systemd for developers

Alison Chaiken
alison@she-devel.com
http://she-devel.com
Feb. 21, 2015

Text in blue is hyperlinked.

On-the-fly audience exercises.

Quiz:

what is the most widely used

Linux init system?

Topics

- Motivation
- *Design* of systemd
- Comparison with sysVinit
- Integration of systemd with kernel features
- Some tips for users and developers

Linux needs to keep innovating

"No one has a guaranteed position in the technology industry." -- Bill Gates, *Pirates of Silicon Valley*

"The only thing that can ever hurt Linux is Linux itself." -- GKH, *Linux Action Show*

"Success is a self-correcting phenomenom." -- Gary Hamel

Design

Philosophy

Extract duplicate functionality from individual daemons and move it to the systemd core or the Linux kernel.

Replace /etc scripts with declarative configuration files in a standard format.

One daemon to rule them all

xinetd: a daemon to lazily launch **internet services** when activity is detected on an AF_INET socket

systemd: a daemon to lazily launch **any system service** when activity is detected on an AF_UNIX socket (oversimplification) systemd is:

- ***modular***;
- ***asynchronous*** and ***concurrent***;
- described by ***declarative*** sets of properties;
- bundled with analysis tools and ***tests;***
- features a fully ***language-agnostic*** API.

sysVinit runlevels ≈ systemd targets

- Targets are *synchronization points* for boot.
- Check */lib/systemd/system/runlevel?.target* symlinks:

  multi-user.target.wants (runlevel 3 == text session)
  graphical.target.wants (runlevel 5 == graphical session)

- Select boot-target :
  - via */etc/systemd/system/default.target* symlink;
  - appending number ('3' or '5') or *systemd.unit*=<*target*> to kernel cmdline;
- Change current target with *runlevel, telinit*
  - or *systemctl isolate <something>.target*

init.d scripts ⇒ systemd units

- Unit's action and parameters: ExecStart=
- Dependencies: Before=, After=, Requires=, Conflicts= and Wants=.
- Default dependencies:
  - Requires= and After= on basic.target;
  - Conflicts= and Before= on shutdown.target.
- Types of unit files: service, socket, device, mount, scope, slice, automount, swap, target, path, timer, snapshot

Understanding dependencies

Try:

```
systemctl list-dependencies basic.target
systemctl list-dependencies –after tmp.mount
```

Understanding dependencies, p. 2

Try:

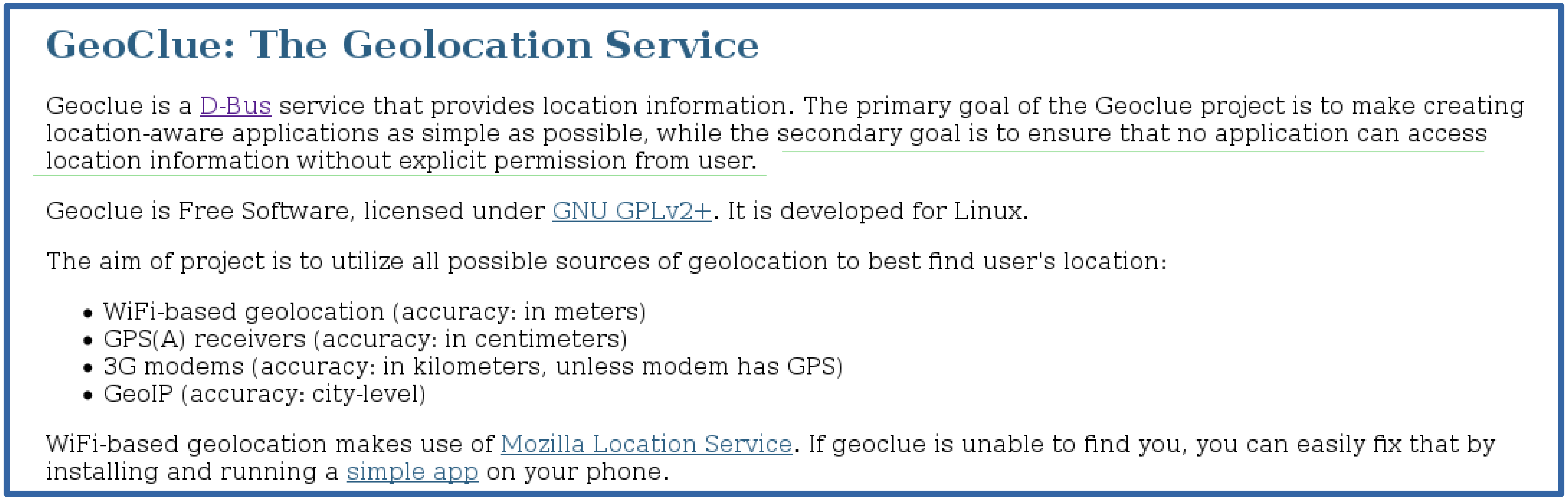
```
systemd-analyze dot  rescue.target


systemd-analyze dot basic.target > basic.dot
dot -Tsvg basic.dot -o basic.svg
eog basic.svg  (or view basic.svg with any web browser)
```

Hierarchy of unit files for system and user sessions

- Organized into *system* and *user* units
- */lib/systemd/system:* systemd upstream defaults for system-wide services
- */etc/system/system:* local customizations by *override* and *extension*
- */lib/systemd/user/*: systemd's upstream defaults for per-user services
- $HOME/.local/share/systemd/user/ for user-installed units
- 'drop-ins' are run-time extensions

[system and user units: gnome-weather demo]



GeoClue: The Geolocation Service

Geoclue is a D-Bus service that provides location information. The primary goal of the Geoclue project is to make creating location-aware applications as simple as possible, while the secondary goal is to ensure that no application can access location information without explicit permission from user.

Geoclue is Free Software, licensed under GNU GPLv2+. It is developed for Linux.

The aim of project is to utilize all possible sources of geolocation to best find user's location:

- WiFi-based geolocation (accuracy: in meters)
- GPS(A) receivers (accuracy: in centimeters)
- 3G modems (accuracy: in kilometers, unless modem has GPS)
- GeoIP (accuracy: city-level)

WiFi-based geolocation makes use of Mozilla Location Service. If geoclue is unable to find you, you can easily fix that by installing and running a simple app on your phone.

sysVinit

systemd

Comparison with sysVinit

SysV already has a big service manager: bash

```
[user@localhost]$ wc -l /sbin/init
    64


[user@localhost]$  wc -l /bin/bash
  4154


[user@localhost]$ wc -l /lib/systemd/systemd
  5944
```

which services are started by sysVinit?

Try: 'ls/etc/init.d'

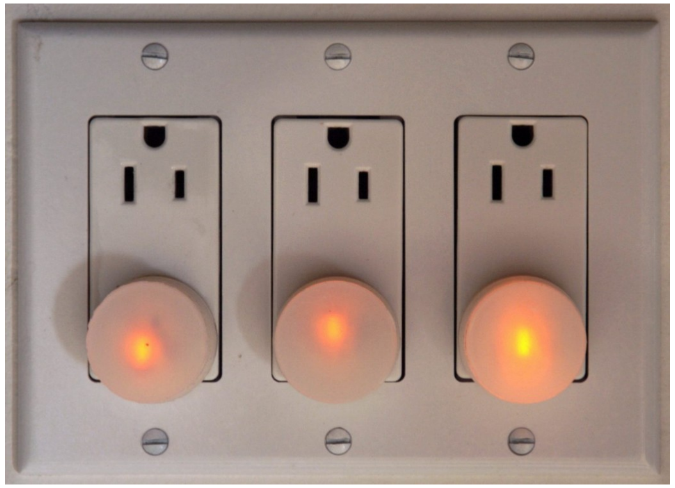
Which daemons started by systemd directly?

Try: 'ls /lib/systemd/system/*.service'

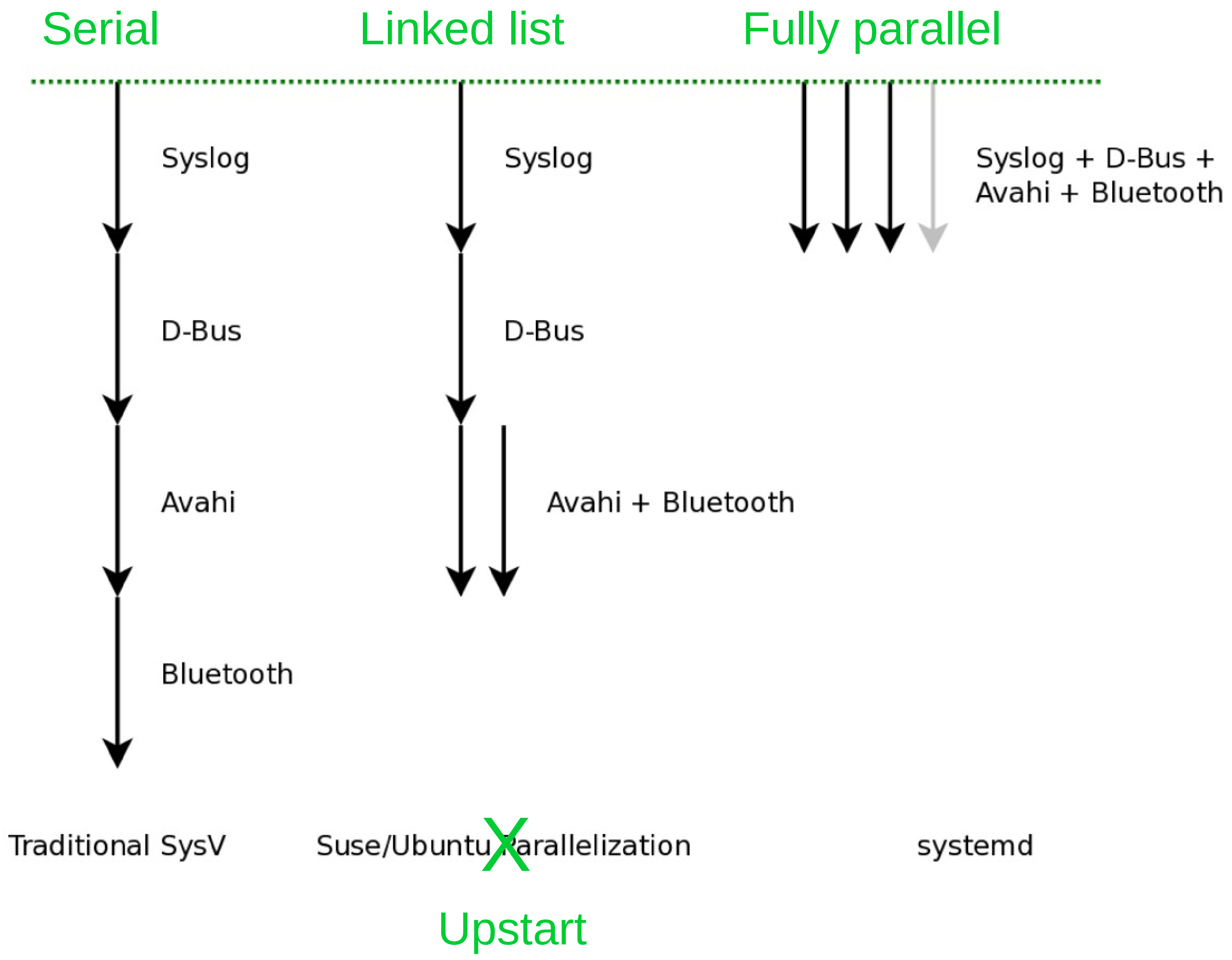
Try: 'systemctl list-sockets'

Major Differences with SysVInit

clean environment



socket-based activation


Serial
Linked list
Fully parallel
Syslog
D-Bus
Avahi
Bluetooth
Syslog
D-Bus
Avahi + Bluetooth
Syslog + D-Bus + Avahi + Bluetooth
Traditional SysV
Suse/Ubuntu Parallelization
X
Upstart
systemd

[Socket activation demo with cups and ncat]

using the systemd journal

- Run “addgroup $USER systemd-journal” for access.
- Can be cryptographically signed.
- Log-reading tools are simple:

```
Try: journalctl -xn
     journalctl -p err
     journalctl  -u cron
     journalctl --list-boots
     systemctl status
     systemctl is-failed bluetooth
     systemctl --failed
```

integration of systemd with kernel features

systemd and cgroups

- cgroups are a kernel-level mechanism for allocating resources like storage, memory, CPU and network
- systemd *slices* are groups of *daemons* whose resources are managed jointly.
- systemd *scopes* are similar groups of *user* processes.
- Can set BlockIOWeight, IOSchedulingPriority, OOMScoreAdjust, CPUShares, MemoryLimit …

Try:

```
sudo systemd-cgls

sudo systemd-cgtop
```

systemd and udev

- udev is a kernel facility that handles device events.
  - merged into the systemd project.
- Rules are enabled by placement in /lib/udev/rules.d, unlike systemd unit enablement.
- Rule loading is ordered by numeric filename prefix, like old sysVinit scripts.

udev is still old-school

Try:

```
ls /lib/udev/rules.d

cat /lib/udev/rules.d/99-systemd.rules
```

systemd and security: granular encapsulationvia kernel's *capabilities*

- PrivateTmp, PrivateDevices, PrivateNetwork
- JoinNamespaces
- ProtectSystem (/usr and /etc), ProtectHome
- ReadOnlyDirectories, InaccessibleDirectories
- systemd-nspawn: systemd's native containers
- Easy configuration of kernel's capability properties

developing systemd

- git clone git://anongit.freedesktop.org/systemd/systemd
- systemd-devel list: submit patches or ask questions
- Impressive and featureful utility library in *src/shared/*

```
#define streq(a,b) (strcmp((a),(b)) == 0)
#define strneq(a, b, n) (strncmp((a), (b), (n)) == 0)
#define strcaseeq(a,b) (strcasecmp((a),(b)) == 0)
#define strncaseeq(a, b, n) (strncasecmp((a), (b), (n)) == 0)
```

- Complex but automated build system with many dependencies.
- 'Plumbing' dev tools in */lib/systemd*, 'porcelain' tools in /bin

```
find /lib/systemd -executable -type f
```

Summary

- Systemd has:
  - a superior design;
  - tight integration with the Linux kernel;
  - a vibrant developer community.
- Control has migrated away from distros toward kernel and freedesktop.org.
- Most users will not notice.
- systemd exemplifies the modernization Linux needs to stay relevant and competitive.

Resources

- Man pages are part of systemd git repo.
- freedesktop.org: systemd mailing list archives and wiki
- At Poettering's 0pointer.de blog
- ➠At wayback machine: “Booting up” articles
- Neil Brown series at LWN
- ➠Fedora's SysVinit to systemd cheatsheet
- Steve Smethurst's Hacker Public Radio episode
- Josh Triplett's Debconf talk video
- Linux Action Show interviews with Mark Shuttleworth and Lennart Poettering

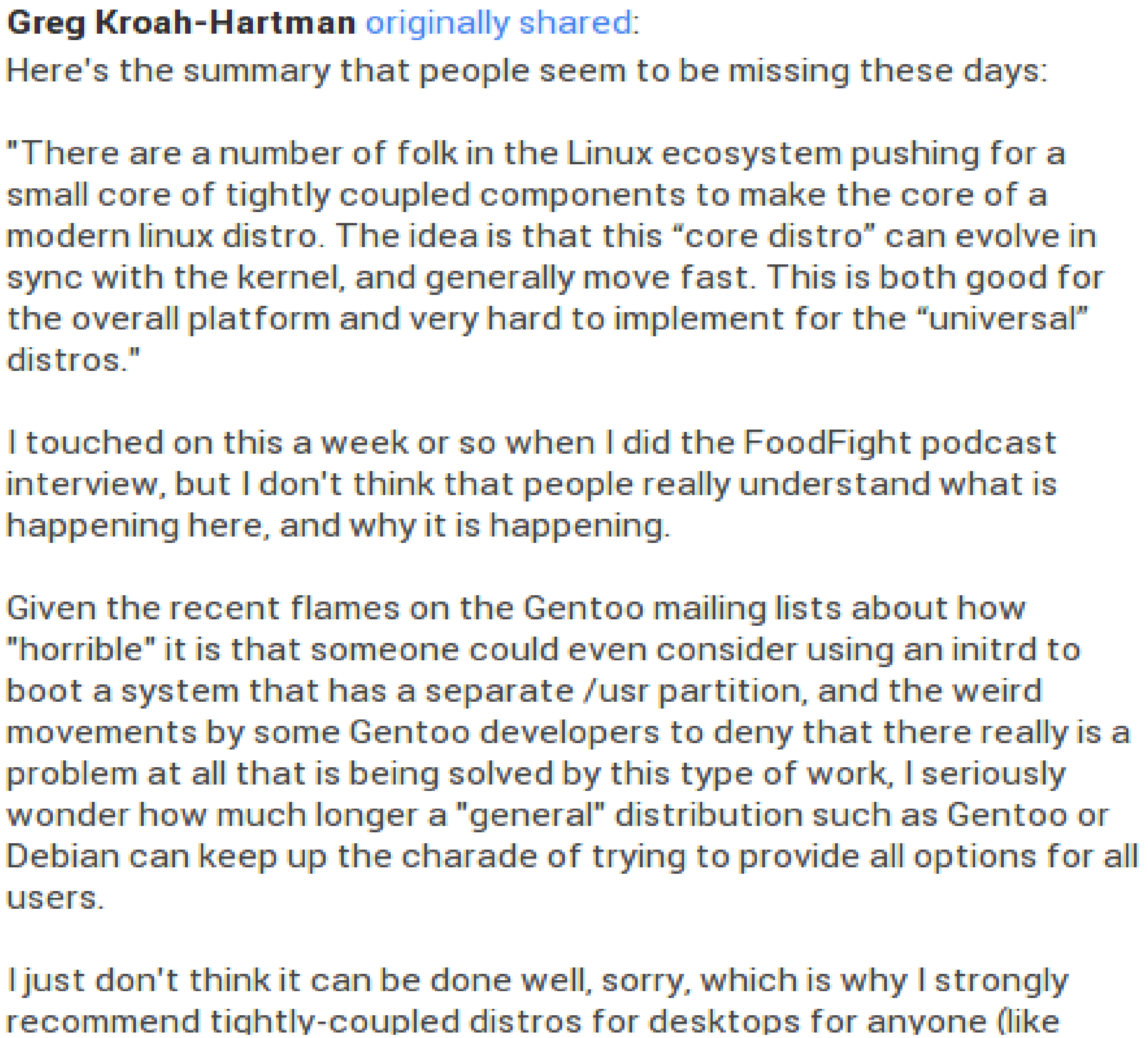
Thanks

- Mentor Graphics for sending me to Germany to hack on systemd.
- Vladimir Pantelic, Tom Gundersen and Lennart Poettering for corrections of an earlier version (without implied 'ack').
- Ivan Shapovalov and Mantas Mikulènas for answering questions.
- Bill Ward and Jym Dyer for use of their images.

photo
courtesy
Jym
Dyer

Leftover Materials

Greg K-H: "Tightly-coupled components"



Greg Kroah-Hartman originally shared:

Here's the summary that people seem to be missing these days:

"There are a number of folk in the Linux ecosystem pushing for a small core of tightly coupled components to make the core of a modern linux distro. The idea is that this "core distro" can evolve in sync with the kernel, and generally move fast. This is both good for the overall platform and very hard to implement for the "universal" distros."

I touched on this a week or so when I did the FoodFight podcast interview, but I don't think that people really understand what is happening here, and why it is happening.

Given the recent flames on the Gentoo mailing lists about how "horrible" it is that someone could even consider using an initrd to boot a system that has a separate /usr partition, and the weird movements by some Gentoo developers to deny that there really is a problem at all that is being solved by this type of work, I seriously wonder how much longer a "general" distribution such as Gentoo or Debian can keep up the charade of trying to provide all options for all users.

I just don't think it can be done well, sorry, which is why I strongly recommend tightly-coupled distros for desktops for anyone (like

photo
courtesy
Bill
Ward

Modularity can produce complexity

systemd and outside projects: CoreOS

- **networkd** was initially contributed by CoreOS developers.
- CoreOS's **fleet** "tool that presents your entire cluster as a single init system" is based on systemd.
  - Spin up new containers due to events on sockets.
- CoreOS devs are outside systemd inner circle.
- systemd has many patches from Arch, Intel, Debian . . .

systemd in embedded systems

- systemd is widely adopted in embedded systems because
  - proper allocation of resources is critical;
  - fastboot is required;
  - customization of boot sequence is common.
- Lack of backward compatibility for older kernels (due to firmware loading) is a pain point.
- Embedded use cases are not always understood by systemd devs.

Try: 'systemctl isolate multi-user.target'
[**warning: KILLS X11**]

[runlevel demo with Fedora Qemu and Firefox]

systemd is easy to use

- systemd utilities:
  - *Try: apropos systemd | grep ctl*
- All-ASCII configuration files: no hidden “registry”.
- Customization is by ***overriding*** default files.
- Many choices are controllable via symlinks.
- Bash-completion by default.
- Backwards compatibility with SysVinit

Override your defaults!

- *Replace* a unit in /lib (upstream) by creating one of the same name in /etc (local changes).
- *Add* services to boot by symlinking them into /etc/systemd/system/default.target.wants.
- 'mask' unit with link to /dev/null.
- *Best practice*: do not change the files in /lib/systemd.
- Read in-use unit with 'systemctl cat'.

photo courtesy
Jym Dyer

Extensions: drop-ins

Try: systemd-delta

Try: systemctl cat <list from 1st command>

Old way	New way	History
X11 manages graphics memory	Kernel's drm manages graphics memory	“Linux Graphics Drivers: an I ntroduction,” p. 26
static /dev, then devfs	udev	
getrlimit, setrlimit	cgroups	
KDE3 and GNOME2	KDE4 and GNOME3	KDE and GNOME
sysVinit	systemd	in progress
X11 client-server model	Wayland compositor	

Crux of the problem: Dave Neary

“There is no freedesktop.org process for proposing standards, identifying those which are proposals and those which are de facto implemented, and perhaps more importantly, there is no process for building consensus around a specification . . .”

(comment regarding GNOME3)

systemd is . . .

- the basis of Fedora, RHEL, CentOS, OpenSUSE, Ubuntu, Debian and much embedded.
- praised by Jordan Hubbard of FreeBSD.
- tightly integrated with Linux kernel cgroups.
- the reference implementation for udev and for kdbus userspace access.

Customizing your installation

- *Replace* a unit in /lib (upstream) by creating one of the same name in /etc (local changes).
- *Add* services to boot by symlinking them into /etc/systemd/system/default.target.wants.
- *Best practice*: do not change the files in /lib/systemd

Sequence of targets on a typical system

```
>$ ls -l /lib/systemd/system/default.target
    /lib/systemd/system/default.target -> graphical.target


>$ cat /lib/systemd/system/graphical.target
    After=multi-user.target


>$ cat /lib/systemd/system/multi-user.target
    After=basic.target


>$ cat /lib/systemd/system/basic.target
   After=sysinit.target sockets.target timers.target paths.target
         slices.target
```

Example: set display manager

```
[user@localhost ~]$ ls -l `locate display-manager.service`

lrwxrwxrwx. 1 root root 35 Dec 11  2013
/etc/systemd/system/display-manager.service ->
/usr/lib/systemd/system/gdm.service
```

```
[user@localhost ~]$ cat /usr/lib/systemd/system/gdm.service
[Unit]
Description=GNOME Display Manager
[ . . . ]
[Install]
Alias=display-manager.service
```

or

```
WantedBy=graphical.target
```

sysinit, sockets and multi-user are composite targets

```
>$ ls /lib/systemd/system/multi-user.target.wants/
    dbus.service@  systemd-ask-password-wall.path@  systemd-
    update-utmp-runlevel.service@ getty.target@

>$ ls /lib/systemd/system/sockets.target.wants:
    dbus.socket@                    systemd-shutdownd.socket@
    systemd-initctl.socket@           systemd-udevd-control.socket@

>$ ls /lib/systemd/system/sysinit.target.wants:
    cryptsetup.target@                systemd-journald.service@
    debian-fixup.service@              systemd-journal-flush.service@
```

Symlinks replace lines of conditional code in SysVinit scripts.

Example: change the default target

```
[alison@localhost ~]$ ls /etc/systemd/system/default.target
/etc/systemd/system/default.target ->
       /lib/systemd/system/graphical.target

[alison@localhost ~]$ sudo rm /etc/systemd/system/default.target
[alison@localhost ~]$ sudo ln -s /lib/systemd/system/multi-user.target
   /etc/systemd/system/default.target

[alison@localhost ~]$ ~/bin/systemd-delta
[ . . . ]
[REDIRECTED]   /etc/systemd/system/default.target →
   /usr/lib/systemd/system/default.target
```

Misconceptions

- systemd is more complex than sysVinit.
- systemd is full of binary configuration files.
- The system log is now unreadable!   And liable to corruption!
- {Fedora/GNOME/RedHat/Poettering} are trying to take over all of Linux.

problems

- systemd *is* modular, but:
  - interopability with other SW may be inadequately tested.
- Potentially rocky piecemeal transition by distros.
  - e.g., Debian installer doesn't warn about a separate /usr partition.
- Merciless deprecation of features (firmware loading, readahead . . .).
- Frequent releases, not particularly stable.

Taxonomy of systemd dependencies

Requires, RequiresOverridable, Requisite, RequisiteOverridable,
Wants, BindsTo, PartOf, Conflicts, Before, After, OnFailure
PropagateReloadsTo, ReloadPropagateFrom,