

# Organização e Arquitetura de Computadores - 1º Trabalho Prático

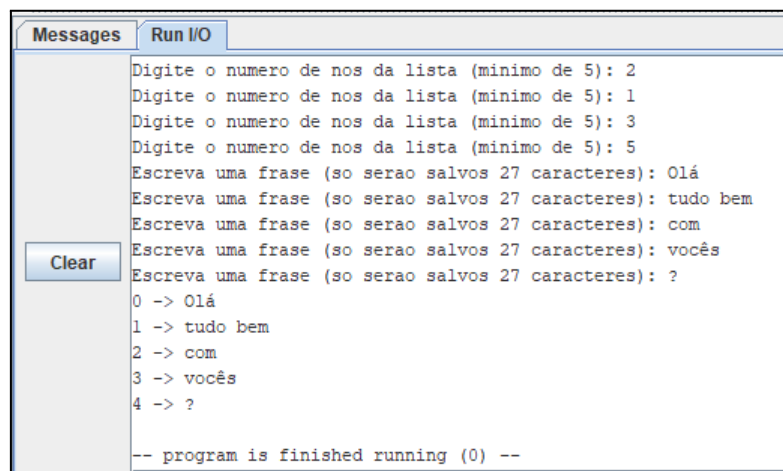
## Implementação de uma lista ligada em Assembly RISC-V

### Integrantes do grupo:

- Enzo Nunes Sedenho (13671810);
- Gabriel da Costa Merlin (12544420);
- João Pedro Mori Machado (13671331);
- Pedro Augusto Monteiro Delgado (13672766).

### 1. CASOS DE TESTE

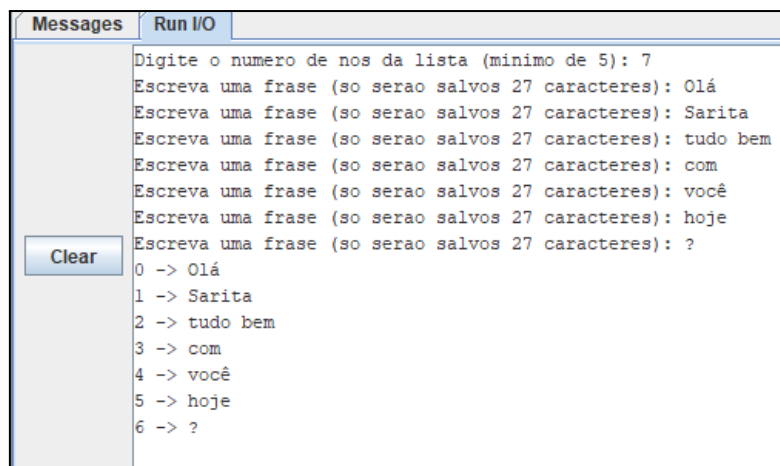
No caso do teste a seguir (figura 1), o usuário tenta inserir um número inválido para a quantidade de nós na lista (menor que 5). Assim, o programa solicita novamente a quantidade de nós na lista até que o número informado pelo usuário seja válido. Após digitar 5 (tamanho válido), o usuário insere as informações dos nós e a lista é impressa em seguida.



```
Messages Run I/O
Digite o numero de nos da lista (minimo de 5): 2
Digite o numero de nos da lista (minimo de 5): 1
Digite o numero de nos da lista (minimo de 5): 3
Digite o numero de nos da lista (minimo de 5): 5
Escreva uma frase (so serao salvos 27 caracteres): Olá
Escreva uma frase (so serao salvos 27 caracteres): tudo bem
Escreva uma frase (so serao salvos 27 caracteres): com
Escreva uma frase (so serao salvos 27 caracteres): vocês
Escreva uma frase (so serao salvos 27 caracteres): ?
0 -> Olá
1 -> tudo bem
2 -> com
3 -> vocês
4 -> ?
-- program is finished running (0) --
```

**Figura 1.** Caso de teste 1 produzido pelo grupo.

No próximo caso de teste (figura 2), o primeiro número informado pelo usuário é válido para a quantidade de nós na lista. Logo, a execução do programa segue para a inserção das informações dos nós, seguida da impressão da lista.



```
Messages Run I/O
Digite o numero de nos da lista (minimo de 5): 7
Escreva uma frase (so serao salvos 27 caracteres): Olá
Escreva uma frase (so serao salvos 27 caracteres): Sarita
Escreva uma frase (so serao salvos 27 caracteres): tudo bem
Escreva uma frase (so serao salvos 27 caracteres): com
Escreva uma frase (so serao salvos 27 caracteres): você
Escreva uma frase (so serao salvos 27 caracteres): hoje
Escreva uma frase (so serao salvos 27 caracteres): ?
0 -> Olá
1 -> Sarita
2 -> tudo bem
3 -> com
4 -> você
5 -> hoje
6 -> ?
```

**Figura 2.** Caso de teste 2 produzido pelo grupo.

## 2. DIFICULDADES ENCONTRADAS

O grupo encontrou dificuldades para a leitura da string informada pelo usuário. Inicialmente, tentou-se utilizar uma posição estática de memória para ler as strings para, depois, copiar byte a byte no nó alocado. Contudo, percebeu-se que essa abordagem é desvantajosa, pois aumenta a complexidade do código em termos de tamanho e de compreensão semântica.

Além disso, as instruções <lw> e <sw> não foram completamente compreendidas antes de serem utilizadas, gerando vários erros durante a implementação do programa. O principal problema foi a relação entre a memória e o banco de registradores durante o uso dessas instruções para manipular o ponteiro para os nós da lista.

## 3. IMPLEMENTAÇÃO DO PROGRAMA

O programa foi implementado em Assembly RISC-V e foi devidamente documentado internamente. Além disso, optou-se por explicitar o uso dos registradores na tabela 1 a seguir.

Tabela 1. Registradores e suas utilizações no programa	
Registrador	Utilização no programa
a0	Manipulação de parâmetros e retornos de funções próprias e de chamadas do sistema (via ecall).
a7	
t0	Contadores para laços de repetição.
t1	
t2	Ponteiro auxiliar para início da lista;
t3	Ponteiro auxiliar para nó alocado dinamicamente.
s0	Armazenamento do número mínimo de nós da lista (constante 5); Ponteiro para nó a ser impresso; ID de um nó a ser impresso.
s1	Armazenamento do número de nós da lista informado pelo usuário.
s2	Ponteiro salvo para o início da lista.