

Sistemas de Tempo Real

Disciplina de Sistemas Embarcados

Prof. Renato Sampaio

Linux em Tempo Real

- Conceitos Gerais
 - Escalonador
 - Gerenciamento de Interrupções
 - Latência
- Características de Tempo-real no Linux
 - RP-PREEMPT
- Extensões de Tempo-real: Xenomai

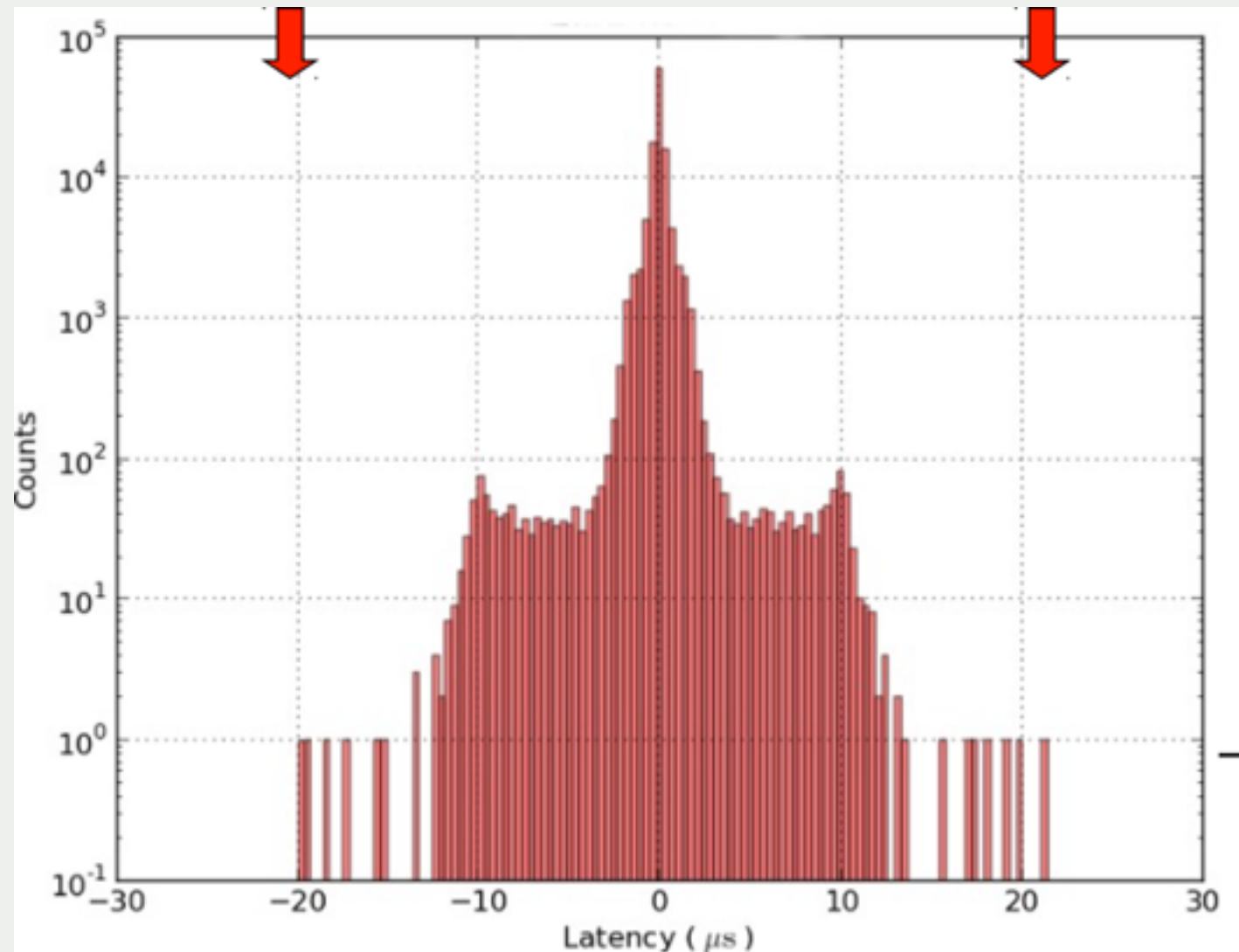
Dificuldades para Atingir Tempo-real

- O Linux é otimizado para **rendimento máximo** (*throughput*) e não **determinismo**.
 - CPUs que rodam Linux utilizam estruturas de memória com muitas camadas (níveis de cache)
 - O uso de Cache impede tarefas RT

Jitter

- O espalhamento do período de controle de todas as tarefas do sistema.
- **Total Jitter**: diferença entre os tempos de resposta **máximo** e **mínimo**.

Jitter



Linux em Tempo-Real

- Duas abordagens

1. RT-Preempt

- Permite preempção das tarefas do kernel para minimizar latências.
- Escalona todas as atividades (incluindo IRQ) como threads escalonáveis.

2. RT Extensions

- Adiciona uma outra camada entre o hardware e o kernel do Linux para tratar das tarefas RT.

Linux em Tempo-Real - Preempt

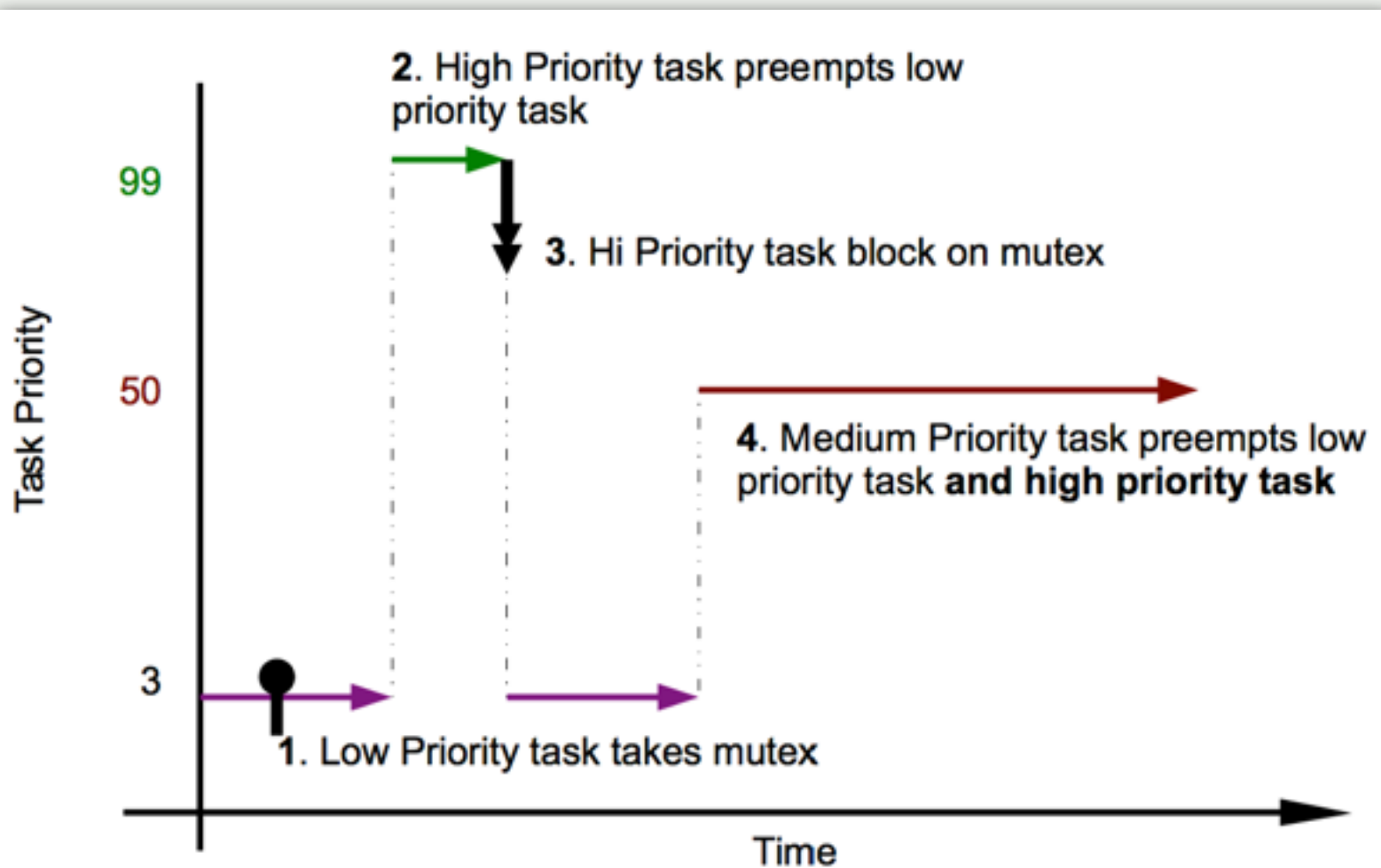
- Escalonador
 - O Kernel suporta vários tipos de escalonadores por extensão
 - SCHED_OTHER (padrão, não-RT)
 - **SCHED_FIFO** (Usa FIFO RT)
 - **SCHED_RR** (Usa RR RT)
 - SCHED_IDLE
 - SCHED_BATCH
 - SCHED_DEADLINE

Linux em Tempo-Real - **Preempt**

- Problemas no Escalonador
 - System calls: no guaranteed response time.
 - Device drivers: System calls handlers or interrupt handlers usually not implemented with care for real-time requirements. Expect uncertainties unless you implemented all kernel support code by yourself.

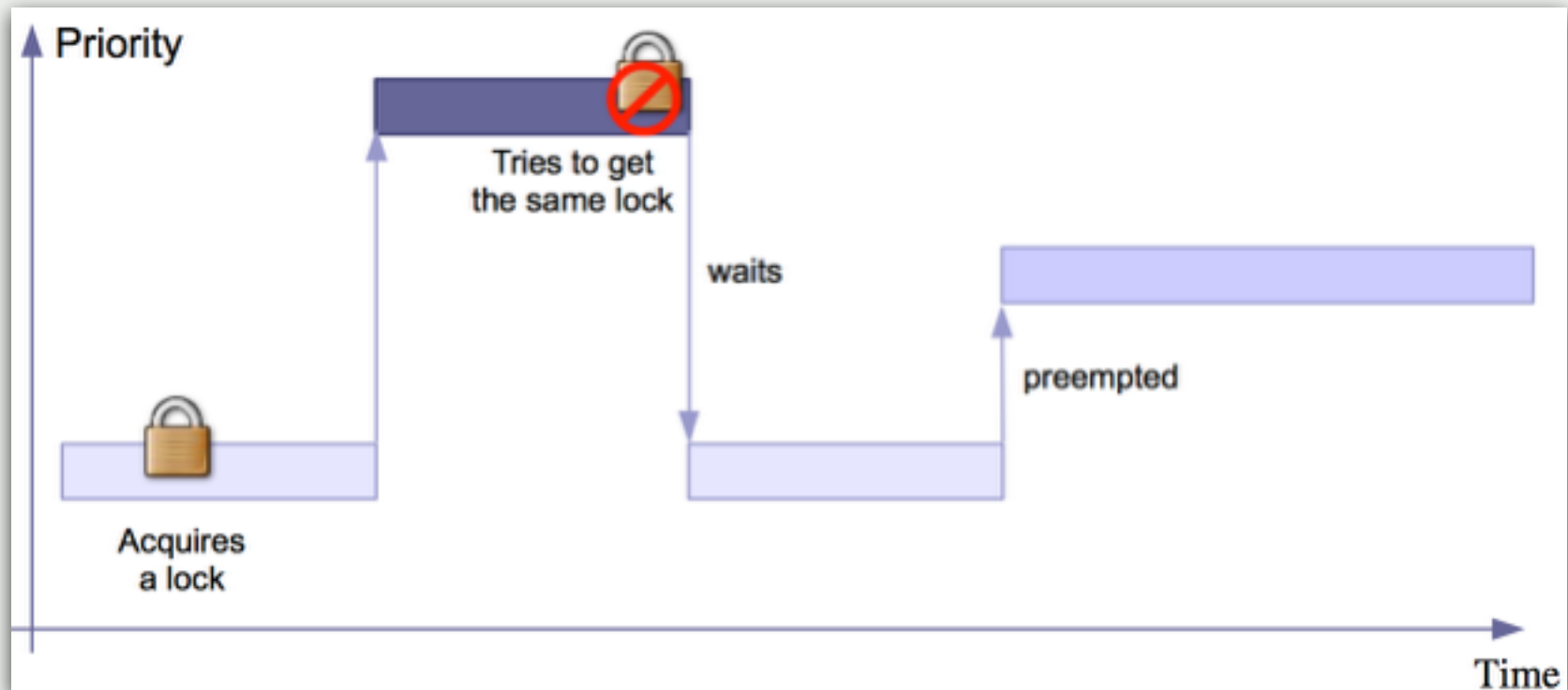
Linux em Tempo-Real - Preempt

- Problemas - Inversão de Prioridades com Mutex



Linux em Tempo-Real - Preempt

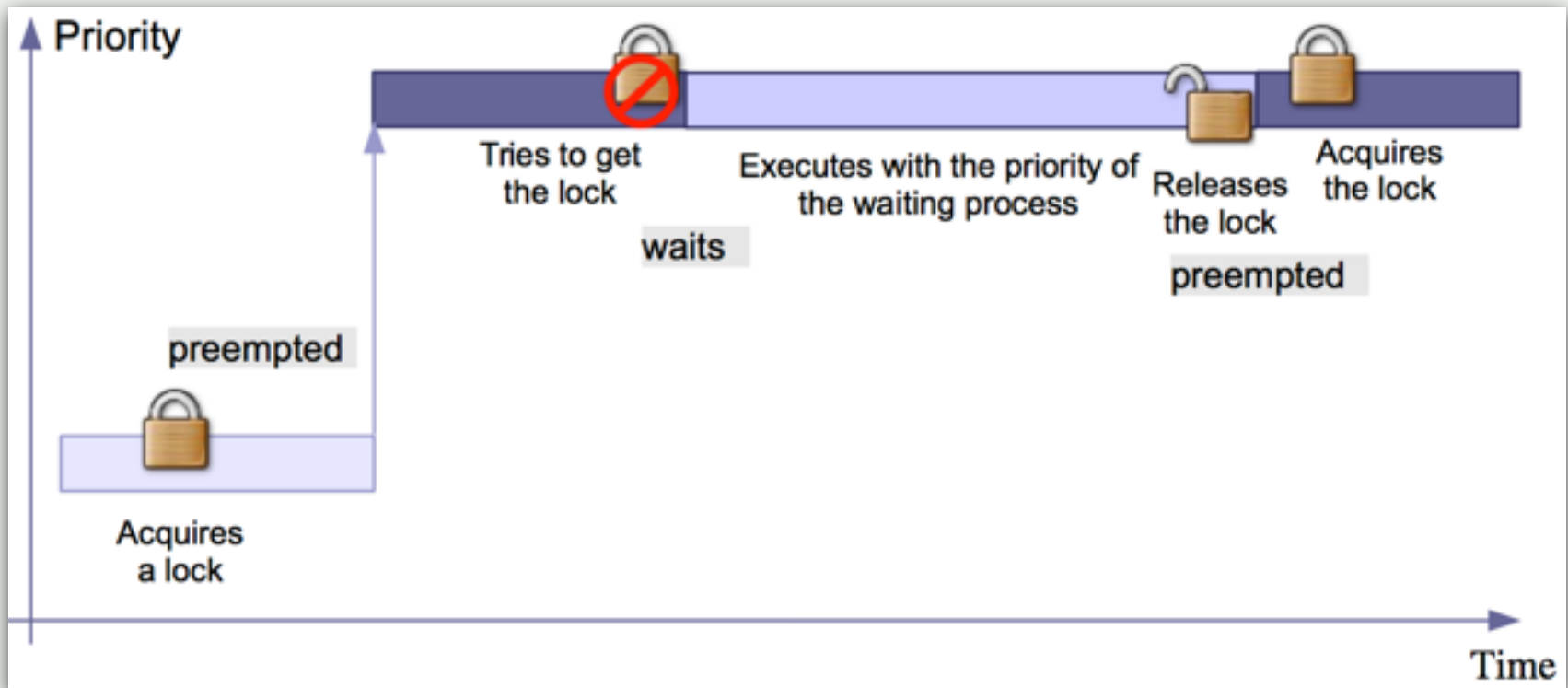
- Problemas - Inversão de Prioridades com Mutex



- Problema: um processo de maior prioridade pode tirar um outro processo que esteja segurando o lock (mutex).

Linux em Tempo-Real - Preempt

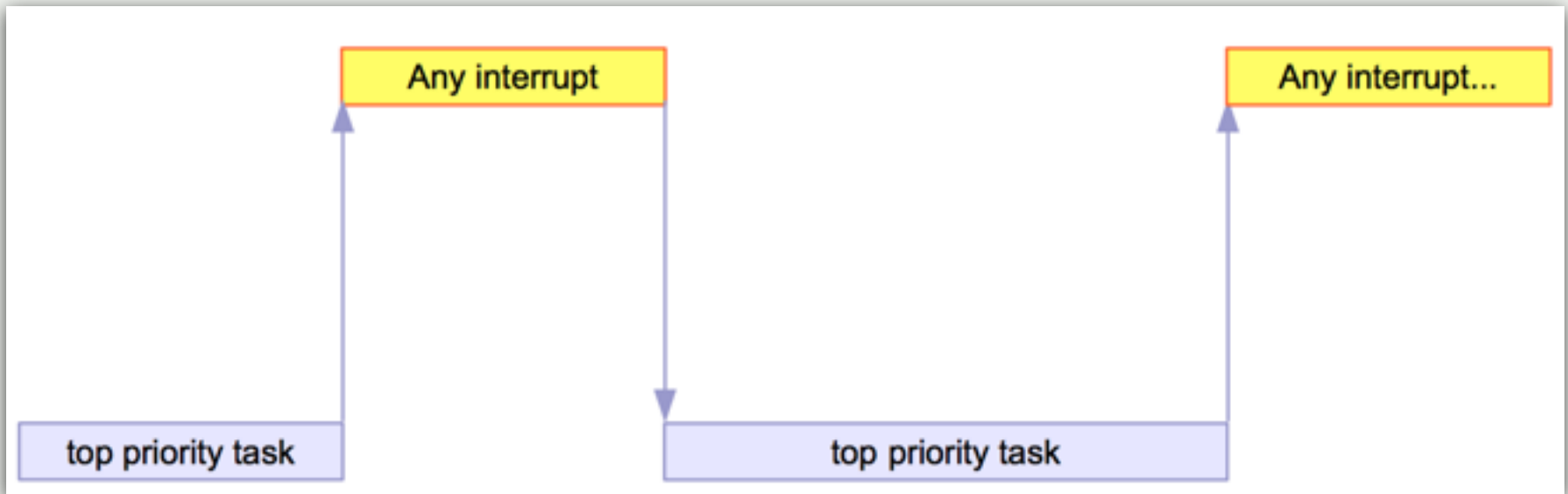
- Problemas - Inversão de Prioridades com Mutex



- Solução:** herança de prioridade. O processo que segura o lock (mutex) ganha alta prioridade.

Linux em Tempo-Real - Preempt

- Problemas - Inversão de Prioridades de Interrupções



- **Problema:** Até a tarefa de mais alta prioridade pode ser interrompida pelo *interrupt handler*.
- **Solução:** interrupção em threads.

POSIX 1003.1b - Extensões RT

- Prioridades no Escalonador
- Sinais de RT
- Clocks e Timers
- Semáforos
- Passagem de Mensagens
- Memória Compartilhada
- E/S Síncrono e Assíncrono
- Locking de Memória

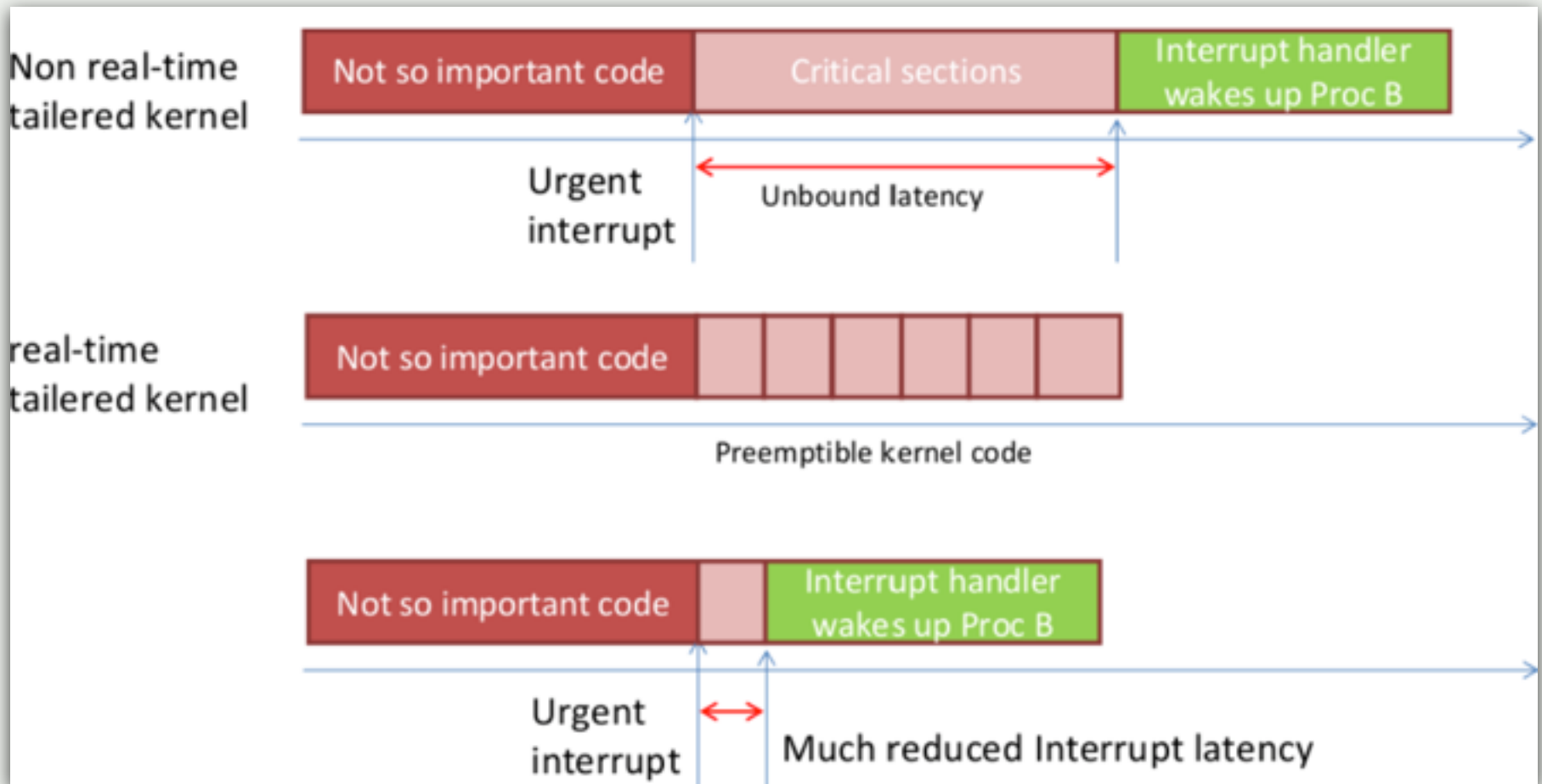
Melhorias do Kernel Linux

- A partir do Kernel 2.6
 - Interrupções em Threads: interrupções podem ser executadas por uma thread do kernel com prioridades
 - RT-Mutex: herança de prioridades
 - BKL-free: Kernel pode ser compilado sem o Big Kernel Lock
 - Aumento de determinismo com a API POSIX RT
 - Chamadas de sistema (System Calls) ainda não preemptíveis por padrão mas extensíveis pelo `CONFIG_PREEMPT` e `CONFIG_PREEMPT-RT`
 - Timers de Alta resolução - *nanosleep()* - *1us*

Kernel RT-Preempt

- Objetivo: Tornar o sistema mais determinístico
 - Kernel preemptivo porém ainda sem garantia total de latência (alocação de memória) —> Usar mlock
 - Drivers (USB, PCI, etc) não implementados em RT.
 - IRQ em Threads
 - Frequência do Timer (100Hz -> 1000Hz) - Resolução de 10ms -> 1ms.
 - Timer de alta resolução (1us)

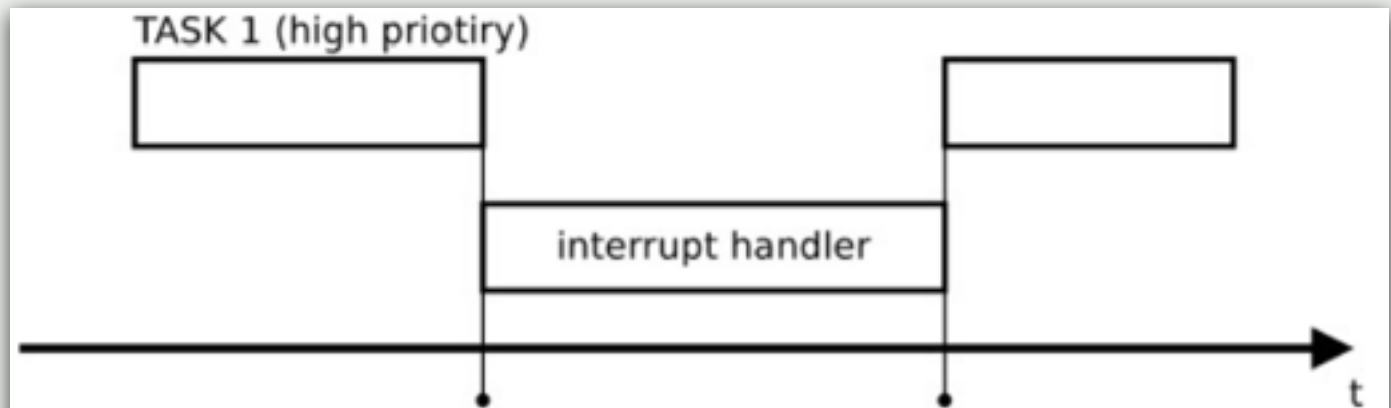
Kernel RT-Preempt



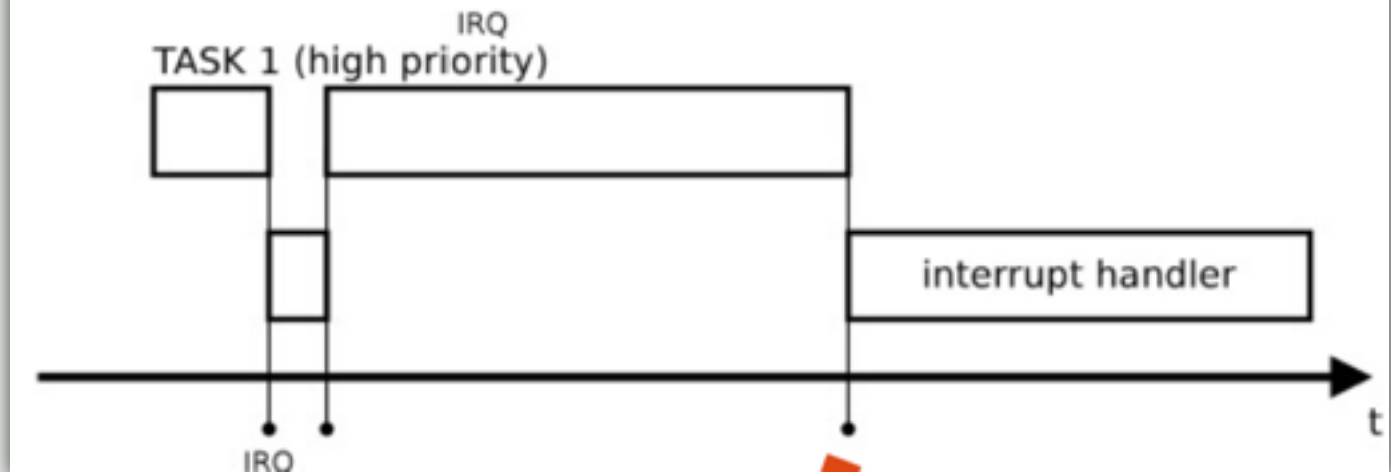
Kernel RT-Preempt

- IRQ em Threads

Kernel Padrão



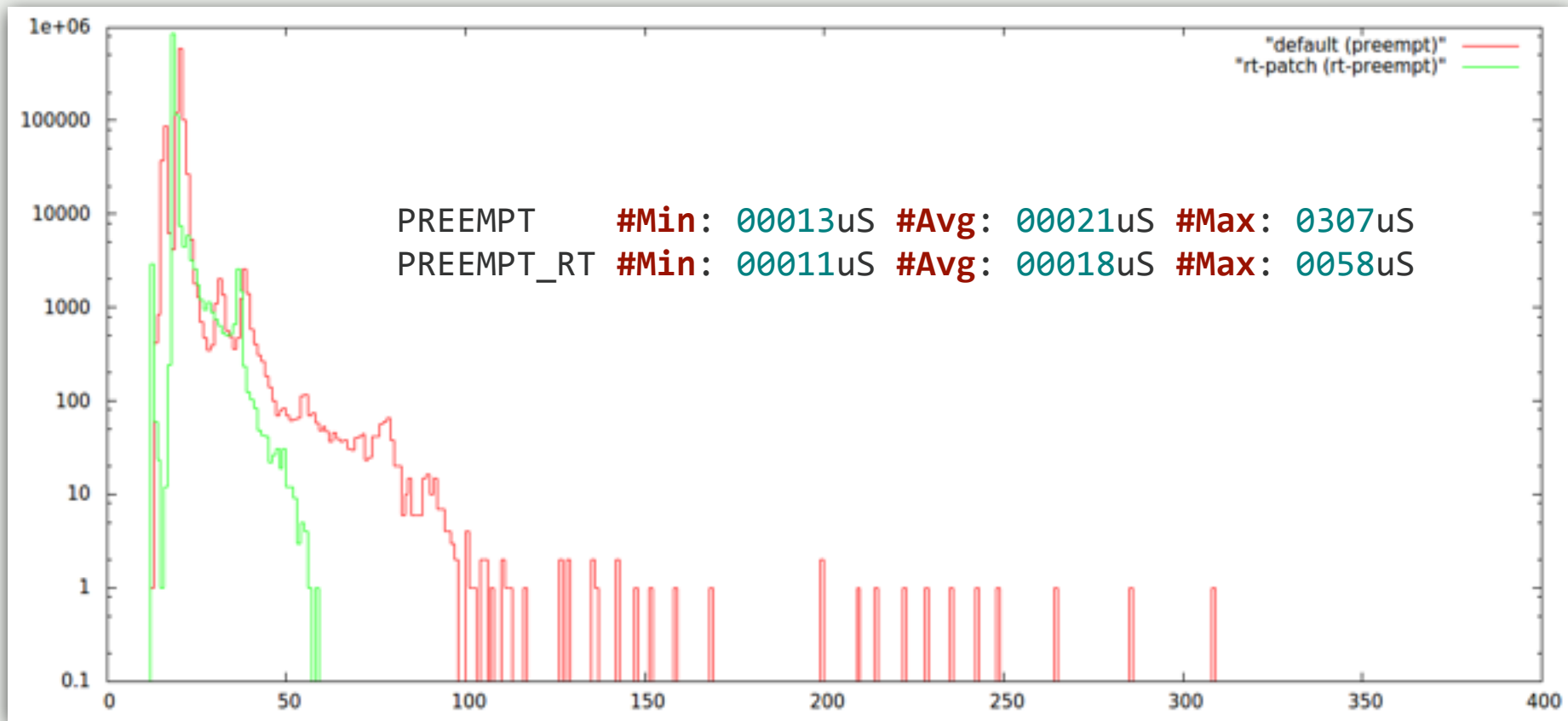
Kernel PREEMPT-RT



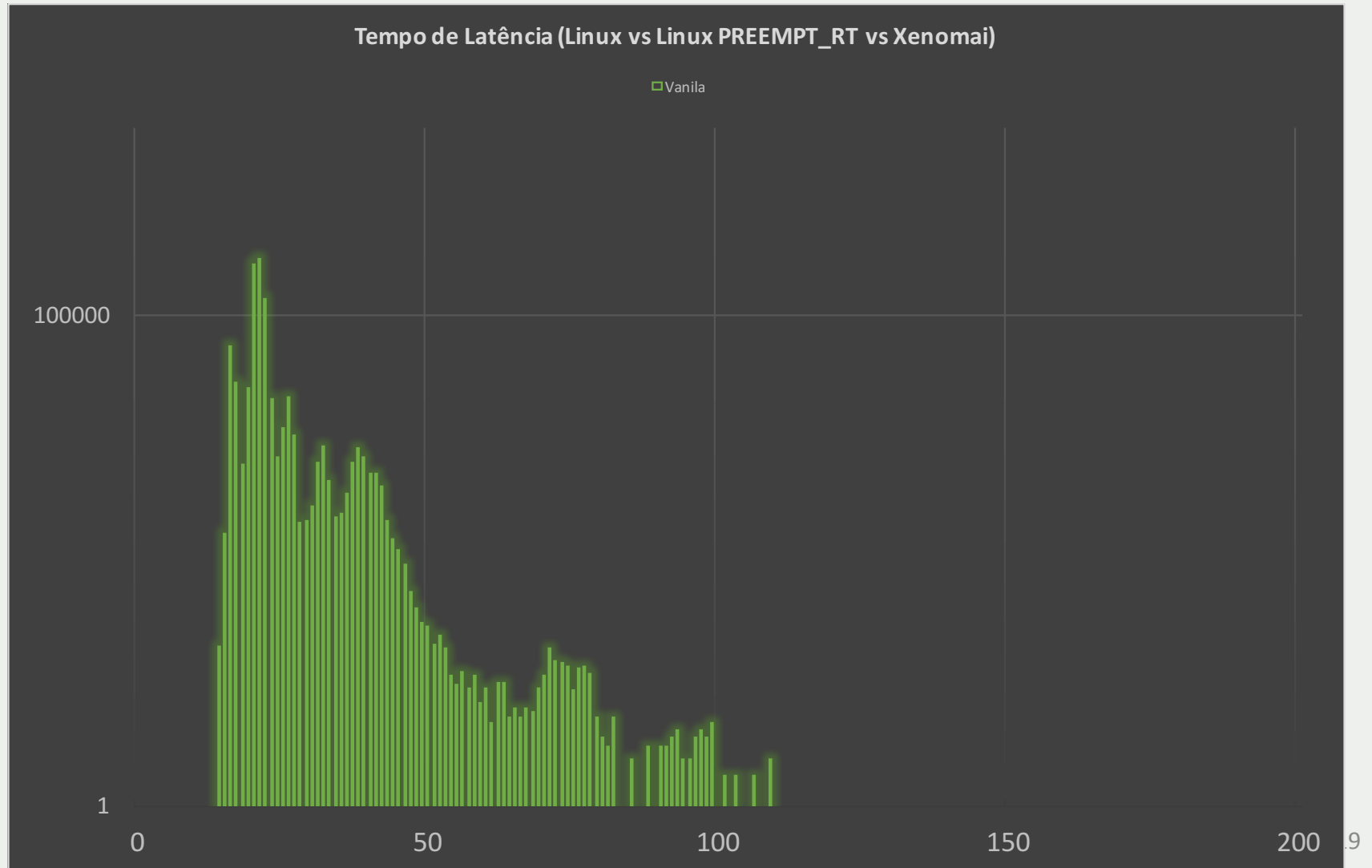
Comparação

- Kernel RT disponível em:

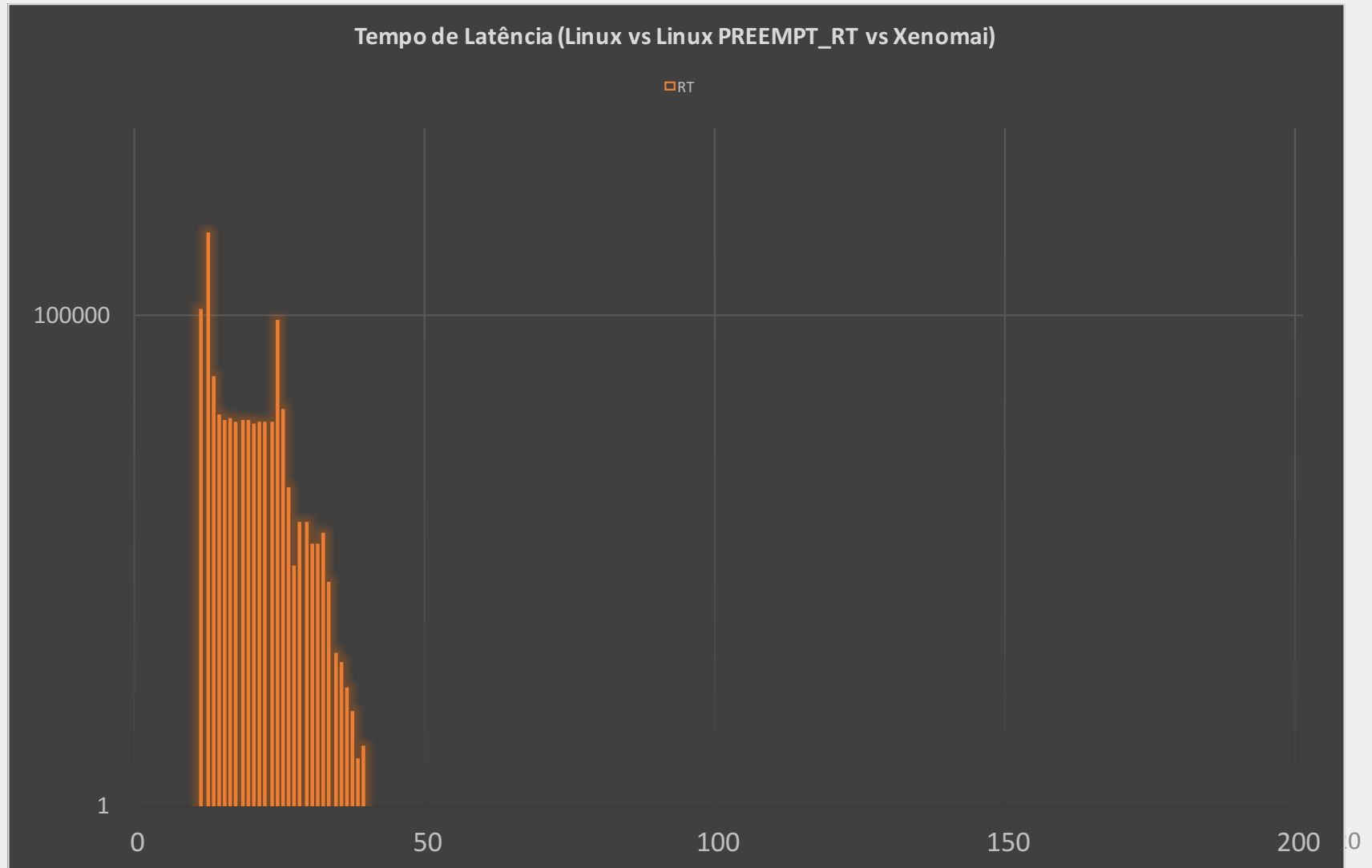
<https://docs.emlid.com/navio/Downloads/Real-time-Linux-RPi2/>



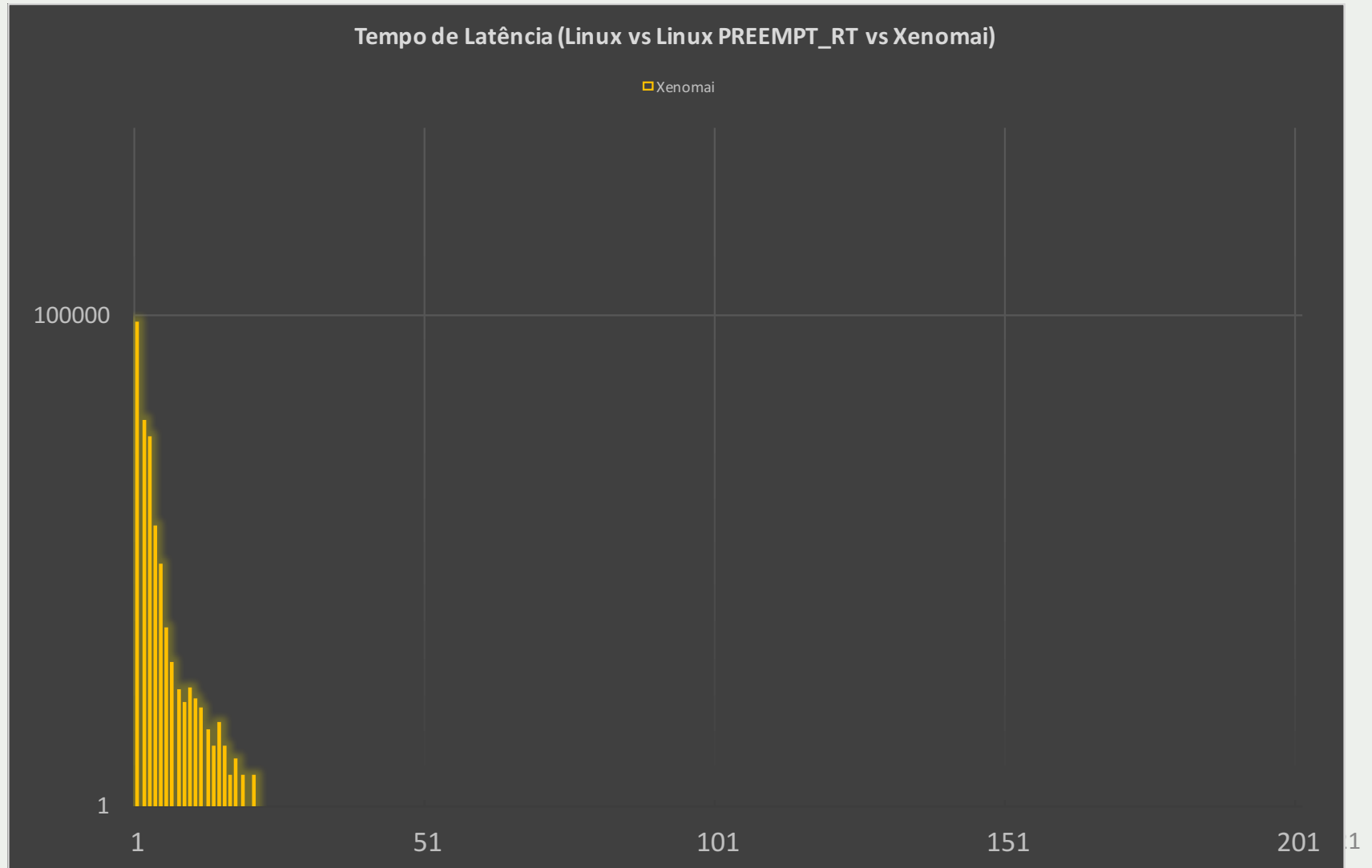
Linux



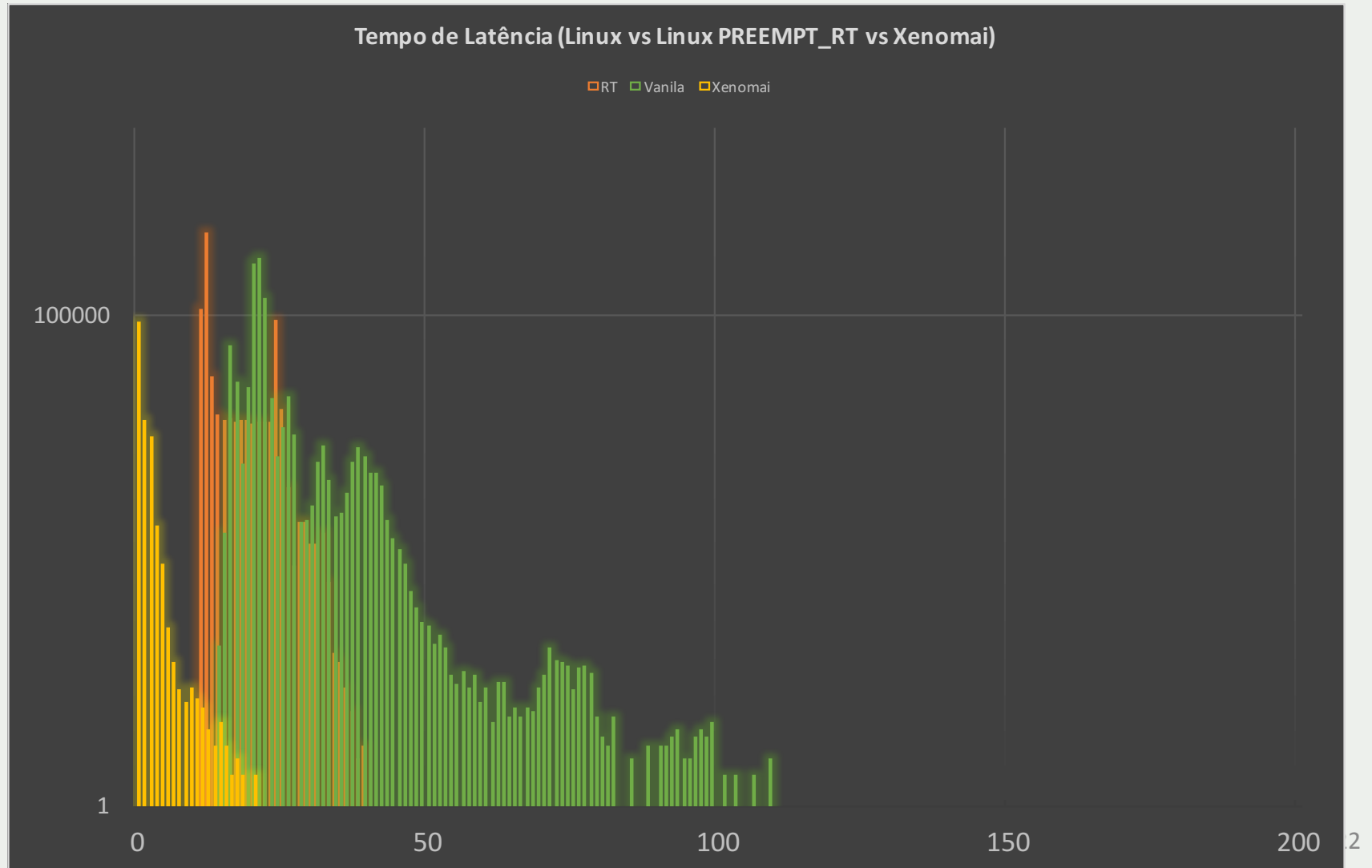
Linux Preempt-RT



Linux Preempt-RT



Linux vs Preempt-RT vs Xenomai



Bibliografia

- Kopetz, Hermann; Real-time Systems: Design Principles for Distributed Embedded Applications, 2nd Ed. Springer, 2011.