

Concorrência

...A utilização concorrente da UCP deve ser implementada de maneira que, quando um programa pede o uso do processador e depois retorna para continuar sua execução, seu estado deverá ser idêntico ao do momento em que foi interrompido. O programa deverá executar a instrução seguinte à última instrução executada, aparentando ao usuário que nada aconteceu.

Concorrência

...O conceito de concorrência é o princípio básico para o projeto e a implementação dos sistemas operacionais multiprogramáveis.

RELEMBRANDO

Funções básicas de um computador são:

Processamento de dados

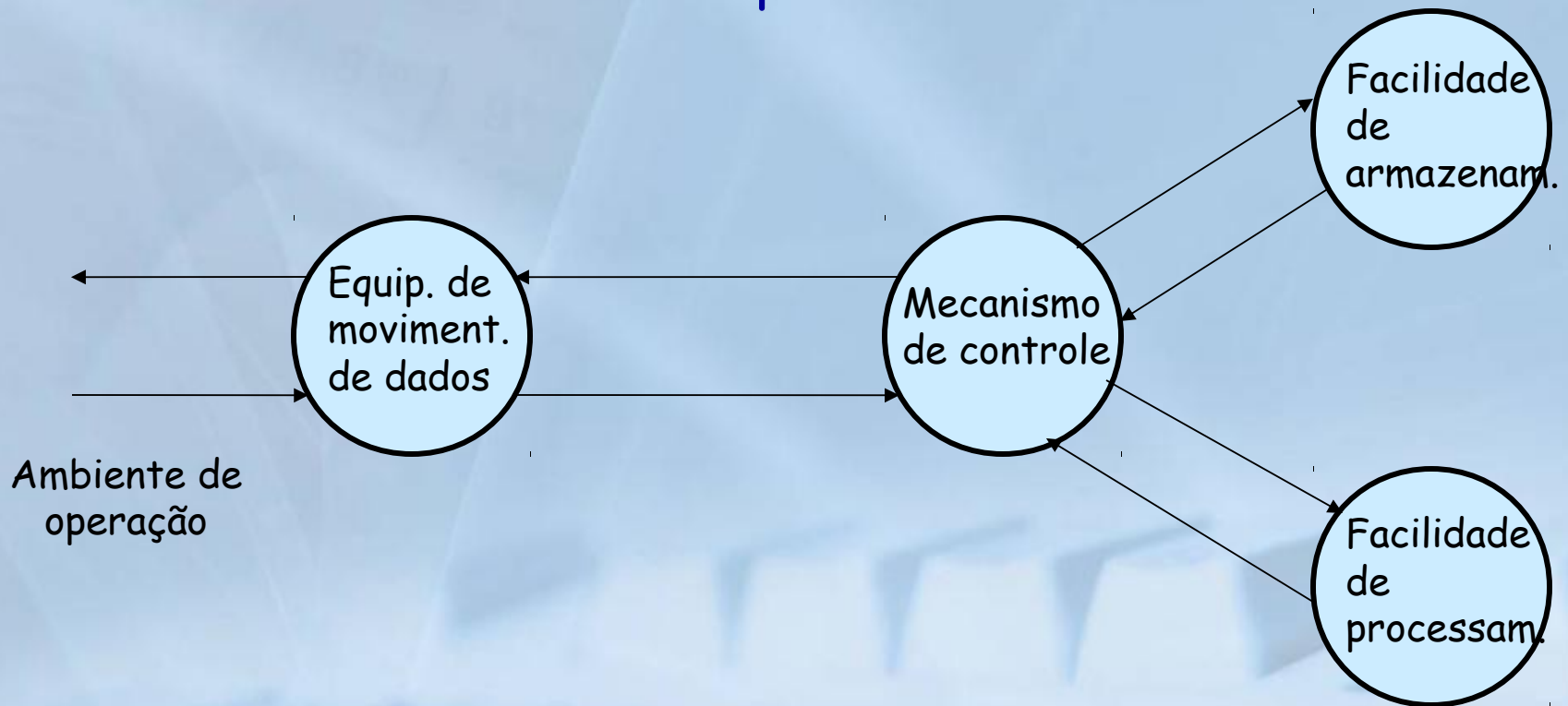
Armazenamento de dados

Movimentação de dados

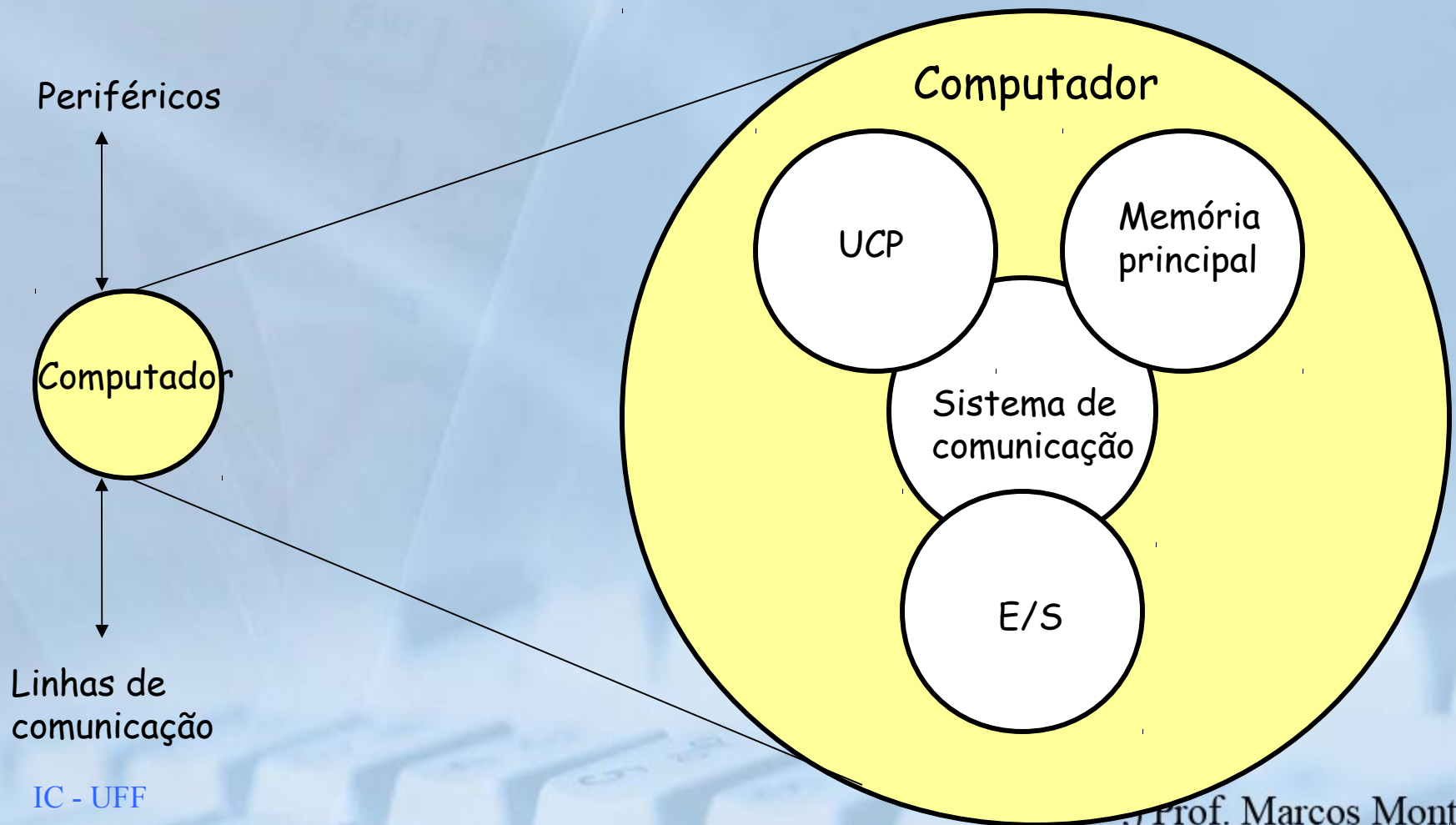
Controle

Visão funcional

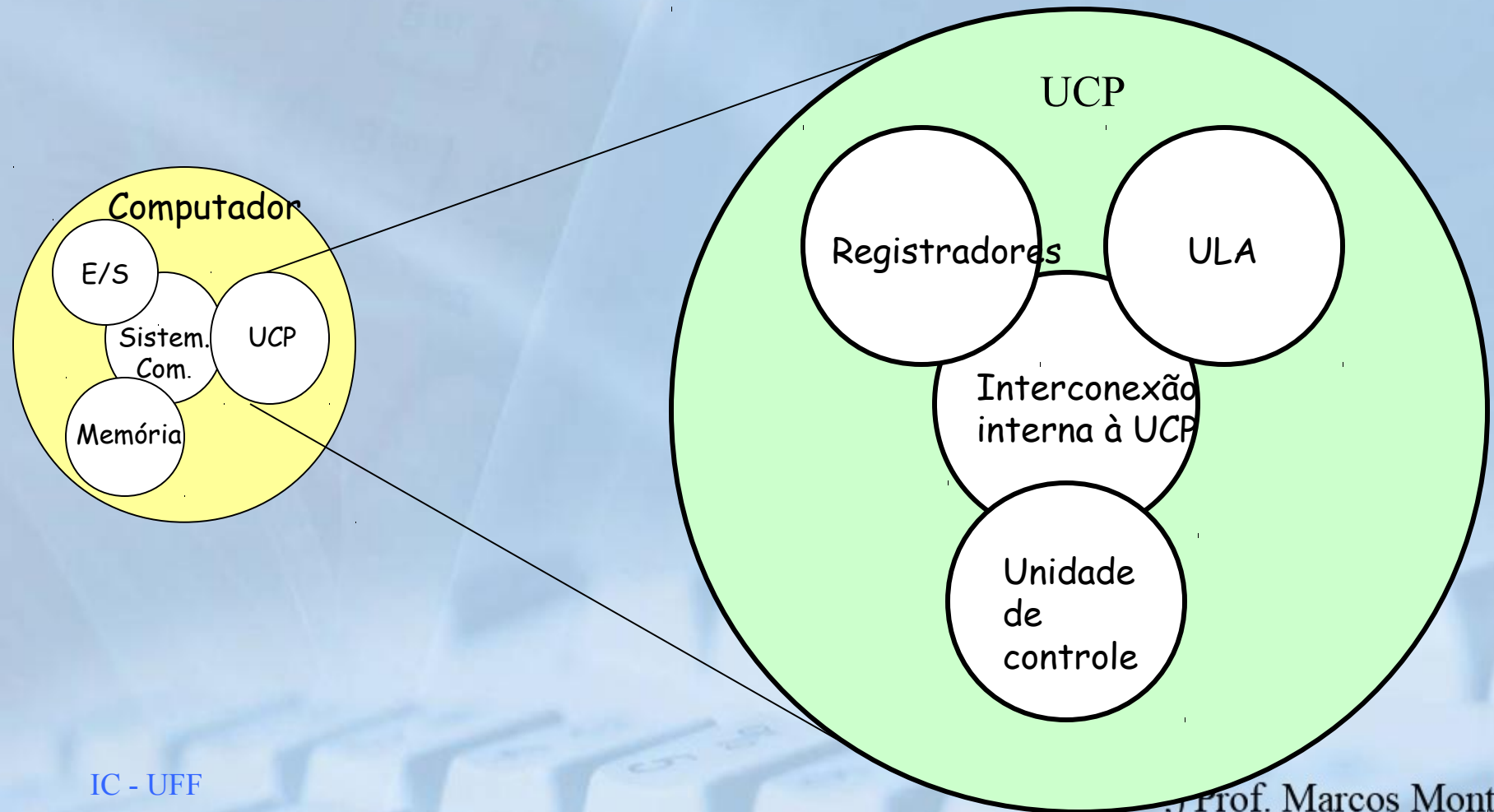
Visão funcional de um computador



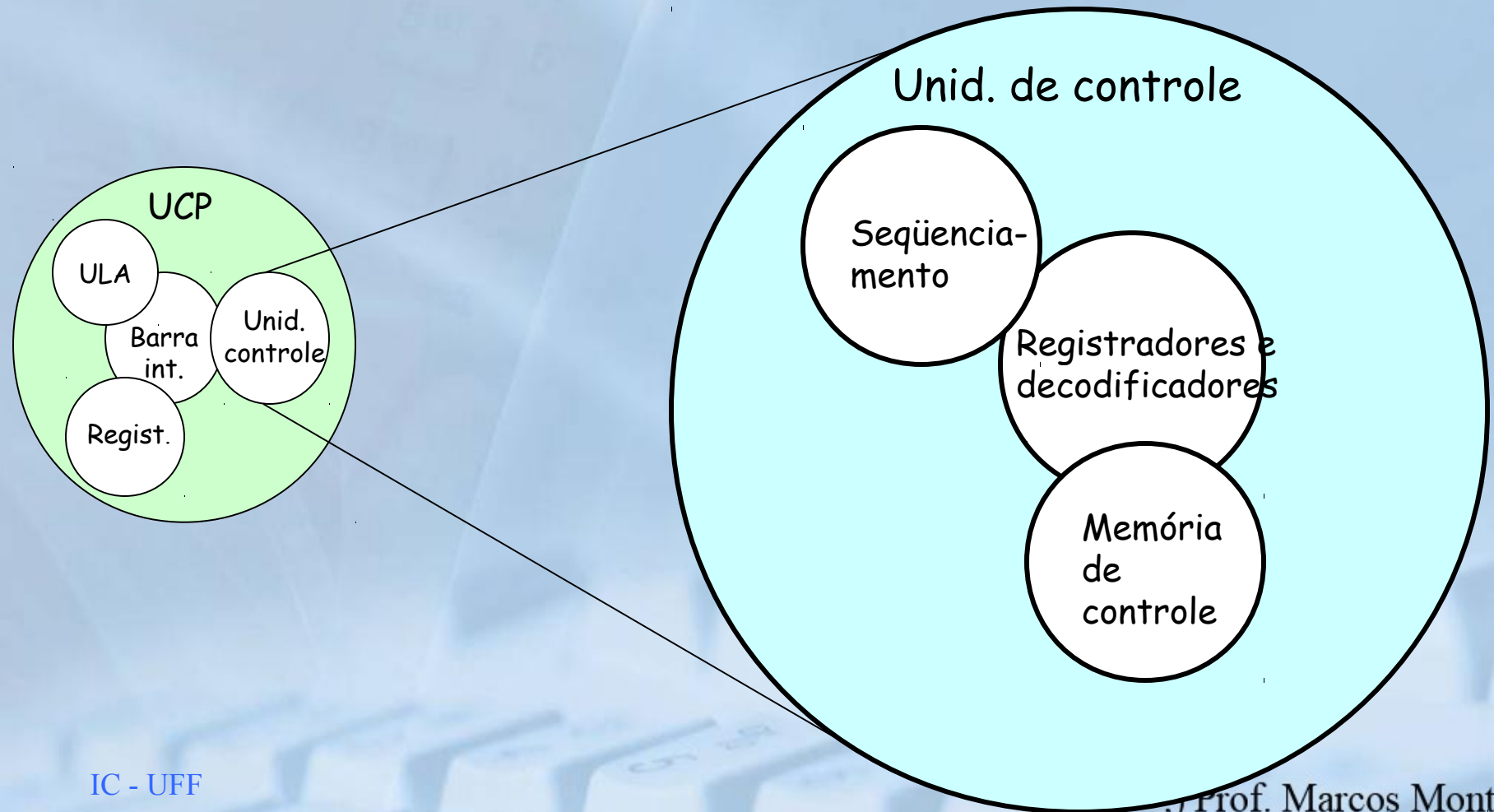
Estrutura: visão macro



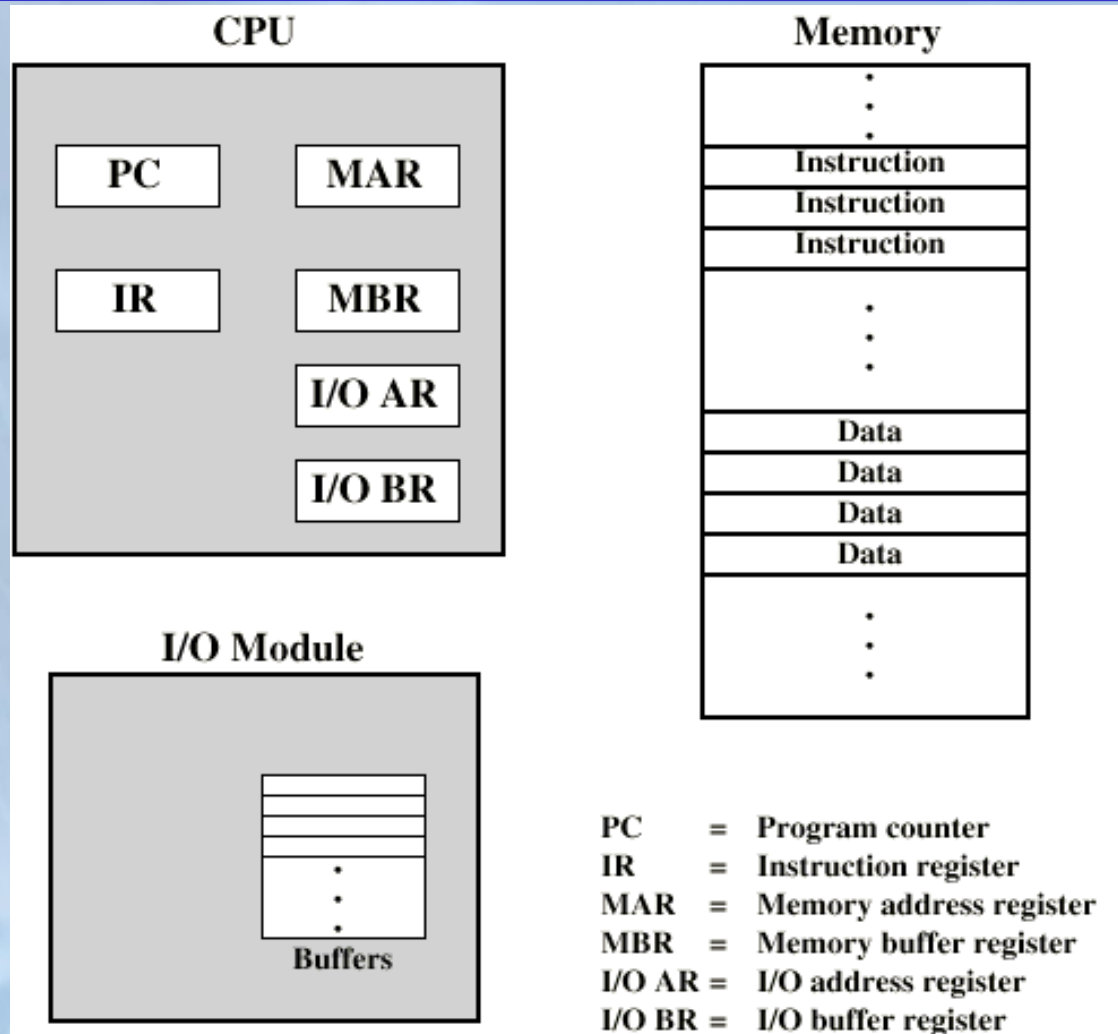
Estrutura: a UCP



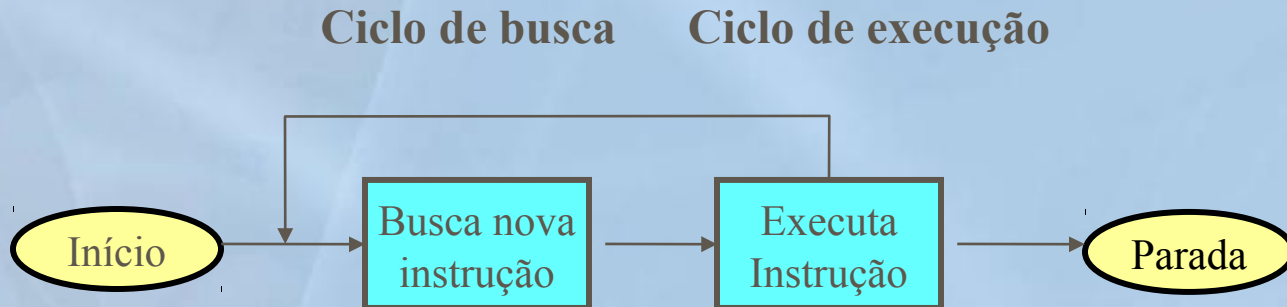
Estrutura: unidade de controle



Computador: visão macro



Ciclo de instrução



Ciclo de busca

PC contém o endereço da próxima instrução

Instrução é buscada e colocada no IR para ser interpretada

PC é incrementado (quase sempre!)

Ciclo de execução

UCP interpreta instrução e executa ação:

UCP memória: transferência de dados

UCP E/S: transferência de dados

processamento de dados: operação lógica ou aritmética

controle: alteração da seqüência de operação

combinação dessas ações

Organização da UCP

Funções da UCP:

buscar instruções

interpretar instruções

buscar dados

processar dados

UCP precisa fazer armazenamento temporário:
registradores

Registradores

Espaço de trabalho temporário

Quantidade e função varia entre processadores

Uma das principais decisões de projeto

Nível superior da hierarquia de memória

Classificação:

vísíveis

restritos

controle

estado

Registradores visíveis

São os que podem ser referenciados através de linguagem de máquina

de uso geral (dados e endereçamento)

de dados (e.g., acumulador)

de endereço: segmento, índice, pilha, ...

códigos de condição (só leitura)

Registradores de controle e estado

visibilidade restrita

essenciais: PC, IR, MAR, MBR

PSW: *Program Status Word*

códigos de condição =

sinal + zero + vai-um + estouro + permissão/inibição de
interrupção + modo supervisor/não

Outros registradores importantes: SP, apontador de
PCB, interrupção vetorizada

Interrupção

Mecanismo pelo qual outros módulos interrompem processamento normal da UCP

Basicamente associado a E/S

Tipos mais comuns de interrupção:

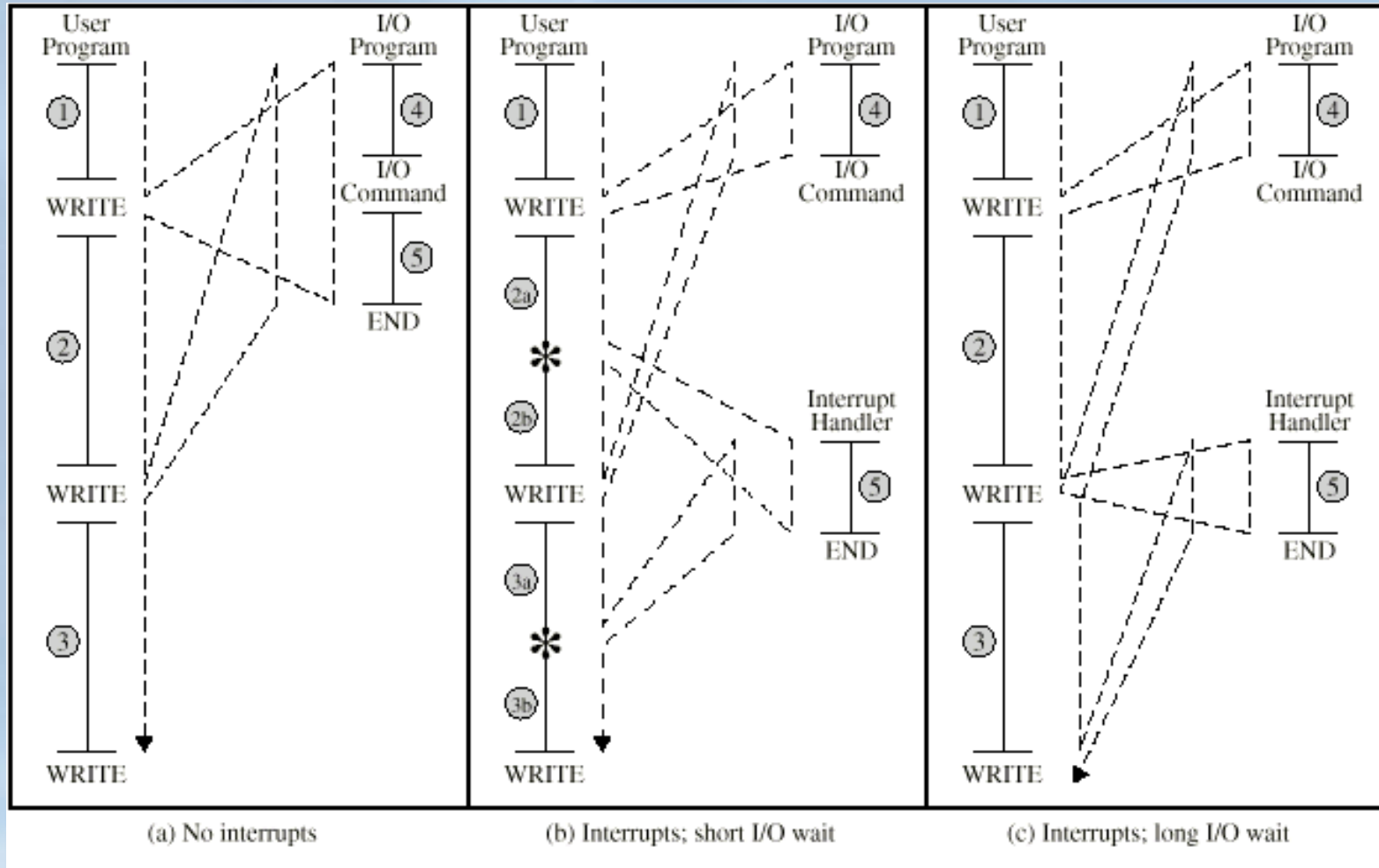
programa (e.g., 0) → *traps*

temporização → escalonamento de processo

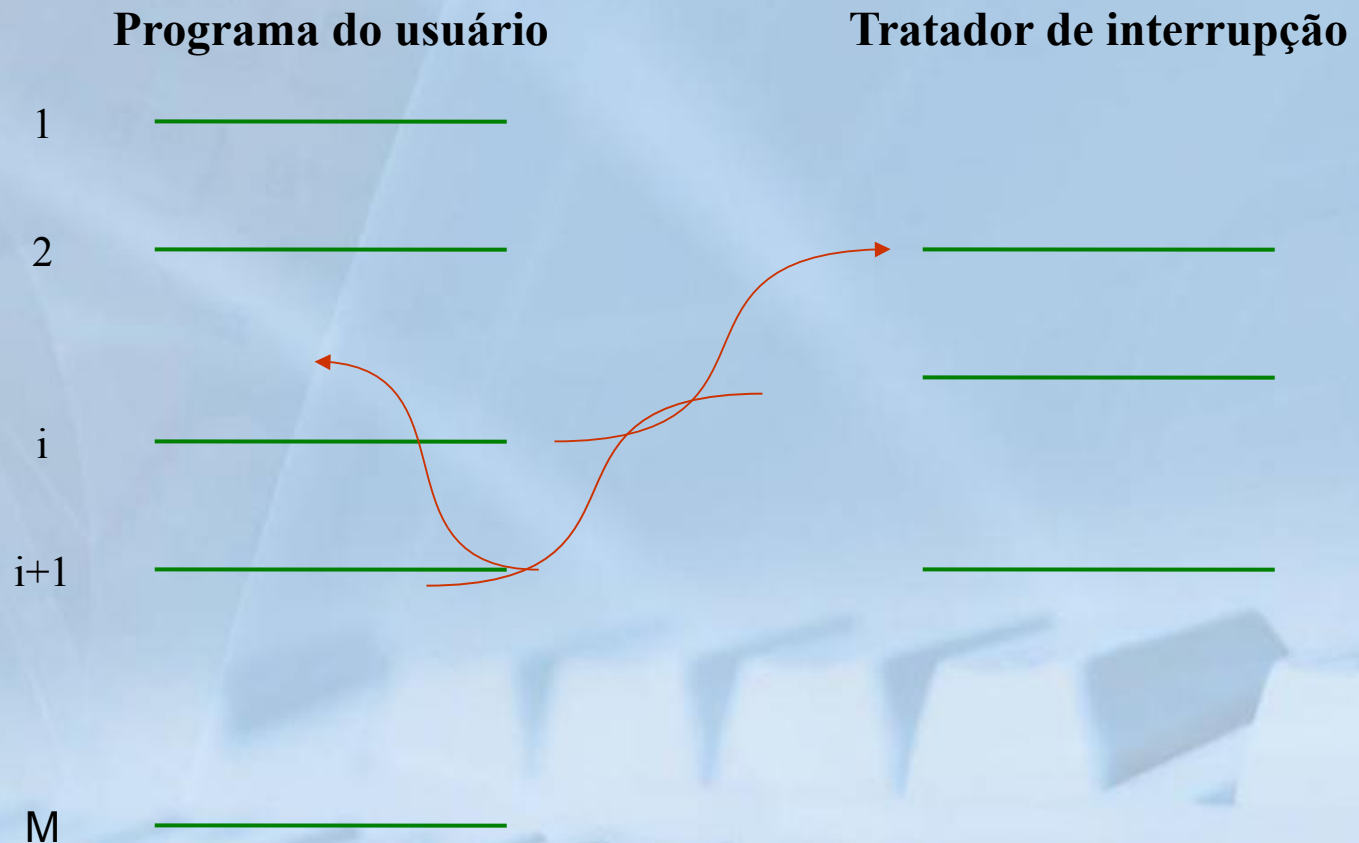
E/S (e.g., fim de escrita em disco)

falha de h/w (e.g., falta de energia)

Fluxo de controle em programa



Transferência de controle



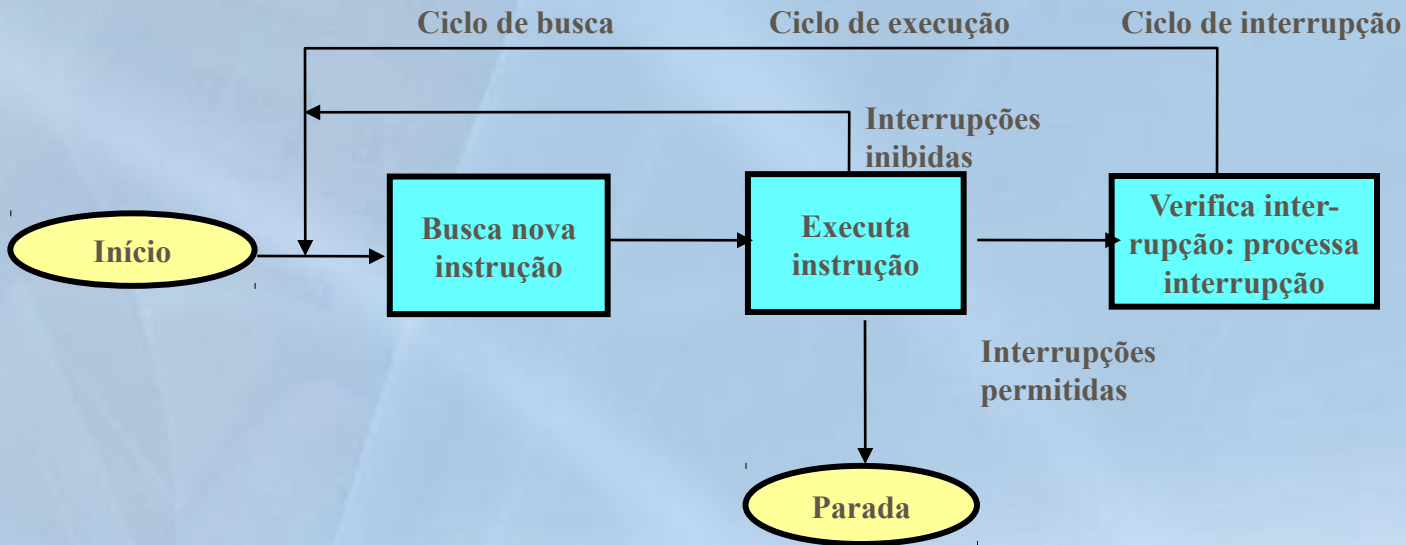
Tratador de interrupção

Programa que determina a natureza da interrupção e que realiza o tratamento adequado

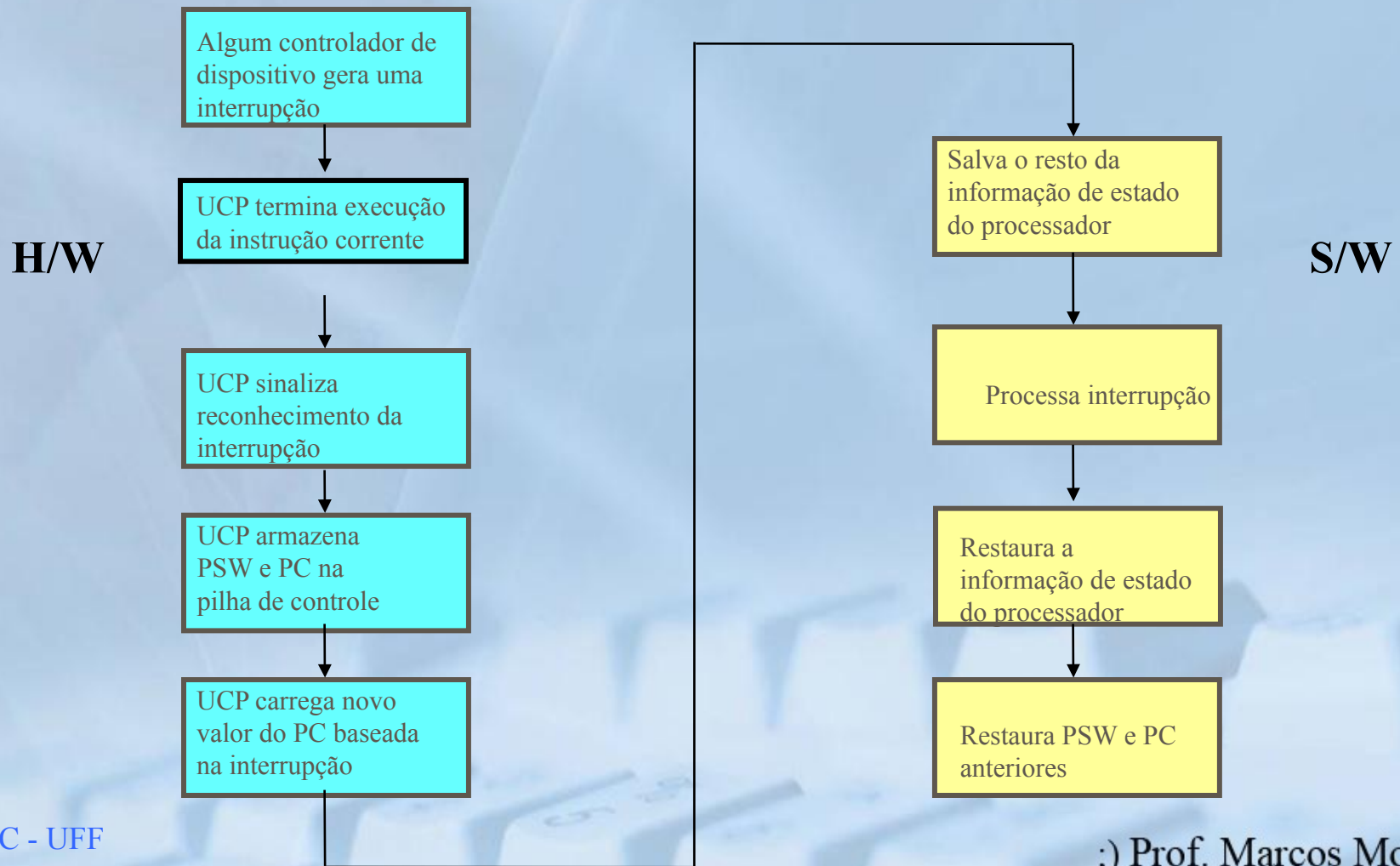
Controle é transferido para este programa após salvamento de algumas informações

É parte do sistema operacional

Ciclo de interrupção



Tratamento da interrupção



Interrupções múltiplas (1)

Desabilita outras interrupções: seqüencial

UCP ignora outras interrupções enquanto processa uma interrupção

interrupções pendentes só são verificadas ao fim do tratamento da interrupção corrente

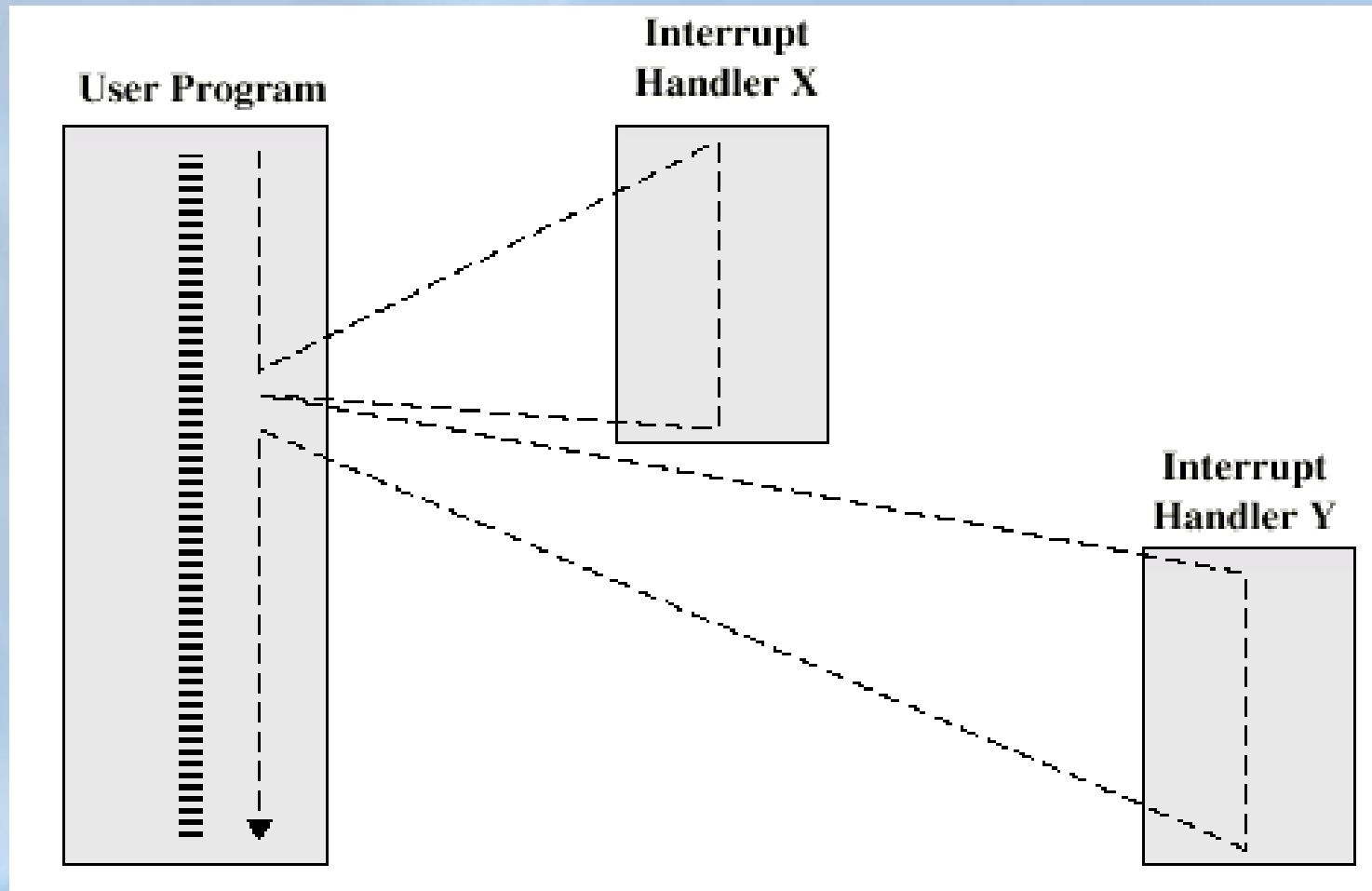
interrupções tratadas na ordem seqüencial de ocorrência

Quando terminar o tratamento, as interrupções são habilitadas.

Vant: simplicidade

Desv.: falta de critério

Tratamento seqüencial



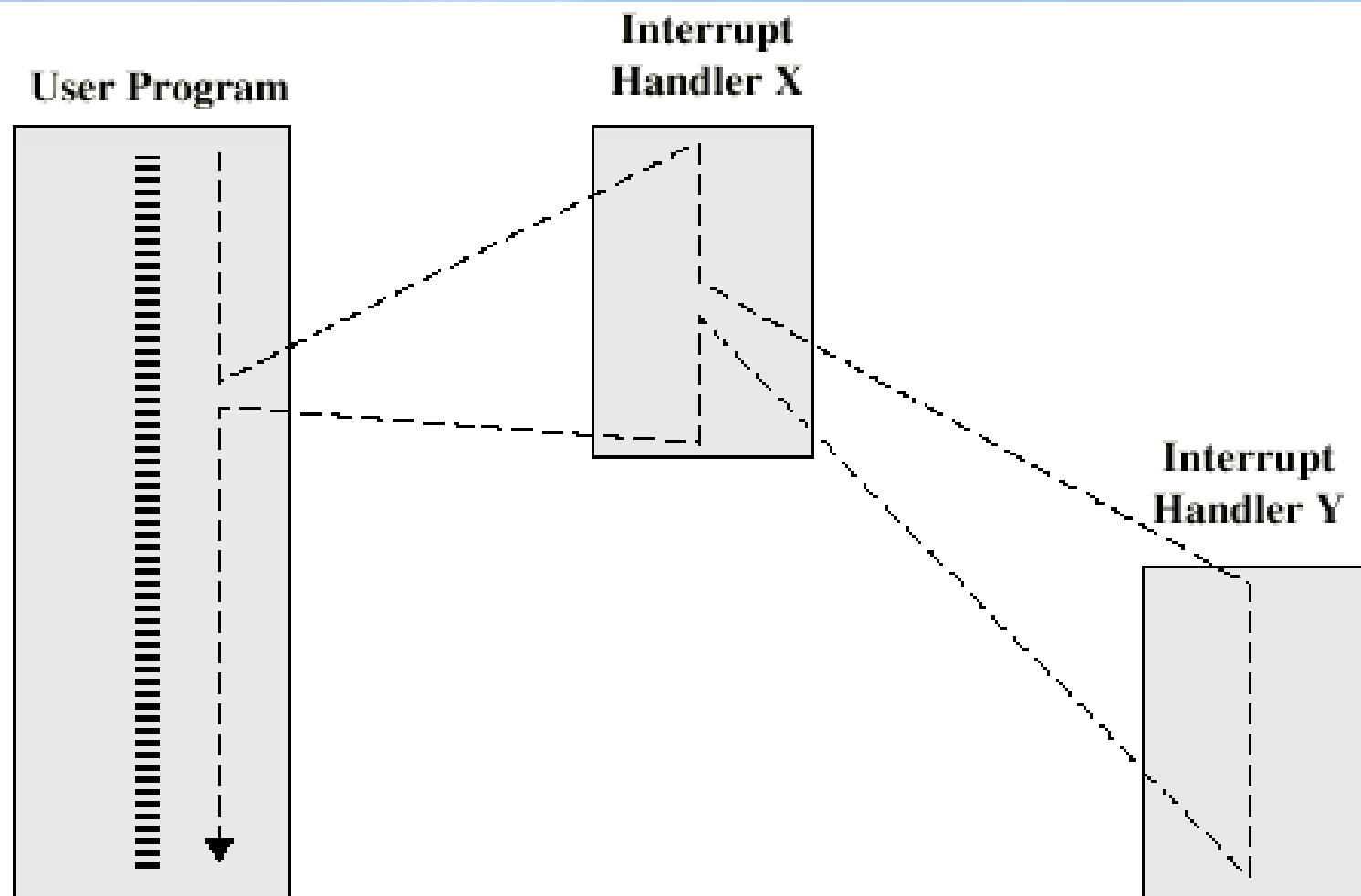
Interrupções múltiplas (2)

Definição de prioridades

interrupções de menor prioridade podem ser interrompidas por interrupções de maior prioridade

quando há o término do tratamento da(s) de maior prioridade, UCP trata a(s) de menor prioridade

Tratamento com prioridades



Tratamento com prioridades

t=0 → instruções sendo executadas

t=10 → interrupção devido impressora

passos de hardware e software

t=15 → rotina de tratamento ainda sendo executada
linha de comunicação interrompe

rotina é interrompida e estado salvo

trata interrupção devido a linha de comunicação

t=25 → termina o tratamento devido a linha de com.
restaura o tratamento devido a impressora

.....

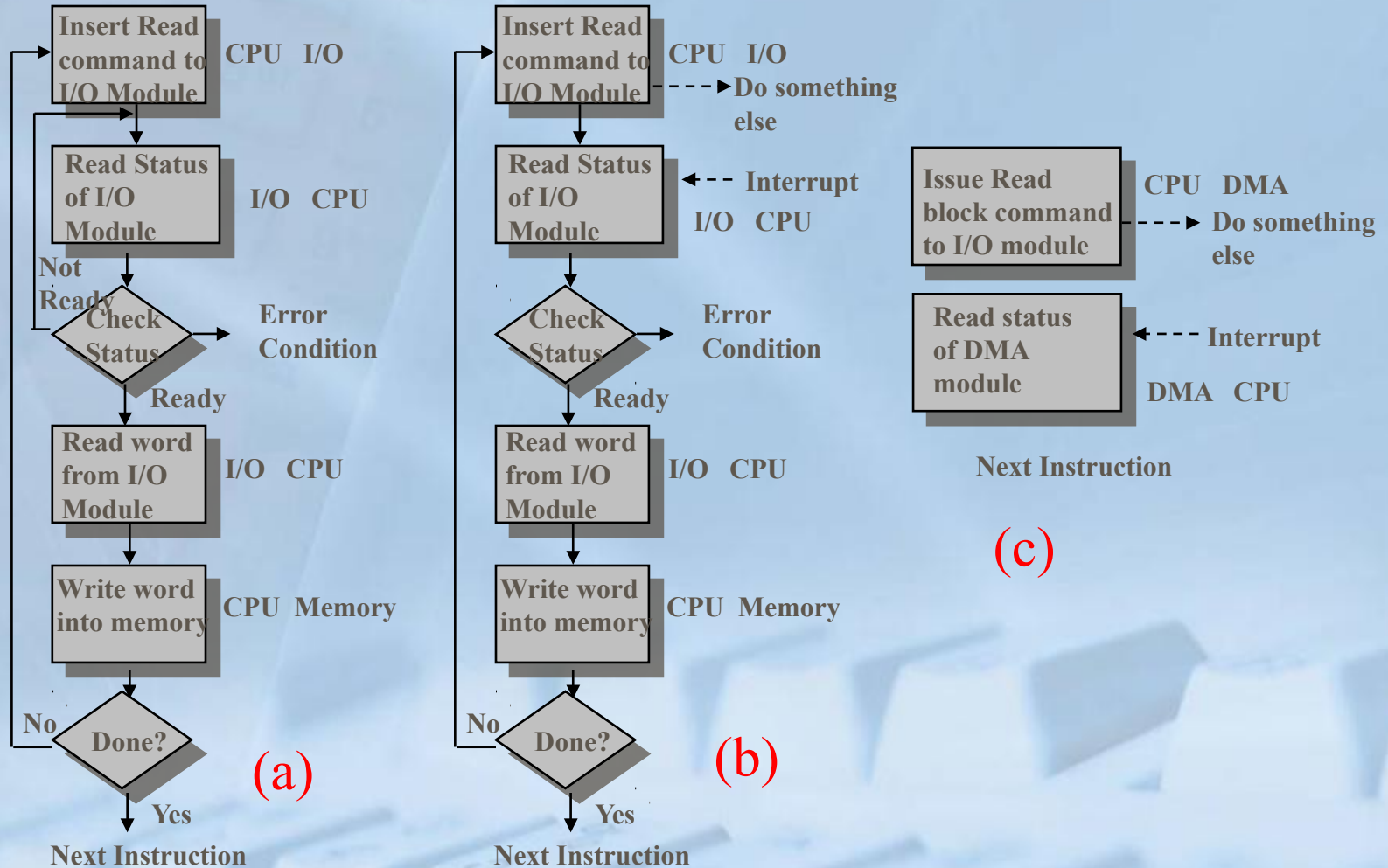
Técnicas de comunicação de E/S

E/S programado: (a)

E/S por interrupção: (b)

E/S por DMA: (c)

Exemplo: leitura de um bloco



Metodos de Interrupção

Vetor de Interrupção

- Contem o endereço inicial de todas as rotinas associadas a cada tipo de evento;

Registrador de Status

- Armazena do tipo de evento ocorrido;

Categories de Interrupções

- **Maskable interrupt (IRQ)**
- **Non-maskable interrupt (NMI)**
- **Inter-processor Interrupt (IPI)**
- **Software Interrupt**
- **Spurious Interrupt: (interrupção falsa)**

Maskable interrupt (IRQ)

...é uma interrupção de hardware que pode ser ignorada por configurar um bit em um registro da máscara de interrupção (IMR) bit-mask.

Non-maskable interrupt (NMI)

...uma interrupção de hardware que carece um bit-mask associado, então isto nunca pode ser ignorado. NMIs são frequentemente usados por timers por exemplo.

Inter-processor Interrupt (IPI)

...é um caso especial que é gerado por um processador para interromper outro processador em um sistema de multiprocessadores.

Software Interrupt

...é uma interrupção gerada dentro de um processador pela execução de uma instrução. Interrupções de software são freqüentemente usadas para implementar chamadas de sistema porque elas implementam uma chamada de subrotina com a mudança de nível da CPU.

Spurious Interrupt: (interrupção falsa)

...é uma interrupção de hardware que é indesejável. Elas são tipicamente geradas por condições do sistema, tais como **interferência elétrica** em uma linha de interrupção ou através de um hardware projetado incorretamente.

Excessão

- >> É semelhante à interrupção, sendo a principal diferença o motivo pelo qual é gerado.**
- >> A exceção é resultado direto da execução de uma instrução do próprio programa.**
- >> É gerada por eventos previsíveis, um único de cada vez.**

Sempre que uma exceção é encontrada, o fluxo do programa é desviado para uma rotina de tratamento de exceção, que muitas vezes, pode ser escrita pelo próprio programador.

Interrupção X Excessão

...INTERRUPÇÕES SÃO GERADAS POR EVENTOS ASSÍNCRONOS, ENQUANTO EXCEÇÕES SÃO GERADAS POR EVENTOS SÍNCRONOS.

OPERAÇÕES DE ENTRADA E SAÍDA

Operações de I/O

Nos primeiros computadores, havia grande dependência entre os dispositivos de entrada e saída e os processadores, visto que havia uma série de instruções especiais para controlar esses periféricos, chamadas de instruções de entrada e saída. Essas instruções carregavam informações específicas de cada periférico.

O controlador permitiu ao processador agir de forma independente dos dispositivos de entrada e saída. Com esse novo elemento, o processador não mais se comunicava com os dispositivos, mas sim, com seus controladores, eliminando a necessidade de especificar detalhes dos periféricos.

...Com o uso de controladores, duas eram as maneiras básicas pelas quais o processador gerenciava as operações de entrada e saída:

controle por programa

e

controle por interrupção.

Controle por programa

O processador se sincroniza com o periférico ao início da transferência de dados e, após iniciada a transferência, o sistema ficava permanentemente testando o dispositivo para saber quando a operação chegaria ao seu fim. Isso resultava em desperdício de tempo da UCP.

Como evolução a este modelo, surge a idéia de que durante a transferência dos dados o processador pudesse permanecer livre para realizar outras tarefas. Assim, **dentro de determinados intervalos de tempo, o sistema operacional deveria testar cada dispositivo para saber o término da operação de entrada e saída. Essa operação inseriu certo grau de paralelismo e permitiu o surgimento dos primeiros sistemas operacionais multiprogramáveis.**

Como desvantagem deste modelo, caso hajam muitos periféricos, o sistema é interrompido freqüentemente para testar os diversos periféricos.

Controle por Interrupção

Com o surgimento de mecanismos de interrupção, as entradas e saídas puderam ser realizadas de forma muito mais eficiente. Em vez de o sistema verificar periodicamente o estado de uma operação pendente, o próprio controlador interrompia o processador para avisar o término da operação.

O controlador, por exemplo, ao receber um sinal de leitura, fica encarregado de ler os blocos do disco e armazená-los na memória ou em registradores próprios. Em seguida, o controlador sinaliza uma interrupção ao processador.

Quando o processador atende à interrupção, a rotina responsável pelo tratamento transfere os dados dos registradores do controlador para a memória principal. Ao término da operação, o processador pode voltar à execução do programa interrompido e o controlador fica novamente disponível para outra operação.

A operação de entrada e saída controlada por interrupção é muito mais eficiente do que a controlada por programa, já que elimina a necessidade de o processador esperar pelo término da operação, além de permitir que várias operações de entrada e saída sejam executadas simultaneamente.

Entretanto, a transferência de um grande número de dados exige muitas intervenções do processador, reduzindo a sua eficiência.

A solução para isso foi a técnica DMA de transferência de dados

DMA – Direct Memory Access

Permite que um bloco de dados seja transferido entre memória principal e dispositivo de entrada e saída, sem a intervenção do processador, exceto no início e no final da transferência.

Quando o sistema deseja ler ou gravar um bloco de dados, o processador informa ao controlador sua localização, o dispositivo de E/S, a posição inicial de memória onde dados serão lidos ou gravados e o tamanho do bloco.

A área em memória utilizada pelo DMA é chamada de buffer de entrada e saída.

No DMA, o controlador deve assumir momentaneamente o controle do barramento, visto que esse pode ser usado apenas por um dispositivo.

Canal de entrada e saída

Introduzido pela IBM, é um processador com capacidade de executar programas de entrada e saída, permitindo o controle total sobre essa operação.

As instruções de entrada e saída são armazenadas na memória principal pelo processador, porém o canal é responsável pela sua execução. Assim, o processador realiza uma operação de entrada e saída, instruindo o canal para executar um programa localizado na memória. Este programa especifica os dispositivos para transferência, buffers e ações a serem tomadas em casos de erros. O canal de E/S realiza transferência e, ao final, gera uma interrupção, avisando o término da operação.

O canal atua como um elo entre o processador principal e os controladores.

Processador de Entrada e Saída

´ É uma evolução do canal de entrada e saída.

Possui a sua própria memória, eliminando a necessidade de os programas serem carregados para a memória principal.

As várias funções de entrada e saída são controladas com mínima intervenção da UCP.

BUFFERING

Consiste na utilização de uma área de memória principal, denominada buffer, para a transferência de dados entre dispositivos de entrada e saída e a memória que:

Libera o dispositivo de entrada para capturar dados logo após que um dado seja transferido inicialmente para o buffer.

O processador manipula os dados que estão armazenados em buffer.

Visa a diminuir a disparidade de velocidade existente entre processador e os dispositivos de entrada e saída.

Visa a manter processador e dispositivos de entrada e saída ocupados a maior parte do tempo.

Assim, no buffer de entrada se localizam arquivos lidos mas ainda não processados e no buffer de saída, arquivos processados mas ainda não gravados.

SPOOLING

Simultaneous peripheral operation on-line

Foi introduzida ao final dos anos 50 para aumentar o grau de concorrência e a eficiência dos sistemas operacionais.

A técnica de spooling foi a base dos sistemas batch.

Inicialmente, o spooling obrigava o processamento estritamente seqüencial, ou seja, o primeiro job a ser gravado na fita seria o primeiro a ser processado.

Com o surgimento de discos e dispositivos de acesso aleatório, tornou-se o spooling mais eficiente, possibilitando o processamento não seqüencial de Jobs (por exemplo, Jobs menores podem ser executados antes dos maiores).

Atualmente, assemelha-se muito à técnica de buffer.

SPOOLING

É utilizada nos sistemas operacionais atuais para o gerenciamento de impressão. Quando um comando de impressão é executado, as informações que serão impressas são antes gravadas em um arquivo em disco, conhecido como *arquivo de spool*, liberando imediatamente o programa para

outras atividades. Posteriormente, o sistema operacional encarrega-se de direcionar o conteúdo para a impressora.

O uso do spooling impede que um programa faça uso exclusivo da impressora. O sistema operacional é o responsável por gerenciar a seqüência de impressões solicitadas pelos programas, seguindo critérios que garantam a segurança e o uso eficiente das impressoras.

Reentrância

É relativa ao uso de um mesmo aplicativo por vários usuários de um sistema, ao mesmo tempo.

Se para cada usuário que usasse estes aplicativos, o código executável fosse recarregado na memória, haveria diversas cópias de um mesmo programa na memória principal, o que ocasionaria um desperdício de espaço.

A reentrância é a capacidade de um código executável (código reentrante) ser compartilhado por diversos usuários, exigindo apenas uma cópia do programa na memória, sendo que cada usuário o executa em um ponto diferente, manipulando dados próprios e exclusivos.

Proporcionam uma utilização mais eficiente da memória principal e aumento no desempenho do sistema.

Proteção do Sistema

Sistemas multiprogramáveis são mais eficientes e também mais complexos, já que alguns problemas de proteção decorrem deste tipo de implementação.

Compartilhamento de recursos deve gerar maior preocupação em garantir a confiabilidade e a integridade dos programas e dados dos usuários, além do próprio sistema operacional.