

Safety Critical Software: Would You Let Linux Drive *Your* Car?



Birds of a Feather Session, LPC 2013

Open source software is sweet...

...too sweet to pass up, even for safety critical apps!

- Medical systems
- Self-driving cars
- UAVs
- Rockets
- Firearms
- Robots

Developing safety-critical software

- There are tried-and-true approaches to developing safety-critical software, but...
- The open source community doesn't use them to develop software

Is open source software reliable enough to risk using it when lives are at stake?

Answer #1: No!

- You need to know whether your application will work out of the box
- Without following strict processes, you can't predict whether the software will work

Answer #2: Yes!

- Open source software has over a billion systems deployed (>900 million Android alone).
- Those of us responsible for fixing the bugs know how rarely it fails.
- Lots of work to do (specifics later)

If lives depend on it, you have to know your product works when you deploy it!

What – me worry?

- Our lives are going to depend on it
- Insurance companies
- We're engineers—it's a cool challenge



What to do?

- Reliability is an attribute of a whole system:
 - Reliability measurements on open source portions (integrate with application-specific pieces)
 - Frameworks: logging, reporting, software restart
 - Interfaces to hardware
 - Development/testing methodologies

Diversity is strength

Open source software overwhelmingly beats proprietary software in breadth of deployments

- Software and hardware configurations
- Tools
- Processes
- Versions
- Application software

Problems exposed:

- Execution timing
- C language “gray” areas
- Configuration interactions
- Resource management
- Hardware specifications
- And much more...

Open source is like ball bearings

- Standard statistical techniques measure the impact of variations on reliability
- Statistical analyses quantify how each variable affects reliability
- **Survival analysis**, widely used in industrial settings, provides predictions for:
 - How long a system will work before crashing—Mean time to failure (MTTF)
 - How long it takes to reboot—Mean time to repair (MTTR) (vital for redundant systems built from open source subsystems)

You get what you measure

- Process:
 - Collect data (just open source components, standard formats, anonymized, etc.)
 - Crunch data
 - Publish analyses
 - Flame, think, talk, study, learn, code
 - Repeat, until perfect

What you get when you measure

- One-of-a-kind systems likely to resemble existing systems
- Applications can choose to resemble known-good applications
- Ultimately:
 - Better predictability, which is great
 - Better reliability, which is even better

What can you do?

- Logging and reporting infrastructure
 - User space crash reporting library
 - User space application crash monitor
 - Kernel space panic/restart log
 - Log forwarder/collector--network and on-device storage for returned devices
- Number crunching
 - Log parser/categorizer
 - Statistics analyzer
- Web site to disseminate information

Some more things you can do

- Frameworks
 - For restartable computing: checkpoint/restart framework
 - For control applications: sensor/actuator integration framework
 - Encourage subsystems with more predictable characteristics, e.g. earliest deadline first (EDF) scheduler.
- Hardware
 - What are the requirements on hardware for when software fails?
 - What gets handled by hardware, what by software (cache error repair vs. reboot, etc)
 - When will a soft reboot work and when must you power cycle?
 - Error checking/correction (ECC, parity, nothing)

Even more things to do...

- Better define goals
- Encourage data sharing by companies/organizations, for example: Google, Samsung, Motorola, Cisco, DARPA, NASA, Ford, Toyota, SpaceX, OSADL, LF Carrier/Auto Grade Linux Workgroups, OpenDO, your neighbor's garage start-up, etc.
- Expand the community—get different perspectives. What about:
 - Academics?
 - People working with proprietary software?
- Organizational framework—Linux Foundation(?)
- Meeting again—Embedded Linux Conference 2014(?)

You tell me!

Insert yourself here

Google group:

- <https://groups.google.com/forum/#!forum/safety-critical-linux>
- ...

