



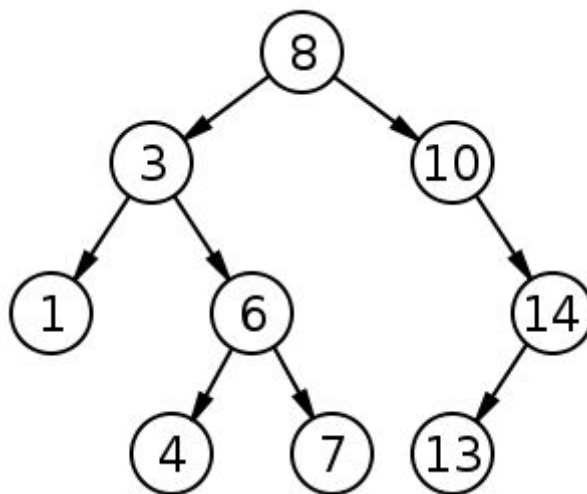
## **SSC0610 - Organização de Computadores Digitais I**

**Docente: Prof. Dr. Francisco José Monaco**

**Monitor PEEG: Vitor P. Ribeiro**

**Monitor: Guilherme Prearo**

### **Relatório - Trabalho 1**



#### **Grupo:**

Adilson Vital Junior - 9278160

César Augusto Lima - 9771525

Flávio Vinicius Vieira Santana - 9866552

Leonardo Carneiro Feltran - 9807430

## 1. Introdução

Este relatório foi elaborado com objetivo de fornecer informações complementares à documentação interna do código, portanto, as funções que não forem aqui explicadas poderão ser compreendidas pelo próprio código.

## 2. Compilação

O código deste trabalho foi dividido em diferentes arquivos para facilitar o trabalho em equipe e melhorar a organização geral. No entanto, é necessário realizar uma configuração adicional na plataforma Mars Mips.

Basta abrir a aba “Settings” na barra superior do programa e marcar a opção “Assemble all files in directory”, conforme destacado na Imagem 1.

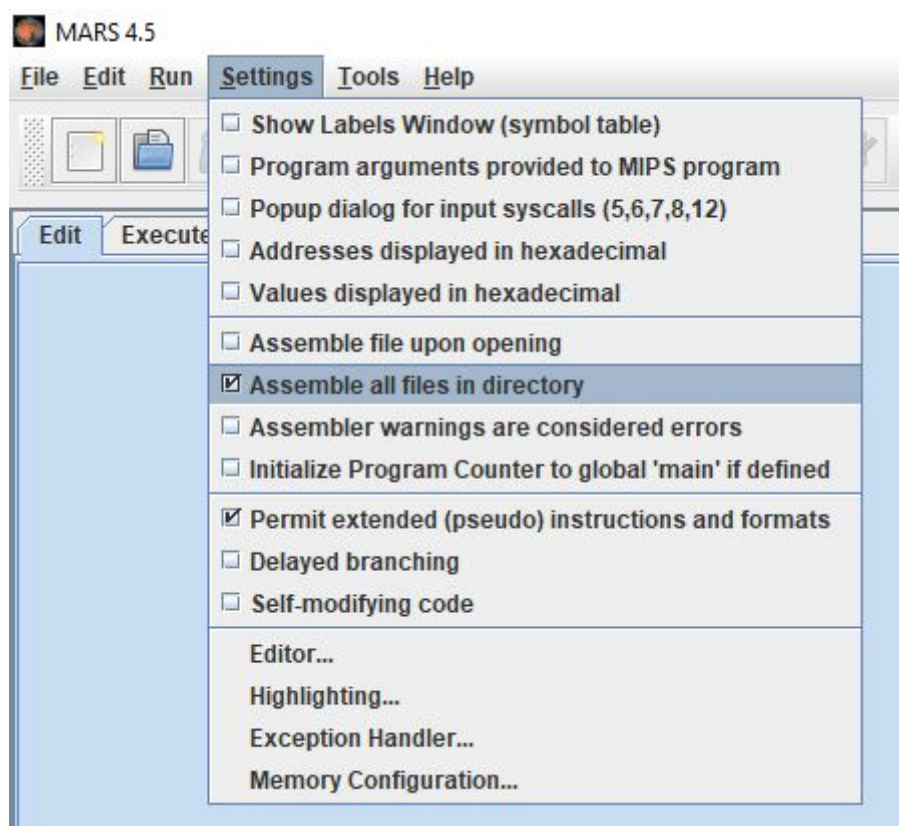


Imagem 1: destaque da configuração no Mars Mips

Feito isso, basta abrir o arquivo “main.asm” e realizar a execução padrão.

### 3. Visualização dos resultados

Para melhor visualização dos resultados do programa, é recomendado expandir a caixa “Run I/O” do Mars Mips. Para isso, basta pressionar a seta para cima, conforme destacado na Imagem 2.

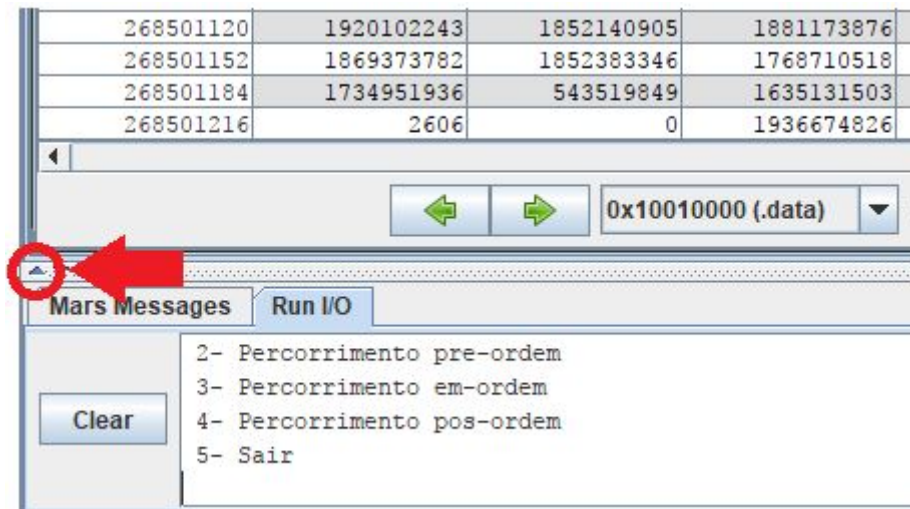


Imagem 2: destaque da seta para expandir os resultados

### 4. Armazenamento da árvore

Para armazenar a árvore dinamicamente na memória, sem comprometer o uso da stack por outras funções, foi utilizada a chamada de sistema “sbrk”. Essa chamada deve receber em \$a0 o número de bytes que serão alocados e irá retornar em \$v0 um endereço de memória com estes espaço disponível.

Neste trabalho cada nó será constituído de 3 valores inteiros, conforme pode ser observado na estrutura equivalente em C:

```
typedef struct{
    int elem;
    No* filhoEsq;
    No* filhoDir;
} No;
```

Portanto o primeiro inteiro será o valor armazenado no nó atual, o segundo inteiro será o endereço de memória do filho à esquerda do nó atual e o terceiro inteiro será o endereço de memória do filho à direita do nó atual.

Sabe-se que cada valor inteiro da plataforma Mars Mips ocupa 4 bytes de memória, portanto cada nó possui 12 bytes alocados na memória dinâmica.Oi