Operating System Support for Embedded Devices

February 7, 2024

Jonas Wozar

Friedrich-Alexander-Universität Erlangen-Nürnberg
Embedded Devices:

- small
- usecase-specific
- limited resources

Operating Systems:

- FreeRTOS
- Zephyr
- ...

Embedded Devices and Operating Systems
Question

Are multi-purpose OSs the best solution for embedded devices?
Related OS Concepts

Tinkertoy

Evaluation

Conclusion
Related OS Concepts
Real Time Operating Systems (RTOS)
  - aimed at time-sensitive operations
Unikernels
  - lightweight
  - designed to run a single application
Exokernels
  - application-level resource management
  - reduced OS abstraction
  - application-specific customization
Library Operating Systems
  - customization with selected libraries
...
Tinkertoy
Overview

Tinkertoy

→ application-specific OS
Overview

Tinkertoy
- set of modules

→ application-specific OS
Overview

Tinkertoy

- set of modules
- components build modules

→ application-specific OS
Overview

Tinkertoy

- set of modules
- components build modules
- building blocks build components

→ application-specific OS
10 modules

- Constraints
- Scheduler
- Memory Allocator
- Context Switcher
- Execution State

- System Call
- Dispatcher
- Kernel Service Routines
- Execution Models
- Task Control Block
10 modules

- Constraints
- **Scheduler**
- Memory Allocator
- Context Switcher
- Execution State

- System Call
- Dispatcher
- Kernel Service Routines
- **Execution Models**
- Task Control Block
Templates
- generic definition of modules

Concepts
- define constraints on types

Functors
- encapsulate building blocks
Example: FIFO Scheduler

using Policy = PolicyWithEnqueueExtensions<FIFO, Counter>;
class CustomFIFOScheduler : public SchedulerAssembler<Policy, 
    TaskCreation::Cooperative::KeepRunningCurrent<Task>, 
    TaskTermination::Common::RunNext<Task>, 
    TaskBlocked::Common::RunNext<Task>, 
    TaskUnblocked::Cooperative::KeepRunningCurrent<Task>, 
    TaskYielded::Common::RunNext<Task>> {}
Execution Models

Event-Driven

- single/few thread(s)
- can be expressed as state-machine
- events define control flow
- example: automatic shades

Thread-Based

- number of threads
- typically short-lived
- concurrent task handling
- example: gateways
Task Control Block

- contain information about a task
- required for every task
- can be built from building blocks
- constraints (stack, system calls, ID, priority, state)
- initializer & finalizer
Evaluation
Setup

- Monitor
- Actuator
- Gateway

Diagram showing the connections between Monitor, Actuator, and Gateway.
Monitor

- event-driven
- measures soil moisture
- informs Actuator
Actuator

- event-driven
- starts/stops watering
- communicates with Gateway
Gateway

- thread-based
- translates CoAP to HTTP
Memory Footprint

Kernel

- Monitor
- Actuator
- Gateway

Memory in KB

Tinkertoy
FreeRTOS
Zephyr
Flash Footprint

Kernel

- Tinkertoy
- FreeRTOS
- Zephyr

Monitor
- Flash in KB: 11

Actuator
- Flash in KB: 6

Gateway
- Flash in KB: 12
Performance

Median

Mean

RTT in ms

0.6

0.4

0.2

0

Median

Mean

Tinkertoy

FreeRTOS

Zephyr
Conclusion
Conclusion

- few lines of code → usecase-specific OS
- significantly smaller memory footprint
- no performance impact
- no networking support
- no synchronization primitives
Questions?