



## Interrupções, Traps e Temporizadores

Universidade Federal de Uberlândia  
Faculdade de Computação  
Prof. Dr. rer. nat. Daniel D. Abdala

## Na Aula Anterior...

- Abstração de Computadores e a Arquitetura von Neumann;
- Sistemas Computacionais (PC);
- ISA – Arquitetura do Conjunto de Instruções;
- Assembly;
- Organização de Computadores – Caminho de Dados;
- Processadores Mono vs Multiciclo;
- Desempenho de UCPs;
- Pipelining;
- Hierarquia de Memórias;
- Memórias Cache.

2

## Nesta Aula

- Interrupções por software e hardware;
- Temporizadores;
  - RTC;
  - Temporizadores via software.

3

## Introdução

- AOC deixa alguns pontos em aberto no que se refere a sua continuidade em SO:
  - Como dispositivos de E/S funcionam?
  - Como o sistema de Interrupções funciona?
  - Instruções de TRAP, para que servem?
  - O que são sinais e como eles são utilizados?
  - O que são temporizadores, RTC e como marcar a passagem do tempo em sistemas computacionais?
- Estudaremos estes pontos nesta aula!

4

## Interrupções

- Dispositivos de E/S funcionam de maneira desacoplada do processador;
- Alguns dispositivos são bem simples no entanto outros são muitíssimos complexos;
- Um disco rígido por exemplo apresenta velocidade de funcionamento na ordem de mili segundos;
- Quando o SO requer que dados sejam lidos do HD, o tempo entre requisição e retorno dos dados é suficiente para executar milhares de instruções pelo processador;
- Ficar com a CPU “travada” esperando que os dados sejam retornados pelo dispositivo é desperdício de tempo de CPU;
- Como aliviar esta situação? → **Interrupções!**

5

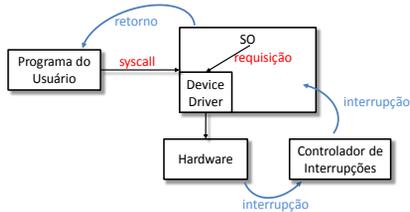
## Interrupções

- Forma de comunicação **assíncrona** entre CPU e dispositivos de E/S;
- Para a maioria dos dispositivos de E/S, interrupções sinalizam que uma operação de E/S finalizou;
- Interrupções podem ser divididas em dois grupos:
  - Interrupções de Hardware;
  - Interrupções de Software;
    - Falta de página.

6

## Interrupções

- Caminhos de Comunicação e Trocas de Mensagens;



7

## Interrupções (Hardware)

- O mecanismo de Interrupções é extremamente dependente do SC e pode variar grandemente;
- O mecanismo de interrupção por hardware pode ser sumarizado da seguinte forma:
  1. Dispositivo de E/S finaliza um trabalho e então gera uma interrupção;
  2. O sinal de interrupção é enviado pela linha do barramento a que está associado para o **controlador de interrupções**;
  3. O controlador de interrupções detecta o sinal e decide como proceder;
    - a) Decide que interrupção atender primeiro;
    - b) Implementa a política de prioridades;
    - c) Gerencia interrupções simultâneas;
  4. Quando uma interrupção é atendida pelo controlador de interrupções, este coloca no barramento de endereços o **ID do dispositivo** e sinaliza a CPU via barramento que uma interrupção ocorreu;

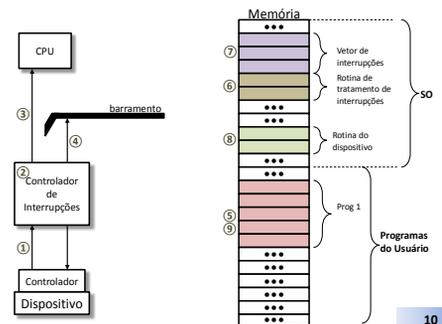
8

## Interrupções (Hardware)

5. Ao receber um sinal de interrupção a CPU procede da seguinte maneira:
  - a) Verifica se interrupções estão habilitadas. Em caso negativo, a interrupção é ignorada e nenhuma confirmação é enviada ao controlador de interrupções;
  - b) Caso habilitadas, ela salta imediatamente para a **rotina de tratamento de interrupções**;
  - c) Nesta rotina, o contexto do processador é salvo na pilha e a seguir ela lê do barramento de endereços o ID do dispositivo e usa o número como um índice para o **vetor de interrupções**;
  - d) Do vetor de interrupções ela obtém um ponteiro para a posição da memória onde se encontra a rotina que processa a o dispositivo requisitante;
  - e) Ao finalizar, esta rotina retorna o controle para a **rotina de tratamento de interrupções**, que por sua vez, restaura o contexto e retorna o controle para o programa previamente interrompido.

9

## Interrupções (Hardware)



10

## Interrupções (Software)

- Qualquer forma de interromper o fluxo normal de execução da CPU;
- Instruções privilegiadas (TRAPS);
- Mecanismo utilizado em muitos SCs para requisitar serviços aos dispositivos de E/S;
- Também utilizada para tratar situações anômalas (Exceptions);
  - Endereços inválidos;
  - Falta de páginas;
  - Erros aritméticos (quando suportados em hardware);
  - etc...

11

## Interrupções Precisas e Imprecisas

- Interrupções podem ser classificadas como precisas ou imprecisas, dependendo do que precisa ser feito quando uma interrupção ocorre;
- Para uma interrupção ser precisa, é necessário:
  1. O PC é salvo em um lugar conhecido;
  2. Todas as instruções anteriores àquela apontada pelo PC foram totalmente executadas;
  3. Nenhuma instrução posterior à apontada pelo PC foi executada;
  4. O estado de execução da instrução apontada pelo PC é conhecido;
- Decorrente em especial devido ao Pipelining!

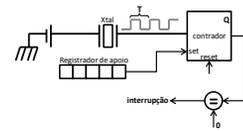
12

## Relógios (Hardware)

- Dispositivos de Hardware que Compõem o sistema Computacional;
  - Usualmente mais de um timer;
  - Usado também para o relógio do sistema;
  - Funciona mesmo com o SC desligado (bateria);
- RTC – Real Time Clock
  - N° de tiques a partir de 12h de 01/01/1979
  - Coordenada Universal de Tempo (UTC)
  - Meio dia de Greenwich

13

## Relógios (Temporizadores)



Controlador de Relógio

14

## Temporizadores e o SO

- Essenciais para o funcionamento de sistemas multiprogramados;
- Evitam que um processo monopolize a CPU;
- Causam interrupções em intervalos bem definidos;
- Usados pelo SO para:
  - controlar o chaveamento de contexto;
  - Limite de tempo de E/S;
  - Chamadas do sistema para controle de tempo;

15

## Armazenando o n° de Tiques

- Quantos bits são necessários para armazenar o tempo do relógio?
- Melhor Pergunta: Quantos tiques são armazenáveis em 32 e 64 bits?
  - 32 bits = 4.294.967.296 tiques
  - 64 bits = 18.446.744.073.709.551.616 tiques
- Considerando um clock de 500MHz  $T = 2\text{ns}$ 
  - 32bits = 8,589.934.592s
  - 64bits  $\approx 36.893.488.147,4\text{s} \approx 614.891.469 \text{ mins} \approx 10.248.191 \text{ horas} \approx 427.007 \text{ dias} \approx 1.169 \text{ anos}$

16

## Relógios (Software)

- Hardware só gera interrupções em intervalos regulares;
- Todo o resto é tratado em software;
- Responsabilidades variam de SO para SO:
  - Manter a hora do dia (tempo real);
  - Controlar o tempo de execução dos processos;
  - Tratar a chamada do sistema <<alarm>> feita pelos processos do usuário;
  - Contabilizar o uso da CPU;
  - Fornecer temporizadores “watch-dog” para partes do sistema computacional;
  - Gerar perfis de execução, realizar monitoramentos e coletar estatísticas.

17

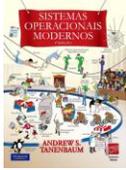
## Temporizadores em Software

- A maioria dos SOs os implementa;
- Forma de expandir o número de temporizadores disponíveis;
- Um temporizador em hardware gera interrupções em intervalos regulares e conhecidos;
- Um processo do SO recebe tais interrupções e em software implementa diferentes temporizadores;
- Quando um temporizador de software deve sinalizar seu tempo, uma interrupção por software é gerada.

18

## Bibliografia

---

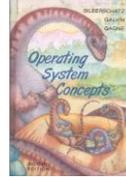


- Ótimo livro texto;
- Referência básica para a disciplina;
- A disciplina é estruturada com base neste e no livro do Silberschatz;

19

## Bibliografia

---



- Ótimo livro texto;
- Referência básica para a disciplina;
- A disciplina é estruturada com base neste e no livro do Tanenbaum;

20