SWP

Graham Leach

Author

Computer Professional

Linux Evangelist

SWP

What We Do Today

1. Define What "Real-Time Operating System" (RTOS) Means
2. Explore The Different Forms of Real-Time Operating Systems
3. Examine Our Options With Respect To Embedded Linux OS
4. Survey The Linux Embedded Linux OS Market
5. Identify The "Top Five" Embedded Linux Systems Suppliers
6. Explain Why Embedded Linux is Important For Hong Kong
7. Meet The Primary Developer of an Embedded Linux Project

SWP

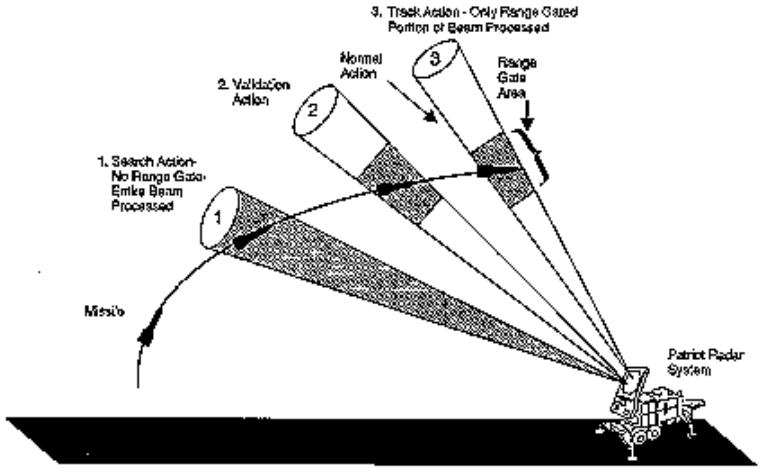
"Real Time" Defined, Described & Illustrated

- A Real Time Operating System provides:
  - **A Guaranteed reaction within a specific timeframe***
- Real Time Systems Are Used For Critical Things:
  - United Nations Video Conferencing Systems
  - Medical Applications (pacemakers)
  - Mass Transit Systems (TKO Line features RT systems)
  - Aircraft Control & Air Traffic Control Systems (Chek Lap Kok)
- Example: The **Patriot Missile Guidance System**

* usually a identified range of microseconds (μ s)

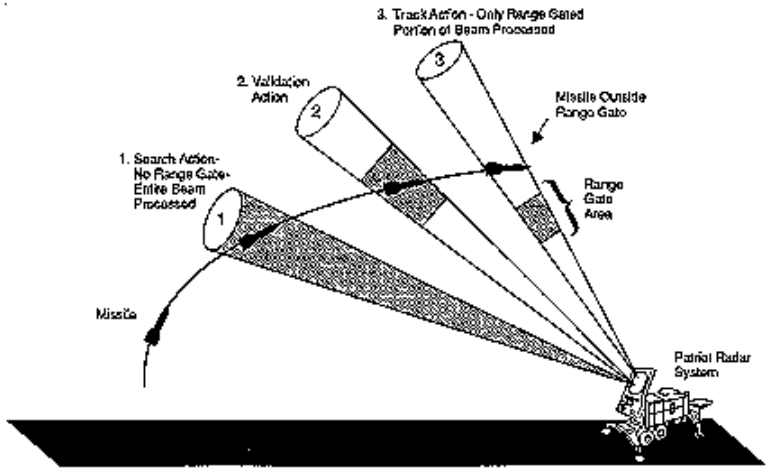
The Patriot Missile Guidance System, I

- Designed to intercept Soviet-era missiles moving at **MACH 3**
- RADAR samples are examined and internal “signatures” are examined to **identify UFO**
- RTOS calculates where the UFO should be next and then re-samples to **track UFO**
- Once UFO is at range, a Patriot is deployed to **intercept UFO**



The Patriot Missile System, II

- On **February 25, 1991** a Patriot system attempted to track a SCUD missile from Iraq during **Desert Storm**
- The SCUD was moving at **MACH 5**
- The Patriot RTOS **did not properly track** the UFO
- The Patriot tracking system **lost** the UFO and **did not deploy**



RESULT: 28 dead US soldiers

SWP

Linux Rarely Gets That Serious

SWP

The Two Forms of Embedded Systems

Old-style “Deep” Embedded Systems

New-style Embedded Systems

Old-style Embedded Systems

- Often termed “deep” Embedded Systems because they lacked significant UI features
- Designed with 8-bit processors@1-4Mhz, resulting in highly constrained designs due to limited resources:
  - 64K RAM
  - Status lights or Hexadecimal displays
- The “OS” was often a simple loop which polled sensors and then drove a display array (EPROM was a popular OS repository)
- Were usually designed as a unique – there was almost never any significant re-use opportunity

SWP

New-style Embedded Systems

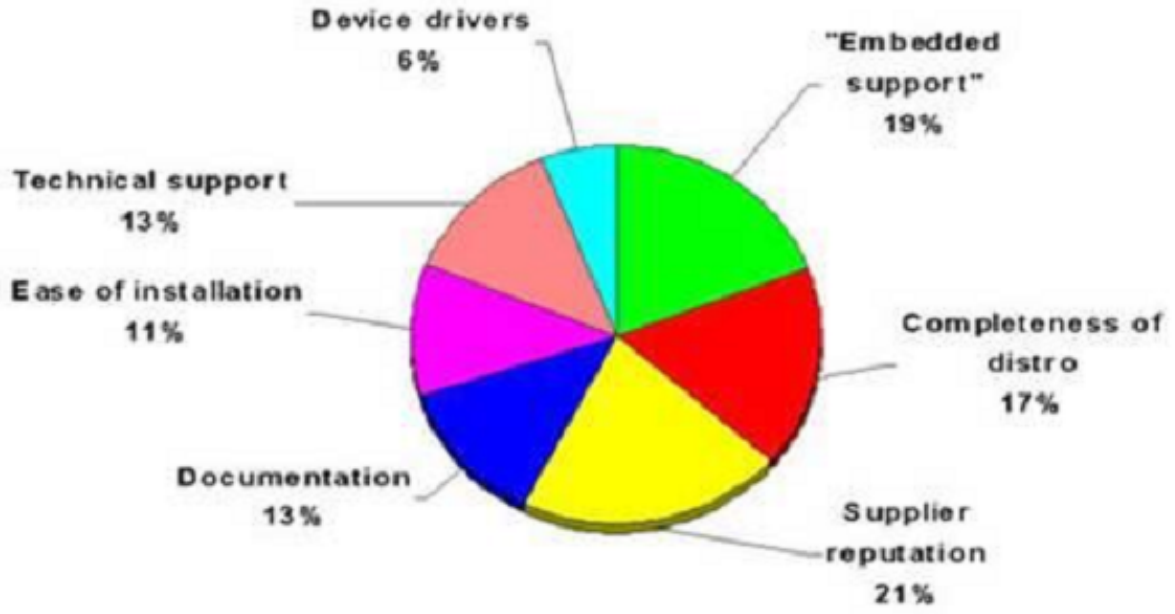
- A Direct Result of Moore's Law
  - Faster 16 and 32-bit CPUs are now cheap commodity items
  - Full-board solutions are common (we'll see one today)
  - Larger maximum RAM allows for far richer OS possibilities
  - Large EEPROM/Flash allows the OS to be persistently stored
- A Direct Result of Pioneers like Bill Joy & Linux Torvalds
  - Showed that (re)writing a world-class OS was not just something for large, well-capitalized companies
- A Direct Result of Advances in Telecommunications
  - Without the collaborative power of the Internet, the "Linux Miracle" would not have happened

SWP

What Do Developers Look For In An Embedded OS?

SWP

Which factor(s) will have the greatest influence on your choice?



Reputation, Completeness, Documentation Are Top Criteria

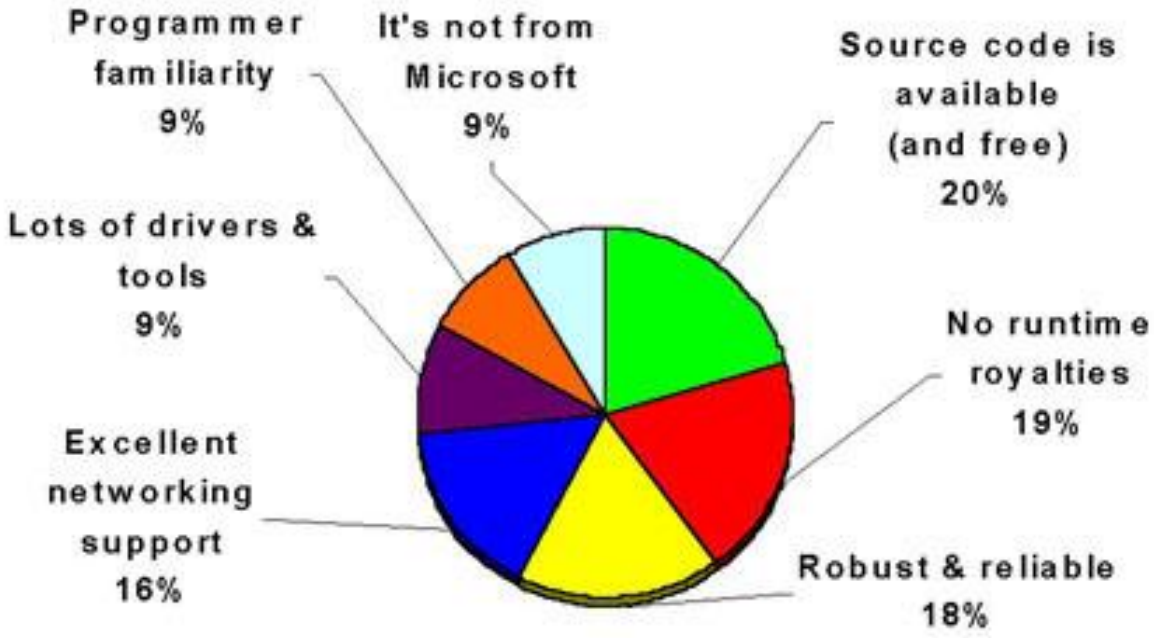
SWP

So Why Choose Linux?

- Proprietary HRT and SRT systems are expensive
- Proprietary HRT and SRT systems cannot match the pace of innovation offered by the Linux Open Source model, not to mention its transparent methods
- Proprietary HRT and SRT systems may not have such broad cross-platform support (x386, AMD, Cyrix, IBM…)
- Proprietary HRT and SRT systems may not offer the strong device support of Linux (in particular networking)
- Proprietary HRT and SRT systems may not have the rich set of development languages and debugging tools that Linux has. They may also lack its deep application pool

SWP

Why are you considering using Linux in an embedded Application?



Traditional Linux Strengths, not Anti-Microsoft Feelings

SWP

Types of **Linux** Real-Time Systems

Non Real Time (NRT)

Soft Real Time (SRT)

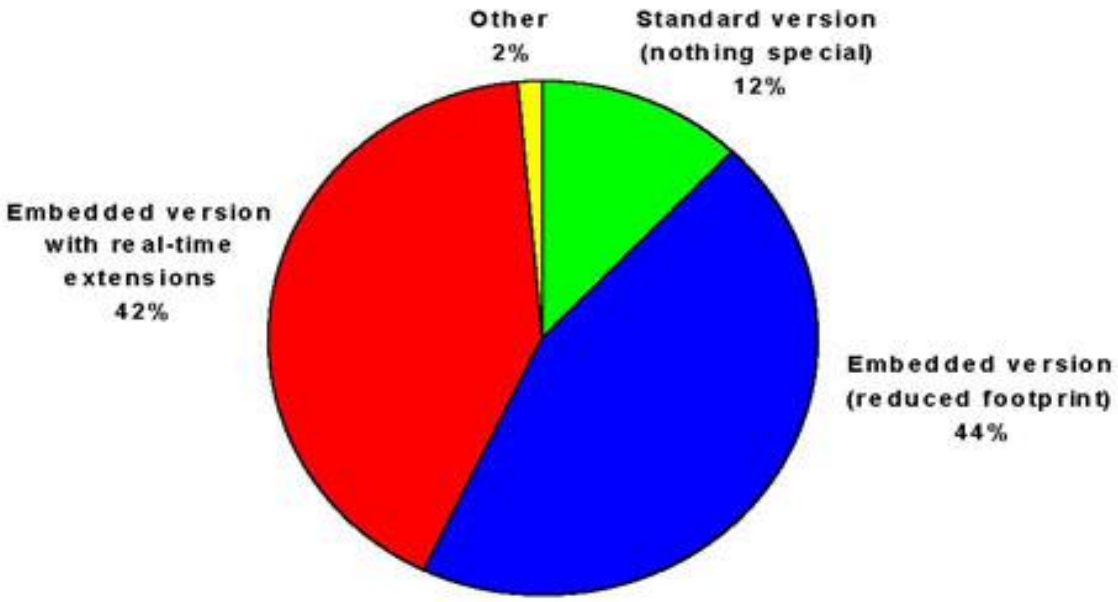
Hard Real Time (HRT)

Linux Non Real Time Systems

- This is the status of the default Linux Kernel
  - Linux Kernel lacks the following functions:
    - Prioritization
    - Preemption functions
    - Offers services on a “best effort” basis
- Just like working with the Linux you know, but
  - Unnecessary drivers removed
  - NO or only a Single shell available
  - Often developed on one platform & deployed to another

SWP

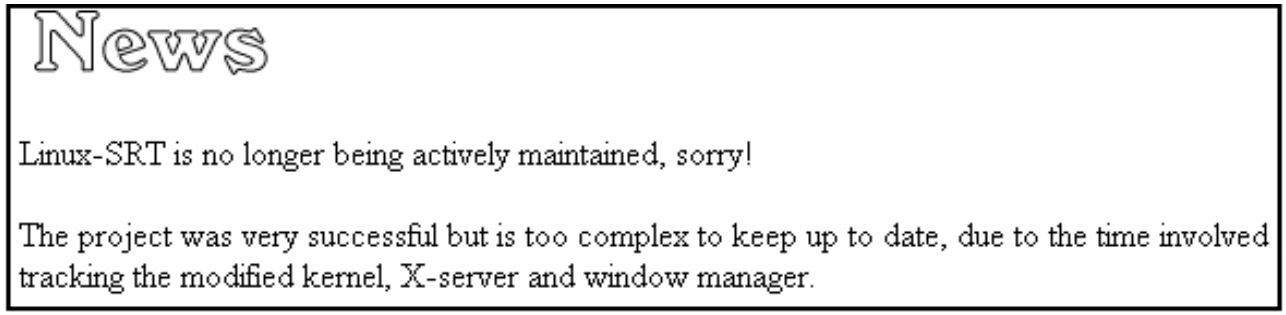
What type of Linux OS will you need?



NRT Embedded Linux May Satisfy up to 56% of Market Need

Linux Soft Real-Time Systems

- Designed to provide SRT (for multimedia application)
  - The problem with Linux-SRT was that it required modifications to the kernel
  - Keeping the kernel up-to-date became a time-consuming and ultimately futile exercise for the author, thus:



News

Linux-SRT is no longer being actively maintained, sorry!

The project was very successful but is too complex to keep up to date, due to the time involved tracking the modified kernel, X-server and window manager.

  - This may not be the disaster it seems, the code is still available and the kernel version relatively recent

Linux Hard Real-Time Systems

- There are many Linux HRT systems
  - POSIX has defined RT services in PSE54
    - POSIX 1003.13[10]
  - A product called RTLinux is available (**commercial**)
    - A Linux kernel is run as a task under a proprietary RTOS, providing a "virtual machine" to operate in.
  - A product called RTAI is available (**open source**)
    - A patch is applied to the kernel to provide a Hardware Abstraction Layer (HAL), plus associates services.

SWP

In Summary:

Regardless Of Your Embedded System Requirements
There Is A Linux For you!

NRT / SRT / HRT

SWP

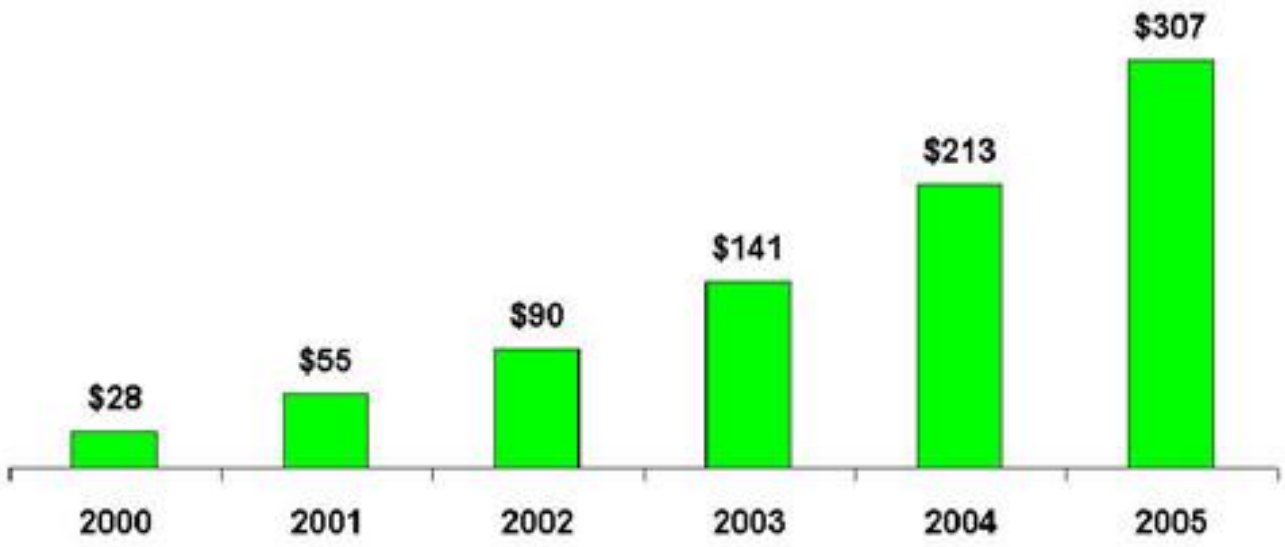
The Embedded Linux Marketplace

Just How Big Is The Embedded Linux Market Right Now?

Where Is The Embedded Linux Market Headed?

SWP

Worldwide Shipments of Embedded Linux OSes, Software Development Tools, and Related Services (in millions of dollars)

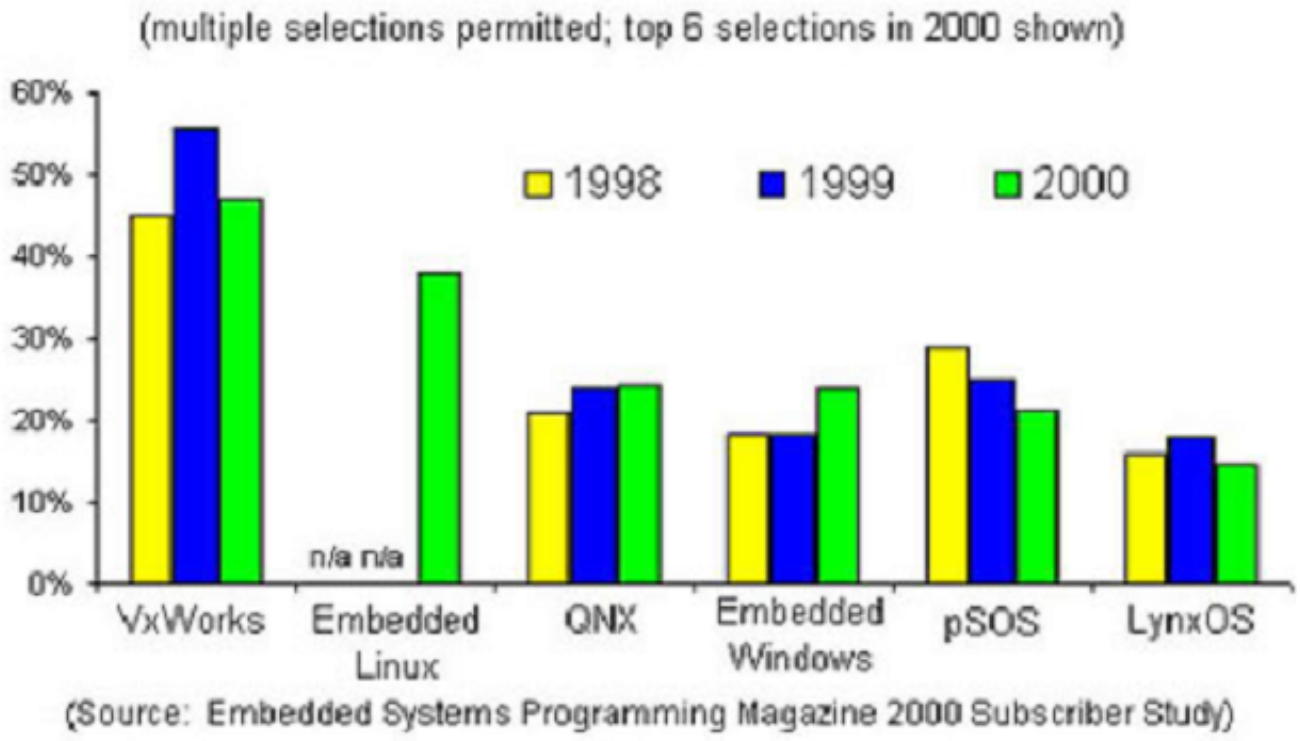


Source: Venture Development Corporation (VDC) 2000 Embedded Linux Market Study

Embedded Linux Market – $90M Now, $307M by 2005

SWP

Which 16 or 32-bit vendors would you consider when purchasing RTOSes or kernels for your embedded projects?



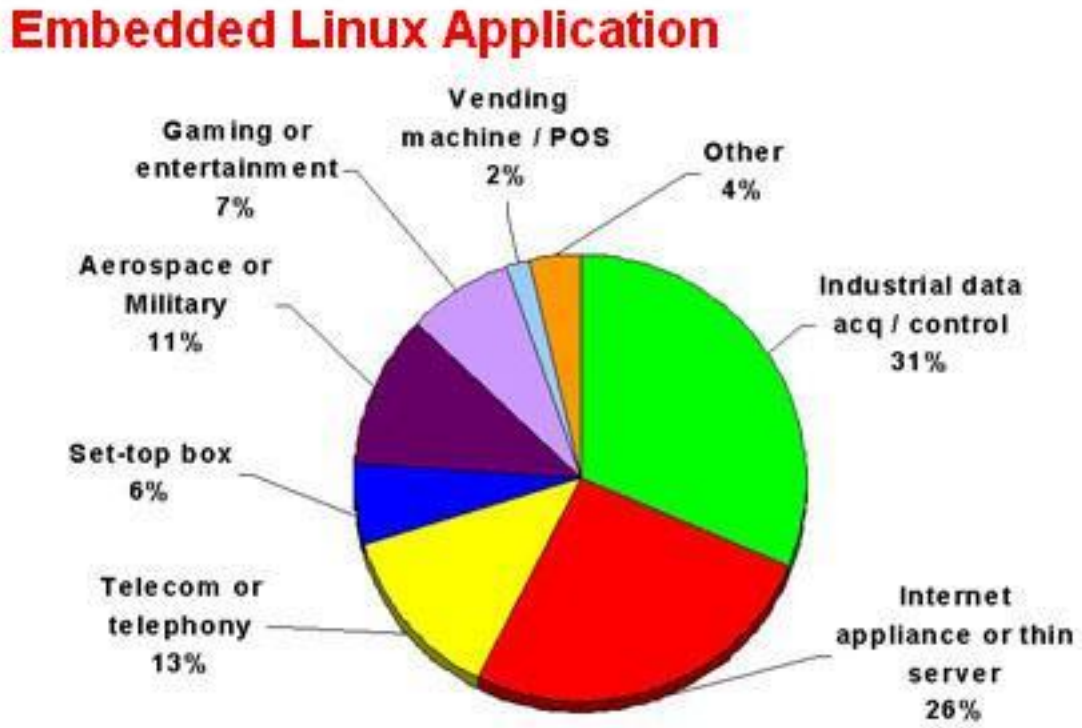
Embedded Linux - 0% Market Choice in 1999 to 40% in 2000

SWP

The Embedded Linux Marketplace

Exactly What Is being Built With Embedded Linux?

SWP



Embedded Linux – Control Systems, Internet, Telco Top List

SWP

Why Embedded Linux Is Important For Hong Kong

SWP

HK: Moving Away From The Entrepot

- HK is no longer a competitive manufacturer
- HK must move towards a Knowledge Economy
- Embedded Linux is a “knowledge product”
- Illustrative Example: Simple robot (toy)
  - It can be manufactured anywhere:
    - Australia, Malaysia, Thailand, China
  - But how does one prevent IP piracy?
    - Produce the physical article in one place
    - Produce the OS in another (HK)!
    - Place tight controls over any “handling” of the OS

Hong Kong's Future Role In Trade, I

An Ideal Scenario: Consumer Product "Coming" From HK

- HK Advantage:
  - Design Expertise
  - Global Business Expertise
  - Financial Infrastructure Maturity
  - Marketing & Trading Expertise
  - Multi-lingual Expertise

The Consumer Product Is Produced "Offshore" (in China)

- CN Advantage:
  - Lower Variable Costs (Labor)
  - Lower Overhead costs (Rent)
  - High quality awareness (ISO)

Hong Kong's Future Role In Trade, II

The Chinese Product Is Shipped To HK, "brainless"

- HK Advantage:
  - Speedy Customs Clearance

The "brain" Is Added To The Product In HK

- HK Advantage:
  - Skilled Workforce (English skills, Technical skills)

The Product Is Sold Abroad

- HK Advantage:
  - Cash Inflow
  - Hong Kong **Retains its Central** Role In The Trading Model

SWP

Picking A Software Development Kit (SDK)

SDK Choices

Our “Top” Picks

Lineo:	Embedix
LynuxWorks:	BlueCcat
Red Hat:	Red Hat Embedded Linux
Transmeta:	Midori
Your Own:	*Description to come at end*

SWP

Our Criteria

- Popularity (Google Search Results)
- Cost
- Platform Support (CPU & Device Drivers)
- Maturity

Head To Head Comparison

Vendor	Product	Popularity	$	CPU	Maturity	URL
ucLinux	**ucLinux**	**890,000**	**$FREE**	**Motorola**	**High**	www.uclinux.com
Lineo	**Embedix**	**8,860**	**$CALL**	**MULTIPLE**	**High**	www.lineo.com
Lynuxworks	**BlueCat**	**2,830**	**$USD2,699**	**MULTIPLE**	**High**	www.lynuxworks.com
Transmeta	**Midori**	**2,020**	**$FREE**	**x86**	**Medium**	www.transmeta.com
Red Hat	**RHEL**	**810**	**$USD2,500**	**MULTIPLE**	**High**	**www.redhat.com**

SWP

Guest: Alan Knowles

PEAR Core Team Member

Experienced Linux Developer

Principal Developer, Embedded Systems, Vending Robotics

SWP

Alan Knowles

- Self-Introduction

- What I will be speaking about:
  - How I Became Involved With Embedded Linux
  - What I Am Developing An Embedded Linux On (CPU)
  - What I developed an Embedded Linux platform For (Need)

  - The Skills I Used To Create My Own Embedded Linux
  - Techniques To Help You Create Your Own Embedded Linux

SWP

How I Became Involved With Embedded Linux

Because Someone Asked Me

SWP

VENDING ROBOTICS PTY. LTD

Australian Company

Produces Vending Machines for Global Clients

At Forefront of "Smart" Vending Machines Development

SWP

The "Smart" Vending Machine

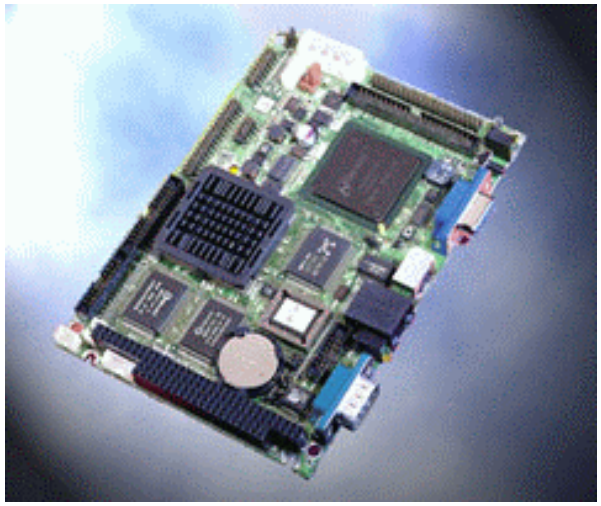
- MPV3000 MkII-R Vendor (vends odd-shapes)
  - Milks, Juices, Plastic Bottles
  - Tetra Paks, Sandwiches
- Uses microprocessor-controlled robotics
- Can retrieve products weighing up to 1Kg
- Has a Product Capacity up to 400 units
- The Unit is completely field programmable and has built-in health, diagnostic and monitoring features.

Patents: **South Africa** 98/8008; **USA** 6,253,955; **New Zealand** 503754; **Australia** S/N 742226

Applications: **Canada** 2,342,804; **China** 98810522.5; **Europe** 98-939-439.0; **Japan** 509062-2000; **Malaysia** PI 9803991; **Philippines** 1-1998-02248; **Singapore** 200000932-4

SWP

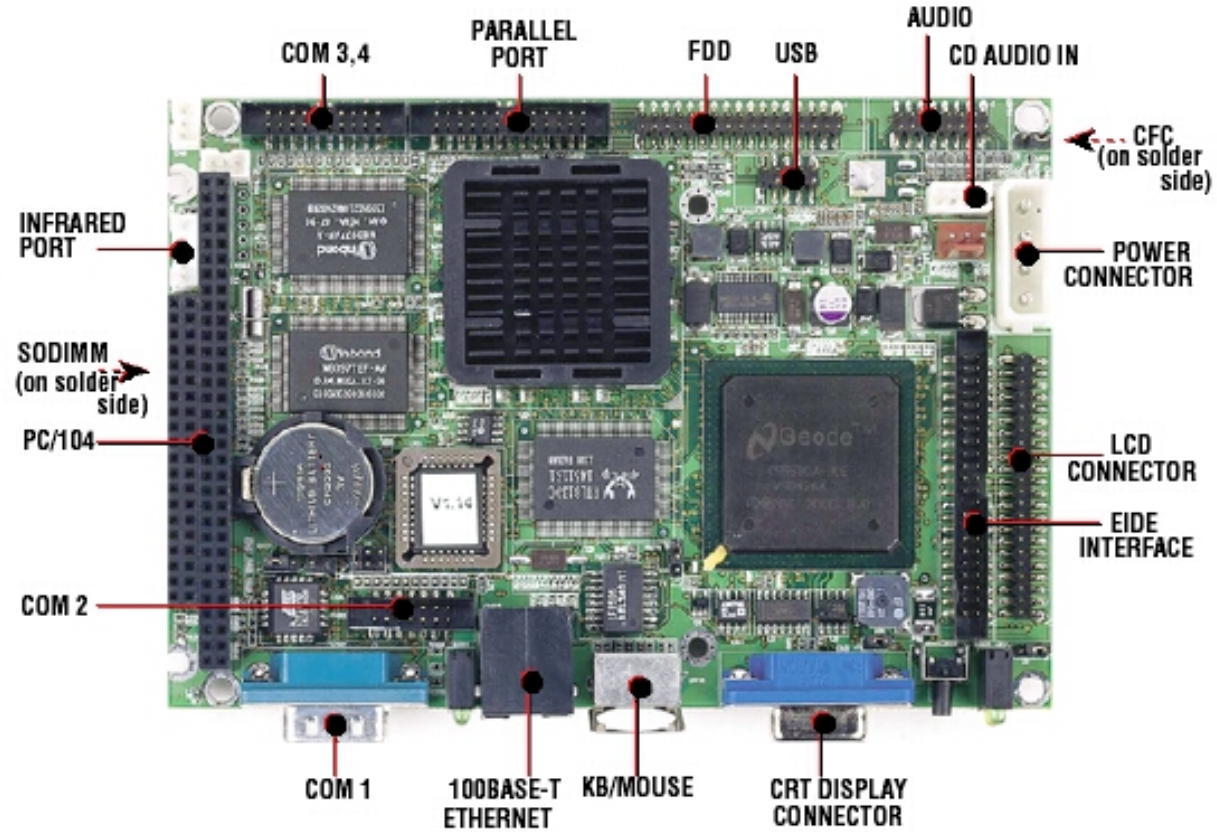
The “brain” – The PCM 5825



- **OS: “stripped” Debian**
- **LILO loads the kernel from SSD**
- **On-board 10/100 (RTL8139)**
- **Compact size (145 x 102 mm)**
- **On-board NS GX1-300 MHz CPU**
- **Base-T Ethernet**
- **Single +5 V power supply**

SWP

PCM-5825: Up Close and Personal



SWP

Going My Own Way

- Motivation
  - Why I chose not to use a commercial SDK
- Development
  - How I began to develop my own "stripped" kernel
  - How I get the "stripped" kernel onto the product
- Operation
  - What happens when the product "boots“
  - Uptime and Performance Statistics
    - 300 Days of cumulative uptime
    - No system crashes whatsoever

SWP

Rolling Your Own Embedded Linux

- Tips & Techniques
  - Remove ALL superfluous devices drivers
  - Use the highest compression available (make bzimage)
  - You **must** use statically linked libraries
    - Watch out for LARGE object files as a result!
- Traps & Pitfalls

SWP

Best Practices

Questions?