

Universidade Federal do Rio Grande do Norte  
Centro de Tecnologia  
Departamento de Engenharia de Computação e Automação

## Exercício de Programação Assembly I

**Disciplina:** DCA3404 - Arquitetura de Computadores

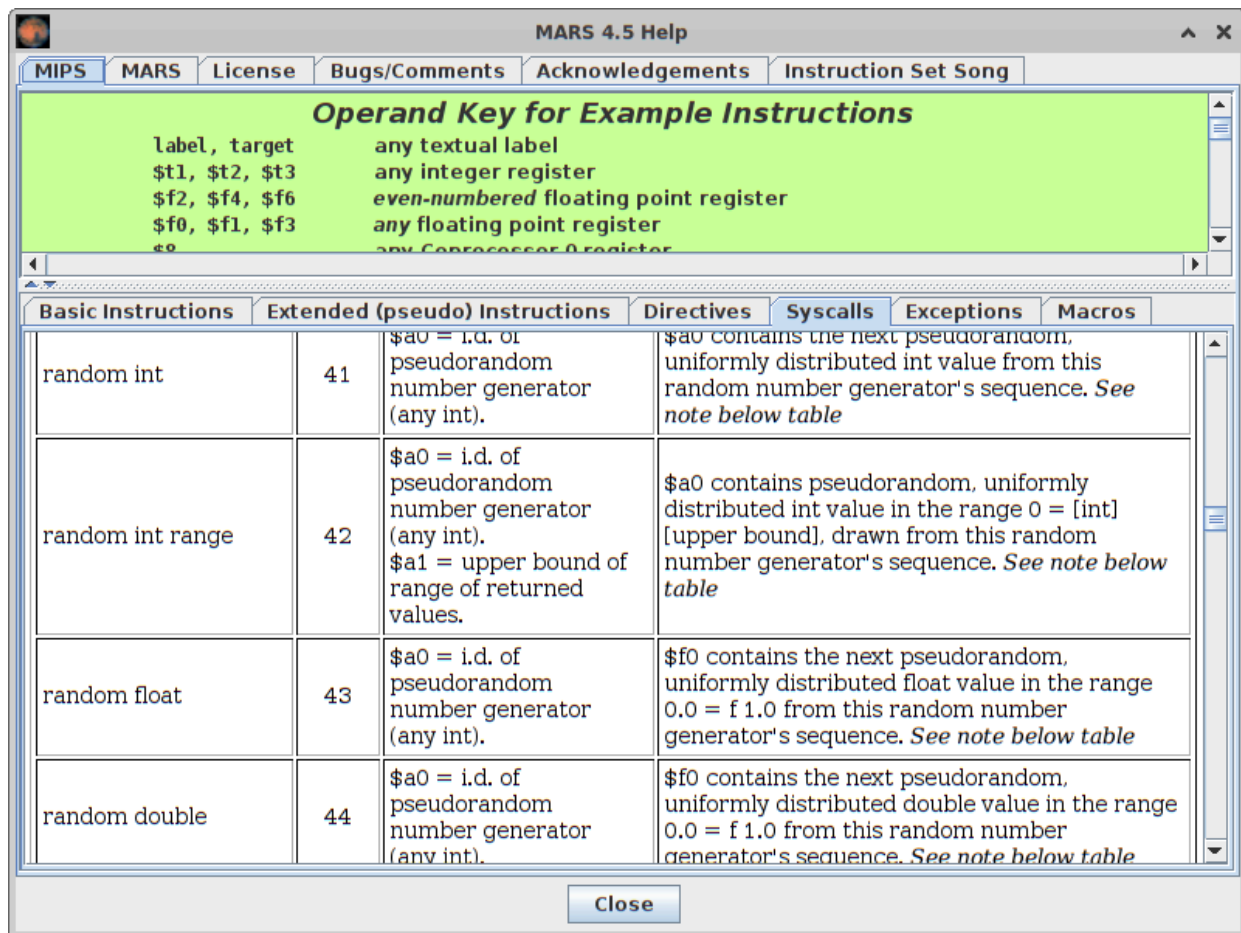
**Turma:** 1, 46T12, sala 4I2, 2024.2

**Professor:** Diogo Pinheiro Fernandes Pedrosa (diogo.pedrosa@ufrn.br)

### Problema

Esta tarefa tem o objetivo de trabalhar com diversos aspectos da programação em baixo nível, para a arquitetura MIPS. Serão abordados operações na memória de dados, chamadas do sistema, laços, entre outros.

Elabore, desenvolva e teste, no simulador MARS, um programa escrito em linguagem *assembly* para a arquitetura MIPS em que seja gerada uma sequência de 10 (dez) números inteiros aleatórios, entre 0 e 255. Para isso, deve-se utilizar a chamada de sistema correspondente. Verifique no menu *Help*, no simulador MARS, a aba *Syscalls* e procure a chamada de sistema número 42 (ver figura a seguir).



Esta sequência deve ser armazenada na memória, em um *array*, e exibida em tela. Assim sendo, alocue previamente com a diretiva do montador *.space*, no segmento de dados, um espaço necessário em *bytes* para realizar esse armazenamento. Perceba que é necessário utilizar a instrução *store word* (*sw*) para isso. Lembre-se, também, de realizar o alinhamento da memória para leitura/escrita de 4 *bytes* na memória (ver diretiva *.align*). A exibição de cada número gerado necessita de uma chamada de sistema (*print integer*).

Após isso, realize a ordenação deste *array*, de forma crescente, e exiba-os em tela.

Para a ordenação, tem-se a sugestão do algoritmo de ordenação bolha. O pseudocódigo a seguir exemplifica esse procedimento.

Considerando um vetor *V*, de *N* elementos...

Para *i* variando de 0 até *N*-1

Para *j* variando de 0 até *N*-*i*-1

Se *V*[*j*] > *V*[*j*+1]

aux = *V*[*j*];

*V*[*j*] = *V*[*j*+1];

*V*[*j*+1] = aux;

```
Fim-Se  
Fim-Para  
Fim-Para
```

## Observações

- Esta é uma tarefa individual;
- Teste, inicialmente, o programa desenvolvido com um vetor pequeno (por exemplo,  $V[3] = \{1, 3, 2\}$ ) e veja se o seu código *assembly* está retornando o esperado. Depois, teste a geração de um número aleatório entre 0 e 255. Por fim, junte os procedimentos e veja o resultado;
- O envio da tarefa será, apenas, o código no formato .ASM. É obrigatório que você coloque seu nome e matrícula (como comentário, no código do arquivo) e que faça os devidos comentários para explicar o que ocorre nos vários trechos do código de baixo nível. Este envio será pelas Tarefas, no SIGAA da turma virtual;
- Lembre-se de testar o código no MARS antes de enviá-lo pelo sistema. Com isso, evita-se que, caso ainda exista algum erro, ele possa ser corrigido antes da submissão.
- Quaisquer dúvidas, enviem mensagem para [diogo.pedrosa@ufrn.br](mailto:diogo.pedrosa@ufrn.br).