Real-Time Systems (in English)
CAS: Benchmarking of RTOSes

Martin Havlík

10 April 2025

RTOS vs GPOS

Real-time vs general-purpose[1].

- precise timing constraints, predictability, stability
- deterministic (priority based) scheduling
- minimal interrupt latency, task switch latency
- hard – aerospace, robotics, industrial control systems
- firm – multimedia, networking
- soft – desktop OS, web server

[1] https://www.ibm.com/think/topics/real-time-operating-system

Select RTOSes

- `RTLinux` – hard RTOS microkernel, runs Linux OS fully preemptive. Robotics, data acquisition, manufacturing, etc.
- `FreeRTOS` – open-source, embedded RTOS, IoT, 40+ proc. architectures, small footprint
- `VxWorks` – commercial RTOS, safety-critical and high-performance embedded systems – aerospace, industrial automation.
- `Keil RTX5` – royalty-free, CMSIS-RTOS API v2 for Cortex-M based devices, µVision IDE/Debugger
- `Zephyr` – embedded, MCU, 750+ boards, highly connected, ecosystem
- `uC/OS-II, uC/OS-III` – portable, ROMable, scalable, preemptive, real-time deterministic multitasking kernel for microprocessors, microcontrollers and DSPs
- `RTEMS` – open source, POSIX, space flight, medical, networking, embedded
- `PREEMPT_RT` – was a set of patches for Linux kernel, hard & soft RT, fully merged into mainline since Sep 2024

Benchmarked properties

Deterministic behaviour

- thread creation
- round robin scheduling
- semaphore & mutex locking/releasing

Timing

- inter-task communication
- queue communication
- stream, message buffers
- task switching
- semaphore acquire release
- task activation from ISR
- task activation jitter induced by priority inversion

Tools and benchmark suites

Worst-Case Execution Time (WCET) analysis

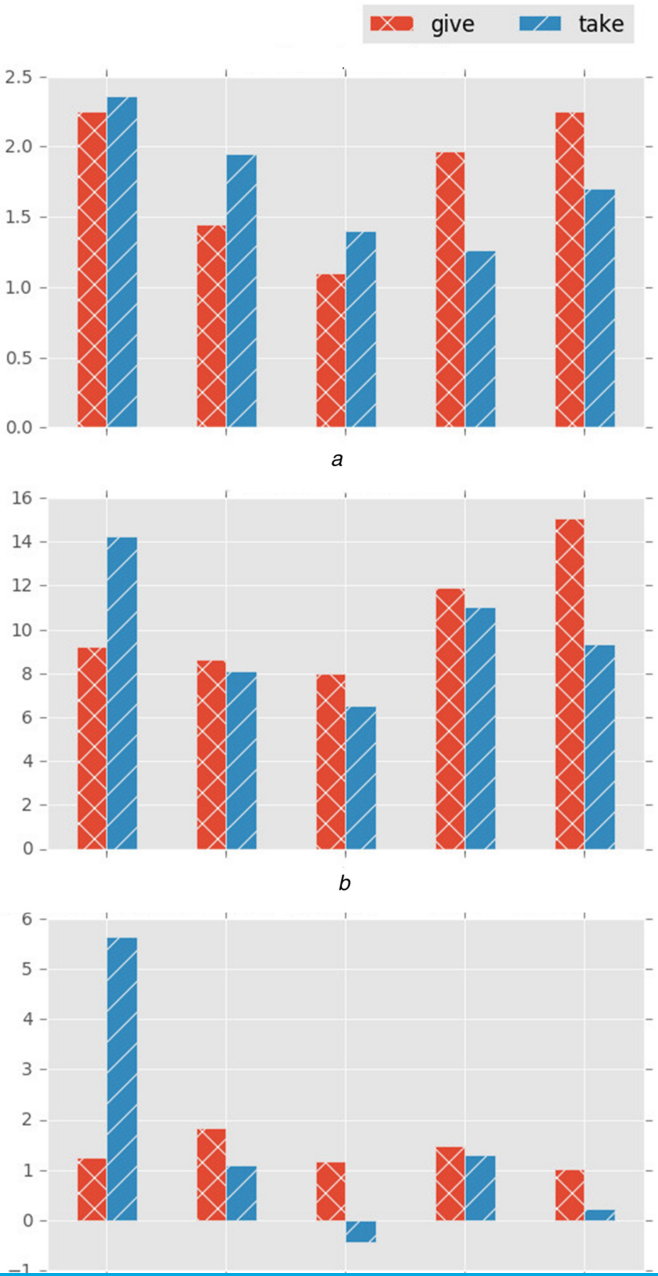
- AbsInt aiT – static analysis of binary executables, abstract interpretation, formal cache and pipeline models, compute tight upper bound for WCET
- Bound-T – static analysis of machine code, WCET and stack usage
- OTAWA – Open Tool for Adaptive WCET Analysis, C++ framework, static analysis

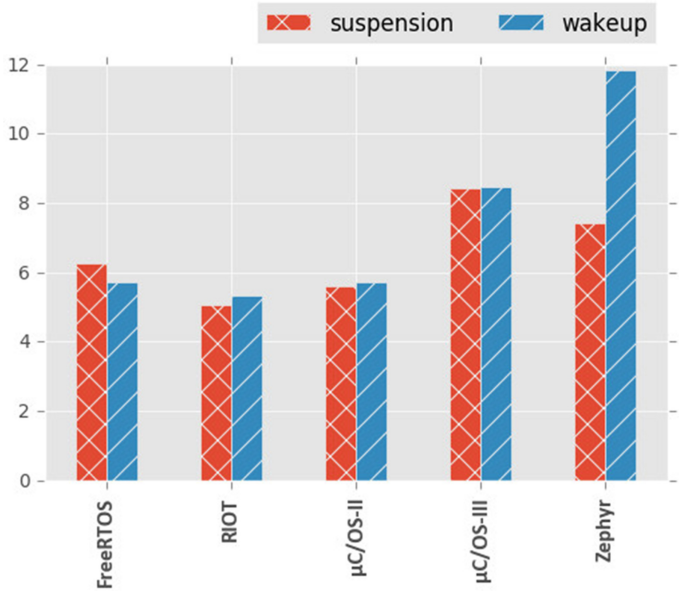
Benchmarks[2]

- Rhealstone – six independent verifiable RT performance measures (task sw & pree, int lat, sem shuff, dead break, inter-t mess lat)
- Thread-Metric – Eclipse Foundation (Azure RTOS → ThreadX, open source); coop & pree sched, mem alloc, synchr, mess pass
- other benchmarks: Dhrystone, Whestone, Hartstone...

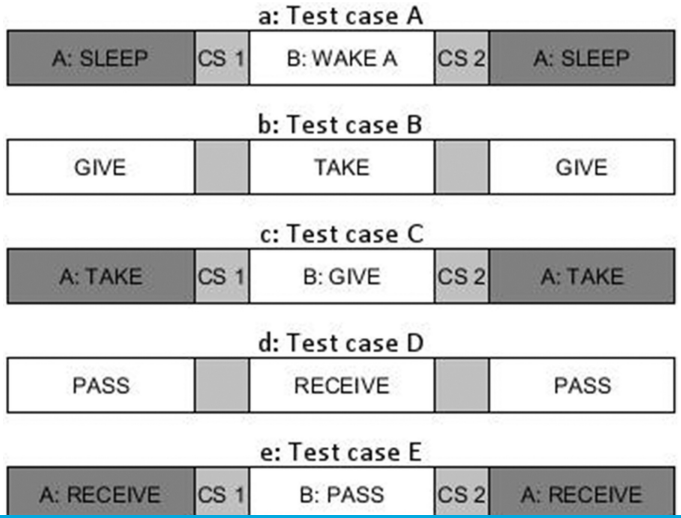
[2] Ján Lorenc: Porovnání vlastností a výkonnosti jader uC/OS-II a uC/OS-III, diplomová práce, Brno, FIT VUT v Brně, 2016

Benchmarks from literature I. a)B,b)C,c)sw+a)vs.b)

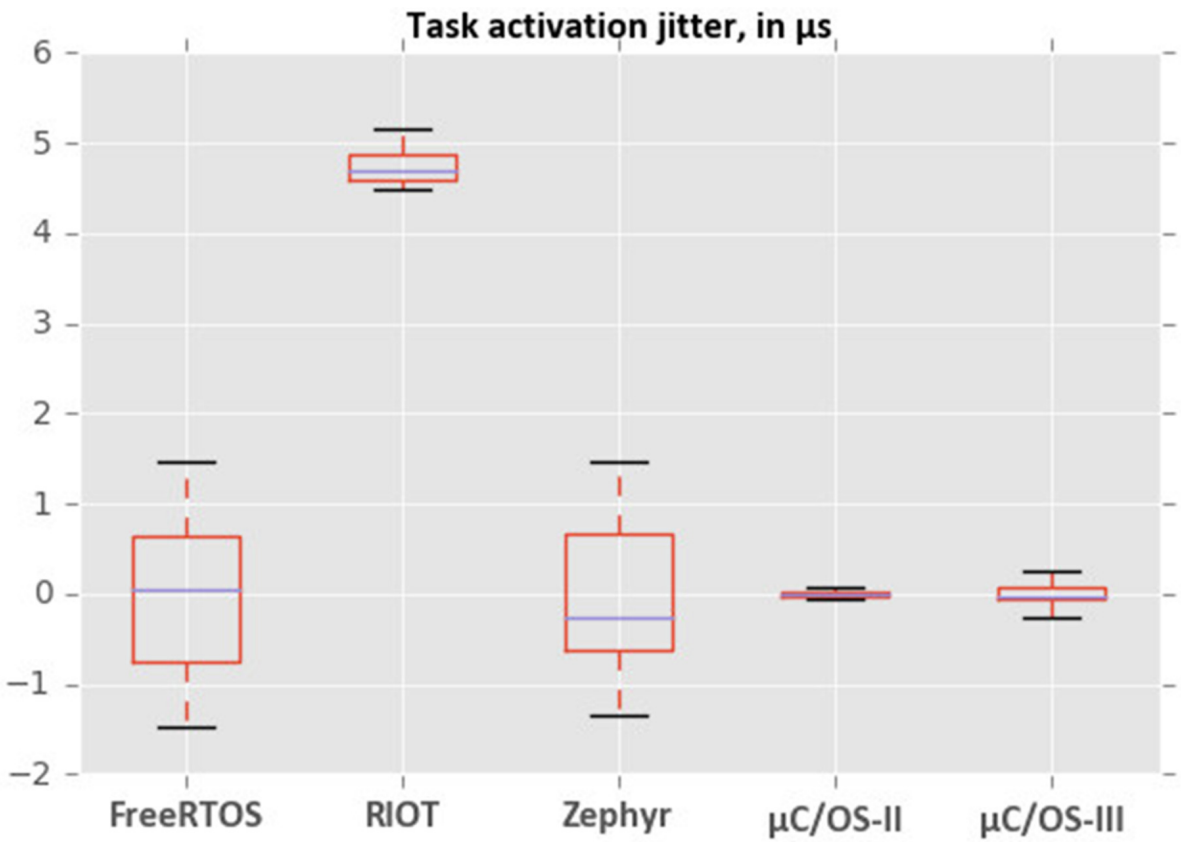




times in μs

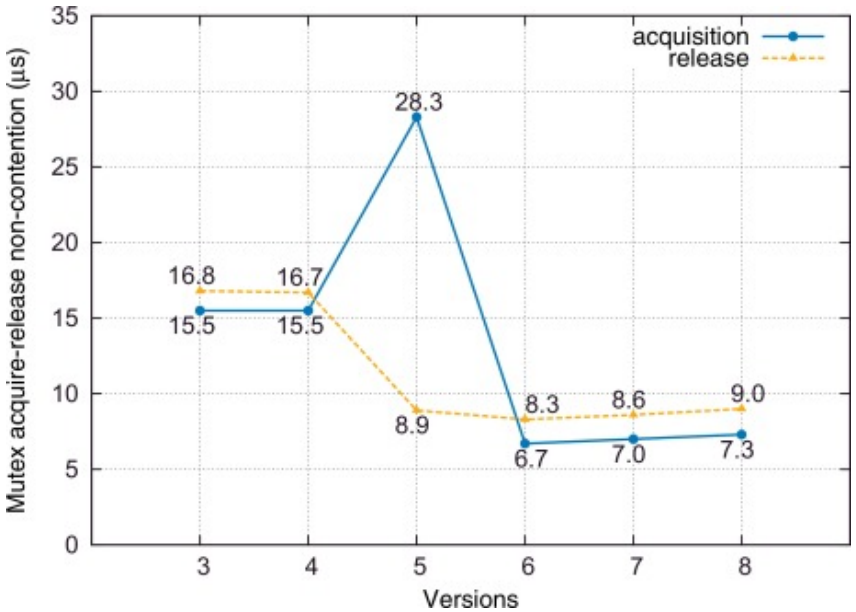


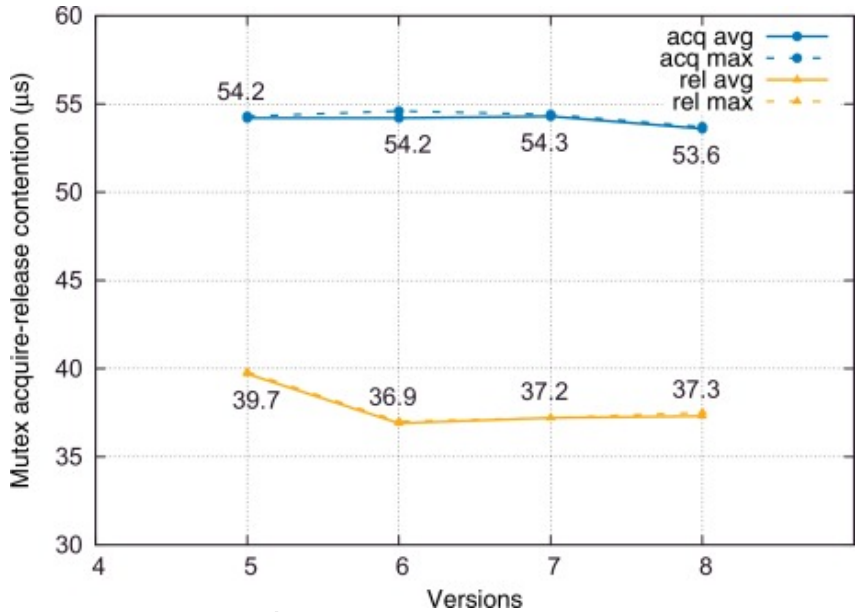
Benchmarks from literature II.



NXP/Freescale FRDM-K64F – Kinetis K64F, ARM Cortex-M4
FreeRTOS v9.0.0, uC/OS-II v2.92.11, uC/OS-III v3.05.11...
RAYMUNDO BELLEZA, Rafael; FREITAS PIGNATON, Edison de. Performance study of real-time operating systems for internet of things devices. IET Software. 2018, vol. 12, no. 3, pp. 176–182.

Benchmarks from literature III.





Non-contention – not owned by any other thread (likely in well-designed software), b) includes thread switch, a) v5 thread suspension&resume unavoidable + clock event yields twice to check ready lists
v3, v4 did not support priority inversion, thus excluded (med prio thread is activated between low-having & high-asking, prio inherit lo→hi)
Renesas RX63N based eval. board YLCDRX63N (32bit MCU 100MHz)

GUAN, Fei; PENG, Long; PERNEEL, Luc; TIMMERMAN, Martin. Open source FreeRTOS as a case study in real-time operating system evolution. Journal of Systems and Software. 2016, vol. 118, pp. 19–35. ISSN 0164-1212

Interesting topics from literature

Robustness benchmark

- fault injection – single & multiple bit flips, data type corruptions; Task Manager, Task Sync, Memory Manager, Timer module
- error model, 7 weighted levels $\rightarrow$ robustness as a quantifiable numerical value

Reducing analysis pessimism for WCET & WCRT ("response")

- system facts, additional system state knowledge parametric source-level annotation language, system aware analysis

Non-linearity, performance, bugs, methodologies

- μC/OS-II vs μC/OS-III – greater complexity $\rightarrow$ slower RT perf vs. advanced mechanisms and structures
- single metrics vs RTOS A "dominating" RTOS B
- possible bugs, long standing issues causing wrong behaviour
- growing complexity $\rightarrow$ slower/halted rate of enhancements

Interesting topics from literature

Robustness benchmark

- fault injection – single & multiple bit flips, data type corruptions; Task Manager, Task Sync, Memory Manager, Timer module
- error model, 7 weighted levels $\rightarrow$ robustness as a quantifiable numerical value

Reducing analysis pessimism for WCET & WCRT ("response")

- system facts, additional system state knowledge parametric source-level annotation language, system aware analysis

Non-linearity, performance, bugs, methodologies

- μC/OS-II vs μC/OS-III – greater complexity $\rightarrow$ slower RT perf vs. advanced mechanisms and structures
- single metrics vs RTOS A "dominating" RTOS B
- possible bugs, long standing issues causing wrong behaviour
- growing complexity $\rightarrow$ slower/halted rate of enhancements

Thank you for your attention