# An introduction to Control Groups (cgroups)

(plus some systemd evangelizing)

# Who am I?

Jonathan maw (not James Thomas)

jonathan.maw@codethink.co.uk

Responsibilities:

- GENIVI Node Startup Controller
- AGL Distro OS/Common Libs maintainer

Automotive experience: since June 2012

cgroup experience: since Sep 01 2015

# What are cgroups?

hierarchical grouping of processes managed by the linux kernel, and exposed through a special filesystem

# cat /sys/fs/cgroup/systemd/system.slice/ssh.service/tasks
622

(systemd-cgls)

/sys/fs/cgroup/
  …
↳blkio
↳cpu
↳memory
↳net_cls
↳systemd
  …
  ↳user.slice
  ↳system.slice
    …
    ↳ssh.service
      ↳cgroup.clone_children
      ↳cgroup.procs
      ↳notify_on_release
      ↳tasks

# Why use cgroups?

subsystems/controllers
https://www.kernel.org/doc/Documentation/cgroups/

Lots of features, the most useful ones I see:

- Control memory usage
- Control how much CPU time is allocated
- Control how much device I/O is allowed
- Control which devices can be accessed

Horror story: memory leak in browser kills system

/sys/fs/cgroup/

…
↳**blkio**
↳**cpu**
↳**memory**
↳**devices**
↳systemd
…
↳user.slice
↳system.slice
…
↳ssh.service
↳cgroup.clone_children
↳cgroup.procs
↳notify_on_release
↳tasks

# Why I’d recommend systemd

Systemd uses cgroups to organise processes (each service is a cgroup, and all processes started by that service use that cgroup)

Systemd handles blkio, cpu, device, and memory accounting for you (http://man7.org/linux/man-pages/man5/systemd.cgroup.5.html)

```
[Service]
ExecStart=/bin/foo
MemoryAccounting=true
MemoryLimit=400K
```

(also systemd-cgtop, systemd-cgls)

# Demonstration

top -d1

echo “+1000” > /proc/$(pidof top)/oom_score_adj
./memory-hog
(OOM kills top, then memory-hog)

cat >/etc/systemd/system/memory-hog.service <<EOF
[Unit]
Description=Memory Hog Service
[Service]
ExecStart=/home/user/memory-hog
MemoryAccounting=true
MemoryLimit=400K

top -d1

echo “+1000” > /proc/$(pidof top)/oom_score_adj

systemctl start memory-hog
(OOM kills memory-hog)

# An example hierarchy for an HMI

system.slice -> weston
dbus
can-message-app
hmiapp.slice -> clock.service
map.service
navigation.service
rtaudio.slice -> alerts.service
music.service
navigation-assistant.service
rt.slice -> rearview.service

# Why not?

systemd:

- Doesn’t do every feature of cgroups.
- e.g. if you want cpu scheduling, you may want to disable systemd’s use of the CPU controller, and handle CPU scheduling, yourself.
- If you want network priority, you’ll have to handle it yourself.
- alternative: http://libcg.sourceforge.net/

cgroups:

- the subsystem controllers (memory, etc.) have a performance cost - is it an acceptable cost?