

Interrupções e Chamadas de Sistema - Definições e Exercícios

Dada a máquina virtual disponibilizada, define-se aqui os tipos de interrupções e chamadas de sistema mínimos para o trabalho. **Estes exercícios integram (são necessários) na primeira entrega de trabalho, que será ainda marcada.**

1. Interrupções

1.1 Tipos de Interrupções

Interrupções previstas (não exaustivo):

- Endereço inválido: programa do usuário acessando endereço fora de limites permitidos;
- Instrução inválida: a instrução carregada é inválida;
- Overflow em operação matemática;
- Final de programa.

Observações:

O programa do usuário pode ficar em loop. Neste momento não trataremos esta situação. Em um sistema real, o programa deve ser monitorado. Para tal faz-se necessário um agente externo que liga uma interrupção de tempos em tempos para gerenciar os processos. Isto será visto junto de estratégias de escalonamento.

1.2 Funcionamento da CPU: o ciclo de instruções revisado

Após carga do programa em memória, o PC é desviado para a primeira posição do programa.

A CPU executa o ciclo de instruções.

Loop

busca a posição de memória apontada por PC, carrega no RI
executa operação, atualiza PC conforme operação.

verifica existência de interrupção e trata. // adicionado

fimLoop

1.3 Exercício

Para cada instrução da VM, prever que interrupções podem ser geradas.
Programar interrupções na VM.

2. Chamadas de Sistema

Além de tratar interrupções, um sistema necessita também oferecer operações aos (programas dos) usuários. Estas operações abstraem o uso de dispositivos (discos, rede, teclado, mouse, etc.), permitem acesso a recursos do sistema (alocar memória, obter dados dos processos, etc.), entre outros.

Estas funcionalidades são oferecidas por um monitor residente ou sistema operacional na forma de chamadas de sistema.

Nesta fase implementaremos a possibilidade de duas operações, na forma de chamadas de sistema:

- Entrada (in): o programa lê um inteiro do teclado
- Saída (out): o programa escreve um inteiro na tela

Entenda que teclado e tela são dispositivos acessados por operações específicas. Estas operações fazem parte do sistema operacional. O programa do usuário chama as mesmas. A interface de chamadas de sistemas pode mudar conforme o HW e o SO. Comumente emprega-se a noção de Trap ou interrupção de SW para desviar para a rotina do sistema operacional. Este mecanismo é muito semelhante ao desvio para uma rotina de tratamento de interrupção - entretanto a Trap é provocada pelo programa do usuário. Nesta versão do sistema:

- adicionamos a operação **TRAP** no conjunto de instruções do processador
- adicionamos um registrador **reg[8]** para codificar a Trap que se quer solicitar
- adicionamos um registrador **reg[9]** que aponta para os parâmetros da Trap
- definimos códigos de traps para **IN (reg[8]=1)** e para **OUT (reg[8]=2)**
- a chamada de sistema **IN** - lê um valor inteiro do teclado, o parâmetro para **IN**, em **reg[9]**, é o endereço de memória para armazenar a leitura
- a chamada de sistema **OUT** - escreve um valor na tela, o parâmetro para **OUT**, em **reg[9]**, é o endereço de memória cujo valor deve-se escrever na tela
- após carga dos registradores, a operação Trap provoca o desvio para respectiva rotina do SO
- esta definição de TRAP permite estender para outros códigos (em **reg[8]**) e para usar **reg[9]** de forma flexível. Por exemplo, se uma trap precisa de vários parâmetros, **reg[9]** pode apontar para a região de memória onde iniciam estes parâmetros armazenados.

Exemplificação.

Programa IN

```
0  LDI 8, 1    // leitura          R8 ← 1
1  LDI 9, 4    // endereço a guardar R9 ← 4
2  TRAP        // rotina realiza leitura de teclado, e valor vai para memória do processo (DMA) na posição dada
3  STOP
4  _____  // valor lido estará armazenado aqui!!!
```

Programa OUT

```
0  LDI 0, 999
1  STD 10, 0
3  LDI 8, 2    // escrita
4  LDI 9, 10   // endereço com valor a escrever
5  TRAP
6  STOP
7
8
9
10 999        // depois da execucao. valor desta posição foi escrito na linha 1.
           // trap faz print do conteúdo da posição 10 do programa, que contém 999
```

2.1 Exercício

Implemente a chamada de sistema. Na CPU a TRAP deve desviar para uma rotina do sistema operacional. Seu sistema operacional está escrito na linguagem de alto nível (Java). Ele inclui as rotinas para fazer entrada e saída conforme definido acima.

Crie ou modifique programas existentes para a máquina virtual para que os mesmos façam operações de entrada e saída, para testar e demonstrar estas funções.