

**Professor:** Nome do Professor, [professor@ifsc.edu.br](mailto:professor@ifsc.edu.br)

**Estudante:**

### Exercício 1: Fundamentos de SOs Embarcados

#### Tópicos:

- Visão geral de SOs embarcados (arquiteturas: monolíticos, microkernel, exokernel)
- Gerenciamento de recursos (CPU, memória, I/O) e proteção
- Processos e threads em sistemas embarcados (contexto, estados, criação/troca)
- Exemplo: Linux embarcado (inicialização, modularidade)

#### Destaque:

- Comparação entre arquiteturas de SO e desafios em sistemas com recursos limitados

### Exercício 2: Escalonamento em Sistemas Embarcados

#### Tópicos:

- Políticas de escalonamento (FCFS, SJF, Round Robin, prioridades)
- Critérios de desempenho (throughput, latência, previsibilidade)
- Escalonamento preemptivo vs. não preemptivo
- Aplicação em sistemas embarcados (trade-offs entre eficiência e determinismo)

#### Destaque:

- Análise de casos com multiprogramação e impacto no uso da CPU

### Exercício 3: Sistemas de Tempo Real (STR)

#### Tópicos:

- Conceitos de STR (hard, soft, no real-time) e métricas (deadlines, laxity)
- Previsibilidade e dificuldades (cache, DMA, interrupções)
- Modelos de tarefas (periódicas, esporádicas, precedência)
- Testes de escalabilidade (exatos, suficientes, necessários)

**Destaque:**

- Discussão sobre inversão de prioridade e bloqueio

**Exercício 4: Sistemas de Tempo Real na Prática****Tópicos:**

- Comparativo entre SOs RTOS:
  - FreeRTOS: Leveza, portabilidade, uso em microcontroladores
  - Zephyr: Suporte a arquiteturas heterogêneas, modularidade
  - RT-Linux: Extensões de tempo real para Linux
  - VxWorks: Usado em missões críticas (aeroespacial, industrial)
- Casos reais: Automação industrial, robótica, dispositivos médicos

**Destaque:**

- Análise de como cada SO aborda previsibilidade e escalonamento (prioridades dinâmicas/estáticas)