

Your Computer; Your Init; Your Choice

By Steve Litt

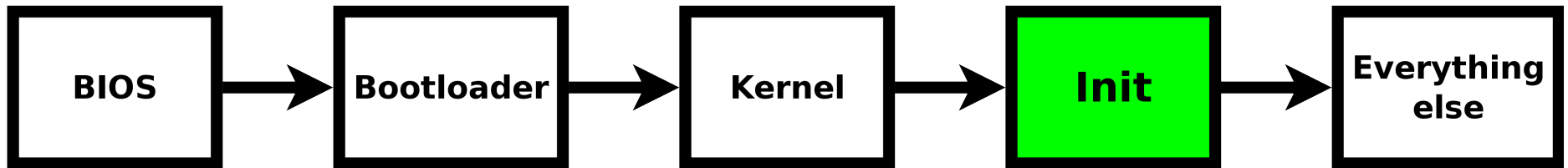
Version 20150108_1348

Copyright © 2015 by Steve Litt
Creative Commons Attribution-NoDerivatives 4.0 International License
<http://creativecommons.org/licenses/by-nd/4.0/legalcode>

Available online at
http://www.troubleshooters.com/linux/presentations/golug_inits/golug_inits.pdf

NO WARRANTY, use at your own risk.

System Overview



- Kernel runs one program, init.
- Everything else run directly or indirectly by init.

Many Different Init Systems

- Epoch
 - OpenRC
 - RichFelker
 - s6
 - sysvinit
 - uselesssd
 - nosh
 - perp
 - runit
 - systemd
 - Upstart
- Many more
 - There's an init for every situation
 - You can make your own

Full vs Partial

- Kernel->full-init at PID1->daemons
 - Systemd, sysvinit, runit, Epoch, Upstart, etc.
- Kernel->PID1->partial-init->daemons
 - OpenRC, daemontools, damontools-encore, etc.

Many Features

- Socket Activation
 - Event controlled
 - Daemontools-like
 - Simplicity
 - Descriptive config
 - Script config
 - Parallel starting
 - Sequential starting
 - Numeric ordering
 - Dependency ordering
 - Work with sysvinit scripts
 - OS toolkit
- Forget features
 - Look for benefits that fit your priorities and situation

Many Routes to Benefits

- Within and outside of init
- With or without sockets
- With or without packaging
- Cutting edge or oldschool

Bogus Characterizations

- ___ is a toy.
 - What does that even mean?
- ___ is not ready for prime time.
 - What does that even mean?
- ___ has an April Fools Joke architecture.
 - What does that even mean?
- ___ lacks features A, B and C.
 - Different inits deliver benefits using different features.
 - Some benefits aren't important in your use case.

YOU Choose Your Init

- It's YOUR computer.
- Keep the default if you like it
- Replace or jumper the default if you don't.
 - Performance
 - Technology
 - Politics
- You owe nobody an explanation

Typical Init Architecture

- Stage 1: Bring up computer
 - No daemons
 - Don't block!
- Stage 2: Manage daemons and ttys
 - Reap zombies
 - Listen for Ctrl+Alt+Del
- Stage 3: Shut down / computer
 - Don't block!

Daemontools Style Inits

- Good stuff!
- Daemon runs in foreground
- Can log stdout and stderr, with timestamps
- Most can parallel start services
- Very simple run scripts
- Examples
 - runit
 - nosh
 - s6
 - perp

Ways to Switch Inits

- Replacement
 - Usually done by packaging
 - When such packaging exists, at will of packagers
 - Removes old init
- Jumper
 - Usually done by compile and copy
 - On almost any distro
 - Old init remains functional if called
 - Additional init can be a huge advantage

Init Jumpering Philosophy

- One way or another, I'll find a way.
 - This almost always pans out.
- Shellscripts are your friend.
 - Test at command prompt.
 - When tested, call from init.
- YOU are CUSTOM building this.
- Simple systems are easier to construct.
- It's amazing how simple you can make it when you eliminate the edge case CYA fluff.

Epoch Init System

- <http://universe2.us/epoch.html>
 - Contents links on the bottom
- <http://universe2.us/collector>
 - Scripts and other stuff.
- Easiest to jumper
- Best docs
- Does respawning:
 - Optionally, per service
- No stage separation

Epoch, More Properties

- Single File conf
- Sequential load
- Numeric order

Epoch Installation

- Be root
- cd /root
- Make sure gcc is installed
- Download latest release tarball
 - From bottom of <http://universe2.us/epoch.html>
- tar xzvf epoch_version_no.tgz
- cd epoch-version_no
- ./buildepoch.sh
- cp -p built/sbin/epoch /usr/bin/epoch
 - Or /sbin if it's not a symlink
- cp -p built/sbin/epoch /e

Epoch Installation Observations

- Builds under /root
- No args needed for buildepoch
- Copy to install
- Adding init=/e inits via Epoch
 - After you make epoch.conf

Hello World epoch.conf

BootBannerText=Sorry, Lennart!
BootBannerColor=CYAN

Hostname=FILE /etc/hostname
DefaultRunlevel=boot
EnableLogging=true
DisableCAD=true
BlankLogOnBoot=true
MountVirtual=procfs sysfs devpts+ devshm+

DefinePriority=**Early_Getty_Start** 40

ObjectID=getty2
ObjectDescription=Early getty on /dev/tty2
ObjectStartCommand=agetty tty2 &
ObjectStopCommand=NONE
ObjectStartPriority=**Early_Getty_Start**
ObjectStopPriority=0
ObjectEnabled=true
ObjectOptions=SERVICE AUTORESTART
ObjectRunlevels=boot

Hello World Observations

- Always use DefinePriority
- tty2 can repeatedly log in and out
- Change command if computer lacks agetty
- No network, sound, or X capabilities
- Reboot with epoch reboot
- Poweroff with epoch poweroff

Full Epoch: Start/Stop ordering

```
DefinePriority=Udev_Start 20
DefinePriority=Rwfs_Start 30
DefinePriority=Sysclock_Start 35
DefinePriority=Early_Getty_Start 40
DefinePriority=Mountall_Start 45
DefinePriority=Network_Start 50
DefinePriority=Wpa_Sup_Start 53
DefinePriority=Dh_Cli_Start 56
DefinePriority=Dev_Mixer_Start 60
DefinePriority=Typical_Daemon_Start 80
DefinePriority=Typical_Getty_Start 70
DefinePriority=Display_Manager_Start 120
```

```
DefinePriority=Typical_Daemon_Stop 10
DefinePriority=Dh_Cli_Stop 20
DefinePriority=Wpa_Sup_Stop 21
DefinePriority=Display_Manager_Stop 40
DefinePriority=Sysclock_Stop 80
DefinePriority=Killall5_Soft_Stop 90
DefinePriority=Killall5_Stop 91
DefinePriority=Rwfs_Stop 100
DefinePriority=Mountall_Stop 101
```

- Hey, where's NetworkManager?
- Where's dbus?
- That's why it starts wpa_supplicant and dhcpcd.

Starting udev

```
ObjectID=udev  
ObjectDescription=Starting udev  
ObjectStartCommand=/root/udev_start.sh  
ObjectStopCommand=NONE  
ObjectStartPriority=Udev_Start  
ObjectStopPriority=0  
ObjectEnabled=true  
ObjectOptions=RAWDESCRIPTION  
ObjectRunlevels=boot
```

```
DefinePriority=Udev_Start 20
```

```
#!/bin/sh  
/usr/lib/systemd/systemd-udev --daemon  
/usr/bin/udevadm trigger --action=add --type=subsystems  
/usr/bin/udevadm trigger --action=add --type=devices  
/usr/bin/udevadm settle
```

- Can swap systemd-udev for vdevd later on.
- Udev necessary for network devices

Starting Wifi Network Device

ObjectID=network

ObjectDescription=Setting up net devices and the network

ObjectStartCommand=/root/upnet.sh

ObjectStopCommand=NONE

ObjectStartPriority=Network_Start

ObjectStopPriority=0

ObjectEnabled=true

ObjectOptions=RAWDESCRIPTION

ObjectRunlevels=boot

```
hostname -F /etc/hostname
```

```
ip link set dev lo up
```

```
ip link set dev wlo1 down
```

```
ip link set dev wlo1 up
```

- Requires run udev or install drivers first
- Distro independent

Starting wpa_supplicant

ObjectID=wpa_supplicant

ObjectDescription=wpa_supplicant

ObjectStartCommand=/usr/bin/wpa_supplicant -B -iwlo1 -c /etc/wpa_supplicant/wpa_supplicant.conf

ObjectStopCommand=killall wpa_supplicant

ObjectStartPriority=Wpa_Sup_Start

ObjectStopPriority=Wpa_Sup_Stop

ObjectOptions=SERVICE AUTORESTART

ObjectEnabled=true

ObjectRunlevels=boot

- Links up wifi (wlo1) with ssid in config file
- Dbus-free substitute for NetworkManager
- Does not obtain an IP address
- Runs as a respawning daemon

Starting dhcpcd

```
ObjectID=dhclient
  ObjectDescription=dhclient
  ObjectStartCommand=/usr/bin/dhclient wlo1
  ObjectStopCommand=PIDFILE /run/dhclient.pid
  ObjectStartPriority=Dh_Cli_Start
  ObjectStopPriority=Dh_Cli_Stop
  ObjectOptions=SERVICE AUTORESTART
  ObjectEnabled=true
  ObjectRunlevels=boot
```

- Wifi device must have been started first
- Runs as a respawning daemon

Starting sshd

```
ObjectID=sshd
  ObjectDescription=Manage sshd daemon
  ObjectStartCommand=/usr/sbin/sshd
  ObjectStopCommand=PIDFILE /run/sshd.pid
  ObjectStartPriority=Typical_Daemon_Start
  ObjectStopPriority=Typical_Daemon_Stop
  ObjectEnabled=true
  ObjectRunlevels=boot
  ObjectOptions=SERVICE AUTORESTART
```

- Same as others
- Runs as respawning daemon
- Wifi must be configged and have IP address

Troubleshooting Epoch

- Docs at <http://universe2.us/epoch.html>
- `/var/log/system.log` provides boot log.
- Get things running on the command line first.
- `epoch status [objectid]`
- `epoch getpid objectid`
- `epoch configreload`
- `ObjectOptions=STOPFAILCRITICAL`
- `ObjectOptions=STARTFAILCRITICAL`

Summary

- Many init programs
- Your choice
 - Keep or switch: Your choice
 - Choose for your use case
 - You owe nobody any explanations
- Replace or jumper
- Daemontools-style have many advantages
- Epoch is easiest to jumper
 - Great as main or emergency B init