VoIP Evaluation Criteria
for Kernels and Development Tools

Comparing Real Time Kernels in a Voice Application;

using a criteria based on features, capabilities, development tools, platforms, cost, performance, and politics

Presented at TI's FAE "Champs" Conference

1:45PM – 2:30PM May 14, 2008

Jim Donovan & Benoit Parrot

Agenda

- Background, and positioning in VoIP marketplace
- An overview of each, what distinguishes platforms
- Criteria checklist, how to aid customers who are weighing the alternatives
- Summarizing each Platform, Environment, and Tools
- Performance comparison
- Durability (of the SW and respective corporations behind the tools)
- Summary, Review, and Q&A

First pass… (reasons an R&D team picks an OS)

- “because we’ve always used x.y. or z.”
- impression of costs (or, lack there-of)
- sense of “real-time” criticality
- growth in memory space requirements
- momentum from existing code base (fear of a “port effort”)
- concern for stability, and long term viability
- rumors and innuendo…

free advice given by technologists who don’t write code for a living

“Hard Real-Time” is critical

“Linux has a large memory footprint”

“Linux is free”

“Linux isn’t free”

“Windriver doesn’t answer the phone”

“Microsoft’s Embedded products are buggy”

“Linux can’t be used in an embedded product”

remember:

ALL GENERALIZATIONS ARE FALSE…

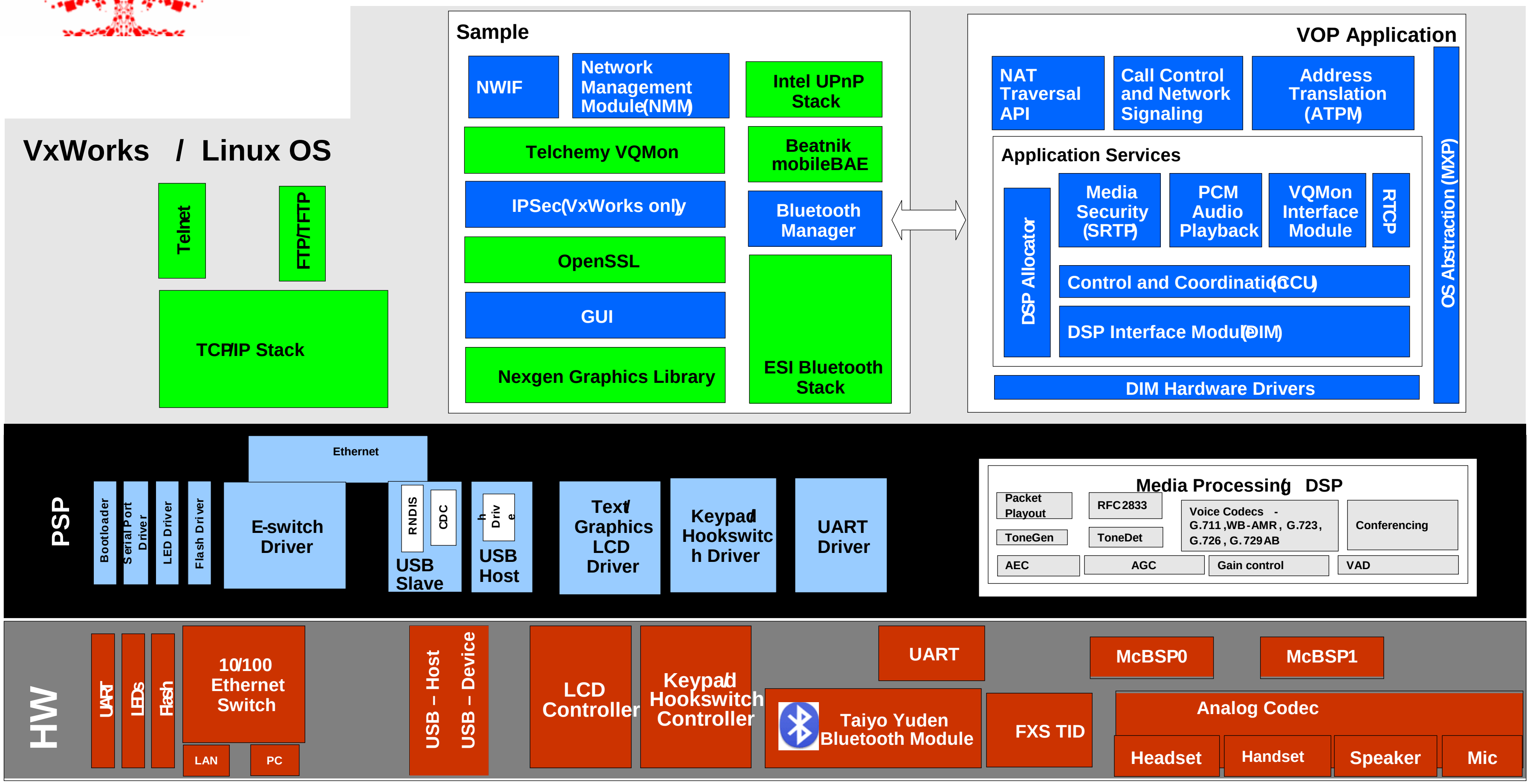
Agenda (recap)

- Background, Positioning in the VoIP Marketplace
  - Comparing: Linux, VxWorks, and Win/CE with in-house kernels for VoIP
  - Microsoft influence in the embedded market
  - Unique requirements of embedded design
    - Peripherals, Memory: flash & RAM options, layout and packaging influence
  - VoIP market place (defining requirements relative to VoIP)
- Criteria checklist, how to aid customers who are weighing the alternatives
- Summarizing each Platform, Environment, and Tools
- Performance comparison
- Durability (of the SW and respective corporations behind the tools)
- Summary, Review, Questions & Answers

VoIP Overview

- Architecture of a VoIP hardware/software
- How VoIP is different
- Influence of kernel selection on a design
- Development steps…
  - Board design
  - Boot monitor, initial drivers
  - Kernel build
  - Application integration, debug, deployment…
- VoIP Market place for Embedded Kernels and Tools

IP Phone Software Architecture



VxWorks / Linux OS

Telnet

FTP/TFTP

TCP/IP Stack

Sample

NWIF

Network Management Module(NMM)

Intel UPnP Stack

Telchemy VQMon

Beatnik mobileBAE

IPSec(VxWorks only)

Bluetooth Manager

OpenSSL

GUI

Nexgen Graphics Library

ESI Bluetooth Stack

VOP Application

NAT Traversal API

Call Control and Network Signaling

Address Translation (ATPM)

OS Abstraction (MXP)

Application Services

DSP Allocator

Media Security (SRTP)

PCM Audio Playback

VQMon Interface Module

RTCP

Control and Coordination (CCU)

DSP Interface Module (DIM)

DIM Hardware Drivers

PSP

Bootloader

Serial Port Driver

LED Driver

Flash Driver

Ethernet

E-switch Driver

RNDIS

CDC

Drive

USB Slave

USB Host

Text/ Graphics LCD Driver

Keypad/ Hookswitch Driver

UART Driver

Media Processing DSP

Packet Playout

RFC2833

Voice Codecs - G.711, WB-AMR, G.723, G.726, G.729AB

Conferencing

ToneGen

ToneDet

AEC

AGC

Gain control

VAD

HW

UART

LEDs

Flash

10/100 Ethernet Switch

LAN

PC

USB – Host

USB – Device

LCD Controller

Keypad/ Hookswitch Controller

UART

Taiyo Yuden Bluetooth Module

FXS TID

McBSP0

McBSP1

Analog Codec

Headset

Handset

Speaker

Mic

🟥	Hardware
🟦	Telogy Micro Module
🟩	3rd Party Stack/Library
⬜	Telogy DSP Moduile
🟦	PSP driver

VoIP Software (unique aspects, use of the kernel)

- VoIP can be considered *Soft* Real Time.
- Packet Based “voice applications” assume they can lose data – this is distinctly different than many other “real time” development efforts!!! (i.e. UDP protocol is used in most every VoIP application)
- Standard SW modules and interfaces/drivers primarily:
- Public domain applications requirement, tends to be limited to GUI, utility applications, and/or drivers (USB, Bluetooth, audio, graphics)
- Embedded Memory requirements (downloadable, and/or “flash”)
- Proprietary blocks of code, are a fact of life for VoIP developers: Acoustic EC, Line Echo Cancellers, G.xyz Voice coders, etc
- Display (if used) have traditionally been LCD character mode
- MP3 Player (for IP Phone ringtone), other music & graphics extras
- other extras: SLIC/SLAC intf. ISDN, T1/E1 framers and control SW

VoIP Software (unique aspects, use of the kernel)

- VoIP is considered *Soft* Real Time.

  "Real Time" criticality here tends to be in the 1 to 10ms resolution range (not uSec).
- Packet Based "voice applications" assume they can lose data – this is distinctly different than many other "real time" development efforts!!! (i.e. UDP protocol is used in most every VoIP application)
- Standard SW modules and interfaces/drivers primarily:
  - Ethernet (IPv6 deserves special mention)
  - other (higher level) network software, like SIP call control, IP Sec for Security, etc
  - Device specific drivers: memory (mainly Flash), USB, BT, LCD, DECT, display
  - Algorithms for Security, Encryption (if not in DSP),
- Public domain applications requirement, tends to be limited to GUI, utility applications, and/or drivers (USB, Bluetooth, audio, graphics)
- Embedded Memory requirements (downloadable, and/or "flash")
  - Flash: ~4MB threshold (NOR Flash), now seeing NAND is common
  - SDRAM: 8-16MB/32MB RAM or 64MB, with DDR memory is commonly 128MB, 256MB or more…
- Proprietary blocks of code, are a fact of life for VoIP developers: Acoustic EC, Line Echo Cancellers, G.xyz Voice coders, etc
- Display (if used) have traditionally been LCD character mode
- MP3 Player (for IP Phone ringtone), other music & graphics extras
- extras: SLIC/SLAC intf. ISDN, T1/E1 framers and control SW

Development Process – VoIP example

- Vendor Selections (device, SW, tools, 3rd Party elements)
- Board Design, module specification
  - Layout often based on “reference design” or prototype
  - Built on previous generation design’s compatibility / new features

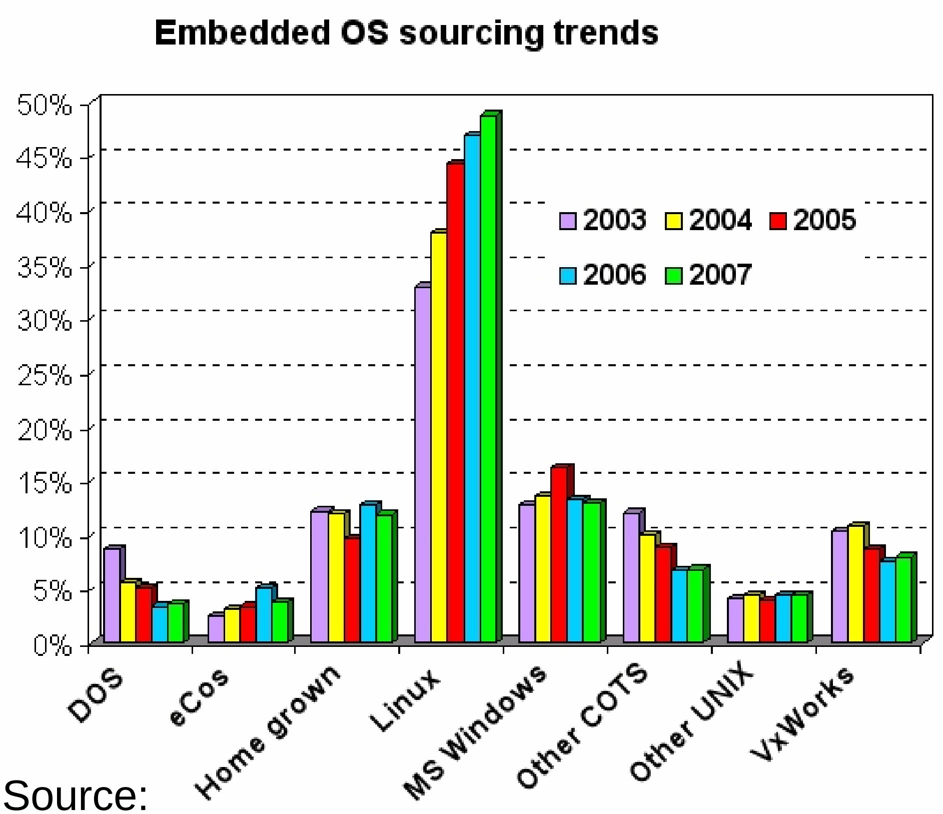
kernel decision
made before this phase

- BSP verification
  - early development efforts could be based upon vendor boards (?)
  - new drivers written for “unsupported” modules versus PSP(from TI)

major effort of
kernel integration

- Application Integration
  - The rest of the development team (beside the BSP effort) commences in earnest… earlier efforts would be primarily concerned with focus areas of high risk, new technologies, or legacy code base compatibility concerns.
- Debug/Test
  - Criteria via established corporate processes, defect count, coverage analysis
  - Release schedule, versus marketing requirement schedules and deliverables
- Release – cycle to next gen requirements
- Support

Market place

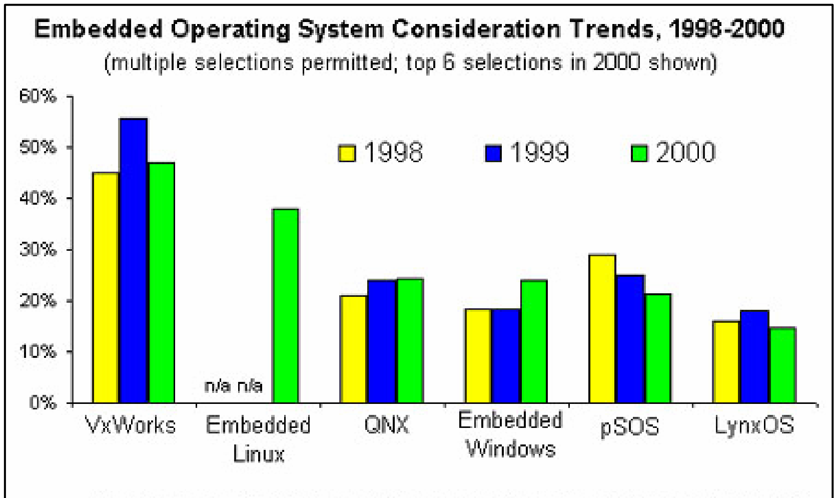


Source:
.linuxdevices.com article

Which embedded OSes are you considering?

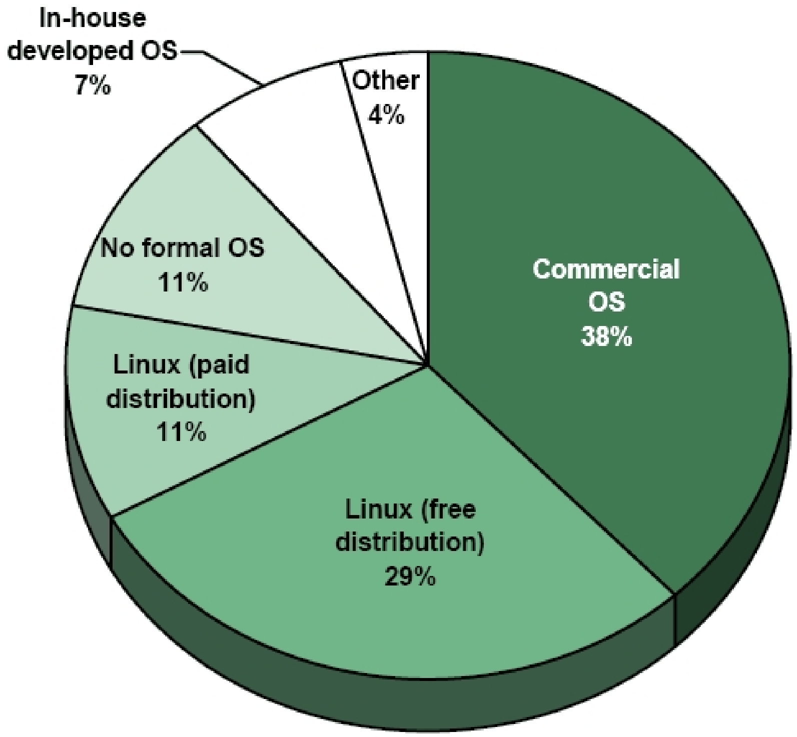
The chart below summarizes the responses to the question "Which 16 or 32-bit vendors would you consider when purchasing RTOSes or kernels for your embedded projects?"

The top six choices are shown here, sorted by their 2000 ranking

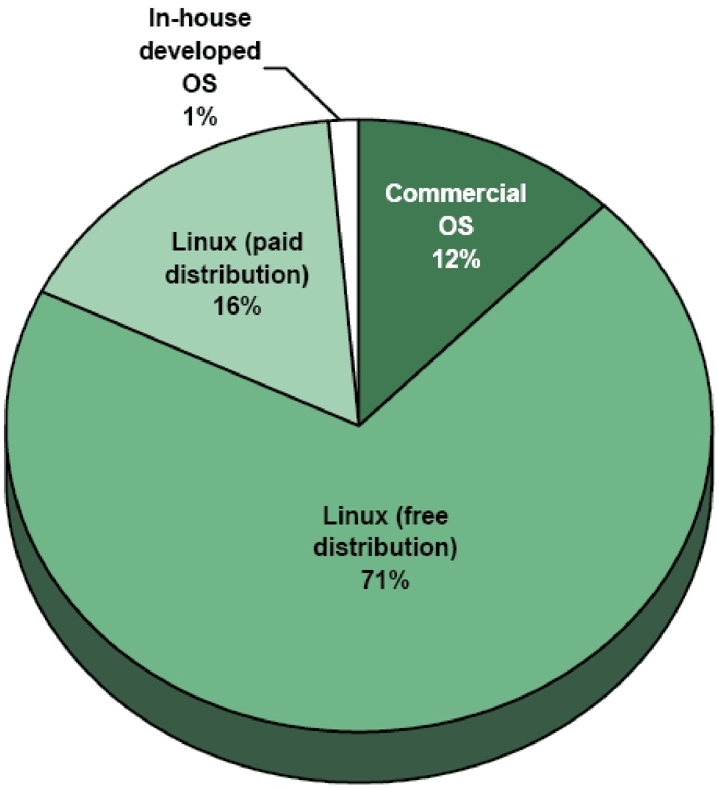


(Source: Embedded Systems Programming Magazine 2000 Subscriber Study)

Embedded Systems Manufacturers Indicate Future Commitment to the Linux Operating System



Operating system used on previous embedded project by teams currently using embedded Linux (percent of respondents)
(Source: VDC Oct 2007)



Operating system planned for next embedded project by teams currently using embedded Linux (percent of respondents)
(Source: VDC Oct 2007)

Agenda

(recap)

- Background, positioning in the VoIP marketplace
- An overview of each, distinguishing features/capabilities:
  - WindRiver, VxWorks
  - Open Source ***<u>or</u>*** MontaVista, Linux
  - Microsoft, Win/CE
  - others: Greenhills INTEGRITY, Thread-X, QNX, Symbian,
  - in-house ("home grown")
- Summarizing each platform and environment
  - main criteria in evaluating kernels/platforms
  - tools and deployment
- Performance, Durability
- Summary, Review, Questions & Answers

Definition of our score card criteria

- Cost
  - Tooling cost, upfront fees, and support fees
  - Run-time (per unit, or one-time-buyout)
- Industrial Grade
  - Capability of being used in “mission critical”, meaning more robust… less defects, no support hassles, more responsive defect fixing, solid code.
- Community of Developers
  - Ability to hire developers experienced with the platform
  - Number of developers who currently develop code for the platform
- Critical Real-time
  - Ability to assure that an interrupt is not ignored
  - Handling of synchronization between processes
  - Co-operative handling of real-world and multi-tasking environments

Definition of our score card criteria

- Memory Efficiency
  - Linux may have a larger footprint, and yet include additional features
  - The management of optimization of unnecessary features should be considered (almost an independent criteria, on its own).
  - As commented earlier, in some applications memory becomes less of a factor than even just a few years ago. In VoIP designs, it is still a concern
- Development Tools
  - Quality and Integration of the development platform: the ability to cross develop code for a target system and also tie into full hosted automated builds.
  - Robustness of the tools, and the cost of the tools
- Support
  - Cost of support, and availability of experts who can support questions when they arise
  - How quickly available is general knowledge, and then specific expertise (i.e. Google on error messages quickly relates someone else's similar experience)

An overview (creating a score card – first pass)

Gathering opinions for discuss, debate, and gain consensus of developers

	Cost tools / run-time	Industrial Grade	cmmnty dev	Critical Real-time	Memory efficiency	Dev Tools	Support
VxWorks	concern			attractive			concern
Linux	attractive	concern	attractive		neutral	good and bad	concern
Win/CE	?	good and bad	attractive		concern		concern

An overview of each (your own score card)

Opinions gathered and summarized to discuss and debate

	Cost tools / run-time	Industrial Grade	cmmnty dev	Critical Real-time	Memory efficiency	Dev Tools	Support
VxWorks	B^- / B	A	B	A^-	B^-	A^-	C
Linux	B^+ / A^-	A^-	A^+	B^+	C	A^-	B
Win/CE	A^- / B	B^-	A^-	B^+	C^-	A^-	B^-
Nucleus	? / ?	A	C	A	B^+	B^+	C
Thread-X	? / ?	C	C	A	B	B	C
QNX	? / ?	A	C	A	B^+	?	?
in-house	C / A	A^-	C^-	B^+	A^-	B	C

An overview of each (your own score card)

Opinions gathered and amalgamated to discuss and debate

	Cost tools / run-time	Industrial Grade	cmmnty dev	Critical Real-time	Memory efficiency	Dev Tools	Support
VxWorks	B^{-} / B	A	B	A^{-}	B^{-}	A^{-}	C
Linux	B^{+} / A^{-}	A^{-}	A^{+}	B^{+}	C	A^{-}	B
Win/CE	A^{-} / B	B^{-}	A^{-}	B^{+}	C^{-}	A^{-}	B^{-}
customer requirement		#1	✓				✓

You don't know which criteria is most important to your project team unless you ask – one factor may outweigh all the others, just because of a given market place.

VxWorks

VxWorks Platform for Consumer Device 3.2

Is a commercial RTOS produced and marketed by WindRiver. It is highly scalable, deterministic, hard real-time kernel. System is optimized to include only the technology required by the application to provide the smallest footprint possible. VxWorks support POSIX and industry standard protocols such as IPv6 and TIPC ensuring code portability and interoperability. VxWorks is backward compatible with previous releases. Supported platforms include variants of ARM and XScale, Freescale, MIPS, PowerPC, and x86 computer architectures. PCD 3.2 supports the listed processors with MMU. VxWorks is royalty based.

Associated tools

- WindRiver Workbench IDE (based on Eclipse) for building application and kernel projects (graphical configuration, version control, compiler/toolchain errors) and debugging the kernel and applications that run on the VxWorks operating system.
- System Event Viewer, Performance Profiler (aka ProfileScope)
- Memory Analyzer (aka MemScope)
- Data Monitor (aka Stethoscope)
- Code Coverage Analyzer
- Function Tracer

Linux

MontaVista Pro 4.0

Is a commercial embedded Linux OS to offer the latest real-time enhancement available in Open Source. Its goal is to provide technology to enable developers the ability to quickly deliver reliable products economically. MV Linux 2.6.10 support Native POSIX Thread Library (NPTL) and industry standard protocols such as IPv6, IPsec and VLAN, USB 2.0 Host/Device, as well as Preemptible Kernel, High Resolution Timer and O(1) scheduler. Supported platforms include variants of ARM and XScale, Freescale, MIPS, PowerPC, and x86 computer architectures. MV Linux is royalty free.

Associated tools

- Montavista's DevRocket is an IDE (based on Eclipse) for building application and kernel projects (graphical configuration, version control, compiler/toolchain errors) and debugging the kernel and applications that run on the Linux operating system.
- MV Pro 4.0 is based on Linux 2.6.10. (and supports MMU)
- GCC 3.4.3
- Large number of Open Source tools and resources are available to the developers.
- uClibc 0.9.27 is also used to replace Glibc in order to minimize memory footprint.

Win/CE – from Microsoft

- Win/CE is NOT Windows Mobile^{TM}
- Current version of Win/CE is v6.0
  note: The “Windows Mobile” version tends to be one version behind the current Win/CE version; for example Windows Mobile 6, is built upon Win/CE v5.0
  - note: Win/CE V7.0 is scheduled for release in early 2009
  - Arm v7 (the core in a ArmCortex-A8) is not supported [yet] in the Win/CE V6.0
  - NEON co-processor, and SGX instructions for Direct/X or OPENGL ES2.0 are not supported
- Using “Platform Builder” builds a kernel from the BSP for a specific architecture and set of peripherals
- The development tools, “plug in” to the MS Visual Studio,
- Platform builder, allows a lower investment for an application developer to get started; and, if a BSP already exists then deploying an executable to a set of hardware is relatively painless; theoretically rapid “startup” with fewer hassles; includes all the “key ingredients”, Serial Port, USB, Ethernet, basic GUI (using web browser implies a higher price), Data Base, Libraries, .

INTEGRITY (GreenHills Software)

INTEGRITY is a real-time operating system (RTOS) produced and marketed by Green Hills Software. It is royalty-free, POSIX-certified, and intended for use in embedded systems needing reliability, availability, and fault tolerance. It is built atop the microkernel and is intended mainly for modern 32- or 64-bit embedded system designs that support MMU. INTEGRITY uses hardware memory protection to isolate and protect itself and user tasks from incorrect operation caused by accidental errors or malicious tampering. Supported platforms include variants of ARM and XScale, Blackfin, Freescale ColdFire, MIPS, PowerPC, and x86 computer architectures.

Associated tools

- MULTI IDE for building INTEGRITY projects (graphical configuration, version control, compiler/toolchain errors) and debugging the kernel and applications that run on the INTEGRITY operating system.
- EventAnalyzer for visual analysis of system level events associated with INTEGRITY tasks and address spaces.
- TimeMachine tool suite for INTEGRITY application execution path analysis, forward/backward execution and single stepping, and other debugging capabilities for INTEGRITY configurations that can acquire instruction and data trace information.
- Green Hills probe and SuperTrace probe for downloading INTEGRITY kernels to target hardware, and examining trace information.
- UML and modeling (Telelogic Rhapsody and IBM Rose RealTime UML).
- Flash programmer

Excerpted from Wikipedia

Performance

- Memory Size / Processor Utilization / “other”
  - we intentionally leave “other” as the criteria which may be more important to specific projects (like, a critical rendezvous measurement for synchronizing a peripheral which may be unique to a customer’s application; or, maybe the nature of a driver which only exists in a given platform and can’t easily be ported)
  - Memory Size may be less a factor than even a couple of years ago (!)
  - Speed is very subjective; objectify it: characterize specific task/functions.

- Measurement of memory requirement, example in VoIP (IP Phone)

- Performance of processor – appropriate to the application with the speed and class of machine given the time allotted for a task needing to be performed

Performance: memory requirement

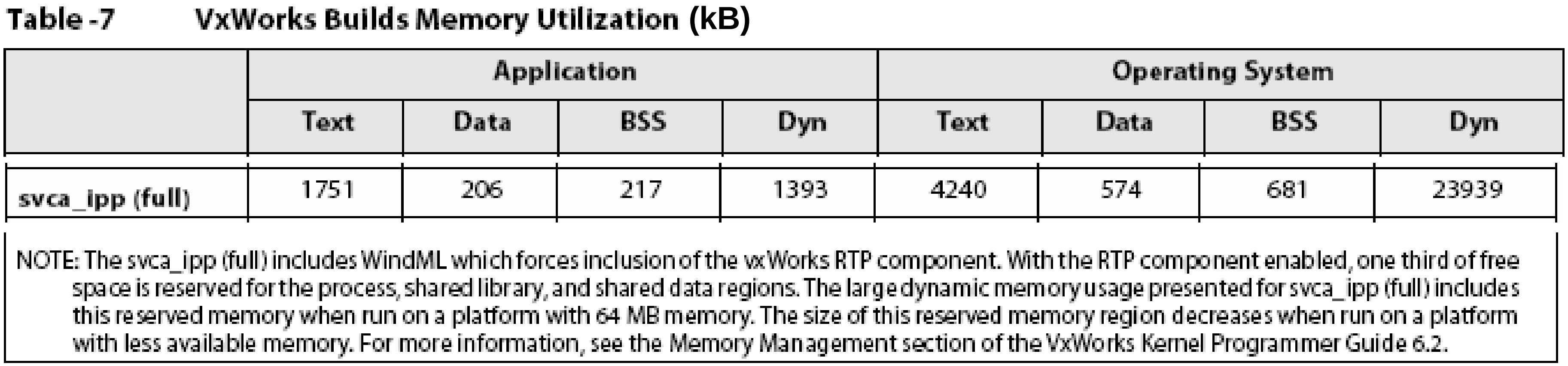


Table -7 VxWorks Builds Memory Utilization (kB)

	Application				Operating System			
	Text	Data	BSS	Dyn	Text	Data	BSS	Dyn
svca_ipp (full)	1751	206	217	1393	4240	574	681	23939

NOTE: The svca_ipp (full) includes WindML which forces inclusion of the vxWorks RTP component. With the RTP component enabled, one third of free space is reserved for the process, shared library, and shared data regions. The large dynamic memory usage presented for svca_ipp (full) includes this reserved memory when run on a platform with 64 MB memory. The size of this reserved memory region decreases when run on a platform with less available memory. For more information, see the Memory Management section of the VxWorks Kernel Programmer Guide 6.2.

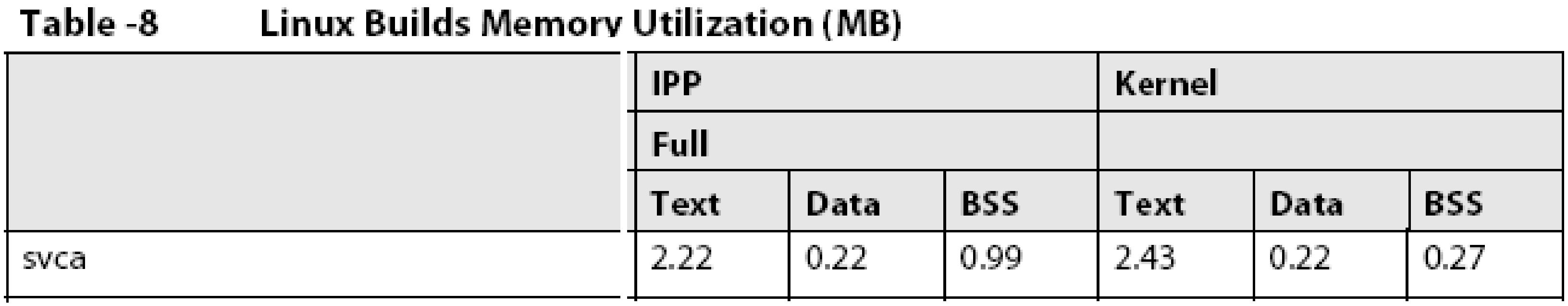


Table -8 Linux Builds Memory Utilization (MB)

	IPP			Kernel		
	Full					
	Text	Data	BSS	Text	Data	BSS
svca	2.22	0.22	0.99	2.43	0.22	0.27

Doc excerpt, “TNETV1051 Performance Guide” for IP Phone, TI Telogy R11.2 May 2007

Performance: memory required (side note)

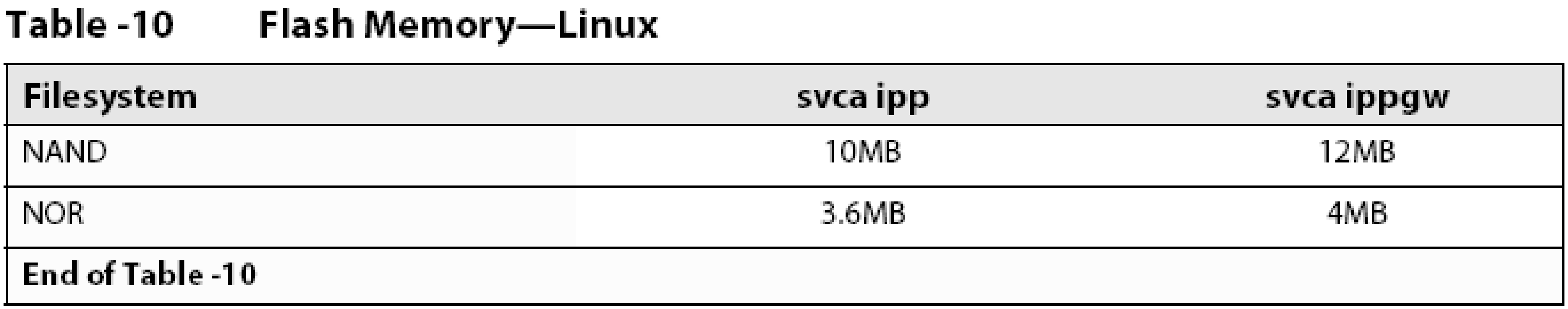


Table -10 Flash Memory—Linux

Filesystem	svca ipp	svca ippgw
NAND	10MB	12MB
NOR	3.6MB	4MB
End of Table -10		

Doc excerpt, “TNETV1051 Performance Guide” for IP Phone, TI Telogy R11.2 May 2007

Performance: processor utilization

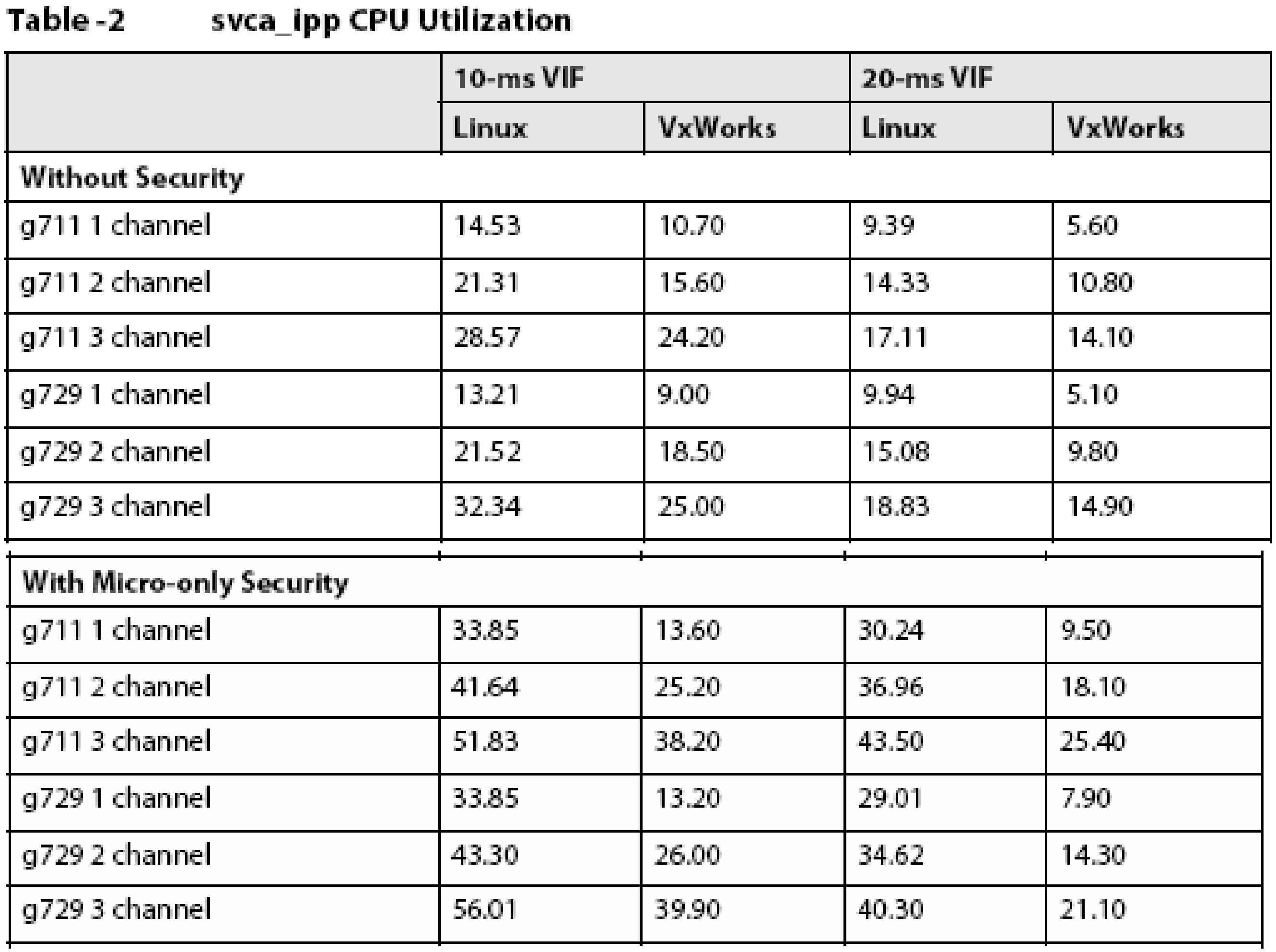


Table -2 svca_ipp CPU Utilization

	10-ms VIF		20-ms VIF	
	Linux	VxWorks	Linux	VxWorks
Without Security				
g711 1 channel	14.53	10.70	9.39	5.60
g711 2 channel	21.31	15.60	14.33	10.80
g711 3 channel	28.57	24.20	17.11	14.10
g729 1 channel	13.21	9.00	9.94	5.10
g729 2 channel	21.52	18.50	15.08	9.80
g729 3 channel	32.34	25.00	18.83	14.90
With Micro-only Security				
g711 1 channel	33.85	13.60	30.24	9.50
g711 2 channel	41.64	25.20	36.96	18.10
g711 3 channel	51.83	38.20	43.50	25.40
g729 1 channel	33.85	13.20	29.01	7.90
g729 2 channel	43.30	26.00	34.62	14.30
g729 3 channel	56.01	39.90	40.30	21.10

Doc excerpt, “TNETV1051 Performance Guide” for IP Phone, TI Telogy R11.2 May 2007

CPU Usage/Idle Measurement

- Traditionally, in the Linux world the “top” utility is used for measuring System Idle time. However the kernel statistics are too coarse to accurately measure idle time in real-time embedded system.
- Also in order to compare a running Linux system to a similar VxWorks one a identical to way calculate overall cpu usage is needed.
- VxWorks already has an Idle Task which can be used to measure how much the CPU is “idle”.
- The idle task is a low priority task which increments a loop counter. Calibration is needed in order to have a baseline for comparison.
- Whenever the system has nothing better to do than the “idle task” is scheduled.

- A similar mechanism has been implemented in Linux for the IP Phone software, so we can actually compare the CPU utilization between VxWorks and Linux.
- This mechanism was used to collect the performance data on the previous slide.

Durability

This characteristic of a platform is the most difficult to define and assess.

- VxWorks – many industrial grade deployments.
- Linux – still yet to be “time tested” in mission critical application
- Win/CE – built on a commonly understood analogous desktop platform.
- other: “purpose built” dedicated platforms

Durability

This characteristic of a platform is one of the most difficult to define and assess. A vendor may be well understood, and yet the software they create takes on behaviors almost from its own merit. Linux has influences beyond any single vendor’s significance. An influence in the market will certainly be felt from Microsoft, but not with any manifest destiny that they might espouse.

- VxWorks – many industrial grade deployments continue to depend upon the certainty of the real-time response time and well proven reliability.
- Linux – still to be time tested in robust real-time mission critical application, but has become popular in consumer goods, and even PDA, phones, and other portable applications, as well as network based apps with GUI. How to assure real-time responsiveness in the platform which is as well know for desktop user platforms remains to be a challenge
- Win/CE – often those environments where the user interface is expected to be leveraging the commonly understood analogous desktop platform.
- other: “purpose built” and dedicated corporate sponsored platforms which depend upon the market and product line for which they were invented.

ScoreCard … an exercise for the reader

- On the analytical side… factor the top criteria that goes into a kernel selection, and give each term a weighting will allow your customer to draw their own conclusion.

- In retrospect: a development team can use the data to draw a consensus or defend a decision… or, just revert back to slide #2 and make a decision based on external factors. (like: to accept an executive directive ☺)

FINAL ANALYSIS: Each tool suite and associated kernel will have benefits and detractors. As long as its applied appropriately each ***<u>can</u>*** accomplish the expected task.

A final perspective

(some more practical observations)

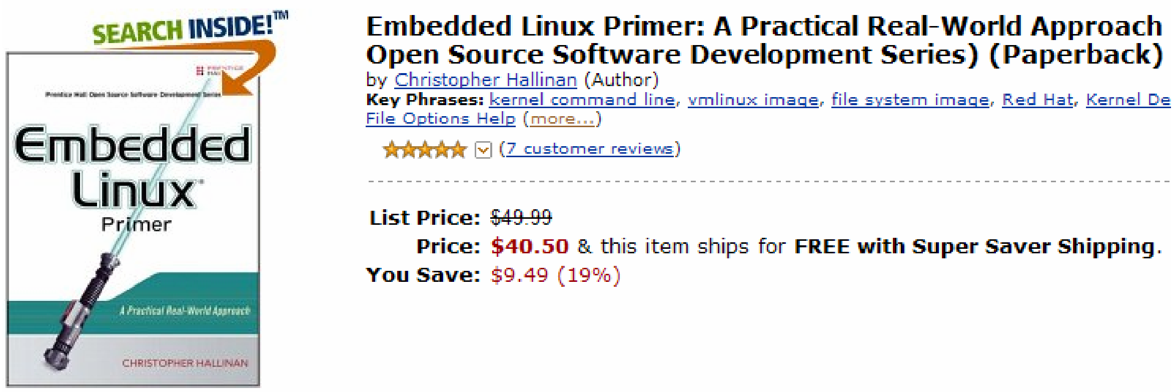
- VxWorks-isms for TI Telogy VoIP projects…
  - Vxworks, tends to have single full image binding of executable.   The tasks spawn as a function, which is managed with their own “cli”
  - As a result, all functions are commands executable from the “->” prompt.
  - TI Telogy VoIP Boot Monitor incorporates minimal (NOR flash) file system which integrated easily into VxWorks file system to “ls”, “rm”, “cp” flash
- Linux-isms for TI Telogy VoIP projects…
  - A “shell” from any target system has the common shell UI which most developers become familiar with quickly
  - TI Telogy’s VoIP Boot Monitor must re-flash the full Linux file system; from the monitor there’s less insight to Linux file system; full memory blocks are fully managed by the Linux environment after it is booted.
  - Using a NAND based flash file system gives target system flexibility, compared with NOR based READ-ONLY “squashfs” file systems which optimized for size. If available a portion of NOR can also be used/formatted as a JFFS2 RW file system.
- Win/CE
  - Fundamentally Microsoft is a sole tool vendor with platform builder on top of Visual Studio as user interface for kernel and application level development environments.
  - A “port” from VxWorks, or Linux deliverables would be needed for a TI VoIP design

References

- http://en.wikipedia.org/wiki/Embedded_system
- Linux Sourcing Trends, 2007
  http://www.linuxdevices.com/articles/AT7065740528.html
- Previous year (2006) trends
  http://www.linuxdevices.com/articles/AT7070519787.html
- Overview of Embedded Linux
  http://www.lynuxworks.com/corporate/news/2004/linux-kernel-2.6.php
- http://www.evansdata.com/reports/viewRelease.php?reportID=17
- http://linuxdevices.com/files/article077/embedded_OS_sourcing_trends.jpg
- http://en.wikipedia.org/wiki/VxWorks

Texts

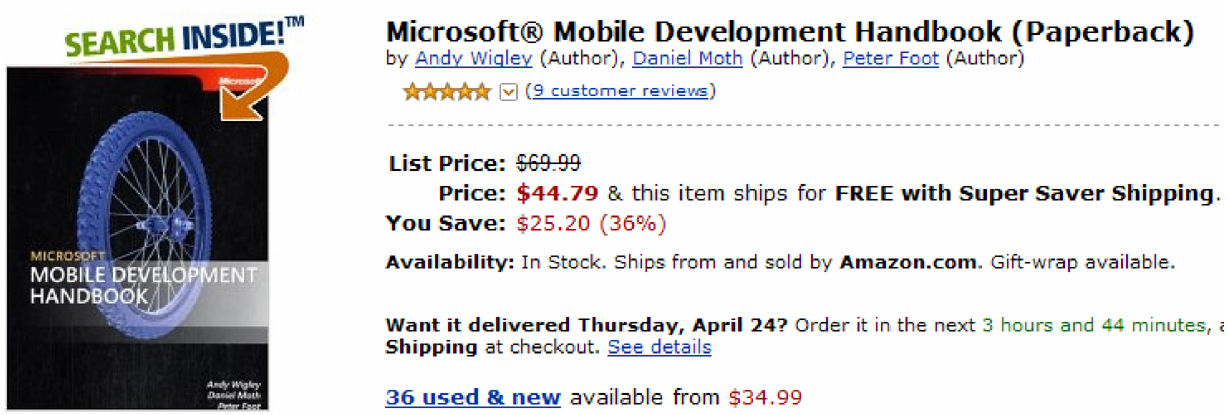
- Embedded Linux Primer: A Practical Real-World Approach



Embedded Linux Primer: A Practical Real-World Approach Open Source Software Development Series) (Paperback)
by Christopher Hallinan (Author)
Key Phrases: kernel command line, vmlinux image, file system image, Red Hat, Kernel De File Options Help (more...)
★★★★★ (7 customer reviews)

List Price: ~~$49.99~~
Price: **$40.50** & this item ships for **FREE with Super Saver Shipping**.
You Save: $9.49 (19%)

- Microsoft® Mobile Development Handbook



Microsoft® Mobile Development Handbook (Paperback)
by Andy Wigley (Author), Daniel Moth (Author), Peter Foot (Author)
★★★★★ (9 customer reviews)

List Price: ~~$69.99~~
Price: **$44.79** & this item ships for **FREE with Super Saver Shipping**.
You Save: $25.20 (36%)

Availability: In Stock. Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Thursday, April 24? Order it in the next 3 hours and 44 minutes, a
Shipping at checkout. See details

36 used & new available from $34.99