



# Relatório Laboratório 06

Maria Eduarda Teixeira Costa e Sofia Gazolla da Costa Silva

Departamento de Informática e Estatística - Universidade Federal de Santa Catarina

Organização de Computadores I

Professor Marcelo Daniel Berejuck

Outubro 2025

## 1 Introdução

Neste laboratório, deveríamos implementar um programa que utilizasse um escalonador de tarefas de um pequeno kernel para lançar duas threads, uma implementando um contador de 1 segundo e a outra um de 3. Por fim, deveríamos usar a ferramenta Digital Lab Sim para exibitessas contagens.

## 2 Desenvolvimento

Ambos os códigos seguem a mesma lógica, com pouquíssimos detalhes diferentes. Primeiramente, o arquivo `labels.asm` é incluso. Após, declaramos `.globl prog2` e `.globl prog3`, para que o kernel consiga chamar essas funções. Então, nos respectivos `.data` de cada programa, são declarados os arrays `numeros1`, que contém os padrões para exibir os números 0 e 1 no display de sete segmentos, e o array `numeros3`, que contém os padrões de 0 a 3.

No programa em si, que segue os nomes definidos pelo `.globl`, o endereço base do array é carregado em um registrador, para que os padrões dos dígitos possam ser carregados da memória e passados para os displays. Depois, no contador de um segundo, o código do

display esquerdo é carregado para um registrador, para que o programa saiba que deve exibir essa contagem no display esquerdo. Já no contador de três segundos, é carregado o código do display direito, para que a contagem seja exibida nele.

O valor de 1000ms (1 segundo) é carregado em um registrador, para poder garantir que o programa atualize o contador no tempo correto. Após isso, o syscall 30 é usado pela primeira vez, para pegar o tempo inicial.

Então, entra-se no primeiro label, `inicializa`. Em `inicializa`, o ponteiro para o endereço é resetado, para começar no primeiro número do array e o índice do número atual também é setado em 0.

Após isso, entra-se no laço principal, chamado de `loop`. Na primeira linha desse laço é onde temos a diferença entre as duas threads. Há uma instrução `branch if equal`, que verifica se todos os números do array já foram percorridos e, portanto, exibidos. Para o contador de 1 segundo, verifica-se se ele já percorreu os dois números do contador (0 e 1) e para o contador de 3, é verificado se os quatro números (0, 1, 2 e 3) já foram percorridos.

Após isso, o syscall 30 é utilizado novamente, para pegar o novo tempo e poder calcular a diferença do tempo inicial para esse novo valor. Então, o resultado dessa subtração é armazenado em um registrador. Caso essa diferença ainda seja menor do que 1000 millissegundos, indicando que um segundo ainda não passou, a instrução de `blt` faz com que o programa volte para o início do laço, mantendo o mesmo valor no display até que o tempo correto tenha passado.

Quando esse tempo houver passado, o valor do display é atualizado. Para isso, o ponteiro para o array dos números é utilizado para acessar o valor do padrão do número atual, usando a instrução `load word`. Após isso, ele é passado para o display usando uma instrução `store word`, atualizando efetivamente o número exibido. Esse laço se repete quando a condição de desvio do início do laço for satisfeita.

Ao sair do laço, o programa vai para o label `fim`, onde há um `jump` para `inicializa`, que irá fazer com que o programa e os contadores se repitam até que o usuário encerre o programa.

Sobre os nomes dos labels, eles são os mesmos para ambos os programas, com sua única diferença sendo que os do contador de 1 segundo são seguidos pelo número 1 e os do contador de 3 são seguidos pelo número 3.

### 3 Materiais e métodos

Para o desenvolvimento dessa atividade foi utilizada a linguagem **Assembly** e a IDE **MARS**, na qual foi realizada toda a escrita, execução e depuração de todos os programas implementados. Além disso, foi utilizada a ferramenta Digital Lab Sim, do próprio **MARS**, para demonstrar o funcionamento dos contadores.

### 4 Resultados

Os resultados obtidos mostraram o funcionamento correto das threads implementadas.

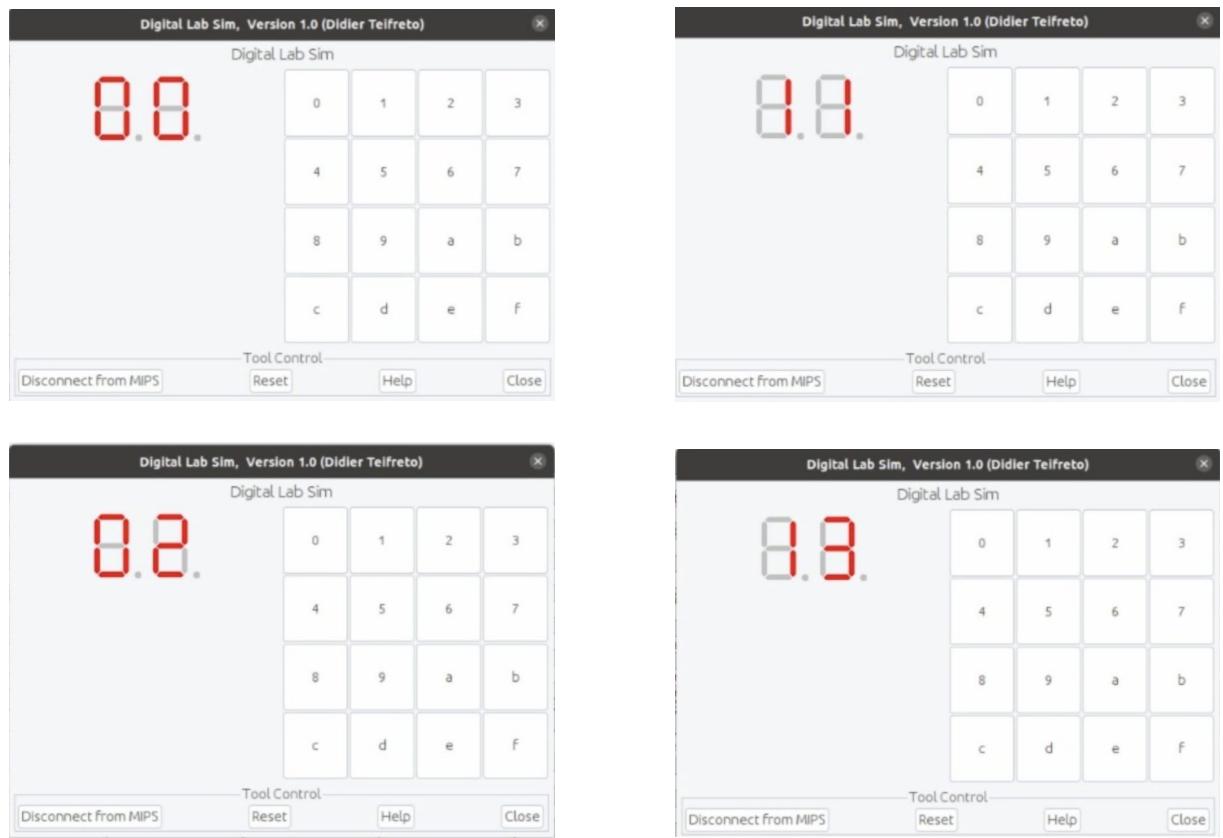


Figure 1: Demonstração do funcionamento das threads, com o contador de 1 segundo no display esquerdo e o de 3 segundos no display direito