

[illegible]

Construa os seguintes programas para a nossa MV, no mínimo os que seguem:

- a) P1: o programa escreve em posições sabidas de memórias os 10 números da sequência de fibonacci. Ou seja, ao final do programa a memória tem estes 10 números em posições convencionadas no programa.
- b) P2: o programa le um valor de uma determinada posição (carregada no início), se o número for menor que zero coloca -1 no início da posição de memória para saída;
se for maior que zero este é o número de valores

da sequência de fibonacci a serem escritos em sequência a partir de uma posição de memória;

- c) P3: dado um inteiro em alguma posição de memória,
se for negativo armazena -1 na saída;
se for positivo responde o fatorial do número na saída.
- d) P4: para um N definido (5 por exemplo)
o programa ordena um vetor de N números em alguma posição de memória;
ordena usando bubble sort
loop ate que nao swap nada
passando pelos N valores
faz swap de vizinhos se da esquerda maior que da direita

1.4 Shell

Construa um shell, ou seja, um terminal interativo que:

- oferece um prompt ao usuário
- o usuário pode solicitar a execução de um programa
- a partir disso
 - o programa é carregado
 - executado
 - ao acabar retorna controle para o shell
- para armazenar o programa, pode-se adotar
 - um arquivo por programa, ou mesmo
 - que existe uma estrutura em memória com os programas que são então carregados para a memória da VM.

2. PROBLEMA

Implemente: a MV descrita acima; o Shell; e rotinas necessárias.

Defina como acontece a carga do programa em memória e o início da execução.

Execute os programas P1 a P4 na MV.

Instrumente para ver o andamento do processamento na MV.

Referências:

[1] Design of a General Purpose 8-bit RISC Processor for Computer Architecture Learning. Antonio Hernández Zavala, Oscar Camacho Nieto, Jorge A. Huerta Ruelas, Arodí R. Carvallo Domínguez. Computación y Sistemas, Vol. 19, No. 2, 2015, pp. 371–385 ISSN 1405-5546 doi: 10.13053/CyS-19-2-1941