



init=/bin/present

Calvin Winkowski



What is init

- init is the first process that runs (pid 1)
- The kernel has a small init “process” that doesn’t count
- Everything practical is ultimately a consequence of init

```
# ls /etc/init.d/
after.local  dhcpd          ntp           rc3.d         snmpd
before.local dhcpd6         postfix       rc4.d         snmptrapd
boot.d       firewall_local postgresql    rc5.d         sshd
boot.dmraid  halt.local    powerd        rc6.d         stunnel
boot.local   ipmi          powerfail     rcS.d         svnserve
boot.md      ipmiseld     raw           rpmconfigcheck tinyproxy
boot.udev    lm_sensors   rc0.d         rsyncd        xymon-client
cron         mdadm        rc1.d         SMagent
dbus         network      rc2.d         SMmonitor
```

Linux processes

- Linux process table contains all processes running on the system
- `#include <linux/sched.h>`

```
struct task_struct(_short) {
    volatile long                state;
    void                          *stack;
    unsigned int                 ptrace;
    unsigned int                 cpu;
    int                           exit_code;
    pid_t                         pid;
    struct task_struct __rcu     *parent;
}
```


Demo!



Everything else

- Populate /dev -- udev, mdev, devfsd, do it yourself with mknod
- Mount file systems -- root, cgroups, proc, nfs, etc.
- Start gettys
- Start X
- Configure network interfaces
- Start logging daemon -- rsyslog or sometimes syslog-ng
- Start various service daemons

State of init on *nix

- systemd dominates linux distros with a few exceptions
- Upstart still has vestiges in old releases of distros
- *BSD package in a “BSD init”
- fleetctl

LoC in inits

- Only including code, not init scripts or unit files
- OpenRC - 16792
- Upstart - 131243
- systemd - 442553

Lennart Poettering: Or How I Learned to Stop Worrying and Love Systemd

- Feature creep
- Attitude of developers
- systemd development is not modular
- A few bad design decisions



systemd feature creep

systemd Utilities

systemctl journalctl notify analyze cglc cgtop loginctl nspawn

systemd Daemons

systemd
journald 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

Things systemd did really right

- Everything uses cgroups
- Unit files are portable and easy to write
- Recognized login needed attention on Linux
- Socket activation to the 21st century
- User daemon management
- Override system unit files
- Really good dependency management
- Daemonizes processes for you
- Trashed runlevels
- Ultimately systemd makes managing a system easier, thus it got adoption
- etc.

The real problem with systemd

- pulseaudio
- systemd
- avahi
- The developer culture is “my way or the highway”
- <https://github.com/tmux/tmux/issues/428>
- <https://lists.fedoraproject.org/pipermail/devel/2012-October/172163.html>
- https://bugs.freedesktop.org/show_bug.cgi?id=74589
- <https://github.com/systemd/systemd/issues/6237>
- <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=825394>
- https://bugs.freedesktop.org/show_bug.cgi?id=64116



References

- <https://utcc.utoronto.ca/~cks/space/blog/linux/SystemdRight>
- <http://ewontfix.com/14/>
- http://without-systemd.org/wiki/index.php/Arguments_against_systemd
- [https://www.agwa.name/blog/post/how to crash systemd in one tweet](https://www.agwa.name/blog/post/how_to_crash_systemd_in_one_tweet)
- [https://wiki.gentoo.org/wiki/Comparison of init systems](https://wiki.gentoo.org/wiki/Comparison_of_init_systems)