

Introdução à Arquitetura de Computadores, Grupo 6

Relatório da 1ª Parte do Projeto “Dino”

Ana Loureiro, nº 98794
Manuel Pereira, nº 98580

No âmbito da cadeira de Introdução à Arquitetura de Computadores, foi-nos pedido que desenvolvêssemos a primeira parte do projeto, que consiste num programa com um conjunto de funções e um simples código de teste, escrito em Assembly do P4, para simular o estado do terreno de jogo.

A nossa primeira abordagem foi começar por escrever a função `geracacto`, visto que esta vai fazer parte da função `atualizajogo`. Segue uma breve descrição de ambas as funções e do programa de teste:

Função `geracacto`:

A função começa por conservar os valores de R1, R4 e R5, pois serão utilizados na função `atualizajogo`. De seguida, copia para R1 o endereço de memória onde se encontra a altura máxima dos cactos e para R2 o endereço de memória onde se encontra a “semente” de valor arbitrário. Depois realizamos uma “tradução” simples do pseudo-código em Python para Assembly, sendo que tivemos de ter especial atenção na parte “if x < 62258” pois, de forma a realizar esta comparação no P4, tivemos de verificar se o bit de estado C era ou não ativado invés do bit de estado N, devido a não estarmos a interpretar os números como complemento para dois. No final da função é guardado o novo valor da “semente” no endereço de memória associado a “semente”, de forma a que na próxima invocação da função seja utilizado este novo valor.

Função `atualizajogo`:

Esta função recebe dois argumentos, o endereço de memória do início do vector com o terreno de jogo (“`inicioterreno`”) e a dimensão do vector que contém o terreno de jogo (“`dim`”), que são copiados para R1 e R2, respetivamente.

A função começa por conservar os valores de R4 e R5 na pilha e em seguida decrementar o valor de R2 em uma unidade, para que o tamanho do vector no loop não seja ultrapassado. Depois, através do loop, começa por mover R1 para R4, incrementando de seguida o valor de R1 em uma unidade, e carregando em R5 o valor que se encontra no endereço de memória em R1. Ainda dentro do loop, é guardado o valor de R5 no endereço de memória guardado em R4 e é feito um decremento de uma unidade ao valor de R2. O loop continuará a correr até que R2 seja igual a 0. Após o loop é chamada a função `geracacto`, sendo o valor de retorno desta última, que se encontra em R3, guardado no endereço de memória que se encontra em R1.

Programa de teste:

O programa de teste trata-se apenas de um loop (“`loopteste`”) em que a função `atualizajogo` é chamada 80 vezes. Para modificar o número de vezes que a função `atualizajogo` é chamada, a altura máxima dos cactos ou a dimensão do terreno de jogo, basta modificar o valor inicial de R4, modificar o valor de “altura”, o valor de “dim” e o valor de “`inicioterreno`”, respetivamente.