



Система инициализации в РЕД ОС

Система инициализации Systemd

Операционной системе нужна **инициализация** - это действия, совершаемые при загрузке системы. Ранее в Linux была система под названием **init (SysVinit)**. Она позволяла запускать **скрипты** и вводила концепцию уровней инициализации (**runlevels**).

Уровни инициализации как безопасный режим в винде, только тут таких режимов больше и они более настраиваемые

Виды уровней инициализации:

- 0 — остановка системы;
- 1 — однопользовательский режим;
- 2/3 — многопользовательский с/без поддержки сети.
- 5 — графический режим
- 6 — перезагрузка

Для каждого из уровней в каталоге /etc был свой каталог с именем rcN.d. Внутри этих директорий символические ссылки на скрипты из каталога /etc/init.d/

Поменять на лету можно командой, например **init 0** - для выключения машины.

Система инициализации Systemd

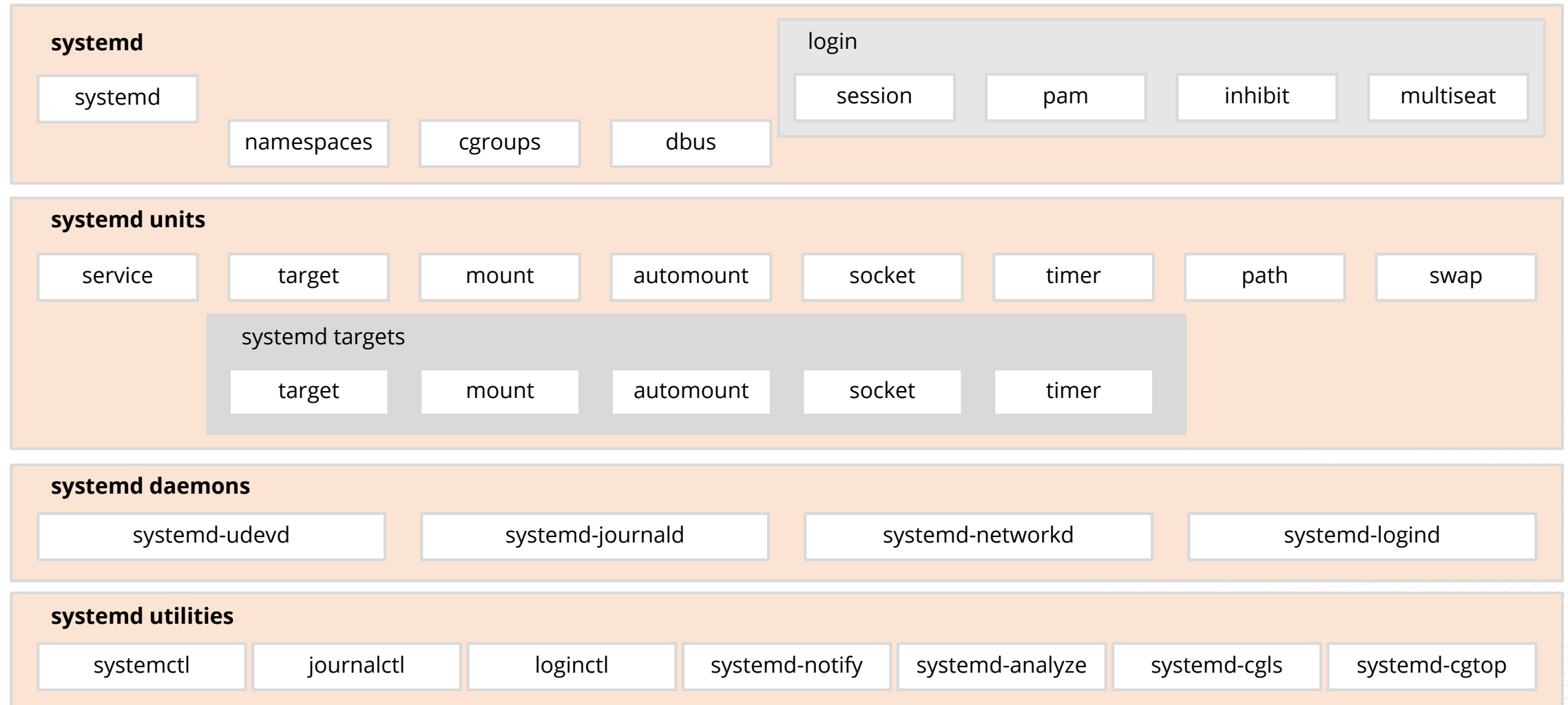
systemd - это "**система инициализации**". Содержит огромную кучу функций. Что нарушает принцип "unix-way".

Основные причины появления systemd - попытка избавиться от shell-скриптов, распараллелить загрузку и иметь единую точку контроля всего и вся.

Его особенностью является интенсивное **распараллеливание запуска** служб в процессе загрузки системы, что позволяло существенно ускорить запуск операционной системы, а так же **упростить настройку и управление**.

Конфигурация состоит из множества файлов, которые называют модулями (**unit**).

Система инициализации Systemd



Конфигурационные файлы `systemd`

Используются Unit-файлы — конфигурационные файлы для `systemd`

Типы unit-файлов:

- **.service** — отвечает за запуск сервисов (служб);
- **.target** — позволяет группировать модули;
- **.path** — управляет иерархией файловой системы.
- **.timer** — позволяет запускать модули по расписанию;
- **.mount** — отвечает за монтирование файловых систем;
- **.automount** — откладывает монтирование до события;
- **.device** — позволяет реагировать на подключение устройств;
- **.swap** — отвечает за подключение файла или устройства подкачки;
- **.socket** — предоставляет службам поддержку механизма сокет-активации;
- **.slice** — отвечает за создание контейнера `cgroups`;

Конфигурационные файлы systemd

Все модули расположены в трёх каталогах:

/usr/lib/systemd/system/ — модули из установленных пакетов

/run/systemd/system/ — модули, созданные в рантайме

/etc/systemd/system/ — модули, созданные администратором

Модуль представляет из себя текстовый файл с форматом, похожим на ini-файлы:

[Название секции в квадратных скобках]

имя_переменной = значение

Target

Цели (Targets)	Описание
runlevel.target, poweroff.target	Выключение
runlevel1.target, rescue.target	Однопользовательский режим
runlevel2.target, runlevel4.target, multi-user.target	Режимы, определяемые пользователем, по умолчанию идентичны цели 3
runlevel3.target, multi-user.target	Многопользовательский режим с поддержкой сети, без графического окружения
runlevel5.target, graphical.target	Графический многопользовательский режим с поддержкой сети. Обычно как цель 3 + графика
runlevel6.target, reboot.target	Перезагрузка

Шаблон Unit-файла (service)

[Unit]

- Description=...
- After=...
- Before=...
- Requires=...
- Conflicts=...

[Service]

- Type=...
- ExecStart=...
- ExecStop=...
- ExecReload=...
- Restart=...

[Install]

- WantedBy=
- RequiredBy=
- Alias=
- Also=

Содержимое файлов модулей

Раздел [Unit] — содержит описание модуля и его связи с другими модулями.

Description: имя и основные функции модуля.

Documentation: указывает местоположение документации.

Requires: перечисление модулей, от которых зависит этот модуль (служба или устройство).

Wants: эта директива похожа на Requires, но менее строгая, потому что не требует обязательного запуска зависимостей для работы модуля.

Before: перечисленные модули не будут запущены до тех пор, пока текущий модуль не будет отмечен как запущенный.

After: перечисленные модули будут запущены до запуска текущего модуля.

Conflicts: модули которые нельзя запускать одновременно с текущим.

Содержимое файлов модулей

Раздел [Service] — содержит информацию о службе и о процессе, которым она Управляет.

Type: тип запуска модуля. Может принимать значение simple, forking, oneshot, dbus, notify или idle.

ExecStart: команда со всеми аргументами, которая выполняется при запуске сервиса.

ExecStartPre, ExecStartPost: дополнительные команды, которые выполняются перед и после команды из ExecStart

ExecReload: команда для перезагрузки службы, если она есть.

TimeoutSec: время ожидания запуска или остановки процесса по истечении которого запуск/остановка службы считаются неудачными.

User, Group: пользователь / группа от имени которых будет запущен процесс службы.

Содержимое файлов модулей

Раздел [Install] — содержит информацию, используемую при включении и отключении автозапуска модуля (enable / disable).

WantedBy: определяет как модуль должен быть включен. Эта директива позволяет указать зависимость аналогично директиве Wants, но при этом включение автозапуска создаст символическую ссылку в каталоге systemd в качестве связи.

Alias: определяет псевдоним (или псевдонимы) модуля.

DefaultInstance: Для модулей-шаблонов, указывает экземпляр, который должен быть включен если пользователь не указал конкретный экземпляр.

Systemctl для unit

В Systemd есть инструмент для управления службами — **systemctl**. У этой утилиты много функций.

systemctl **опции** **команда** **служба** **служба...**

Не полный перечень команд, опции практически не используются

list-units — посмотреть все службы (юниты), аналог опции -t

list-sockets — посмотреть все службы сокетов

start — запустить службу linux

stop — остановить службу linux

reload — обновить конфигурацию службы из файла юнита

restart — перезапустить службу

status — посмотреть состояние и вывод службы

enable — добавить службу в автозагрузку

disable — удалить службу из автозагрузки

Systemctl для unit

В Systemd есть инструмент для управления службами — **systemctl**. У этой утилиты много функций.

systemctl **опции** **команда** **служба служба...**

Не полный перечень команд, опции практически не используются

cat — посмотреть файл юнита

kill -s HUP — убить модуль (определенный сигнал с помощью опции -s)

daemon-reload — перезагрузка конфигурации systemd

list-dependencies — дерево зависимостей юнита

show — низкоуровневые свойства юнита. Ключи в формате **ключ=значение**

mask — заблокировать юнит и сделать его не доступным

umask — разблокировать юнит и сделать его доступным

get-default — посмотреть модуль загружаемый по умолчанию

set-default — установить модуль загружаемый по умолчанию

Управление службами systemd

Рекурсивно перечислить все зависимости, нужно добавить флаг all

systemctl list-dependencies multi-user.target --all

Показать обратные зависимости (юниты, зависящие от юнита)

systemctl list-dependencies nginx.service --reverse

Просмотр списка модулей с типом таргет:

systemctl list-units -t target --all

Просмотр статуса системы:

systemctl status

Редактирование юнита

systemctl --full edit <имя_юнита>

Создание юнита

systemctl --force --full edit <имя_юнита>

Активные юниты

systemctl

Пример файла сервиса

[Unit]

Description=My Hello world service

Documentation=Нет документации

After=syslog.target network.target

[Service]

Type=oneshot

ExecStart=/bin/echo "Hello, world!"

User=dima

[Install]

WantedBy=multi-user.target

Пример файла сервиса

[Unit]

Description="Записывает в файл mylog каждые 10 секунд текущее время."

After=syslog.target network.target

[Service]

Type=simple

ExecStart=/usr/bin/forktest.sh

Restart=always

RestartSec=10

[Install]

WantedBy=multi-user.target

```
systemctl daemon-reload
systemctl start forkingservice.service
за файлом я так слежу из консоли:
tail -f /var/log/mylog
```

```
Скрипт
/usr/bin/forktest.sh
#!/bin/bash
date >> /var/log/mylog # Дописываем дату
```

Анализ systemd

Анализируем загрузку групп сервисов

systemd-cgtop

Проверить вхождение в группы – **systemd-cgls /init.scope**

Control Group	Tasks	%CPU	Memory	Input/s	Output/s
/	955	33,8	4.5G	0B	180.3K
user.slice	697	27,0	3.4G	-	-
user.slice/user-1000.slice	697	27,1	3.4G	-	-
user.slice/user-1000.slice/user@1000.service	210	15,3	1.0G	-	-
user.slice/user-1000.slice/session-3.scope	487	11,8	2.4G	-	-
system.slice	104	11,4	577.6M	-	-
system.slice/sddm.service	17	11,3	152.5M	-	-
system.slice/NetworkManager.service	3	0,1	11.3M	-	-
system.slice/chronyd.service	1	0,0	3.2M	-	-
system.slice/dbus-broker.service	2	0,0	6.9M	-	-
system.slice/tuned.service	4	0,0	30.6M	-	-
system.slice/wpa_supplicant.service	1	0,0	4.4M	-	-
system.slice/rtkit-daemon.service	3	0,0	832.0K	-	-
system.slice/systemd-userdbd.service	4	0,0	5.2M	-	-
system.slice/rsyslog.service	3	0,0	2.9M	-	-
init.scope	1	-	57.9M	-	-
system.slice/ModemManager.service	4	-	7.8M	-	-
system.slice/accounts-daemon.service	4	-	3.9M	-	-
system.slice/alsa-state.service	1	-	416.0K	-	-

Анализ systemd

Инструмент анализа загрузки. Позволяет просматривать и изменять настройки управления загрузкой

bootctl — проверка статуса загрузки

```
[dima@denisov ~]$ bootctl
systemd-boot not installed in ESP.
System:
  Firmware: n/a (n/a)
  Secure Boot: disabled
  Setup Mode: user
  Boot into FW: supported

Current Boot Loader:
  Product: n/a
  Features: X Boot counting
           X Menu timeout control
```

Анализ **systemd**

Анализируем скорость загрузки Linux — **systemd-analyze**
время загрузки операционной системы:

systemd-analyze time

большие задержки загрузки:

systemd-analyze blame | head -n 10

графическая карта анализа загрузки Linux:

systemd-analyze plot > boot_analysis.svg

Практическая работа

1. С помощью `systemctl` вывести список всех служб (имеющих тип `service`)
2. Вывести список всех служб, которые не смогли запуститься (состояние `failed`)
3. Посмотреть состояние службы `cups`
4. Остановить службу `cups` и открыть в браузере на его страницу (`localhost:631`)
5. Остановить сокет `cups` и открыть в браузере на его страницу (`localhost:631`)
6. Найти файл `systemd`-службы фаервола — `iptables.service`
7. Изменить в нём описание службы (`Description`)
8. Выполнить перезагрузку конфигурации `systemd`
9. Посмотреть статус службы `iptables` и убедиться что описание изменилось



Спасибо за внимание!

www.red-soft.ru
redos@red-soft.ru

