arm

Embedded Linux

Introduction

Goals

To introduce the structure of the Embedded Linux Online Course

To introduce the concept of embedded systems

To provide a few illustrative examples of Linux-based embedded systems

Summary

Course structure

Introduction to embedded systems

Linux in embedded systems

Summary

Course structure

Introduction to embedded systems

Linux in embedded systems

Course Structure

Module 1: Linux in Embedded Systems

- Definition of embedded systems
- Examples (in real world)

Module 2: Linux-based Embedded System Component Stack

- Bootloader
- Kernel
- Root file system
- Device tree
- System programs
- Application

foundament embedded linux system

arm

Course Structure

Module 3: Anatomy of a Linux-based system

- The Linux Kernel internals
- Device tree
- System programs and BusyBox

Module 4: Configuration & Build Process of an Embedded Linux System

- Buildroot
- Yocto - useful for setting the process

Module 5: Introduction to Linux Kernel Modules

- CPU – I/O interface
- I/O taxonomy
- Linux devices
- Virtual file system abstraction
- Linux Kernel modules

Course Structure

Module 6: Communication Between Kernel and User Space

- Module level communication point of view
- User level communication point of view

Module 7: Application Demo: Building a Ranging Sensor Kernel Module

- The sysfs file system
- Building Linux support for the HC-SR04 ultrasonic ranging sensor

Module 8: System Debugging & Profiling lab exercises

The majority of the theoretical lectures will be complemented with lab exercises.

Summary

Course structure

Introduction to embedded systems

Linux in embedded systems

What Is an Embedded System?

computer
processor
memory
input / output device

ใช้ resource ให้น้อยที่สุดที่จะ
fullfill requirement

It is a special-purpose computer designed for a specific application.

Example of application:
internal combustion engine (ICE)

Example of embedded system:
electronic control unit for ICE

Embedded System Components

Two main components

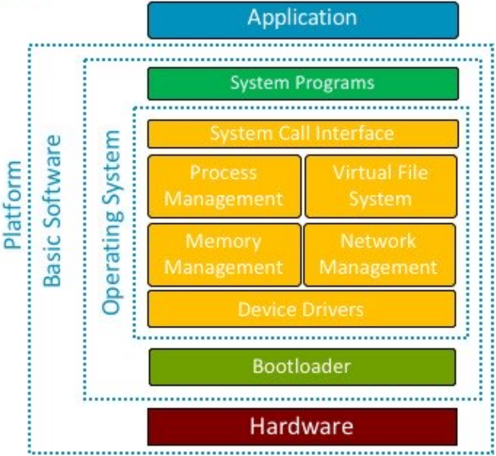
- Application software ที่มี service ตามที่ user ต้องการ
- Platform : software , hardware ที่เป็น resource ให้ทำ task ได้

Application

- Software that implements the functionalities for which the embedded system is intended (e.g., to control an ICE)

Platform

- Combination of hardware and basic software components that provides the services needed for the application to run
- Basic software includes system programs, operating system, bootloader

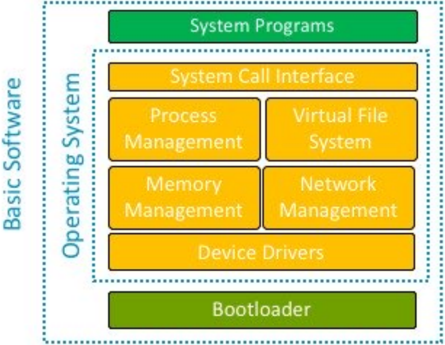


arm

Basic Software

Abstracts the hardware details by providing easy-to-use functionalities, such as:

- Access to the resources through user-friendly utilities known as system programs
- Example: `ls` to list the content of a directory
- Efficient access to the resources provided by the hardware through the operating system
- Example: CPU real-time scheduling, device driver management
- Initialization of hardware resources at power-up and execution of the operating system through the bootloader



Operating Systems for Embedded Systems

There are many solutions available which serve different purposes depending on the requirements of the application.

- Example 1
  ควรเป็นอย่างแรกที่ถูก
  พิจารณาเพื่อต่อไปเลือก OS
- Needs: deterministic real-time operating system for low-cost devices, with little memory footprint
- Possible solutions: ARM RTX, Micrium µC/OS, FreeRTOS, and others
- Example 2
- Needs: multi-core and networking support, advanced graphics, and complex device handling
- Possible solutions: Linux, Android, and Windows

Example 1: deterministic real-time system

Example 2: in-vehicle infotainment

http://linuxgizmos.com/linux-based-in-vehicle-infotainment-on-the-rise/

Summary

Course structure

Introduction to embedded systems

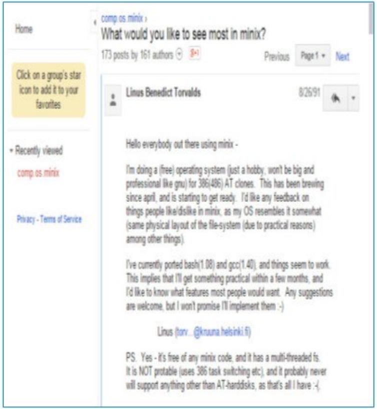
Linux in embedded systems

Why Linux-based Embedded Systems? (สาเหตุที่ embedded linux เป็นที่นิยม)

1. Open Source (under GNU General Public License v2.0 : GPLv2)
   - The full source code is available for learning and adaptation
2. Engaged community maintaining and improving Linux regularly
   - Companies
   - Individuals
   - Academics
   - Hobbyists
3. Flexible and adaptable: supports many hardware/System-on-Chip (SoC) configurations
   - Based on ARM, x86, PowerPC, SPARC, etc. (อยู่ได้ในทุกๆ architecture)
4. Proven in many different scenarios (see next slides) → สามารถ test ได้เยอะ
5. Supported by a very large ecosystem of software
   - Bootloader, system programs, networking services, advanced graphic services, etc.
6. Royalty-free

Linux Evolution

August 26, 1991: everything started with this post to comp.os.minix (เมื่อก่อนเป็นอันใหญ่ๆ อันเดียว)



What would you like to see most in minix?

Linus Benedict Torvalds 8/26/91

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

แต่ละ: categories refer ถึง [illegible] level of code base ?

Today several kernel categories exist, including:

- Prepatch or "RC" kernels, which are pre-releases maintained and released by Linus Torvalds.
- Mainline kernel is maintained by Linus Torvalds, and is where all new features are introduced. New mainline kernels are released every 2-3 months.
- Long-term kernels are older releases subject to "long-term maintenance". Important bug fixes are applied to such kernels.

Longterm release kernels

Version	Maintainer	Released	Projected EOL
4.4	Greg Kroah-Hartman	early 2016	Feb, 2018
4.1	Greg Kroah-Hartman	2015-06-21	Sep, 2017
3.18	Sasha Levin	2014-12-07	Jan, 2017
3.14	Greg Kroah-Hartman	2014-03-30	Aug, 2016
3.12	Jiri Slaby	2013-11-03	2016
3.10	Greg Kroah-Hartman	2013-06-30	End of 2015
3.4	Li Zefan	2012-05-20	Sep, 2016
3.2	Ben Hutchings	2012-01-04	May, 2018
2.6.32	Willy Tarreau	2009-12-03	Early 2016

https://www.kernel.org/category/releases.html

Linux-based Embedded System: Example 1

In-flight entertainment systems

> “Linux is particularly suited for in-flight entertainment because it's simple, not weighed down by accompanying programs, and easily adaptable to many environments.”

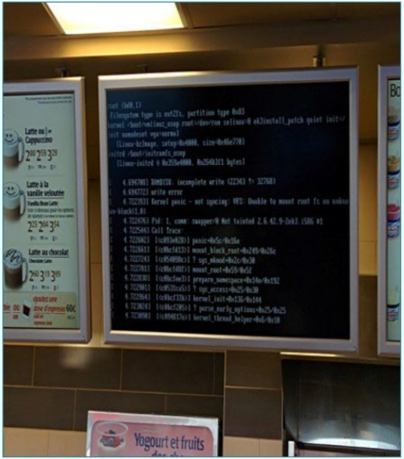
http://www.linuxinsider.com/story/The-Flying-Penguin-Linux-In-Flight-Entertainment-Systems-65541.html

http://www.linuxinsider.com/story/The-Flying-Penguin-Linux-In-Flight-Entertainment-Systems-65541.html

Linux-based Embedded System: Example 2

Tim Horton's Café and Bake Shop

The screen displays the messages Linux produces during boot-up. In particular, we can recognize a kernel panic, as the kernel is not able to find the root file system.



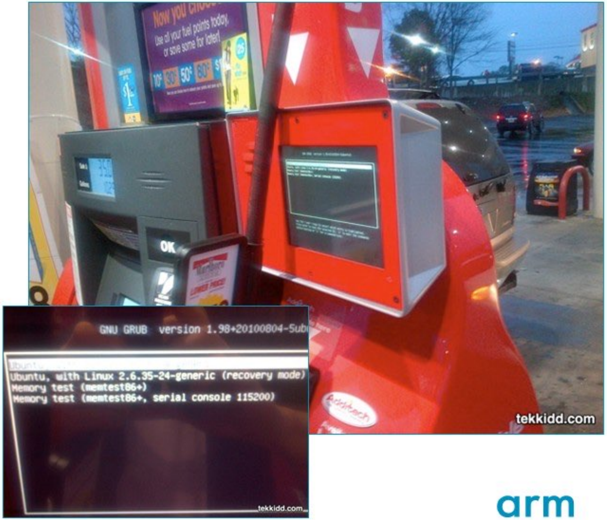
```
oot (hd0,1)
Filesystem type is ext2fs, partition type 0x83
ernel /boot/vmlinuz_osmp root=/dev/ram selinux=0 ek3install_patch quiet init=/
nit nomodeset vga=normal
   [Linux-bzImage, setup=0x4000, size=0x46e778]
nitrd /boot/initramfs_osmp
   [Linux-initrd @ 0x355e4000, 0x2b4b3f1 bytes]

[    4.694700] RAMDISK: incomplete write (22343 != 32768)
[    4.694772] write error
[    4.722393] Kernel panic - not syncing: VFS: Unable to mount root fs on unkno
n-block(1,0)
[    4.722476] Pid: 1, comm: swapper/0 Not tainted 2.6.42.9-2ek3.i586 #1
[    4.722544] Call Trace:
[    4.722682]  [<c093e02b>] panic+0x5c/0x26e
[    4.722661]  [<c08cf413>] mount_block_root+0x249/0x26c
[    4.722724]  [<c05409bc>] ? sys_mknod+0x2c/0x30
[    4.722781]  [<c08cf40f>] mount_root+0x59/0x5f
[    4.722830]  [<c08cfee3>] prepare_namespace+0x14e/0x192
[    4.722901]  [<c0531ca5>] ? sys_access+0x25/0x30
[    4.722964]  [<c08cf33b>] kernel_init+0x136/0x144
[    4.723024]  [<c08cf205>] ? parse_early_options+0x25/0x25
[    4.723098]  [<c094017e>] kernel_thread_helper+0x6/0x10
```

Linux-based Embedded System: Example 3

A gas station pump

The screen displays the messages of a Linux bootloader.
This gas station is powered by Linux Ubuntu distribution with Kernel 2.6.35.



```
        GNU GRUB  version 1.98+20100804-5ub

 [illegible]
 Ubuntu, with Linux 2.6.35-24-generic (recovery mode)
 Memory test (memtest86+)
 Memory test (memtest86+, serial console 115200)
```

tekkidd.com

tekkidd.com