Workshop on Dependable Operating Systems

Ishigaki Island, Japan
January 21-25, 2010

A few loose notes

Paulo Veríssimo
Univ. Lisbon – Portugal *(NOT INESC! :-)*
www.di.fc.ul.pt/~pjv

DEOS – Dependable Embedded Operating Systems proj.
Mario Tokoro (coord.) et al.

- Fundamental questions in road to dependable OS
  - Feasibility: can we really build a dependable system?
  - Validity: one that we can prove is dependable?
- Need a link from the technical to the societal facet of dependability:
  - Accountability
  - Assurance
- Evidence-Based Computing as “dependability attestation”
  - Implies real-time monitoring of computing units or clusters
- A comment - How trustworthy is monitoring, and monitors?
  - “Who guards the guardian”
  - “What can be used can be abused”
- A comment - what about *dependable adaptation* :
  - Dependability as a continuum, evolving with changing of environment
  - A possible way to reconcile uncertainty with predictability

“Connecting your coffee-shop laptop to a life-critical system” *David Powell (LAAS-CNRS, France)*

- Managing to build systems composed of critical and non-critical parts, acted upon by trusted and non-trusted components
  - Hybrid distributed/modular systems models
- Putting your personal laptop to talk to an avionics system is possible... if properly done:
  - using virtualized machines which implement multi-level integrity/ confidence models handling the flows of information
  - relyíng on a reduced footprint set of TCBs as root of trust
- A comment - root of trust can be made intrusion tolerant

"Formally-Verified OS Kernel—A basis for reliable systems?" *Gernot Heiser (NICTA, Australia)*

- Verified micro-kernel (seL4)

"Testing and Evaluating OS's Dependability: The Joys & Pitfalls of Experimental Approaches" Neeraj Suri (TU Darmstadt, Germany)

- reflections on experimental evaluation

"Improving OS safety using the Coccinelle Program Matching and Transformation Tool" *Gilles Muller (LIP6, INRIA, France)*

- Coccinele: bug-eating "bug"

- A comment - for as much as FIT (Fault and Intrusion Tolerance) is the way to go for "automatic" security and dependability, threats are increasingly powerful, so we must continue betting on :
  - Fault/Vulnerability prevention and removal

"The elimination of a monolithic operating system in the GENESYS MPSoC architecture" *Hermann Kopetz (TU Vienna, Austria)*

- Impacts on dependability:
- shared memo (SMA) vs. message passing (DSA)
  - from depend. viewpoint, DSA much better
- from large monolithic OS to modular OS
  - modules reside on partitioned HW, so can reboot OS partially
- A comment:
  - same concept as DOS but cast into SoC
  - + : performance near monolythic ;
  - - : single point of failure

the end.