UNDERSTANDING SYSTEMD

THE MODERN LINUX SERVICE MANAGER

GARRET ARCORACI,
ROCHESTER INST. OF TECHNOLOGY

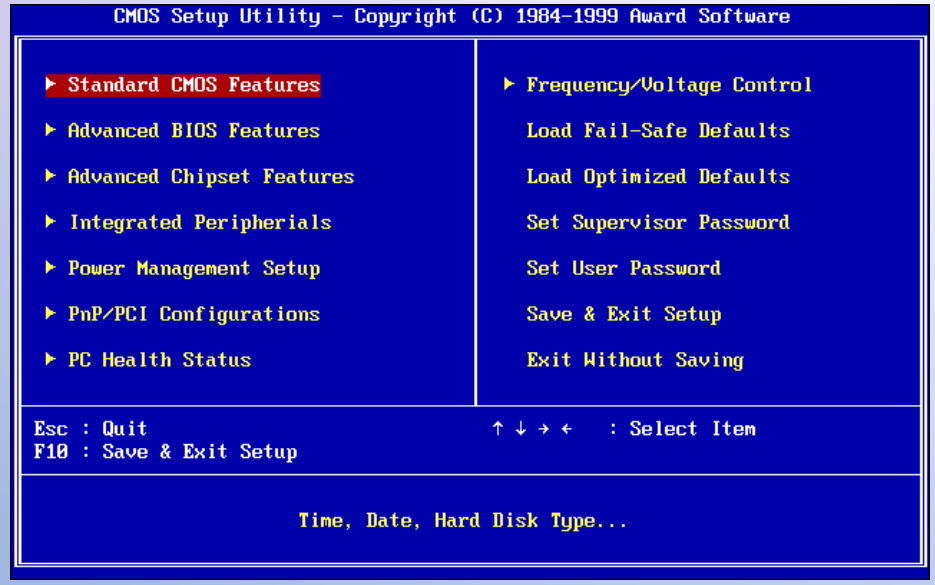
INTRODUCTION

- ME AND SYSTEMD
- WHERE DOES SYSTEMD FIT

IN THE BEGINNING…

http://www.uefi.org/

TRADITIONAL BIOS



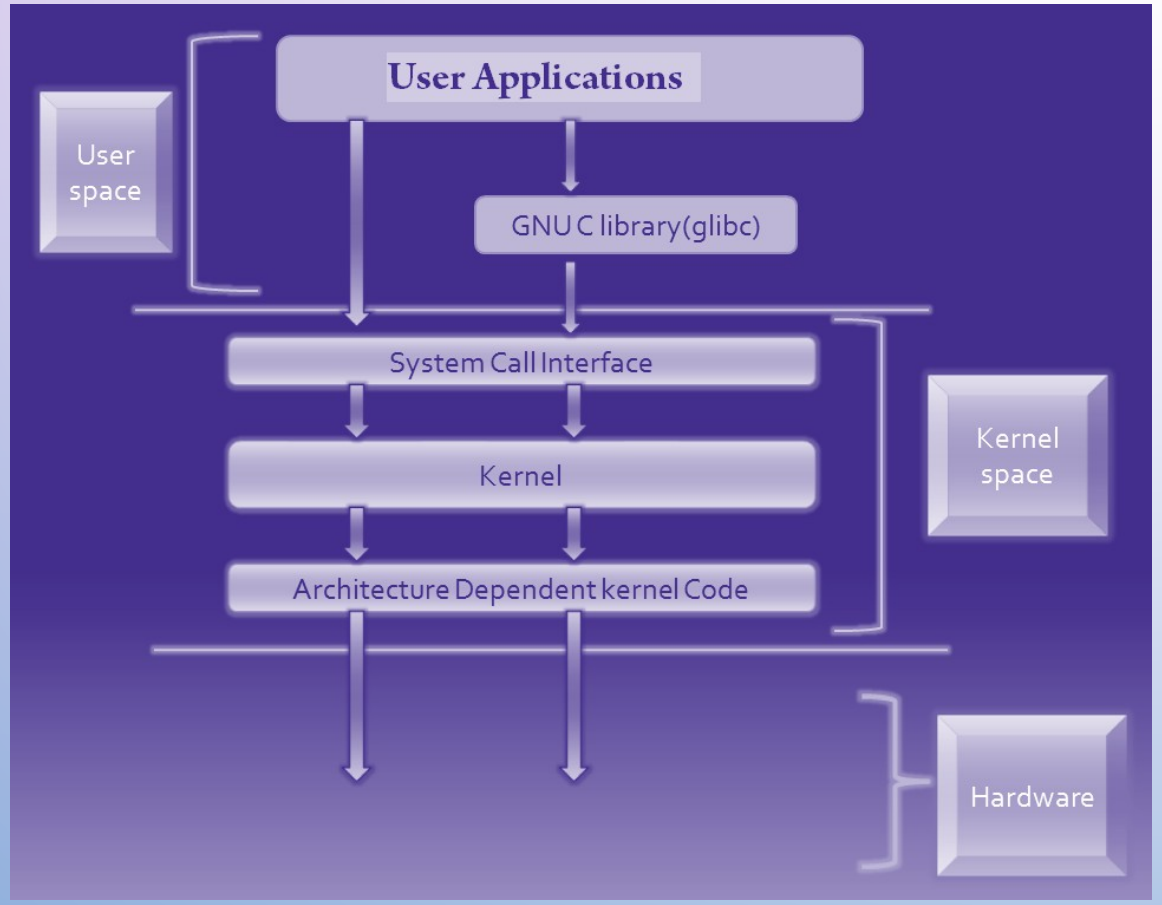
UEFI

WHERE IS THE OS?

BOOT LOADER

Chainload into GRUB 2
When you have verified GRUB 2 works, you can use this command to complete the upgrade: upgrade-from-grub-legacy
Debian GNU/Linux, kernel 2.6.28-11-generic
Debian GNU/Linux, kernel 2.6.28-11-generic (recovery mode)
Debian GNU/Linux, kernel memtest86+
Other operating systems:
Windows Vista (loader)
Use the ↑ and ↓ keys to sele
Press enter to boot the
commands before boo
Select operating system
TecMint.com
Linux Howto's Guide
Best Linux Boot Loaders
GNU GRUB version 0.97 (638K lower / 52217
Fedora (2.6.21-1.3194.fc7)
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line
fedora
Mageia Linux
Advanced options for Mageia Linux
Microsoft Windows XP Professional (on /dev/sda1)
Mageia 2 (2)
Advanced options for Mageia 2 (2)
Mageia 3 (3)
Advanced options for Mageia 3 (3)
Mageia 3 (3)
Advanced options for Mageia 3 (3)
Mageia 3 (3)
Advanced options for Mageia 3 (3)
Advanced options for Mageia 3 (3)
Mageia Grub 2 Menu
mageia
Select an item with the arrow keys and press Enter to boot.
Press 'c' for command line, 'e' to edit.
GNU GRUB
Ubuntu, with Linux 2.6.38-8-generic
Ubuntu, with Linux 2.6.38-8-generic (recovery mode)
Chainload to rEFIt
Chainload to ELILO

STARTS THE KERNEL



Initialize the File System in RAM `initramfs`

LET'S REVIEW

UEFI/ BIOS

- Firmware Interface
- Find bootable Disk

Boot Loader

- GRUB 2

Kernel

- Initramfs
- Loads drivers

init

- **Systemd**

Everyth ing

- System Prerequisites
- Services
- Shell

THE USER SPACE

- What gets loaded…
  - Low-level services that must always be running, such as udevd, syslog, file system mounts (fstab)
  - Network configurations are loaded
  - High-Level services, such as cron, sssd, cups, or a web service
  - Presented with the login prompt
- The order is important because sometimes serial loading is required for dependencies
- Parallel loading can be used to speed up certain services

INIT, UPSTART, & SYSTEMD

- Once the kernel has loaded, it searches the init process /sbin/init
- The init process is responsible for starting the user space environment
- Traditionally, the system V init procedure was used to start services
- /sbin/init can be linked to Upstart or systemd
- The intent of using upstart and systemd was to make the loading system configuration settings more efficient

SYSTEMD

Date Wed, 2 Apr 2014 11:57:41 -0700
Subject Re: [RFC PATCH] cmdline: Hide "debug" from /proc/cmdline
From Linus Torvalds <>

On Wed, Apr 2, 2014 at 11:42 AM, Steven Rostedt <rostedt@goodmis.org> wrote:
>
> The response is:
>
> "Generic terms are generic, not the first user owns them."

And by "their" you mean Kay Sievers.

Key, I'm f*cking tired of the fact that you don't fix problems in the code *you* write, so that the kernel then has to work around the problems you cause.

Greg - just for your information, I will *not* be merging any code from Kay into the kernel until this constant pattern is fixed.

This has been going on for *years*, and doesn't seem to be getting any better. This is relevant to you because I have seen you talk about the kdbus patches, and this is a heads-up that you need to keep them separate from other work. Let distributions merge it as they need to and maybe we can merge it once it has been proven to be stable by whatever distro that was willing to play games with the developers.

But I'm not willing to merge something where the maintainer is known to not care about bugs and regressions and then forces people in other projects to fix their project. Because I am *not* willing to take patches from people who don't clean up after their problems, and don't admit that it's their problem to fix.

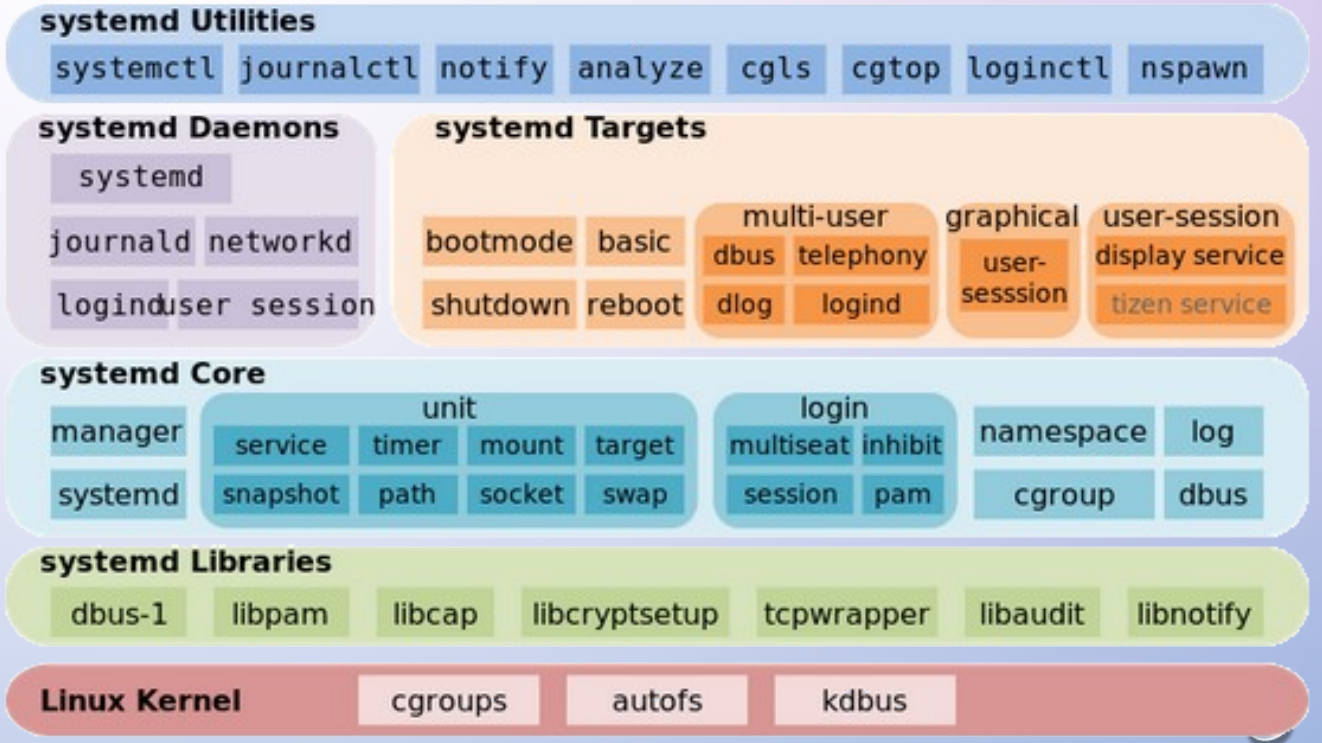
Kay - one more time: you caused the problem, you need to fix it. None of this "I can do whatever I want, others have to clean up after me" crap.

Linus

systemd is about managing stuff!

Goals

- Provide a unified OS to run on top of the Linux kernel.
- Systemd is control!
- Improves dependency requirements for all services.
- Systemd is the current standard on all major modern Linux distributions.
- Standardize the scripts.



UNIT FILES

- Systemd unit files defines what needs to be started
- Different types of unit files exist
  - Service
  - Mount
  - Timer
  - Automount
  - Target
  - Path
- Unit files are stored in **/usr/lib/systemd/system** and /etc/systemd/system

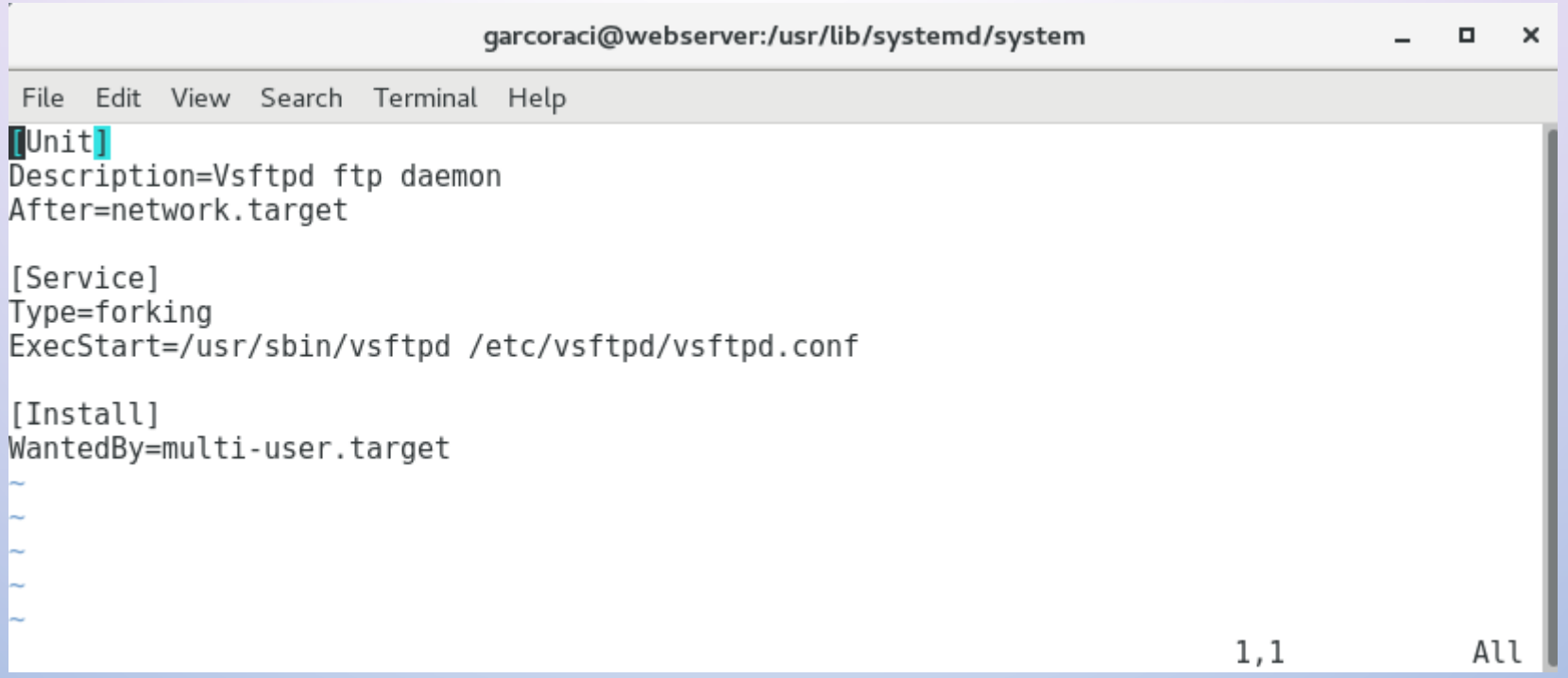
UNIT TYPES

Unit Type	Description
service	A system service
target	A group of systemd units
automount	A file system automount point
device	A device file recognized by the kernel
mount	A file system mount point
path	A file or directory in a file systems
scope	An externally created process
slice	A group of hierarchically organized units that manage system processes
snapshot	A save state of the systemd manager
socket	An inter-process communication socket
swap	A swap device of a swap file
timer	**A systemd timer**

DEPENDENCIES

- SYSTEMD CATEGORIZES DEPENDENCIES
  - REQUIRES – DEFINES UNITS THAT MUST BE LOADED TO LOAD THIS UNIT
  - WANTS – TARGETS THAT DEFINE WHICH UNITS SHOULD BE LOADED BUT IF NOT DO NOT CAUSE THIS UNIT TO FAIL
  - REQUISITE – A DEFINED UNIT MUST BE ACTIVE OTHERWISE THIS UNIT FAILS TO LOAD
  - CONFLICTS – A UNIT MAY NEVER BE ACTIVE WHEN THIS UNIT IS LOADED
  - BEFORE – THE CURRENT UNTIL WILL ACTIVATE BEFORE THE LISTED UNITS
  - AFTER – THE CURRENT UNTIL WILL ACTIVATE AFTER THE LISTED UNITS

EXAMPLE UNIT FILE



FEATURES/BENEFITS

- Logging – all messages are stored in the new systemd journal
- Dependencies – an explicit set of dependencies is defined for each service
- Cgroup – allows for every piece or part of a service to be managed
- Services – Services can be started based on needs
- Units – systemd can manage various units types
- Resource Management – more control over resources used by each service

RUN LEVELS

- RUN LEVELS DEFINE THE STATE A SERVER SHOULD BOOT IN AND DETERMINE WHICH SERVICES MUST BE LOADED.

Common Commands

- `systemctl stop`
- `systemctl start`
- `systemctl enable`
- `systemctl disable`
- `systemctl list-dependencies`
- `systemctl list-units --type service`
- `systemctl list-unit-files`