

Работа с systemd

Цель работы: получить практические навыки по управлению ОС Linux с помощью systemd.

Краткие теоретические сведения:

Самой распространенной системой инициализации ОС Linux является systemd. Она заменила подсистему init и в отличие от последней позволяет реализовать параллельный запуск служб в процессе загрузки системы.

Основной сущностью systemd является – юнит. Юниты бывают нескольких типов:

service - самый используемый вид юнитов, описывает службы, или приложения, работающие в фоновом режиме (демоны). socket - юниты для описания IPC, сетевых сокетов и FIFO буферов.

device - автоматически создаваемые юниты, описывающие файлов устройств, которые необходимы для systemd.

mount - юниты для описания точек монтирования.

automount - юниты для автоматического монтирования файловых систем. Зависят от mount юнитов и выполняются после них.

swap - описывает файл подкачки.

target - то, что раньше имело название runlevel, юниты, описывающие базовое состояние системы. path - юниты, для отслеживания изменения путей в файловой системе.

timer - юниты для описания задач по расписанию, аналог cron

slice - юниты для управления группой процессов с помощью Linux CGroups (man systemd.slice).

scope - не конфигурируются с помощью файлов, генерируются автоматически на основе данных из системной шины. Используются для управления группами процессов, созданных извне.

Для управления systemd служат ряд утилит, среди них:

- `systemctl` – для управления загрузкой и получение информации о юнитах, управление загрузкой системы, управление отдельными юнитами
- `journalctl` – для работы с бинарным журналом systemd
- `systemctl-analyze` – для получения информации о времени загрузки системы

Если необходимо создать свой сервис, то он должен быть описан через файл юнита `service` в каталоге `/etc/systemd/system`.

Service-файла в systemd обычно состоит из трех секций:

- `[Unit]` - описание сервиса
- `[Install]` – конфигурационные параметры сервиса например зависимости
- `[Service]` – команды сервиса и правила обработки сервиса со стороны systemd.

Справку можно получить в справочных страницах `man`, например так: `man systemd.unit`

Перечислим основные параметры раздела `[Unit]`:

- `Description` - краткое описание юнита.
- `Documentation` - список ссылок на документацию.
- `Before, After` - порядок запуска юнитов.
- `Requires` - если этот сервис активируется, перечисленные здесь юниты тоже будут активированы. Если один из перечисленных юнитов останавливается или падает, этот сервис тоже будет остановлен.
- `Wants` - слабые зависимости. Если один из перечисленных юнитов не может успешно запуститься, это не повлияет на запуск данного сервиса. Это рекомендуемый способ установления зависимостей.
- `Conflicts` - если установлено что данный сервис конфликтует с другим юнитом, то запуск последнего остановит этот сервис и наоборот.

К основанным в секции `[Install]` относятся параметры:

- `Alias` - дополнительные имена сервиса, разделенные пробелами. Большинство команд в `systemctl`, за исключением `systemctl enable`, могут использовать альтернативные имена сервисов.
- `RequiredBy, WantedBy` - данный сервис будет запущен при запуске перечисленных сервисов. Для более подробной информации смотрите описание опций `Wants` и `Requires` в секции `[Unit]`.
- `Also` - определяет список юнитов, которые также будут активированы или деактивированы вместе с данным сервисом при выполнении команд `systemctl enable` или `systemctl disable`.

Основные параметры секции [Service]:

Type - настраивает тип запуска процесса. simple (по умолчанию) — запускает сервис мгновенно. Предполагается, что основной процесс сервиса задан в ExecStart. forking — считает сервис запущенным после того, как родительский процесс создает процесс-потомка, а сам завершится. oneshot — аналогичен типу simple, но предполагается, что процесс должен завершиться до того, как systemd начнет отслеживать состояния юнитов (удобно для скриптов, которые выполняют разовую работу и завершаются). dbus — аналогичен типу simple, но считает сервис запущенным после того, как основной процесс получает имя на шине D-Bus. notify — аналогичен типу simple, но считает сервис запущенным после того, как он отправляет systemd специальный сигнал. idle — аналогичен типу simple, но фактический запуск исполняемого файла сервиса откладывается, пока не будут выполнены все задачи.

ExecStart - команды вместе с аргументами, которые будут выполнены при старте сервиса. Опция Type=oneshot позволяет указывать несколько команд, которые будут выполняться последовательно. Опции ExecStartPre и ExecStartPost могут задавать дополнительные команды, которые будут выполнены до или после ExecStart.

ExecStop - команды, которые будут выполнены для остановки сервиса, запущенного с помощью ExecStart.

ExecReload - команды, которые будут выполнены чтобы сообщить сервису о необходимости перечитать конфигурационные файлы.

Restart - если эта опция активирована, сервис будет перезапущен если процесс прекращен или достигнут timeout, за исключением случая нормальной остановки сервиса с помощью команды systemctl stop

RemainAfterExit - если установлена в значение yes (True), сервис будет считаться запущенным даже если сам процесс завершен. Полезен с Type=oneshot. Значение по умолчанию False.

User и Group — задает пользователя и группу в контексте безопасности которых будет запущен сервис.

Инструментальные средства:

Утилиты:

systemctl, journalctl, systemctl-analyze,
logger, date

Файлы:

Утилиты работы с текстом:

echo, grep, sed, awk, cut

Редакторы:

vi, nano

Часть 1. Получение информации о времени загрузки.

1. Выведите информацию о времени, затраченном на загрузку системы
2. Выведите список всех запущенных при старте системы сервисов, в порядке уменьшения времени, затраченного на загрузку сервиса.
3. Выведите список сервисов, запуск которых с необходимостью предшествовал запуску сервиса `sshd`.
4. Сформируйте изображение в формате `svg` с графиком загрузки системы, сохраните его в файл.

Часть 2. Управление юнитами.

1. Получите список всех запущенных юнитов сервисов
2. Выведите перечень всех юнитов сервисов, для которых назначена автозагрузка.
3. Определите от каких юнитов зависит сервис `sshd`.
4. Определите запущен ли сервис `cron`, если нет, запустите его.
5. Выведите все параметры юнита `cron`, даже те, которые были назначены автоматически, и не были прописаны в файле юнита.
6. Запретите автозагрузку сервиса `cron`, но оставьте ему возможность запускаться по зависимостям.

Часть 3. Создание сервиса

1. Создайте собственный сервис `mymsg`. Сервис `mymsg` должен:
 - a. при старте системы записывать в системный журнал дату и время
 - b. должен запускаться только если запущен сервис `network`.

Примечание: *писать в системный журнал позволяет команда `logger`. Проверить корректность юнит-файла `service` позволяет команда `systemdanalyze`.*

2. Настройте автоматический запуск сервиса `mymsg` при старте системы.
3. Запустите сервис.

Часть 4. Работа с системным журналом

1. Выведите на консоль системный журнал. Убедитесь, что сервис `mysmsg` отработал корректно.
2. Выведите на консоль все сообщения системного журнала, касающиеся сервиса `mysmsg`.
3. Выведите на экран все сообщения об ошибках в журнале.
4. Определите размер журнала.

Часть 5. Создание и настройка `.mount` юнита

1. Подготовьте файловую систему.
 - a. Создайте новый раздел на диске или используйте существующий.
 - b. отформатируйте его в файловую систему `ext4`.
 - c. Создайте директорию для монтирования `/mnt/mydata`
2. Создание `.mount` юнита
 - a. Создайте файл `.mount` юнита в `/etc/systemd/system/mnt-mydata.mount`.
 - b. Настройте юнит следующим образом:
 - i. Добавьте описание юнита в секцию `[Unit]`.
 - ii. В секции `[Mount]` укажите устройство, точку монтирования, тип файловой системы и опции.
 - iii. В секции `[Install]` укажите, что юнит должен быть активирован при достижении `multi-user.target`.
 - c. Сохраните файл и выйдите из редактора.
3. Запуск и проверка `.mount` юнита
 - a. Включите и запустите юнит.
 - b. Проверьте статус юнита.
 - c. Убедитесь, что раздел смонтирован.

Часть 6. Использование .automount для отложенного монтирования

1. Подготовьте соответствующий .mount-юнит
 - a. После выполнения Части 5 у вас должен был остаться юнит для монтирования /mnt/mydata
 - b. Убедитесь, что при остановке раздел отмонтируется, а монтируется обратно только при запуске юнита или перезагрузке системы
2. Создание .automount юнита
 - a. Создайте файл .automount юнита в /etc/systemd/system/mnt-mydata.automount.
 - b. Настройте юнит:
 - i. В секции [Unit] добавьте описание.
 - ii. В секции [Automount] укажите точку монтирования и время до размонтирования (TimeoutIdleSec).
 - iii. В секции [Install] укажите, что юнит должен быть активирован при достижении multi-user.target.
 - c. Сохраните файл и выйдите из редактора.
3. Запуск и проверка .automount юнита
 - a. Включите и запустите .automount юнит.
 - b. Проверьте статус юнита и убедитесь, что раздел монтируется при обращении к точке монтирования.
 - c. Убедитесь, что раздел размонтируется после завершения работы.

Вопросы и задания:

1. Чем отличаются команды `systemctl restart` и `systemctl try-restart`?
2. Как с помощью `systemctl` запустить Linux в однопользовательском режиме?
3. Пусть вам нужно создать еще один сервис `mysrv`, который не будет запускаться автоматически, и может быть выполнен, только если сервис `mysrv` будет принудительно остановлен уже после старта системы. Приведите параметры и их значения из описания юнитов `mysrv` и `mysrv`, которые обеспечат выполнения этих условий.