@vbox: ~
File Edit View Search Terminal Help
root@vbox:/home/qwe# sudo systemctl daemon-reload
root@vbox:/home/qwe# sudo systemctl enable sample_app
Created symlink /etc/systemd/system/multi-user.target.wants/sample_app.service → /etc/systemd/system/sample_app.service.
root@vbox:/home/qwe# sudo systemctl start sample_app
root@vbox:/home/qwe# systemctl status sample_app
● sample_app.service - Simple App Service
   Loaded: loaded (/etc/systemd/system/sample_app.service; enabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Wed 2025-05-21 19:37:23 MSK; 1min 2s ago
     Process: 2599 ExecStart=/usr/bin/python3 /opt/sample_app/app.py (code=exited, status=200/CHDIR)
    Main PID: 2599 (code=exited, status=200/CHDIR)

May 21 19:37:23 vbox systemd[1]: sample_app.service: Main process exited, code=exited, status=200/
May 21 19:37:23 vbox systemd[1]: sample_app.service: Failed with result 'exit-code'.
May 21 19:37:23 vbox systemd[1]: sample_app.service: Service RestartSec=100ms expired, scheduling
May 21 19:37:23 vbox systemd[1]: sample_app.service: Scheduled restart job, restart counter is at
May 21 19:37:23 vbox systemd[1]: Stopped Simple App Service.
May 21 19:37:23 vbox systemd[1]: sample_app.service: Start request repeated too quickly.
May 21 19:37:23 vbox systemd[1]: sample_app.service: Failed with result 'exit-code'.
May 21 19:37:23 vbox systemd[1]: Failed to start Simple App Service.
lines 1-14/14 (END)
```

Рисунок 5 - Определение процесса управления приложением

```
qwe@vbox: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/systemd/system/sample_app.service

[Unit]
Description=Simple App Service
After=network.target

[Service]
ExecStart=/usr/bin/python3 /opt/sample_app/app.py
Restart=always
User=nobody
WorkingDirectory=/opt/sample_app
PIDFile=/run/sample_app.pid
StandardOutput=journal
StandardError=journal
MemoryLimit=500M
CPUQuota=50%

[Install]
WantedBy=multi-user.target

[ Read 18 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line
```

Рисунок 6 - Описание сервиса

## **Контрольные вопросы**

### **1. Что такое служба ОС?**

Служба операционной системы (или демон в Linux) — это фоновый процесс, который работает без взаимодействия с пользователем и выполняет важные функции, такие как управление сетью, обновления программ, мониторинг ресурсов и многое другое.

### **2. Зачем нужны службы в ОС?**

Службы обеспечивают автоматизацию работы ОС, управляют важными процессами (без необходимости вмешательства пользователя), поддерживают работоспособность системных компонентов, а также повышают безопасность и стабильность.

### **3. Что такое юнит в Linux?**

Юнит — это конфигурационный объект в systemd, который управляет различными ресурсами системы (службы, монтирование файловых систем, таймеры и др.).

### **4. Какие юниты доступны в Ubuntu?**

В systemd есть несколько типов юнитов:

- Service (.service) — управление службами.
- Timer (.timer) — планирование выполнения задач.
- Mount (.mount) — монтирование файловых систем.
- Socket (.socket) — управление сетевыми соединениями.
- Target (.target) — группировка юнитов.

### **5. Инструмент инициализации systemctl**

systemctl — это утилита для управления службами, предоставляемая systemd. Она используется для запуска, остановки, проверки статуса и настройки служб в Linux.

### **6. Основные параметры команды systemctl**

- systemctl start <служба> — Запустить службу.
- systemctl stop <служба> — Остановить службу.
- systemctl restart <служба> — Перезапустить службу.
- systemctl enable <служба> — Автозапуск службы при загрузке ОС.

- `systemctl disable <служба>` — Отключить автозапуск.
- `systemctl status <служба>` — Проверить статус службы.

## 7. Как запустить или остановить службу в Ubuntu?

Запуск:

`sudo systemctl start <служба>`

Остановка:

`sudo systemctl stop <служба>`

## 8. Назначение служб в Windows

Службы Windows выполняют системные функции, например управление принтерами, обновлениями, сетевыми соединениями и защитой ОС.

## 9. Графический инструмент для управления службами в Windows

`services.msc` — встроенный инструмент Windows для просмотра, запуска и настройки служб.

## 10. Основные командлеты PowerShell для управления службами Windows

- `Get-Service` — Просмотр списка служб.
- `Start-Service <служба>` — Запуск службы.
- `Stop-Service <служба>` — Остановка службы.
- `Restart-Service <служба>` — Перезапуск службы.
- `Set-Service -StartupType Automatic <служба>` — Настроить автозапуск.

## 11. Как запустить или остановить службу в Windows?

Через PowerShell:

`powershell`

`Start-Service <ИмяСлужбы>`

`Stop-Service <ИмяСлужбы>`

Через `services.msc`:

1. Открыть `services.msc` (Win + R → `services.msc` → Enter).

2. Найти нужную службу.
3. Нажать "Запустить" или "Остановить".