Interrupções

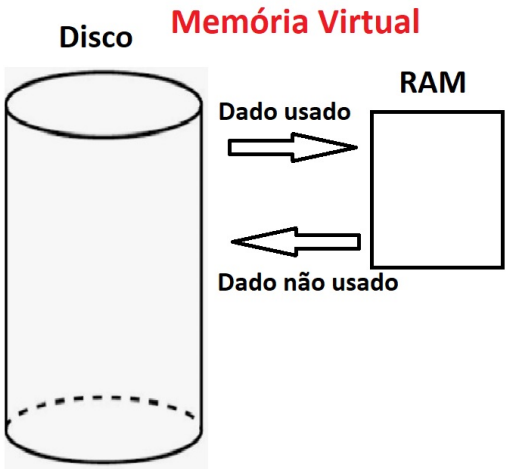
MAC0344 - Arquitetura de Computadores
Prof. Siang Wun Song
Slides usados: https://www.ime.usp.br/∼song/mac344/slides08-interrupts.pdf

Baseado parcialmente em W. Stallings - Computer Organization and Architecture

Interrupções

- O assunto Interrupções é da disciplina Sistemas Operacionais - S.O.
- Apoio de hardware é essencial. Por isso está na ementa desta disciplina.
- Interrupção é um evento que requer a atenção do S. O.
- Uma interrupção pode se originar de várias maneiras. As mais comuns são:
  - Interrupção externa: um dispositivo necessidade de atenção.
  - Interrupção interna ou de software: a execução de um comando dá origem a uma interrupção.

Interrupções



- Interrupção não significa necessariamente erro. Exemplos:
- Numa interrupção externa, um dispositivo (e.g. teclado) precisa de atenção pois um dado foi entrado.
- Acesso a uma página que não está na memória. Uma interrupção (*page fault*) traz a página do disco para a memória.
- Interrupção de relógio: um processo atingiu o tempo máximo de ocupar a CPU. Uma interrupção retira temporariamente o processo para deixar um outro entrar em execução.

Interrupções

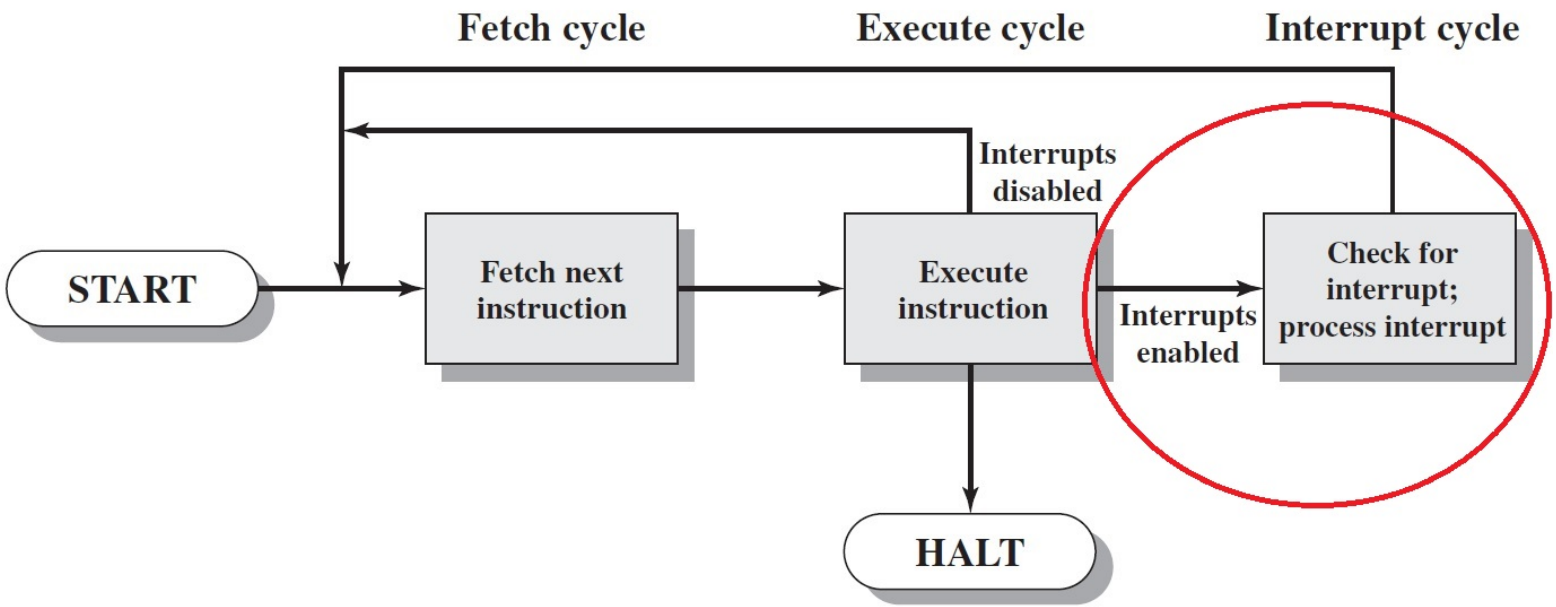
x = 0

y = z / x

Interrupção!

- Algumas interrupções podem sinalizar situações de erro. Exemplos:
  - Uma interrupção interna: Um processo realiza uma operação inválida, e.g. dividir por zero, ou uma tentativa de acesso a uma posição de memória de outro processo.
  - Queda iminente de energia elétrica: uma interrupção de altíssima prioridade ocorre alguns milissegundos antes da queda a fim de avisar ao S.O. para salvar certos contextos e dados essenciais.

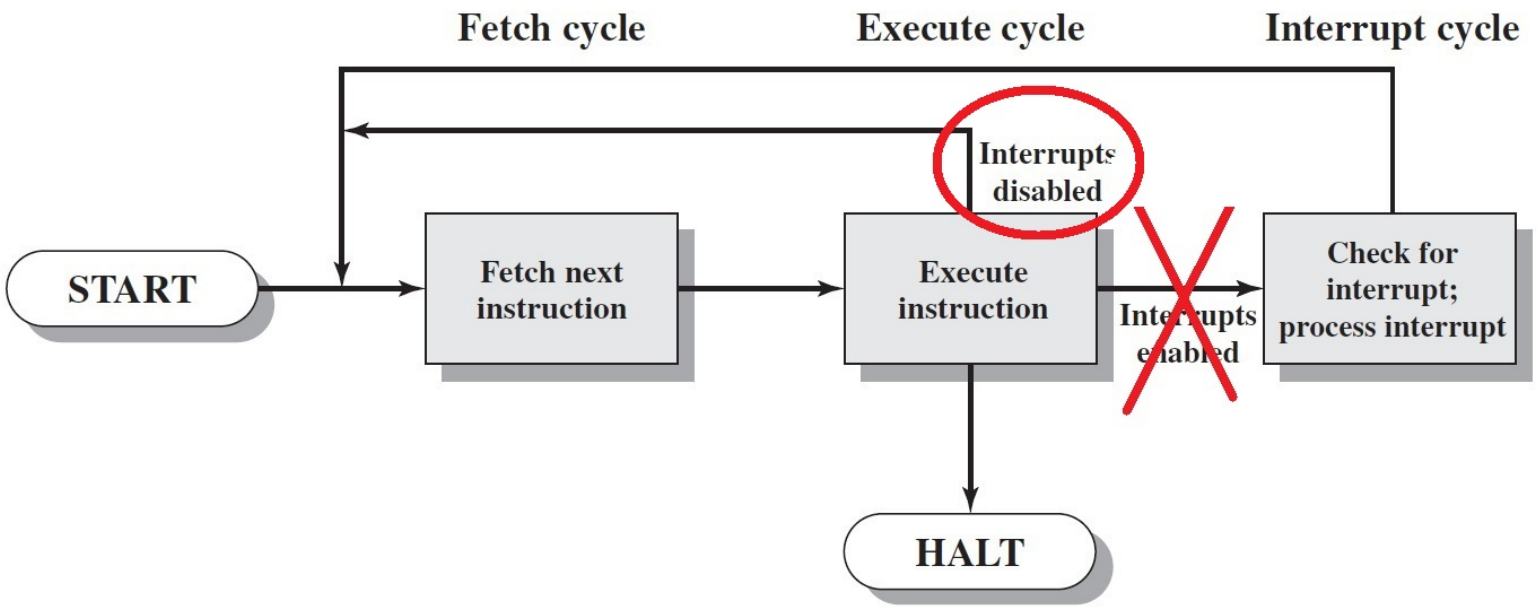
Ciclo de execução de uma instrução



Source: W. Stallings

- Um dispositivo precisa de atenção do S.O.: liga um sinal para pedir interrupção.
- Interrupção pode estar habilitada (o processo em execução pode ser interrompido) ou inibida (o processo em execução não pode ser interrompido).
- No final de um ciclo de execução de uma instrução, se interrupção está habilitada, então é verificado se há pedido de interrupção. Se sim, passa-se ao tratamento da interrupção.
- Se interrupção está inibida, não se verifica se há pedido de interrupção.

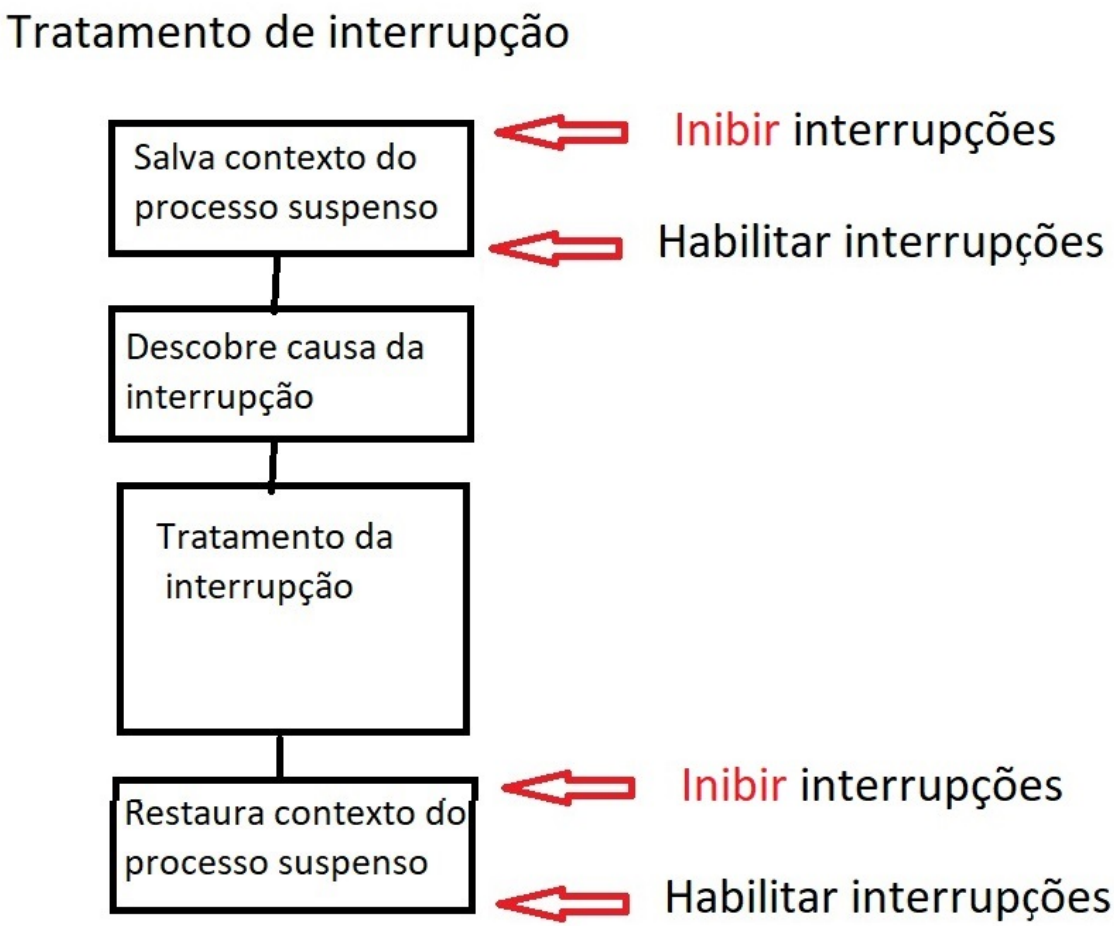
Ciclo de execução de uma instrução



Source: W. Stallings

- Ao atender a uma interrupção, o sistema suspende o processo corrente. Mas o PC deste processo suspenso já está preparado para a próxima instrução.
- Esse valor de PC precisa ser salvo para poder ser restaurado para continuar a sua execução após o término da interrupção.
- Além do PC, é preciso salvar o contexto (registradores etc) do processo sendo suspenso para poder restaurar ao mesmo contexto quando retornar.
- Enquanto se salva o contexto do processo, novas interrupções são inibidas.

Tratamento de interrupções

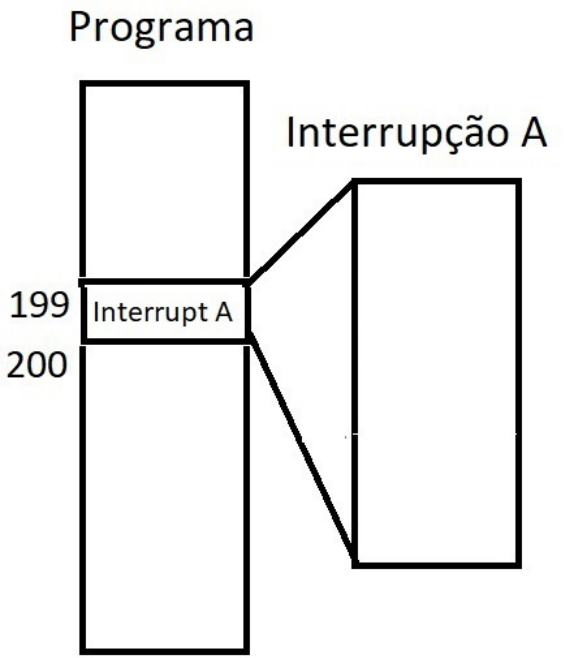


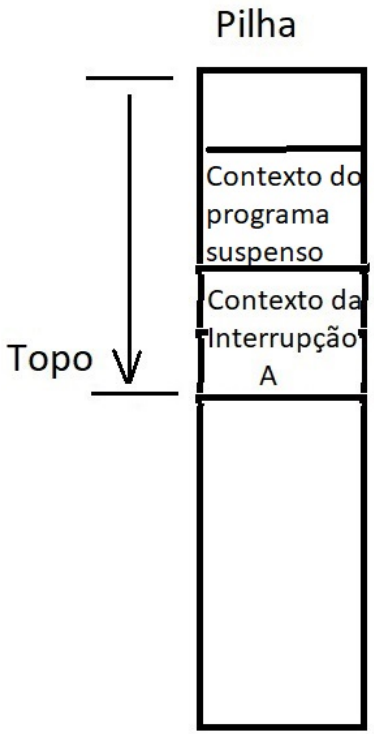
- O S.O. inicia o tratamento salvando o contexto do processo suspenso. Nessa etapa, novas interrupções são inibidas.
- A seguir, interrupções são habilitadas para que a presente interrupção possa ser interrompida por uma nova interrupção de maior prioridade.
- Finalmente durante a restauração do contexto do processo suspenso, interrupções são novamente inibidas.

Tratamento de Interrupções

- Vejamos como o S.O. descobre a causa da interrupção para a devida providência.
- Há duas maneiras para descobrir a causa:
  - Através de *polling*: o S.O. consulta cada dispositivo que pode causar a interrupção para conhecer a origem e a causa da interrupção.
  - Através da chamada *interrupção vetorada*: o dispositivo que precisa de atenção fornece já o endereço para tratar a interrupção.

Interrupções



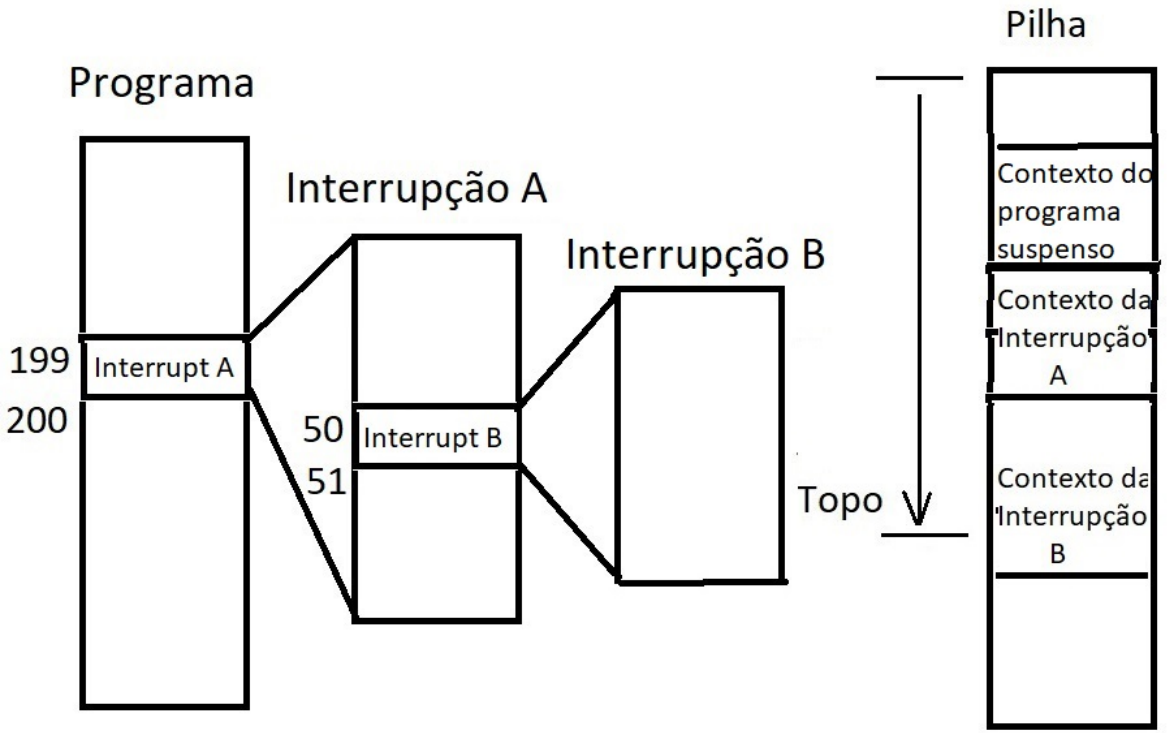


- Tratar uma interrupção é semelhante a chamar um procedimento. Ambos precisam salvar dados do processo que será suspenso.
- Na figura, um processo precisa ser interrompido por causa da interrupção A.
- O S.O. entra em ação para tratar da interrupção A.
- A interrupção A tem o seu contexto próprio. Então o contexto do processo suspenso precisa ser salvo.

Encadeamento de interrupções

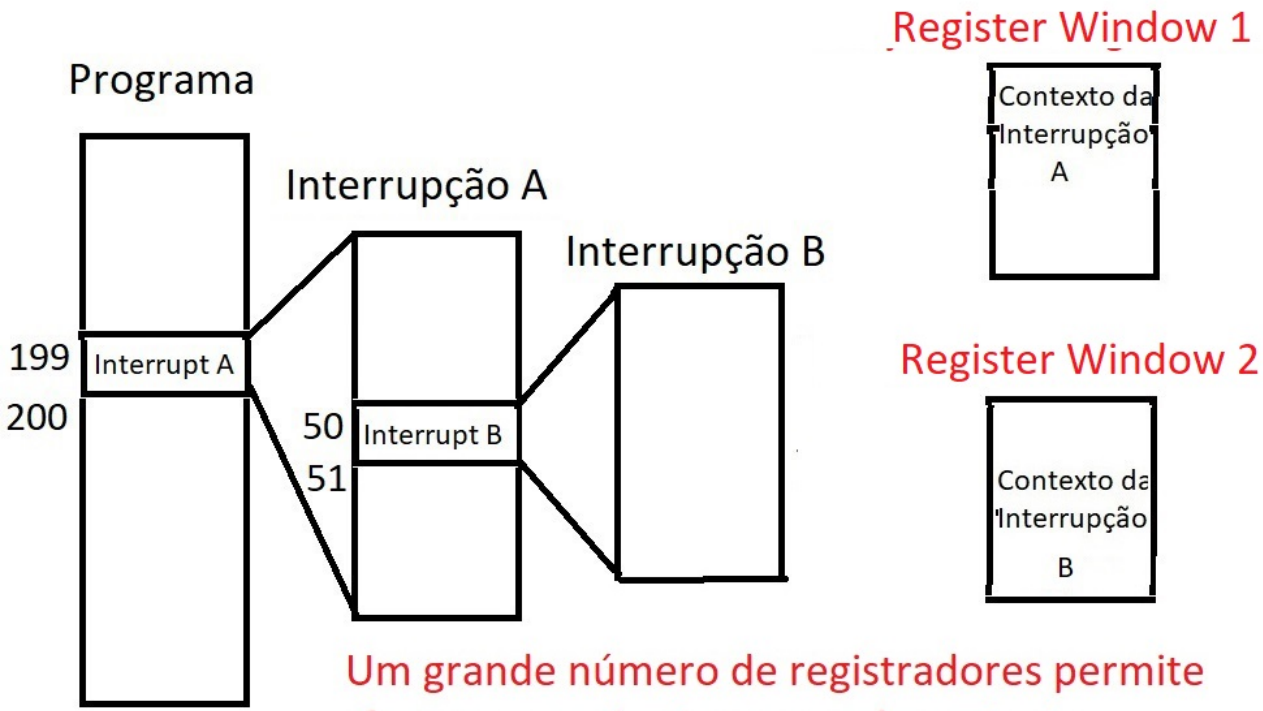
- Uma interupção de prioridade maior pode interromper uma de prioridade inferior, dando origem a um encadeamento de interrupções. Exemplo:
  - Um processo provoca uma interrupção (*page fault*) ao necessitar de uma página que não está na memória. O S.O. entra para tratar da interrupção.
  - Um dispositivo que controla a temperatura de um forno causa uma interrupção de alta prioridade para ter a atenção do S.O. pois a temperatura está alta demais. O S.O. passa a tratar essa nova interrupção.
  - Tendo tratada a nova interrupção, o S.O. retorna ao tratamento da interrupção anterior.
- O encadeamento de interrupções se assemelha a chamada de procedimentos quando um procedimento chamado pode chamar outros.

Interrupções



- Quando o S.O. está tratando a interrupção A, ocorre uma interrupção B de prioridade maior.
- O S.O. passa a tratar da interrupção B, guardando o contexto da interrupção A para poder restaurar mais tarde.
- A interrupção B tem seu contexto próprio.

Interrupções



Um grande número de registradores permite alocar um conjunto para cada contexto.
A passagem de um contexto para outro é rápido, basta mudar um ponteiro a outro Register Window.

- Um computador RISC possui um grande número de registradores.
- Conjuntos diferentes de registradores, chamados *Register Windows*, podem ser alocados para representar os contextos de cada interrupção.
- Estando os registradores dentro do processador, o acesso ao contexto de cada interrupção fica ágil e basta mudar um ponteiro ao Register Window desejado.

Arquitetura RISC

- Veremos a seguir a arquitetura RISC.
- Por não possuir microprograma, sobra espaço no processador para mais registradores.
- Não é incomum um processador RISC possuir centenas de registradores.
- Vimos um uso dos registradores para agilizar troca de contexto em interrupções e chamadas de procedimentos.
- De modo geral, com mais dados e instruções armazenados em registradores, há menos acessos à memória.