

Создание простого systemd unit



Дата создания: 2023/04/22 17:28 (C) mihanik



Источники

- <https://newadmin.ru/sozдание-prostogo-systemd-unit/?ysclid=lgs8xf61vl418127116>

Что такое systemd?

systemd — это набор базовых строительных блоков для системы Linux. Он предоставляет диспетчер систем и служб, который работает как PID 1 и запускает остальную часть системы. systemd предоставляет агрессивные возможности распараллеливания, использует активацию сокетов и D-Bus для запуска служб. Предлагает запуск демонов по требованию, отслеживает процессы, использующие группы управления Linux. Поддерживает точки монтирования и автоматического монтирования. Реализует сложную логику управления службами на основе транзакционных зависимостей. systemd поддерживает сценарии SysV и LSB init и работает как замена для sysvinit. Другие части включают демон ведения журнала, утилиты для управления базовой конфигурацией системы. В него входит: имя хоста, дата, локаль, ведение списка зарегистрированных пользователей и запущенных контейнеров. А также виртуальных машин, системные учетные записи, каталоги времени выполнения и параметры. Не забудем о демонах управления простой конфигурацией сети, синхронизацией сетевого времени, пересылкой журналов и разрешением имен.

unitd работает с сервисами описанными в своей конфигурации. Основой для создания сервиса служит юнит (unit) — это текстовый файл с описанием на подобии ini файла в системе windows. Конфигурационный файл состоит из секций. Внутри каждой секции указываются необходимые параметры. Обязательными во всех юнитах являются две секции, остальные используются в зависимости от типа юнита.

Каталоги хранения юнитов

- */usr/lib/systemd/system* — юниты поставляемые вместе с системой и устанавливаемыми приложениями
- */run/systemd/system* — юниты созданные динамически (в рантайме)
- */etc/systemd/system* — юниты системного администратора (тут и будем хранить наши)

- **Wants** — еще одна зависимость, означает желательно. В примере `Wants=sshd-keygen.target`, т.е. желательно чтобы было запущено `sshd-keygen.target`. Желательно но не обязательно.

[Service]

Type — типы запуска служб. Могут быть

- **simple** (по умолчанию) — происходит незамедлительный запуск этой службы, с учетом того что процесс не разветвляется (`fork`). Не используйте `simple` если пользуетесь очередностью запуска. Одно исключение это активация сокета.
- **forking** — служба считается запущенной после того, после разветвления процесса с завершением родительского процесса. Используется для запуска классических демонов исключая случаи, когда в таком поведении процесса нет необходимости. Также желательно указать `PIDFile=`, чтобы `systemd` мог отслеживать основной процесс.
- **oneshot** — удобен для скриптов, которые выполняют одно задание и завершаются. При необходимости можно задать параметр `RemainAfterExit=yes`, чтобы `systemd` считал процесс активным даже после его завершения.
- **notify** — идентичен параметру `simple`, но с оговоркой, что демон пошлет `systemd` сигнал о своей готовности. Эталонная реализация данного уведомления представлена в `libsystemd-daemon.so`.
- **dbus** — служба считается находящейся в состоянии готовности, когда указанный параметр `BusName` появляется в системной шине `DBus`.
- **idle** — откладывает выполнение двоичного файла службы до момента выполнения всех остальных задач. В остальном поведение аналогично `simple`.

Далее в разделе **Service**

- **EnvironmentFile** — файлы переменного окружения
- **ExecStart** — полный путь к исполняемому файлу программы с параметрами запуска
- **ExecReload** — полный путь к исполняемому файлу программы с параметрами перезапуска программы
- **KillMode** — указывается как будет завершен процесс. В данном случае параметр `process` говорит о том что будет закрыт только главный процесс
- **Restart** — перезагрузка процесса, параметр `on-failure` указывает на автоматическую перезагрузку в случае отказа процесса
- **RestartSec** — время ожидания через которое процесс должен перезагрузиться

[Install]

- **WantedBy** — указывает на каком уровне запуска стартует сервис, параметр `multi-user.target` указывает на запуск в многопользовательском режиме без графики

Файлы с юнитами кладем в каталог `/etc/systemd/system/`

Итак начнем создание простого `systemd` unit, подопытным будет утилита мониторинга системы `nmon`. Откроем на редактирование файл юнита

```
mcedit /etc/systemd/system/nmon.service
```


