

# Trabalho Final

MC404 - 2023 - 2s (Prof. Edson Borin)

## Objetivo

O objetivo deste trabalho é avaliar os conceitos apresentados no decorrer da disciplina.

## Enunciado

Para realizar este trabalho, você deve:

- Revisar os conceitos no livro [An Introduction to Assembly Programming with RISC-V](#);
- Verificar o manual de periféricos do ALE: [ALE Peripherals MMIO Manual - v1.0](#)
- Realizar o projeto final "FP 1: Driving Through the City", descrito no livro de exercícios do simulador ALE: [2023-ALE-Exercices-Book \(public\)](#)

## Entrega e Avaliação

A avaliação deste trabalho ocorrerá em duas partes, descritas abaixo. Apesar de haver testadores automáticos, a avaliação final e **atribuição de notas será feita manualmente pelos monitores** e poderá considerar outros aspectos, como, por exemplo, a consistência de seus resultados. **Os trabalhos serão corrigidos nas máquinas da sala 300 e a nota será atribuída em função do desempenho do código nestes sistemas computacionais. Não submeta seu trabalho sem testar nos computadores da sala 300.**

- Parte 1 (60%): Teste unitário das funções da API de controle e Syscalls.
  - Um código de testes será compilado e ligado às suas implementações das subcamadas ACOS e CoLib.
  - Assistentes:
    - Teste para as syscalls:
      - Assistente: [Link](#)
    - Teste para a AP de controle:
      - As operações podem ser vistas [neste arquivo](#).
      - Assistente: [Link](#)
  - O resultado dos assistentes é um bom indicativo da corretude de seu código, entretanto:
    - Alguns testes têm entradas aleatórias. Na correção, executaremos seu código com os mesmos assistentes 3 vezes e calcularemos a média das execuções.
    - Nem todas as funções são testadas por eles. As demais funções serão testadas em testes fechados durante a correção e corresponderão a aproximadamente 30% da nota desta parte do trabalho.
- Parte 2 (40%): Teste na pista.
  - Nós fornecemos uma lógica de controle básica que *deve* funcionar se suas funções apresentarem desempenho razoável. Você pode alterá-la livremente

e cabe a você fazer o carro percorrer o trajeto corretamente (do ponto onde ele está inicialmente até o próximo cruzamento).

- O código para a camada de controle está disponível [neste arquivo](#).
  - Note que nós alteramos ligeiramente a assinatura de algumas funções. Isso não deve prejudicar ou requerer quaisquer modificações em sua implementação.
  - Note também que nem todas as funções foram utilizadas, mas você pode utilizar as demais conforme for necessário. Em particular, as funções `puts` e `atoi` podem ser bastante úteis para depuração.
- Os assistentes para o teste em pista são [Rota A](#), [Rota B](#), e [Rota C](#). Sua solução será considerada correta se percorrer os trajetos imprimindo os logs corretamente.
- Os trajetos podem ser visualizados nos vídeos [Rota A](#), [Rota B](#) e [Rota C](#).
- Sua solução deve ser consistente: conseguir percorrer o trajeto corretamente uma única vez não é suficiente para garantir a nota. Sua solução deve funcionar na maioria das vezes (durante a correção manual, seu código terá apenas duas chances para fazer o percurso corretamente).

Você deve submeter no Moodle os códigos das três partes implementadas (ACOS, CoLib e CoLo).

#### Prazos de Entrega:

- Prazo 1 (valendo 100% da nota):
  - **Até 22/11 às 23h59**
- Prazo 2 (Fator multiplicativo: 0.8):
  - **Até 24/11 às 23h59**
  - **Em função do prazo extra, talvez não seja possível divulgar a nota antes da semana de estudos. Neste caso, a depender da correção, a nota pode ser divulgada até 1 dia antes do exame.**
- Prazo 3 (Fator multiplicativo: 0.6):
  - **Até 26/11 às 23h59**
  - **Em função do prazo extra, talvez não seja possível divulgar a nota antes da semana de estudos. Neste caso, a depender da correção, a nota pode ser divulgada até 1 dia antes do exame.**
- OBS: Trabalhos entregues após o dia 26/11 não serão aceitos.

#### Observações:

- **Teste seu trabalho nas máquinas da sala 300, estas serão utilizadas para a correção do trabalho e a nota será atribuída em função do desempenho do código nestes sistemas computacionais.**
- Você deve submeter os três arquivos **colib.s**, **acos.s** e **colo.c** no Moodle.
- Atente-se aos nomes dos arquivos, não segui-los implicará em penalização de um fator multiplicativo de 0.9.
- Os arquivos **colib.s** e **acos.s** devem ser desenvolvidos em linguagem de montagem - A submissão de programas em linguagem de programação de alto nível, como C, ou de programas gerados por ferramentas de compilação serão consideradas fraude.
- Os arquivos base são: [colo.c](#), [control\\_api.h](#) e [linked\\_lists.s](#).
- Os arquivos **control\_api.h** e **linked\_lists.s** não devem ser modificados.