Introduction: Embedded Operating Systems and Linux
Outline

• What is an embedded Operating System
• Linux as an embedded OS
• Is Linux Real-Time?
  • Can it be and How?
• Embedded Linux Distributions
• What is the Cost of Embedded Linux
• Tools and development
• Embedded Linux Resources
What is an Operating Systems?

• “A Very Special Program” that provide a software platform on top of which other programs, called application programs, can run.

• The application programs must be written to run on top of a particular operating system.

• Your choice of operating system, therefore, determines to a great extent the applications you can run.
Various OS Abstractions

- applications
- databases
- command interpreters
- compilers
- runtime libraries
- networking
- file systems
- virtual memory
- inter process comm.
- processes
- device drivers
- trap/arch interface

large O/S
[OS/360, VAX/VMS]

kernel-based O/S
[UNIX]

micro-kernel O/S
[Mach, Chorus]

submicro-kernel O/S
[SPACE]
OS Varieties

• Desktop
  • Windows (9X, XP Home, XP/2000 Prof)
  • Mac

• Server
  • Windows (XP/2000 Server & Advanced Server)
  • Unix Varieties

• Embedded
  • VxWorks, Windows CE, Embedded Linux, ….
What is an Embedded OS?

• An embedded OS is an operating system which runs on any embedded platform.

• Embedded platforms are generally required to function without human intervention.

• A typical embedded system consists of a single-board microcomputer or SOC with an OS and some software loaded in ROM.
  • It will not usually have any of the normal peripherals such as a keyboard, monitor, serial connections, mass storage, etc. or any kind of user interface software unless these are required by the overall system of which it is a part.

• Often an embedded OS must provide real-time response to perform its requirements.
What makes a good Embedded OS?

- Modular
- Scalable
- Configurable
- Small footprint
- CPU support
- Device drivers
- etc, ...
What is Real Time?

• The ability of the operating system to provide a required level of service in a bounded response time. - POSIX Standard 1003.1.

• Not the CPU speed

• But the Time Constraints.
Hard vs. Soft Real Time

• Hard
  • *guaranteed worst-case* response times
  • absolutely, positively, first time every time
• Soft
  • Kinda, sorta, usually
What makes a good RTOS?

• Multi-threaded and preemptible
• Thread/Task priority-based scheduling
  • Static or Dynamic
• Must support predictable thread synchronization mechanisms
• Should have a system of priority inheritance
  • To solve priority inversion problems
Interrupt latency

• Traditional UNIX Operating systems suffer from large interrupt latency

• How to reduce the interrupt latency?
  – Make kernel highly preemptible by changing its internal structure (minimizing interrupt disabling) or adding a set of preemption points
  – Microkernel approach
Definition of Interrupt Latency

- Interrupt latency ($t_L$), response ($t_R$), and recovery ($t_{RC}$) times, $T_1$: time for saving CPU contexts, $T_2$: time for restoring CPU contexts
• Is Linux Real-Time?
  • In a nut shell: **NO, BUT...**

• Why?
  • (Monolithic Kernel) The Linux kernel uses coarse grained synchronization, which allows a kernel task exclusive access to some data for long periods. This could delay the execution of any POSIX real-time task that needs access to that same data.
  • (NotPreemptible in Kernel Mode) The Linux kernel does not preempt the execution of any task during system calls. If a low priority process is in the middle of a system call and a message is received for a real-time process, the message will unfortunately be held in the queue until the system call completes, despite its low priority.
Linux and Real-Time

• (Resource Lock) Linux makes high priority tasks wait for low priority tasks to release resources. For example, if any process allocates the last network buffer and a higher priority process needs a network buffer to send a message, the higher priority process must wait until some other process releases a network buffer before it can send its message.

• (Priority Scheduling) The Linux scheduling algorithm will sometimes give the most unimportant and nicest process a time slice, even in circumstances when a higher priority process is ready to execute.
Who are the Embedded OS Vendors?

- Wind River Systems
  - VxWorks
  - pSOS
- QNX Software Systems
  - QNX
- Microsoft
  - Windows CE, …
Who are the Embedded OS Vendors? -2

• Mentor Graphics
  • VRTX
• Palm Computing
  • PalmOS
• Symbian
  • SymbianOS
Microsoft

- Embedded NT/XP
  - “Real-time” control
- Windows CE (CE.NET)
  - Internet devices
- Pocket PC 2002
  - Handheld PC’s and PDA’s
Commercial Embedded Linux

- Lineo Embedix
  - supports real time and high availability apps
- LynuxWorks BlueCat
  - general purpose embedded solution
- MontaVista Linux
  - general purpose embedded solution
BlueCat Embedded Linux
Open Source Embedded Linux

• Embedded Debian Project
  • convert Debian to an embedded OS
• uCLinux
  • for microprocessors that don’t have MM
Commercial Linux RTOS

- FSMLabs - Open RT Linux
- Lineo - Embedix Realtime
- LynuxWorks - BlueCat RT
- MontaVista Software - Real Time Extensions
- TimeSys - Linux/Real-Time
Open Source RTOS

- Linux-based RTOS
  - RTLinux
  - RTAI - “hard” Real Time Application Interface
  - KURT - event schedules with 10us resolution
- Other embedded RTOS
  - eCos
RTLinux

- A “hard real-time” mini operating system
- runs Linux as it’s lowest priority execution thread
- Linux thread completely preemptible
- Real time threads and interrupt handlers never delayed by non-realtime operations
- Supports user level programming
- MiniRTL implementation fits on a floppy
KU Real-Time Linux (KURT)

- Kernel modifications in
  - Increasing the resolution of the Linux software clock
  - Scheduling algorithm for firm real-time applications
- KURT scheduler: an explicit plan scheduler, requiring real-time applications to state explicitly the times at which events are to occur.
eCOS: free real-time OS from Red Hat

- eCOS (an embedded configurable operating system)
- Host platforms:
  - Red Hat Linux 5.x, 6.x
  - Window NT 4.0
- Target Architectures: ARM7, 9, PowerPC, MIPS, i386,…
eCos Architecture
Advantages of Linux

• Multiple choices vs. sole source
• Source code freely available
• Robust and reliable
• Modular, configurable, scalable
• Superb support for networking and Internet
• No runtime licenses
• Large pool of skilled developers
Issues in Linux

• Lack of hardware device drivers
• Competing and/or lacking standards
• No formalized qualification testing
• No single source for marketing
• GPL license issues
• Startup vendors with shaky futures
What CPU’s will it run on?

• Intel X86
• MIPS
• ARM
• StrongARM
• PowerPC
• Hitachi SuperH
Any Development Tools Available?

- QT/Embedded
- Other GUI/Windowing toolkits
- Arcom Control Systems
- GNUPro Tools
- Vendor specific
- Standard Linux toolset
C or C++ for Development?

• In general C is a better choice
  • Advanced OOP features can cause code bloat
  • C++ compilers can generate many routines for a single function
  • Virtual methods and polymorphism slow program launch times significantly

• Size really does matter
What is the Cost of Embedded Linux

• Build Embedded Linux OS yourself.
  Advantage:
  • Cost
  • Educational Process
  • Ultimate Control
  Disadvantage:
  • Not very easy
  • Can be a problem to maintain latest changes
  Recommendation:
  • Getting one of the commercially available embedded Linux OS products.
  • Open source projects hosted by organizations
What’s It Being Used For?

• Control and Monitoring Applications
• Industrial Controllers
• TV Set Top Boxes (TiVO)
• Handheld PDA’s
• Automobile Computers
• Telecomm and Networking Hardware
• Myriad and sundry other uses...
An example of networking environment
Sharp Zaurus

- Lineo Embedix
- 206 MHz StrongARM
- 64 MB DRAM 16MB Flash
- 3.5” display (320x240) 64K colors
- Opera browser & Qtopia
- QT/Embedded GUI
Cell and Web Phones

Telepong Mobile Phone

GITWiT Mobile Phone

Aplio/PRO IP Phone
TiVO Set Top Box

- Home grown port of Embedded Linux
- 54MHz PowerPC
- Multi GB hard disk
Axis 2120 Network Camera

- uCLinux
- Built-in Ethernet port
- 100 MHz ETRAX CPU
- 16 MB RAM
Humanoid Robots

- Univ. of Tokyo/Kawanda Ind.
- Dual Pentium CPU
- RT-Linux
- Height: 53 inches
- Weight: 121 lbs.

- Isamu
Humanoid Robots

- HOAP
- Fujitsu
- RT-Linux
- Height: 48 cm
- Weight: 6 kg
- 100 units/yr
References

• LinuxDevices.com
  http://www.linuxdevices.com/

• Embedded Linux Consortium
  http://www.embedded-linux.org/

• All Linux Devices
  http://alllinuxdevices.com/

• Embedded Linux StarGate
  http://linux-embedded.com/
References

• Dr. Dobbs Embedded Systems
  http://www.ddjembedded.com/about/

• Embedded Linux Journal
  http://embedded.linuxjournal.com/

• Embedded Systems Programming
  http://www.embedded.com/