

Relatório Técnico do Projeto Raptor

1. Introdução

O *Zephyr OS*® é um sistema operacional de tempo real baseado em um kernel de pequeno porte, amplamente utilizado em aplicações de sistemas embarcados com recursos limitados e suporte a diversas arquiteturas. Apesar de pequeno, ele oferece um conjunto abrangente de serviços do kernel, que incluem desde gerenciamento de energia até algoritmos de escalonamento múltiplo, possibilitando a multiprogramação.

Neste projeto, foi desenvolvido para Raptor Air, uma solução em ambiente de simulação usando o QEMU®, para exibir a mensagem *"Hello World! This is Raptor"* simulando o funcionamento de um hardware em ambiente de virtualização.

3. Implementação

3.1 Estrutura do Projeto

O *Zephyr OS*® fornece, em sua instalação, uma série de bibliotecas e APIs que facilitam significativamente seu uso no desenvolvimento de aplicações. Além disso, conta com uma documentação completa para consulta e utilização de suas estruturas iniciais. Dessa forma, o projeto foi estruturado de acordo com a *topologia 2*, seguindo o exemplo de aplicação fornecido no repositório do *GitHub* do Zephyr, conforme ilustrado na *Imagem 1* abaixo.

```
west-workspace/
├── application/          # .git/
│   ├── CMakeLists.txt
│   ├── prj.conf          never modified by west
│   ├── src/
│   │   └── main.c
│   └── west.yml          # main manifest with optional import(s) and override(s)
├── modules/
│   └── lib/
│       └── zcbor/        # .git/ project from either the main manifest or some import.
└── zephyr/              # .git/ project
    └── west.yml          # This can be partially imported with lower precedence or ignored.
                        # Only the 'manifest-rev' version can be imported.
```

Imagem-1: Estrutura de diretórios do Projeto.

O desenvolvimento do projeto foi estruturado com base no *Git Flow*, um modelo de ramificação para o Git que organiza o fluxo de trabalho a fim de facilitar seu acompanhamento, tornando cada etapa do projeto mais visível para quem desejar compreendê-lo.

3.3 Features Desenvolvidas

1. Feature 1: Hello-World

- Implementa exibição da mensagem “Hello World! This is Raptor”, em ambiente de virtualização, fornecido pelo QEMU.

2. Feature 2: Programar-e-Bacana

- Implementa exibição da mensagem “Programar eh Bacana!”, em ambiente de virtualização, fornecido pelo QEMU.

3. Feature 3: Programar-em-C

- Implementa exibição da mensagem “Programar em C eh muito mais bacana!!!”, em ambiente de virtualização, fornecido pelo QEMU.

4. Desafios

O grande desafio do projeto foi lidar uma tecnologia nova, entender sua documentação, estruturação, montar o projeto no meu ambiente de desenvolvimento e lidar com algumas falhas que ocorreram durante esse processo.

Foram duas as maiores falhas durante a clonagem do projeto no meu ambiente e montagem, onde o *west*, a ferramenta utilizada pelo *Zephyr* para facilitar a configuração e gerenciamento dos ambientes, não estava conseguindo concluir com sucesso o download do projeto completo ocasionando a não inicialização do *zephyr*. Devido a esse problema, algumas depêndencias ficaram faltando no momento de realizar a instalação da *Zephyr SDK*, sendo necessário realizar a instalação manual dessas mesmas dependências.

Outro problema ocorrido foi ao ter conseguido realizar a primeira execução do projeto corretamente, não ter finalizado o ambiente do QEMU corretamente, impossibilitando sua execução para novos testes. Foi resolvido apenas reiniciando o ambiente.

O ultimo desafio e que levou mais tempo para ser solucionado, foi o projeto não poder ser submetido ao meu repositório remoto, devido a ele pertencer à um outro. Não foi possível submeter os arquivos da implementação nova do projeto. Este problema foi solucinado removendo os arquivos de configurações de *git* do diretório do projeto exemplo, evitando a necessidade de adicionar o projeto exemplo como

submódulo, simplificando assim a sua submissão, possibilitando a visualização todas as alterações no histórico de desenvolvimento.

5. Instruções para execução do projeto

Dependências

Para executar o projeto, primeiramente você ter instalado em instalado as ferramentas na versão mínima do Cmake 2.20.5, Python 3.10 e Devicetree Compiler 1 .4.6.

Deve ter instalado também gerenciador de ambiente virtuais *venv* e também o *west* e *Zephyr SDK*.

As intruções para instalação dessas dependências podem ser encotradas na documentação do *Zephyr*.

https://docs.zephyrproject.org/latest/develop/getting_started/index.html

Com todas as dependencias instaladas, basta seguir os passos abaixo:

1. Clone o repositório <https://github.com/renato5alves/Raptor-Zephyr>
2. Acesse diretório *example-application*
 - *cd raptor-workspace/example-application*
3. Monte o ambiente virtual QEMU e compile o projeto utilizando o comando:
 - *west build -b qemu_x86 app*
4. Execute o projeto
 - *west build -t run*

7. Referências

<https://docs.zephyrproject.org/latest/>