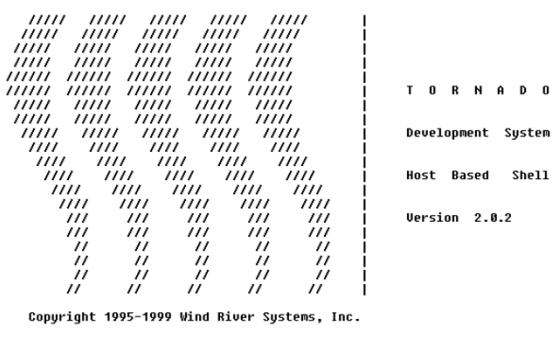
UNIVERSITY OF OSLO

FYS 4220/9220 – 2011 / #6

Real Time and Embedded Data Systems and Computing

Real-Time Operating systems
in particular VxWorks



T.B. Skaali, Department of Physics, University of Oslo)

UNIVERSITY OF OSLO

What makes an Operating System Real-Time ?

- In previous lectures we have defined some key characteristics for Real-Time processing and discussed mechanisms, such as interprocess communication, that are essential for building Real-Time / embedded systems, without “binding” the functionalities to a particular operating system.
- The next pages are from a recent lecture on RTOS at the School of Informatics at the University of Edinburgh, which summarizes very well requirements for a Real-Time OS.
  - The author, prof. Michael O’Boyle, is director of Institute for Computing Systems

Lecture 8
Real-Time Operating Systems

M O'Boyle

February, 2010

School of informatics

OS for Embedded Systems

- Embedded Systems, like traditional ones, need OS functions & services to manage h/w & s/w resources of computer
- Two ways of achieving this functionality:
  1. provide it within the programming language (e.g. some Ada implementations code this into the RTSS)
  2. provide it through a separate *Real-time Operating System (RTOS)*
- In both approaches, OS acts as interface between h/w below & application programs (periodic + sporadic processes) above
- Both have arguments in favour of & against

Requirements of RTOS Predictability and Control

- **Predictability** - most critically w.r.t. *time*
  - all services executed *within* bounded & known times & *at* controlled & known times
  - other resources, such as, files, I/O devices, etc. as well as fault management, should be predictable

- **Visibility & Control** for system components - RTOS user must be able to access & control h/w & system behaviour:
  - necessary to guarantee predictability
  - at the same time, level of abstraction for handling these should be convenient

Requirements of RTOS - Flexibility

- **Openness** - RTOS should be an *open* system
  - should define a flexible set of mechanisms without forcing a particular policy on a user
  - e.g. should allow choice of different policies for task scheduling

RTOS Functions & Services

- Functions to access & control both absolute & relative time -
  - many clock-related operations execute at highest priority & not-interruptible
  - OS timers also need to return values of a fine granularity
- Process & Thread management
  - operations to create, initialise, activate, terminate, communicate & synchronise between tasks
  - possible support for periodic & sporadic processes
  - facilities for task scheduling with a specifiable scheduling policy
- Operations for generating & handling s/w interrupts & context switches - to implement *exception handling* services

RTOS Functions & Services: Time

- Device management utilities must be accessible at the application-RTOS interface:
  - required to control & access sensors, timers & conventional I/O devices
  - e.g. to initiate an I/O operation, read state of a device, defining & connecting interrupt handlers
  - also facilities for attaching new real-time devices

- RTOS cannot be given the responsibility of performing main memory management transparently:
  - for control purposes, users must do it themselves
  - RTOS needs to provide suitable primitives to manage memory

Synchronisations & Communications in RTOS

- Lowest kernel functions must often be executed atomically or as locked critical sections
- For single-processor systems, indivisibility can be achieved by disabling interrupts during that function execution
- RTOS provide range of synchronisation facilities, such as, locks, semaphores, signals, messages, etc. - predictability of timing behaviour a key feature
- Different types of timeout mechanisms also provided

Additional wish-list for RTOS's

- The possibility of building user specific RT kernels to match the wide range of processing machines for RT/Embedded, from microcontrollers and soft core processors to large distributed systems
- Quality assured
- An excellent Development Environment
- Debugging and test tools
- RTOS support of that particular computer architecture you want to use
  - See following pages
- Portability, that is, POSIX compliant
- And of course, "at the end of the day", the price is important!

ARM The Architecture for the Digital World®

Contact ARM
中文 | 日本語

Search our site

Login email... ...and password

ABOUT PRODUCTS MARKETS COMMUNITY SUPPORT REGISTER Why? Help

Homepage > Community > Software Enablement > Real-Time Operating Systems (RTOS)

RSS updates Share this

Quick Links

- Processors
- Multimedia
- Tools
- Find an ARM Partner

Software Enablement

- Adobe
- Google
- Linux
- Microsoft
- Real-Time Operating Systems (RTOS)
- Symbian Foundation
- UEFI

Real-Time Operating Systems (RTOS)

The ARM architecture is supported by all major vendors of Real-Time Operating Systems (RTOS).

Many embedded systems require software to respond to inputs and events within a defined short period of time. Such systems can be categorised as **hard real-time,** where missing a response deadline is unacceptable (for example an anti-lock braking system), and **soft real-time,** where hitting a deadline is desirable but not critical. In both types of system, a degree of **determinism** is important.

(View Larger Real-Time Operating Systems (RTOS) Image)

RTOSes are designed to control an embedded system, and to deliver the real-time resposiveness and determinism required by the controlled device. Applications run under the control of the RTOS, and their allocated CPU time is scheduled by the RTOS kernel.

In modern systems, a RTOS consists not only of a real-time kernel, but also higher-level functions such as device management (USB, UART, Ethernet, LCD etc), file systems, protocol stacks (CAN, TCP/IP, HTTP etc) and graphical user interfaces (GUI).

See the "RTOS vendors" tab below to see a table of ARM Connected Community Partners whose RTOSes support the ARM architecture.

ARM Advantage | RTOS vendors

Featured Products

AMBA Cache Controllers
Level-2 Cache Controllers for AHB and AXI processo...

Mali Software
Optimized 3D graphics software that works with or ...

Logic IP - Standard Cell Libraries, Power Management Kit
Optimal processor performance and power efficiency...

Cortex-A5 Processor
ARM's most power efficient, low cost processor ena...

News & Events

ARM, Dolby and Ittiam Collaborate To...
11 Sep 2010

NOV 09th ARM Technology Conference
The technical prog...

ARM Advantage | RTOS vendors

The ARM architecture is supported by all popular RTOS vendors in the embedded market. Below is a table showing the RTOS companies, their proc
an indication of which ARM processor families they currently support:

Company	RTOS	Cortex-			Classic		
		A	R	M	ARM11	ARM9	ARM7
Altreonic	OpenComRTOS			•			
American Megatrends	MegaRAC					•	
CMX Systems	CMX-RTX	•	•	•		•	•
eCosCentric	eCos			•		•	•
eForce	µC3			•			•
ENEA	OSE			•	•	•	•
eSOL	eT-kernel	•			•	•	•
Express Logic	ThreadX	•	•	•	•	•	•
FreeRTOS.org	FreeRTOS			•		•	•
Green Hills Software	INTEGRITY, VelOSity	•	•	•	•	•	•
Huone Inc	ionESS						
Hyctron Electronic	EmbeddedOS				•	•	
IAR Systems	PowerPAC			•	•	•	•
KADAK	AMX RTOS				•	•	•
Keil	Keil RL			•		•	•
LynuxWorks	LynxOS, Blue Cat	•				•	•
Mentor Graphics	Nucleus OS	•	•	•	•	•	•
Micrium	µC/OS-III			•	•	•	•
Micro Digital Inc	SMX RTOS			•		•	•
OpenSynergy GmbH	COQOS					•	
Pengutronix	OSELAS		•	•	•	•	•
Phoenix Technologies	HyperSpace					•	•
QNX Software Systems	Neutrino	•			•	•	
Quadros Systems	RTXC	•		•	•	•	•
Quantum Leaps	QP-nano			•		•	•
Radisys	Microware OS-9					•	•
RISC OS Ltd	RISC OS					•	•
Rowebots	Unison v4			•			
rt-labs AB	rt-kernel			•		•	•
SCIOPTA	SCIOPTA	•	•	•	•	•	•
Segger	EmbOS	•		•	•	•	•
Semihalf	FreeBSD	•				•	
SYSGO	PikeOS	•				•	
Wind River Systems	VxWorks	•			•	•	•
Wittenstein	OpenRTOS, SafeRTOS			•		•	•

UNIVERSITY OF OSLO

Real-Time OS for the ARM architecture

The NIOS II soft core embedded processor

- **Nios II** is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGAs
- Nios II is comparable to MicroBlaze, a competing softcore CPU for the Xilinx family of FPGA
- Key features:
  - the Nios II architecture is a RISC soft-core architecture which is implemented entirely in the programmable logic and memory blocks of Altera FPGAs. The soft-core nature of the Nios II processor lets the system designer specify and generate a custom Nios II core, tailored for his or her specific application requirements
  - By using custom instructions, the system designers can fine-tune the system hardware to meet performance goals and also the designer can easily handle the instruction as a macro in C/C++
  - Introduced with Quartus 8.0, the optional MMU enables Nios II to run operating systems which require hardware-based paging and protection, such as the Linux kernel. Without an MMU, Nios is restricted to operating systems which use a simplified protection and virtual memory-model: e.g., µClinux and FreeRTOS
  - http://www.altera.com/products/ip/processors/nios2/tools/ni2-development_tools.html

NIOS II embedded Operating system support

Embedded Operating System Support

Altera's embedded software partners provide an array of operating systems for use with the Nios II processor. Table 1 shows the operating system support available for the Nios II processor.

Table 1. Embedded Operating System Support for the Nios II Processor

Operating System	Supplier
eCos	eCosCentric
eCos	Zylin
embOS	Segger
Erika Enterprise	Evidence
Euros RTOS	Euros
Linux	Timesys
Linux	Wind River
Linux	SLS
Linux	CodeSourcery
Linux	Open Source Community
MicroC/OS-II *(1)*	Micrium
osCAN *(2)*	Vector
ThreadX	Express Logic
µCLinux	SLS
µCLinux	Open Source Community

Choice of OS for the Real-Time lab exercises

- Why use VxWorks
  - VxWorks from Wind River is an industry leader in Real-Time systems
  - It offers a very wide choice of software components (system calls)
  - Based on host – target configuration. The development and cross compilation are done on a host computer, typically a PC. The excutable code is downloaded to the target processor and linked with the target resident (mini)kernel
  - Excellent Development Environment based on the Eclipse Platform.
    - The Eclipse Platform is an open and extensible platform.
  - However, VxWorks is a rather expensive solution, however, free as a Universty program
  - Very good tech support in my experience
  - For more info on Wind River products, see http://www.windriver.com/
  - However, nobody is perfect, see next two pages page
- VxWorks Application Programmers Guide
  - Selected sections presented in the following
    - The VxWorks-6.2_Application_Programmers Guide can be downloaded from the home page

DeviceLINE

SECURITY NEWS FOR THE INTERNET OF THINGS

mocana

« The Smart Meter "Off Switch"

New Software Promises iPhone Passwords Without Jailbreak »

Flaws Uncovered in Popular RTOS

Recently presented at the Security B-Sides and DEFCON conferences in Las Vegas, two critical vulnerabilities have been discovered in VxWorks, Wind River's popular embedded OS that is used in tens of thousands of designs for "smart devices" from organizations including Cisco, Apple and even NASA.

As reported in SC Magazine, one of the vulnerabilities allows hackers to leverage the RTOS's embedded debugging services to take (unauthorized) control of the device.

> VxWorks has a service enabled by default that provides read or write access to a device's memory and allows functions to be called.... The vulnerable service, called WDB agent, is a "debugger" for the VxWorks operating system that is used to diagnose problems and ensure code is working properly when a product is being developed. [...]

These two bugs are "just the tip of the iceberg," Moore wrote in a blog post on Monday, August 2nd. 2010

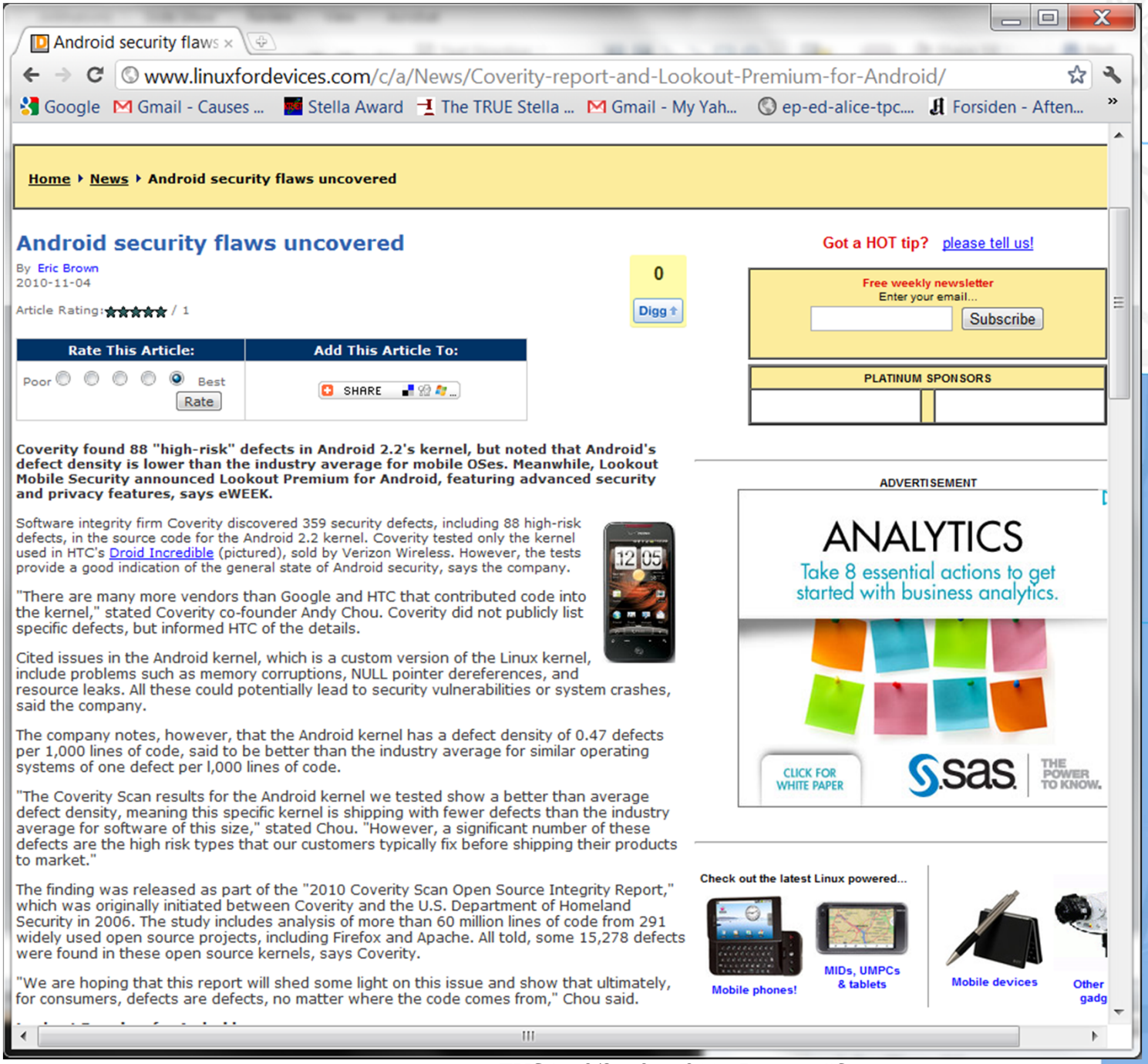
Follow Mocana

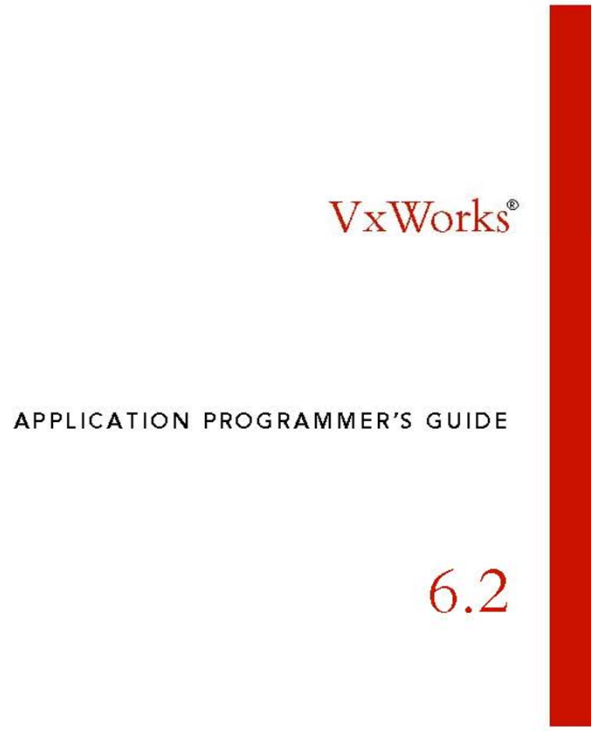
Mocana

- >DeviceLINE Home
- >Mocana Home
- Buzz
- Company
- Downloads
- Industries
- Partners
- Products

Recent DeviceLINE Posts

- Preparing for International Cyber War
- High-Def Video Encryption Cracked
- Growing Security Concerns Block Enterprise Use of Consumer Devices
- Disney to Set Web Toy Standards?
- Plane Crash Possibly Linked to Malware
- Your Smart Grid Security Bill: $3 per Meter


Android security flaws
www.linuxfordevices.com/c/a/News/Coverity-report-and-Lookout-Premium-for-Android/
Google
Gmail - Causes ...
Stella Award
The TRUE Stella ...
Gmail - My Yah...
ep-ed-alice-tpc....
Forsiden - Aften...
Home ▸ News ▸ Android security flaws uncovered
Android security flaws uncovered
By Eric Brown
2010-11-04
Article Rating: / 1
Rate This Article:
Poor
Best
Rate
Add This Article To:
SHARE
0
Digg
Got a HOT tip? please tell us!
Free weekly newsletter
Enter your email...
Subscribe
PLATINUM SPONSORS
Coverity found 88 "high-risk" defects in Android 2.2's kernel, but noted that Android's defect density is lower than the industry average for mobile OSes. Meanwhile, Lookout Mobile Security announced Lookout Premium for Android, featuring advanced security and privacy features, says eWEEK.
Software integrity firm Coverity discovered 359 security defects, including 88 high-risk defects, in the source code for the Android 2.2 kernel. Coverity tested only the kernel used in HTC's Droid Incredible (pictured), sold by Verizon Wireless. However, the tests provide a good indication of the general state of Android security, says the company.
"There are many more vendors than Google and HTC that contributed code into the kernel," stated Coverity co-founder Andy Chou. Coverity did not publicly list specific defects, but informed HTC of the details.
Cited issues in the Android kernel, which is a custom version of the Linux kernel, include problems such as memory corruptions, NULL pointer dereferences, and resource leaks. All these could potentially lead to security vulnerabilities or system crashes, said the company.
The company notes, however, that the Android kernel has a defect density of 0.47 defects per 1,000 lines of code, said to be better than the industry average for similar operating systems of one defect per l,000 lines of code.
"The Coverity Scan results for the Android kernel we tested show a better than average defect density, meaning this specific kernel is shipping with fewer defects than the industry average for software of this size," stated Chou. "However, a significant number of these defects are the high risk types that our customers typically fix before shipping their products to market."
The finding was released as part of the "2010 Coverity Scan Open Source Integrity Report," which was originally initiated between Coverity and the U.S. Department of Homeland Security in 2006. The study includes analysis of more than 60 million lines of code from 291 widely used open source projects, including Firefox and Apache. All told, some 15,278 defects were found in these open source kernels, says Coverity.
"We are hoping that this report will shed some light on this issue and show that ultimately, for consumers, defects are defects, no matter where the code comes from," Chou said.
ADVERTISEMENT
ANALYTICS
Take 8 essential actions to get started with business analytics.
CLICK FOR WHITE PAPER
sas
THE POWER TO KNOW.
Check out the latest Linux powered...
Mobile phones!
MIDs, UMPCs & tablets
Mobile devices
Other gadg



Note! This release of VxWorks runs under Workbench Development platform. An earlier release of VxWorks is also installed in the FYS4220 lab, and runs under the Tornado Development platform