# ARQUITETURA DE COMPUTADORES

2021-2022

Laboratório 3 – Funções

Este laboratório destina-se a consolidar conhecimentos de introdução à programação na linguagem Assembly da arquitetura RISC-V, utilizando o simulador Ripes[1].

O trabalho deve ser realizado fora do horário de laboratório, destinando-se este à demonstração e avaliação do trabalho realizado. No final da aula de laboratório deverá submeter o código Assembly no Fénix.

Para garantir a correção da solução, deverá validar a sua solução do laboratório no emulador (Ripes) e confirmar os resultados, observando os valores finais dos registos ou o conteúdo da memória. Pode igualmente colocar pontos de paragem no código (de tal forma que o emulador parará a execução sempre que atingir um destes pontos) clicando nos números das linhas correspondentes.

## Exercício 1

Considere a função `dist` apresentada de seguida.

```
int dist(x1,y1,x2,y2){
  int dx = x1-x2;
  int dy = y1-y2;
  return dx*dx+dy*dy;
}
```

Escreva duas implementações da função acima descrita:

- `dist_reg`: passagem de parâmetros por registo, seguindo a convenção descrita no "RISC-V Reference Guide"
- `dist_stack`: passagem de parâmetros pela pilha

Teste o seu código com diferentes exemplos e mostre ao docente no início da aula. Discuta com o docente as diferenças de complexidade relativamente ao número total de instruções e número de acessos à memória (load ou store).

[1] https://github.com/mortbopet/Ripes

## Exercício 2

Considere uma sequencia de pontos representados num plano xy, cujas coordenadas são {(x0,y0),(x1,y1),(x2,y2),(x2,y3),…}. Pretende-se implementar uma função `find_neareast` que determina o índice i do ponto mais próximo de uma dada referência ($x_{ref}$,$y_{ref}$).

Considere que as coordenadas da lista de pontos se encontram em dois arrays vx e vy, o primeiro com as coordenadas no eixo dos x, o segundo com as coordenadas no eixo dos y.

Escreva o código Assembly correspondente à função `find_nearest` apresentada de seguida.

```
int find_nearest(int xref, int yref, int *vx, int* vy, int N){
    int k, index = 0;
    int d, dmin = dist(xref,yref,vx[0],vy[0]);
    for (k=1; (k<N) && (dmin>0); k++)
        d = dist(xref,yref,vx[k],vy[k]);
        if (d<dmin) {
            dmin = d;
            index = k;
        }
    }
}
```

Considere por exemplo a seguinte invocação da função:

```
int d1[] = {-1, 3,7,-2, 4,1,5,9,1,-5};
int d2[] = { 6,-3,2,-2,-3,2,1,0,4,-2};
int N = 10; // número de elementos nos vetores A e B

i=find_nearest(3,4,d1,d2,N);
```

Teste o seu código com diferentes exemplos e mostre ao docente no início da aula. Escolha uma implementação na passagem de parâmetros: por registo ou pela pilha.

## Exercício 3

Este é um exercício surpresa que será divulgado pelo docente durante a aula.

## Método de avaliação

O laboratório será avaliado numa escala de 0-8, com a seguinte ponderação:

- Exercício 1 – 1 ponto
- Exercício 2 – 1 ponto
- Exercício 3 – 3 pontos
- Mini-Teste (Moodle) – 3 pontos