

systemd

The standard Linux init system

January 22, 2016

Introduction

WHAT IS AN INIT SYSTEM?

An init system is the first userspace process (PID 1) started in a UNIX-like system. It handles:

- Starting system processes and services to prepare the environment
- Adopting and “reaping” orphaned processes

CLASSICAL INIT SYSTEMS

Init systems before systemd - such as SysVinit - were very simple.

- Services and processes to run are organised into “init scripts”
- Scripts are linked to specific runlevels
- Init system is configured to boot into a runlevel

systemd

CAN WE DO BETTER?

- “legacy” init systems have a lot of drawbacks
- Apple is taking a different approach on OS X
- Systemd project was founded to address these issues

SYSTEMD DESIGN GOALS

- Expressing service dependencies
- Monitoring service status
- Enable parallel service startups
- Ease of use

SYSTEMD - THE BASICS

- No scripts are executed, only declarative units
- Units have explicit dependencies
- Processes are supervised
- cgroups are utilised to apply resource limits
- Service logs are managed and centrally queryable
- Much more!

SYSTEMD UNITS

Units specify how and what to start. Several types exist:

- `systemd.service`
- `systemd.target`
- `systemd.timer`
- `systemd.path`
- `systemd.socket`
- `systemd.device`
- `systemd.mount`
- `systemd.swap`
- `systemd.slice`

RESOURCE MANAGEMENT

Systemd utilises Linux **cgroups** for resource management, specifically CPU, disk I/O and memory usage.

- Hierarchical setup of groups makes it easy to limit resources for a set of services
- Units can be attached to a **systemd.slice** for controlling resources for a group of services
- Resource limits can also be specified directly in the unit

Systemd comes with an integrated log management solution, replacing software such as `syslog-ng`.

- All process output is collected in the journal
- `journalctl` tool provides many options for querying and tailing logs
- Children of processes automatically log to the journal as well
- **Caveat:** Hard to learn initially

A variety of CLI-tools exist for managing systemd systems.

- `systemctl`
- `journalctl`
- `systemd-analyze`
- `systemd-cgtop`
- `systemd-cgls`

Let's look at some of them.

Demo

Controversies

Systemd has been heavily criticised, usually focusing around a few points:

- Feature-creep: Systemd absorbs more and more other services

SYSTEMD CRITICISM

systemd Utilities

systemctl journalctl notify analyze cglsg cgtop loginctl nspawn

systemd Daemons

systemd
journalld networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
dbus telephony display service
shutdown reboot dlog logind user-session tizen service

systemd Core

manager unit login namespace log
service timer mount target multiseat inhibit
systemd snapshot path socket swap session pam cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

SYSTEMD CRITICISM

Systemd has been heavily criticised, usually focusing around a few points:

- Feature-creep: Systemd absorbs more and more other services
- Opaque: systemd's inner workings are harder to understand than old `init`
- Unstable: development is quick and breakage happens

SYSTEMD ADOPTION

Systemd was initially adopted by RedHat (and related distributions).

It spread quickly to others, for example ArchLinux.

Debian and Ubuntu were the last major players who decided to adopt it, but not without drama.

Questions?
