Predictable Virtualization on Memory-Protection Unit-Based Microcontrollers

Runyu Pan, Gregor Peach, Yuxin Ren, Gabriel Parmer
The George Washington University
panrunyu@gwu.edu
Reliability
Reliability
Tesla Model 3 gets hacked, reveals more details and great potential for dual motor/performance versions

Fred Lambert - Feb. 25th 2018 10:31 am ET - @FredericLambert

Security
Security
Flexibility
Flexibility
RTOS

- Control task 1
- Control task 2
- Control task 3
- GUI interface
- Input framework
- Networking stacks
- Scheduler
- Memory management
- Driver frameworks

RTOS
RTOS

Big market share
RTOS

Rich frameworks

Big market share
RTOS

Rich frameworks

μC/OS
RTOS and Stacks

Big market share

Research features

Domain-specific

Big market share

Hobbyists
Hobbyist apps

RT-Thread frameworks

Existing applications

Certified applications

Bare-metal applications

Native interface applications

Arduino

RT-Thread

FreeRTOS

μC/OS
Complex codebase
Complex codebase

Security vulnerabilities
Complex codebase
Security vulnerabilities
Not real-time
Protection

MMU support

HW support

Virt. extensions

Mem size

Large memory
Protection

MMU support

MPU support

HW support

Virt. extensions

Mem size

Large memory
No virtualization, but still have protection
Protection  MMU support  MPU support

HW support  Virt. extensions

Mem size  Large memory

FreeRTOS
32k ROM 16k RAM

Large memory

Virt. extensions

Mem size

FreeRTOS
32k ROM 16k RAM

Large memory
Protection

MMU support

MPU support

HW support

Virt. extensions

Mem size

Large memory

FreeRTOS

16k RAM

32k ROM

Mem size

Large memory
VT-x, VT-d, VT-i
AMD-V
VT-x, VT-d, VT-i
AMD-V

No hardware assistance
Protection

MMU support

MPU support

HW support

Virt. extensions

Mem size

Large memory

FreeRTOS

32k ROM

16k RAM
Protection | MMU support | MPU support
---|---|---
HW support | Virt. extensions | Para-virtualization
Mem size | Large memory
Protection

MMU support

MPU support

HW support

Virt. extensions

Para-virtualization

Mem size

Large memory

FreeRTOS

32k ROM

16k RAM

Large memory

Virt. extensions

Mem size
<table>
<thead>
<tr>
<th>Item</th>
<th>RAM</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>8 GB</td>
<td>2 TB</td>
</tr>
<tr>
<td>GPOS</td>
<td>1 GB</td>
<td>32 GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>RAM</th>
<th>Flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>512 KB</td>
<td>2 MB</td>
</tr>
<tr>
<td>RTOS</td>
<td>16 KB</td>
<td>32 KB</td>
</tr>
</tbody>
</table>
Protection

MMU support

MPU support

HW support

Virt. extensions

Para-virtualization

Mem size

Large memory
Protection

MMU support

MPU support

HW support

Virt. extensions

Para-virtualization

Mem size

Large memory

RTOS footprint
VM vectors  VM threads  

FreeRTOS

Paravirtualization extensions

μC/OS

Paravirtualization extensions

MPU-imposed isolation
- VM vectors
- VM threads
- FreeRTOS
- Paravirtualization extensions
- MPU-imposed isolation
- CPU & I/O virtualization
- Scheduler
- I/O manager
- Virtual machine monitor
- Composite user-level components
Scheduler

VM scheduler

TCap

Temporal capability (RTSS'17)
Scheduler

VM scheduler

TCap

Temporal capability (RTSS'17)

RTOS scheduler

RTOS scheduler

RTOS scheduler

TCap

TCap

TCap
I/O

Hardware
MPU

Cortex-M memory

I/O manager

VM B

Shared memory

VM A

May share memory
I/O manager shares memory with every VM.

May share memory.

Cortex-M memory

- Shared memory
- I/O manager
- VM B
- VM A
- Shared memory
MPU

Cortex-M memory

- Shared memory
- I/O manager
- VM B
- VM A

Size constraints: power of 2

I/O manager shares memory with every VM

May share memory
MPU

Cortex-M memory

Shared memory

I/O manager

Shared memory

VM B

Shared memory

VM A

I/O manager shares memory with every VM

May share memory

Start address constraints: align to size

Size constraints: power of 2

[Image of diagram with various memory blocks and labels]
MPU

Cortex-M memory

- Shared memory
- I/O manager
- Shared memory
- VM A
- VM B

I/O manager shares memory with every VM

Disabled subregions

MAY SHARE MEMORY

Start address constraints: align to size

Size constraints: power of 2

Subregions: disable or enable
MPU

Cortex-M memory

- Shared memory
- I/O manager
- Shared memory

VM B

VM A

- I/O manager shares memory with every VM
- Disabled subregions
- May share memory
- Start address constraints: align to size

Total region number limit: 4 - 16

Subregions: disable or enable

Size constraints: power of 2
Challenge:

1. Unify MMU/MPU

2. MPU region placement
Path-compressed Radix Trie

0x0000  0x1000
Path-compressed Radix Trie

0x0000

0x0800

0x1000

0x0000
Path-compressed Radix Trie

0x0000   0x1000

0x0000   0x0800

0x0000   0x0400   0x0800   0xC00
Path-compressed Radix Trie

Diagram of a path-compressed Radix Trie with node addresses 0x0000, 0x0000, 0x0800, 0x0800, 0x1000, 0x1000, 0x0400, 0x0400, 0x0C00, 0x0C00, 0x1400, 0x1400.
Path-compressed Radix Trie

Diagram showing path-compressed Radix Trie with addressing values.
Path-compressed Radix Trie
2 Region, 4 Subregions per Region

- Shared by 3
- Shared by 2
- Private
- Empty
2 Region, 4 Subregions per Region

- Shared by 3
- Shared by 2
- Private
- Empty
## 2 Region, 4 Subregions per Region

<table>
<thead>
<tr>
<th>Shared by 3</th>
<th>Shared by 2</th>
<th>Private</th>
<th>Empty</th>
</tr>
</thead>
</table>

- **Shared by 3**: 3 stars
- **Shared by 2**: 2 stars
- **Private**: 1 star
- **Empty**: 0 stars
2 Region, 4 Subregions per Region

Shared by 3

Shared by 2

Private

Empty

Requires 3 regions - Ran out of regions!
Algorithm

Goal: layout to best use limited #regions
Algorithm

Goal: layout to best use limited #regions

Find a VM that have max #regions

Pick a region that have max. #VMs

Merge with a region w/most similar sharing
Algorithm

Goal: layout to best use limited regions

Find a VM that have max regions

Pick a region that have max. VMs

Merge with a region w/most similar sharing

Intuition: shrink as many regions as possible!
2 Region, 4 Subregions per Region

Requires 3 regions - Ran out of regions!
Requires 3 regions - Ran out of regions!
2 Region, 4 Subregions per Region
2 Region, 4 Subregions per Region

Requires 3 regions - Ran out of regions!
2 Region, 4 Subregions per Region

Requires 3 regions - Ran out of regions!
2 Region, 4 Subregions per Region
2 Region, 4 Subregions per Region

Requires 2 regions - Placement feasible!
Toolchain

**VM specs**
- Private
- Shared

**MPU specs**
- Size
- Align
- Subregion
Toolchain

VM specs
- Private
- Shared

MPU specs
- Size
- Align
- Subregion

Algo
Toolchain

VM specs
- Private
- Shared

MPU specs
- Size
- Align
- Subregion

Algo

Linker script

Memory manager
## Eval: Memory Overhead

<table>
<thead>
<tr>
<th>Components</th>
<th>ROM Size</th>
<th>RAM Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>135 KB</td>
<td>37 KB</td>
</tr>
<tr>
<td>Scheduler</td>
<td>65 KB</td>
<td>24 KB</td>
</tr>
<tr>
<td>I/O manager</td>
<td>0.5 KB</td>
<td>0.5 KB</td>
</tr>
<tr>
<td>FreeRTOS/VM</td>
<td>41 KB</td>
<td>30 KB</td>
</tr>
</tbody>
</table>
## Eval: Memory Overhead

<table>
<thead>
<tr>
<th>Components</th>
<th>ROM Size</th>
<th>RAM Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>135 KB</td>
<td>37 KB</td>
</tr>
<tr>
<td>Scheduler</td>
<td>65 KB</td>
<td>24 KB</td>
</tr>
<tr>
<td>I/O manager</td>
<td>0.5 KB</td>
<td>0.5 KB</td>
</tr>
<tr>
<td>FreeRTOS/VM</td>
<td>41 KB</td>
<td>30 KB</td>
</tr>
</tbody>
</table>

Scale up to 8 VMs on 512 KB SRAM
## Eval: Memory Overhead

<table>
<thead>
<tr>
<th>VM</th>
<th>ROM Size</th>
<th>RAM Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>basicmath</td>
<td>62 KB</td>
<td>30 KB</td>
</tr>
<tr>
<td>dijkstra</td>
<td>64 KB</td>
<td>70 KB</td>
</tr>
<tr>
<td>gsm</td>
<td>87 KB</td>
<td>30 KB</td>
</tr>
<tr>
<td>pbmsrch</td>
<td>81 KB</td>
<td>31 KB</td>
</tr>
</tbody>
</table>
## Eval: Memory Overhead

<table>
<thead>
<tr>
<th>VM</th>
<th>ROM Size</th>
<th>RAM Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>basicmath</td>
<td>62 KB</td>
<td>30 KB</td>
</tr>
<tr>
<td>dijkstra</td>
<td>64 KB</td>
<td>70 KB</td>
</tr>
<tr>
<td>gsm</td>
<td>87 KB</td>
<td>30 KB</td>
</tr>
<tr>
<td>pbmsrch</td>
<td>81 KB</td>
<td>31 KB</td>
</tr>
</tbody>
</table>

VM + Applications still have low mem. overhead
## Eval: Paravirtualization Effort

<table>
<thead>
<tr>
<th>Item</th>
<th>Source lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeRTOS HAL</td>
<td>483</td>
</tr>
<tr>
<td>FreeRTOS/MPU HAL</td>
<td>622</td>
</tr>
<tr>
<td>FreeRTOS/VM HAL</td>
<td>363</td>
</tr>
</tbody>
</table>
### Eval: Paravirtualization Effort

<table>
<thead>
<tr>
<th>Item</th>
<th>Source lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeRTOS HAL</td>
<td>483</td>
</tr>
<tr>
<td>FreeRTOS/MPU HAL</td>
<td>622</td>
</tr>
<tr>
<td>FreeRTOS/VM HAL</td>
<td>363</td>
</tr>
</tbody>
</table>

Paravirtualization effort is small
Eval: Cortex-M7 @ 216MHz
Eval: Cortex-M7 @ 216MHz
Eval: Cortex-M7 @ 216MHz
Eval: Cortex-M7 @ 216MHz

RT-Linux futex context switch : > 25000 Cycles
Eval: Cortex-M7 @ 216MHz

Overhead percentage (%) vs. Interrupt inter-arrival time (ms)

- Composite
- FreeRTOS
- FreeRTOS/MPU
- FreeRTOS/VM

Events:
- Motion HID
- Sensor
- CAN Motors
Eval: Cortex-M7 @ 216MHz

Max. 3% overhead for apps listed above
Conclusion
Conclusion

Paravirtualization infrastructure for MCUs
Memory management across MPU & MMU
Algorithm to enable efficient placement
Conclusion

Paravirtualization infrastructure for MCUs

Memory management across MPU & MMU

Algorithm to enable efficient placement

Evaluation of the system - fit for realistic settings
?? & /* */

composite.seas.gwu.edu