

# Laboratório 2

## Sub-rotinas e Pilha

Este laboratório discute os conceitos de sub-rotinas, pilha e passagem de parâmetros. Usaremos o ambiente com o processador Nios II para ilustrar esses conceitos. Como no laboratório anterior, faremos uso do sistema *DE2 Basic Computer*. O seu relatório deve incluir um resumo das atividades que você desenvolver nas partes I e II a seguir.

### Parte I

Desejamos ordenar uma lista de números não sinalizados de 32 bits em ordem decrescente. Um algoritmo simples para efetuar essa ordenação é conhecido como *Selection Sort*, mostrado logo abaixo.

```
// vector é o vetor com os elementos a serem ordenados
// size é o tamanho do vetor

for (i=0; i < size; i++) {
    indice_max = i;
    for (j=i+1; j < size; j++) {
        if (vector[j] > vector[indice_max]) {
            indice_max = j;
        }
    }
    if (i != indice_max) { // se maior foi encontrado, troca
        temp = vector[indice_max];
        vector[indice_max] = vector[i];
        vector[i] = temp;
    }
}
```

Sua tarefa é implementar esse algoritmo de ordenação em linguagem de montagem do Nios II. Você deve implementá-lo como uma sub-rotina, que será invocada externamente por um programa disponibilizado na página da disciplina, chamado de *main.s*. Não se esqueça de seguir as convenções para uso de registradores determinadas pela ABI, conforme visto em aula.

Prossiga da seguinte forma:

1. Crie um arquivo novo, chamado *sort.s*, para implementar o algoritmo de ordenação. Não se esqueça de criar um símbolo global chamado `SORT`, para que o código em *main.s* consiga invocar sua sub-rotina.
2. Crie um projeto novo, escolhendo o sistema *DE2 Basic Computer*. Os arquivos que farão parte deste projeto serão o baixado da página da disciplina (*main.s*) e o que você escreverá (*sort.s*).
3. Compile o projeto na placa usando o Altera Monitor para verificar que está tudo compilando corretamente. Não execute ainda!
4. Escreva o programa de ordenação. Assuma que o tamanho e a posição da lista a ser ordenada são parâmetros passados para a sub-rotina através de registradores, tal que:

- O parâmetro *tamanho* é dado pelo conteúdo do registrador *r4* do Nios II.
- O endereço da primeira entrada da lista é dado pelo conteúdo do registrador *r5*.

A sub-rotina deve salvar na pilha o conteúdo dos registradores que ela usa, e restaurá-los antes de retornar o fluxo de execução ao programa principal. Planeje seu código de forma a usar os registradores adequados, conforme definido pela ABI e visto em aula.

5. Antes de executar o seu código você precisa carregar para a memória a lista com os números a serem ordenados. Carregue o arquivo `small.dat` (disponível na página da disciplina) na memória a partir do endereço `0x500`.

Nota: O arquivo que contém a lista pode ser carregado para a memória através do programa monitor, como explicado na seção 11 do documento “*Altera Monitor Program Tutorial*”.

6. Execute seu programa e certifique-se que funciona corretamente. Posteriormente, carregue o arquivo `large.dat` (também na página da disciplina) e reexecute a ordenação, garantindo que ela continue funcionando.

## Parte II

O processador Nios II usa o registrador *ra* (*r31*) para armazenar o endereço de retorno quando a sub-rotina é chamada. No caso de sub-rotinas aninhadas, onde uma sub-rotina chama outra, é necessário garantir que o endereço de retorno original não seja perdido quando um novo endereço de retorno é colocado no registrador *ra*. Isso pode ser feito ao armazenar o endereço de retorno original na pilha e recarregando-o no registrador *ra* após o retorno da sub-rotina chamada.

Para demonstrar o conceito de sub-rotinas aninhadas, vamos usar a computação do *n-ésimo* número da sequência de Fibonacci. O *n-ésimo* número de Fibonacci é computado como

$$Fib(n) = Fib(n - 1) + Fib(n - 2)$$

Note que  $Fib(0) = 0$  e  $Fib(1) = 1$ .

Escreva um programa que use recursão para computar o *n-ésimo* número de Fibonacci. O programa deve incluir uma sub-rotina, chamada FIBONACCI, que invoca ela mesma, repetidas vezes, até que o número desejado seja computado. O programa principal deve passar *n* como parâmetro para a sub-rotina na pilha. A sub-rotina deve retornar o resultado através do registrador *r2*.

Compile, carregue e execute seu programa. Verifique que seu funcionamento está correto para diferentes valores de *n*.