Colin Guthrie (Tribalogic Ltd), Timo Müller (BMW Car IT GmbH)

SMART SYSTEM SHUTDOWN.

How systemd inhibitors can be used to handle eCall and telephony scenarios?

BMW GROUP
BMW Car IT GmbH

TALKING TO YOUR FRIEND.
REGULAR PHONECALL.

MEDICAL EMERGENCY.
MANUAL eCALL.

CAR ACCIDENT.
AUTOMATIC eCALL.

HOW CAN WE USE SYSTEMD IN THESE CASES? LET'S HAVE A LOOK AT INHIBITORS.

RESOURCES.

www.press.bmwgroup.com

systemd: controlling stopping distance

Colin Guthrie
Mageia Contributor

Who am I

- Linux enthusiast for many years
- Contributor to Mandriva and co-founder of community-run Mageia
- Was responsible for X and Sound in Mandriva
- Upstream PulseAudio maintainer
- Now responsible for boot+init in Mageia
- Integrated systemd
- Involved in upstream and attend several conferences and hackfests

What is systemd

- Modern, stateful init system (PID 1)
- But more!
- Base toolkit for building an OS
- Essential building blocks for bootstrapping any modern linux system

PID 1

- Kernel hands over control
- Responsible for starting and stopping subsystems
- And, with various constraints and time limits, for shutting down the system cleanly

Inhibitors

- Ability to block certain system operations
  - Shutdown
  - Sleep
  - Hardware-led actions (Lids on Laptops, Suspend Keys etc)
- NOT part of PID 1
- Implemented on top of systemd as in logind

logind

- Proxy for many raw operations of systemd with logic and authorisation (polkit) taken into account
- Tracks user sessions – who is active and who has access to various h/w resources (via udev/uaccess)
- High level concepts not considered core part of PID1 e.g. inhibiting various operations

Inhibitors

- Important point: For the system to work with inhibitors, all operations must go via logind
  - If something uses PID1 directly to shutdown, inhibitors will be ignored
- Two types of inhibitors
  - Delay
  - Block

Inhibitors: delay

- Only applicable to sleep and hibernate
- Allows callbacks to be triggered before state is entered
  - Instant Messaging clients can logout/set “away”
  - Lock screens can lock screen before suspending!
- Subject to a (global) InhibitDelayMaxUSec setting (logind.conf) which defaults to 5s

Inhibitors: block

- Prevent native handling of operation
  - i.e. stop logind proxying request to PID1
- In Desktop context, DE may want to handle lid-switch on laptops to implement it's own high-level policy
  - e.g. do not suspend on lid closure if external monitor is attached
  - logind default is to “play it safe”

Handling calls

- Ensure all shutdown triggers go via logind – not directly asking PID1
- When call in in progress request a “block” inhibit
  - Recommended: via DBus
  - `systemd-inhibit --mode block --what shutdown:sleep:idle --who “Telephony” --why “Call in progress”`

Handling calls

- To block or delay?
- Block calls will simply block while the lock file descriptor is open – no callback i.e. no way to know whether a shutdown request has come in during call, and thus no way to continue it after call has ended
- Delay of 5s is barely enough time to emit a blood curdling scream before the call ends!
- Delay probably still more appropriate

Loooong delay?

- Possibly increase default delay to very long value (longest possible phone call)
- Is it too long for other uses of delay inhibitors (i.e. do you trust the other code not to block too long?)
- Perhaps (slight) API change needed to allow requesting a per-inhibition delay up to the Max value to protect other, less trustworthy code

Implementation

- PrepareForShutdown() & PrepareForSleep()
  - Callbacks triggered at appropriate times
- Could only inhibit when call is initiated
  - Would generally ignore callback
- Could inhibit at all times (from boot)
  - Callback could do some checks and close fd if no call in progress
  - Retake it later on resume (or GOTO 10 on boot)

Resources

- man (1) systemd-inhibit
- freedesktop.org/wiki/Software/systemd/inhibit
- systemd-devel@lists.freedesktop.org

Questions?

Any questions?