

An Introduction to the Operating Systems of the IoT

Elahe Jalalpoor and Parham Alvani

Amirkabir University of Technology

`elahejalalpoor@gmail.com`

`parham.alvani@gmail.com`

Jun 18, 2015

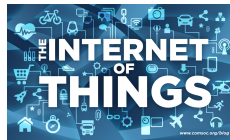


What is IoT...

The Internet of Things (IoT) is a scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

What is IoT...

The Internet of Things (IoT) is a scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.



Open Source Operating Systems for the IoT

- ▶ FreeRTOS
- ▶ RIOT
- ▶ Contiki
- ▶ TinyOS
- ▶ Embedded Linux
- ▶ OpenWSN



FreeRTOS

- ▶ FreeRTOS is designed to be **small** and **simple**.
- ▶ The kernel itself consists of only three or four C files.
- ▶ It provides methods for multiple threads or tasks, mutexes, semaphores and software timers.
- ▶ Key features are **very small memory footprint**, **low overhead**, and **very fast execution**.



- ▶ RIOT is a **real-time multi-threading** operating system.
- ▶ RIOT is based on design objectives including:
 - Energy-Efficiency
 - Reliability
 - Real-Time Capabilities
 - Small Memory Footprint
 - Modularity
 - Uniform API Accessindependent of the underlying hardware
(this API offers partial POSIX compliance)



- ▶ Contiki is an open source operating system for **networked**, **memory-constrained** systems
- ▶ Contiki provides three network mechanisms:
 - The uIP stack, which provides IPv4 networking,
 - The uIPv6 stack, which provides IPv6 networking,
 - The Rime stack, which is a set of custom lightweight networking protocols designed specifically for low-power wireless networks.

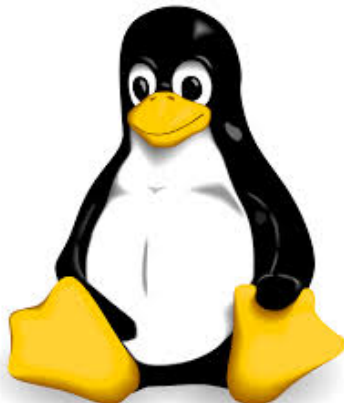


- ▶ TinyOS is a **component-based** operating system and platform targeting wireless sensor networks.
- ▶ TinyOS is an embedded operating system written in the **nesC programming language** as a set of cooperating tasks and processes.



Embedded Linux

- ▶ Embedded Linux is created using OpenEmbedded, the build framework for embedded Linux.
- ▶ OpenEmbedded offers a best-in-class cross-compile environment.



- ▶ The goal of the OpenWSN project is to provide open-source implementations of a complete protocol stack based on Internet of Things standards, on a variety of software and hardware platforms.



Comparison

| OS | Min RAW | Min ROM | C Support | C++ Support |
|---------|--------------|------------|-----------------|--------------|
| Contiki | $< 2kB$ | $< 30kB$ | Partial support | No support |
| Tiny OS | $< 1kB$ | $< 4kB$ | No support | No support |
| Linux | $\sim 1MB$ | $\sim 1MB$ | Full support | Full support |
| RIOT | $\sim 1.5kB$ | $\sim 5kB$ | Full support | Full support |



Comparison

| OS | Multi-Threading | Modularity | Real-Time |
|---------|-----------------|-----------------|-----------------|
| Contiki | Partial support | Partial support | Partial support |
| Tiny OS | Partial support | No support | No support |
| Linux | Full support | Partial support | Partial support |
| RIOT | Full support | Full support | Full support |



Why Not Linux?

Real-Time Linux

Controlling a laser with Linux is crazy, but everyone in this room is crazy in his own way. So if you want to use Linux to control an industrial welding laser, I have no problem with your using PREEMPT_RT.

- Linus Torvalds



Why Not Linux?

- ▶ Linux certainly is a robust, developer-friendly OS
- ▶ Linux has a disadvantage when compared to a real-time operating system:
 - Memory footprint
 - It simply will not run on 8 or 16-bit MCUs



Internet Usage and Protocols for the IoT

Web Hundreds / thousands of bytes



- Inefficient content encoding
- Huge overhead, difficult parsing
- Requires full Internet devices

Internet of Things Tens of bytes



- Efficient objects
- Efficient Web
- Optimized IP access

Questions?