|  |  |
| --- | --- |
|  | UNIVERSIDADE FEDERAL DO AMAZONAS  FACULDADE DE TECNOLOGIA  DEPARTAMENTO DE ELETRÔNICA E COMPUTAÇÃO  PROGRAMAÇÃO DE SISTEMAS DE TEMPO REAL  PROF. ANDRÉ CAVALCANTE  SEMESTRE 2024/2 |

**LABORATÓRIO 5 - Compilação de um RTOS**

**Aluno: Matheus Carvalho Reges – 22152027**

**1. Descrição dos Objetivos**

Instalar e configurar um kernel Linux com suporte a tempo real (RTOS) usando o patch **PREEMPT\_RT**, a fim de habilitar funcionalidades de computação em tempo real para simulações e aplicações que demandam baixa latência e alta previsibilidade.

**2. Descrição do Problema Proposto**

A versão padrão do kernel Linux não é projetada para tempo real, apresentando limitações em situações que exigem predição determinística. O objetivo é modificar o kernel padrão do Ubuntu para um kernel de tempo real, garantindo maior controle sobre preempção e tempos de resposta. O desafio inclui aplicar corretamente o patch PREEMPT\_RT, configurar o kernel para atender às demandas do hardware e realizar testes de desempenho.

**3. Descrição dos Procedimentos Realizados**

* 1. **Criação do Ambiente de Trabalho**

1. Criação de um diretório para o kernel:

* mkdir ~/kernel
* cd ~/kernel

**3.2 Download dos Arquivos Necessários**

1. Obtenção dos códigos fonte do kernel e do patch PREEMPT\_RT:

* wget https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.15.170.tar.gz
* wget <https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/5.15/patch-5.15.170-rt81.patch.xz>

1. Extração dos arquivos:

* tar -xzf linux-5.15.170.tar.gz
* xz -d patch-5.15.170-rt81.patch.xz
* cd linux-5.15.170
* patch -p1 < ../patch-5.15.170-rt81.patch

**3.3 Configuração do Kernel**

1. Copiar a configuração padrão do kernel:

* cp /boot/config-6.8.0-48-generic .config

1. Atualizar e instalar dependências:

* sudo apt update
* sudo apt install make gcc libncurses-dev libssl-dev flex libelf-dev bison

1. Configurar as opções do kernel:

* make menuconfig
* Habilitar a opção **Fully Preemptible Kernel (Real-Time)** em "General setup > Preemption Model".

**3.4 Compilação e Instalação do Kernel**

1. Compilar o kernel:

* sudo make

1. Instalar os módulos do kernel:

* sudo make modules\_install

1. Instalar o kernel:

* sudo make install

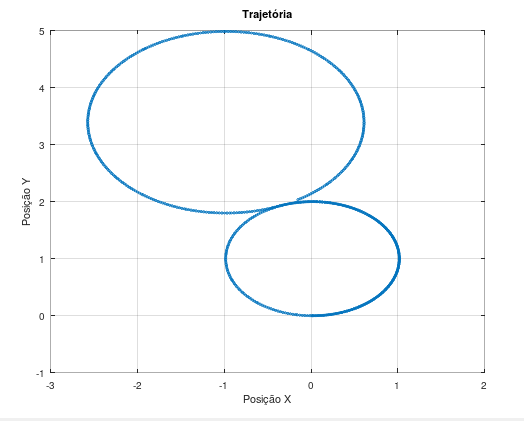
**3.5 Reinicialização e Seleção do Kernel**

 Reiniciar o sistema e selecionar o kernel com suporte a tempo real no **GRUB**.

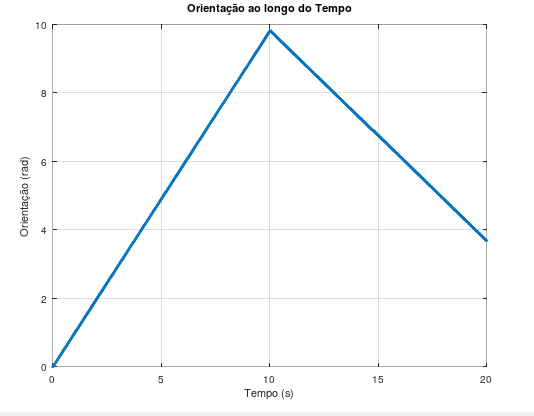
Durante esta etapa, foi encontrado o erro: "bad shim signature - you need to load the kernel first". Para solucionar, foi necessário acessar a BIOS da placa-mãe e desativar a opção de Secure Boot, permitindo assim o carregamento do kernel.

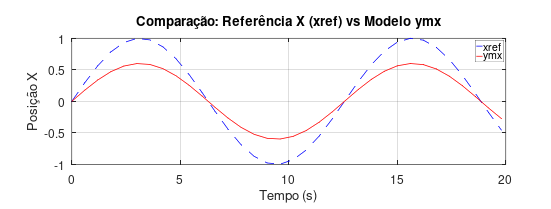
**4.Saídas dos Programas de Testes**

Para os resultados a seguir foi usado alpha1 e alpha2 igual a 3.00

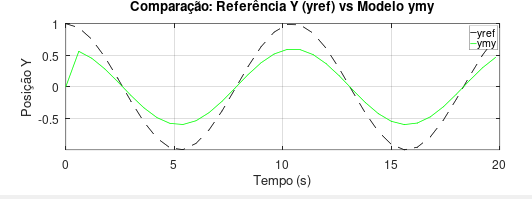
**4.1 Grafico y(t)  
**

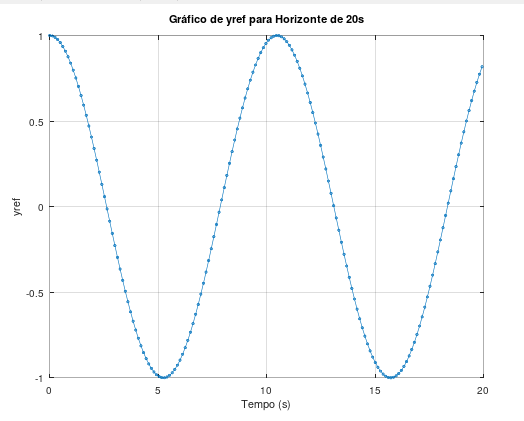
**4.2 Orientação ao longo do tempo**

**  
4.3 xref x ymx**

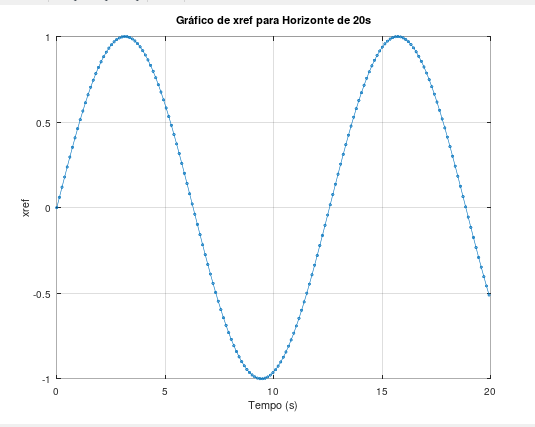
****

**4.4 yref x ymy**

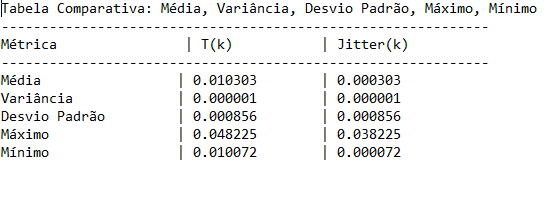
****

**4.5 yref **

**4.6 xref**

****

**4.7 Tabela com Tempo(k) e jitter(k)**

****

**5. Conclusão**

Durante o processo de configuração, foi identificado que os links fornecidos no tutorial original para os códigos fonte do kernel e do patch PREEMPT\_RT eram inválidos, impossibilitando o download. Foi necessário ajustar os links para as versões mais recentes disponíveis, substituindo:

* wget <https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.15.96.tar.gz> por wget https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.15.170.tar.gz
* wget https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/5.15/patch-5.15.96-rt61.patch.xz por wget https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/5.15/patch-5.15.170-rt81.patch.xz

Essa modificação foi essencial para o sucesso da compilação e aplicação do patch.

Adicionalmente, durante a configuração do kernel, foi necessário substituir o comando:

* cp /boot/config-5.15.0-43-generic .config por
* cp /boot/config-6.8.0-48-generic .config

devido à mudança de versão do kernel. Essa adaptação foi essencial para evitar inconsistências na configuração do kernel.

Além disso, durante a etapa de extração dos arquivos, foi necessário adaptar os comandos devido às alterações nos nomes dos arquivos. Os comandos originais:

* tar -xzf linux-5.15.96.tar.gz
* xz -d patch-5.15.96-rt61.patch.xz
* cd linux-5.15.96
* patch -p1 < ../patch-5.15.96-rt61.patch

foram substituídos por:

* tar -xzf linux-5.15.170.tar.gz
* xz -d patch-5.15.170-rt81.patch.xz
* cd linux-5.15.170
* patch -p1 < ../patch-5.15.170-rt81.patch

Finalmente, ao reiniciar e selecionar o kernel, foi encontrado o erro "bad shim signature - you need to load the kernel first". A solução foi desativar o Secure Boot na BIOS da placa-mãe, permitindo o carregamento correto do kernel.

Essas alterações foram necessárias para refletir os arquivos disponíveis e garantir a continuidade do processo.

**6. Referências**

1. Tutorial "Building a Real-time Linux Kernel in Ubuntu with PREEMPT\_RT": <https://www.acontis.com/en/building-a-real-time-linux-kernel-in-ubuntu-preemptrt.html>.
2. Documentação oficial do projeto Real Time Linux: https://wiki.linuxfoundation.org/realtime/start
3. Documentação do Ubuntu sobre configuração do kernel: <https://ubuntu.com>