



Sistemas Operacionais

PUC Minas

UNIDADE 1: Estrutura de Sistemas de Computação

- Sumário:
- Operações
 - Interrupções
 - Estrutura de I/O
 - Métodos de de I/O
 - Acesso Direto a Memória (DMA)
 - Estrutura de Armazenamento
 - Proteção de Hardware:
 - Operação em Modo Dual
 - Proteção de I/O
 - Proteção de Memória
 - Hardware de Proteção
 - Proteção de CPU
 - Arquitetura Geral do Sistema





PUC Minas

INTRODUÇÃO

Estrutura de Sistemas de Computação

Um sistema de computação moderno

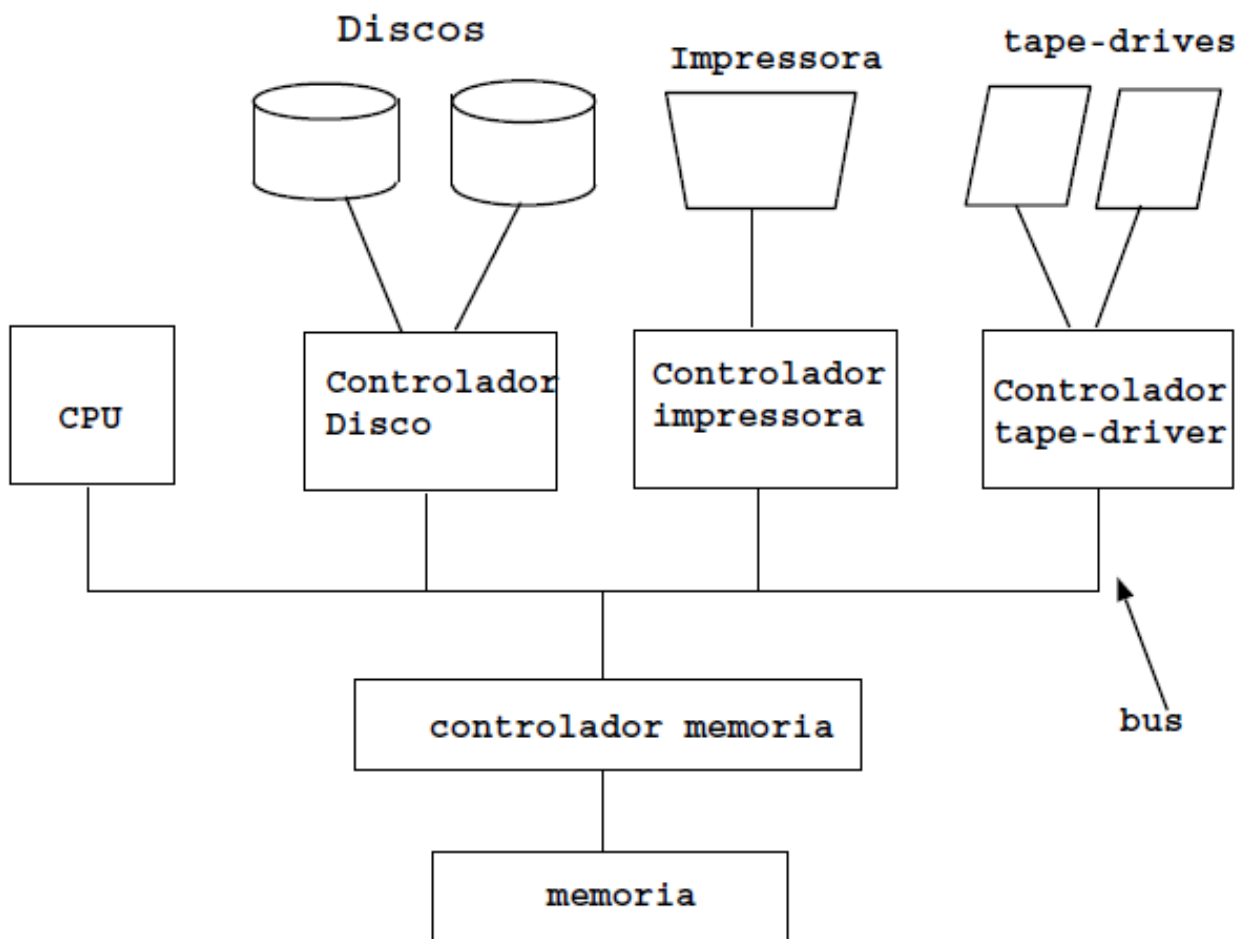


Figura: Arquitetura de um Computador



Operações dos Sistemas Computacionais

Operações dos Sistemas Computacional

- Dispositivos de I/O e a CPU podem executar de forma concorrente
- Cada controladora de dispositivo cuida de um tipo específico de dispositivo
- Cada controladora de dispositivo tem um *buffer* local
- A CPU transfere dados entre os *buffers* locais e a RAM
- O I/O se dá entre o dispositivo e o *buffer* local
- A controladora informa a CPU que terminou sua operação, através de uma **interrupção**



Interrupções

- Os SO.s modernos são baseados em interrupções
- Interrupções transferem o controle para a rotina de serviço da interrupção, através do vetor de interrupções, o qual contém os endereços de todas as rotinas de serviço. Antes da execução da rotina, o endereço da instrução interrompida deve ser salva.
- Segmentos de código separados determinam qual ação deve ser executada para cada interrupção



Interrupções

- O SO preserva o estado da CPU, armazenando os valores dos **registradores** e o contador de programa (***program counter***)
- Outras interrupções ficam desabilitadas até que a interrupção seja processada
- Uma exceção ou ***trap*** é uma interrupção gerada por *software* que sinaliza um erro ou requisição do usuário

Estrutura de I/O

Síncrona: Após o início da operação de I/O, o controle retorna ao programa do usuário apenas após o término da operação

A instrução de espera (***wait***) deixa a CPU inativa até a próxima interrupção

No máximo um pedido de I/O fica pendente em determinado momento. Não há processamento simultâneo de I/O

Assíncrona: Após o início da operação de I/O, o controle retorna ao programa do usuário mesmo antes do término da operação

System call – Uma requisição ao SO será necessária, caso o programa do usuário deseje esperar pela conclusão da operação de I/O

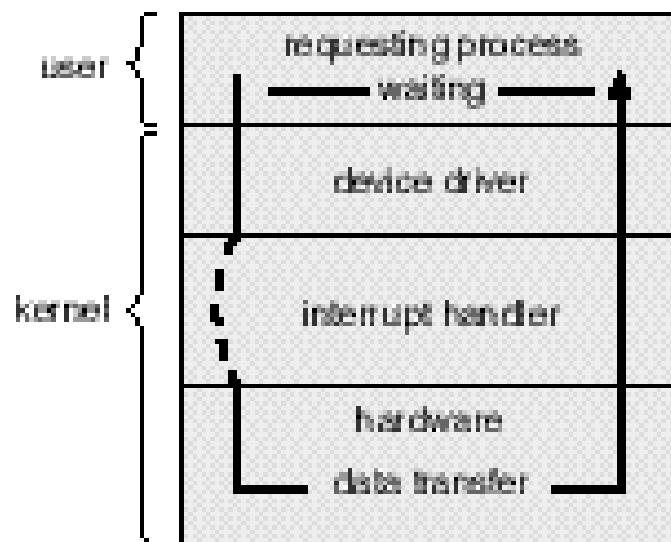
Tabela de status de dispositivos contém entrada para cada dispositivo de I/O, indicando seu tipo, endereço e estado.

O SO indexa a tabela para determinar o status do dispositivo e incluir interrupções (fila de espera).



Dois métodos de I/O

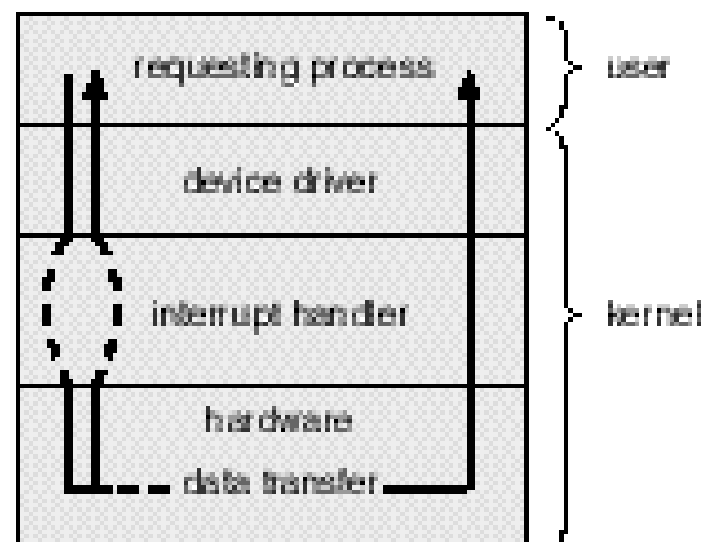
Synchronous



time →

(a)

Asynchronous



time →

(b)

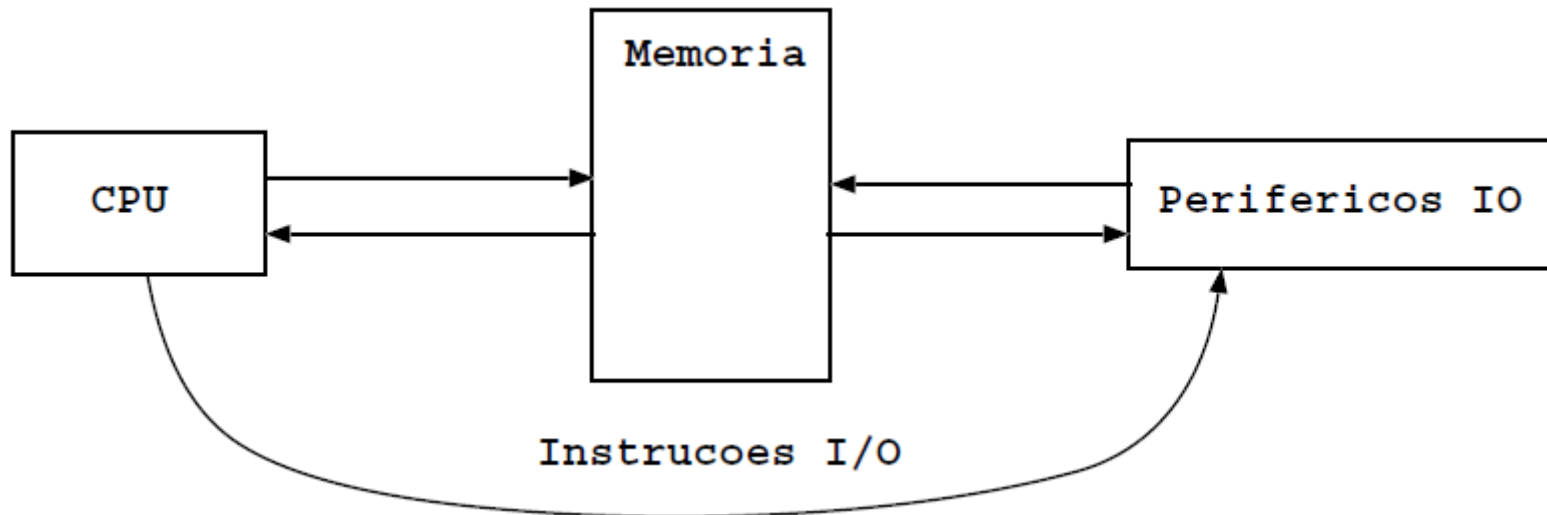


Acesso Direto a Memória (DMA)

Usado para permitir a dispositivos de I/O de alta velocidade transmitirem informação em velocidade comparável à da memória

Controladora do dispositivo transfere blocos de dados do *buffer* diretamente à memória principal, sem a intervenção da CPU

Apenas uma interrupção é gerada por bloco, ao invés de uma interrupção por byte



Memória Principal – única grande área de memória acessada diretamente pela CPU

Memória Secundária – extensão da memória principal, não volátil

Sistemas de Armazenamento são organizados hierarquicamente.

Velocidade

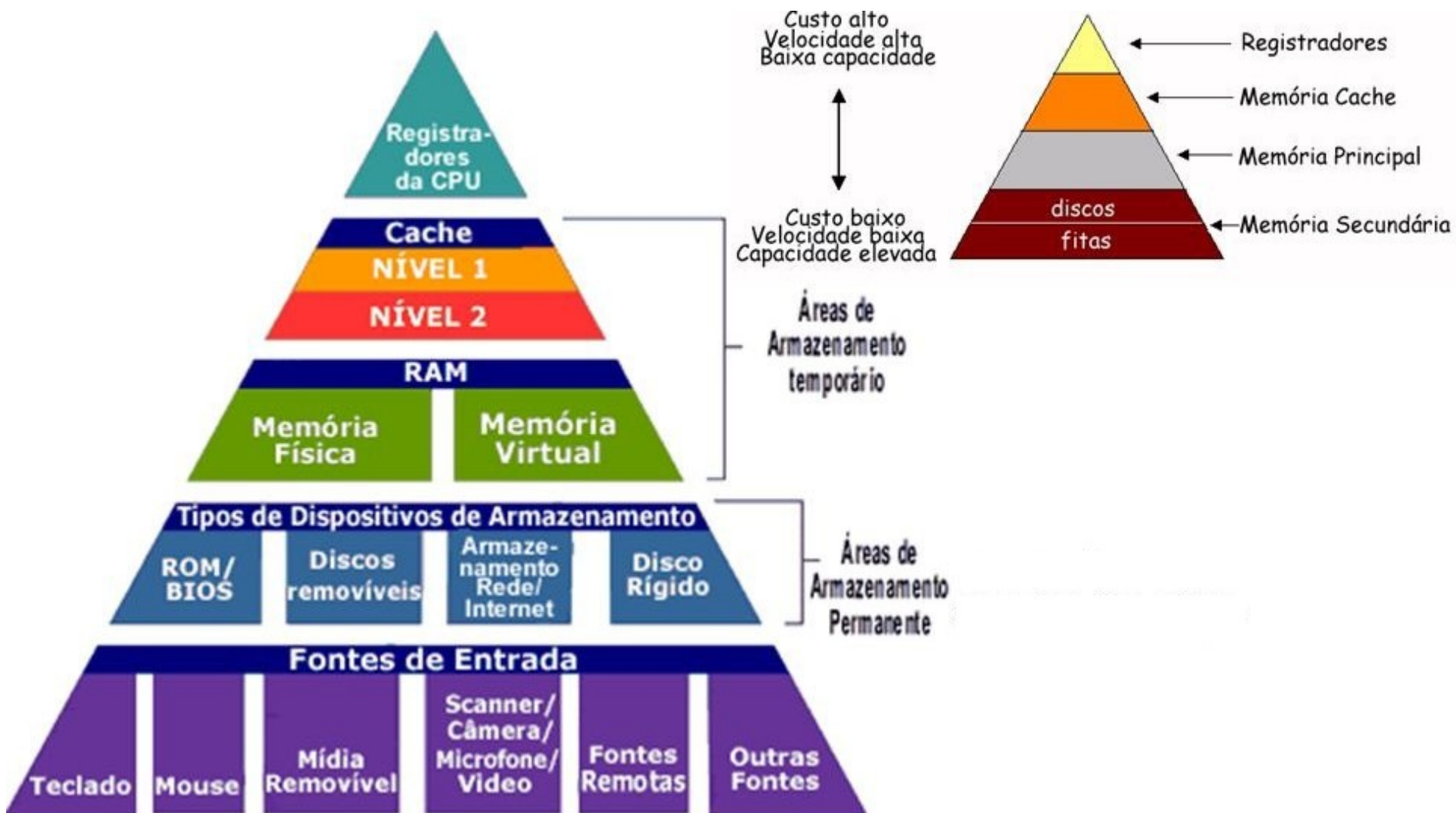
Custo

Volatilidade

Caching – cópia de informação em diferentes níveis de memória

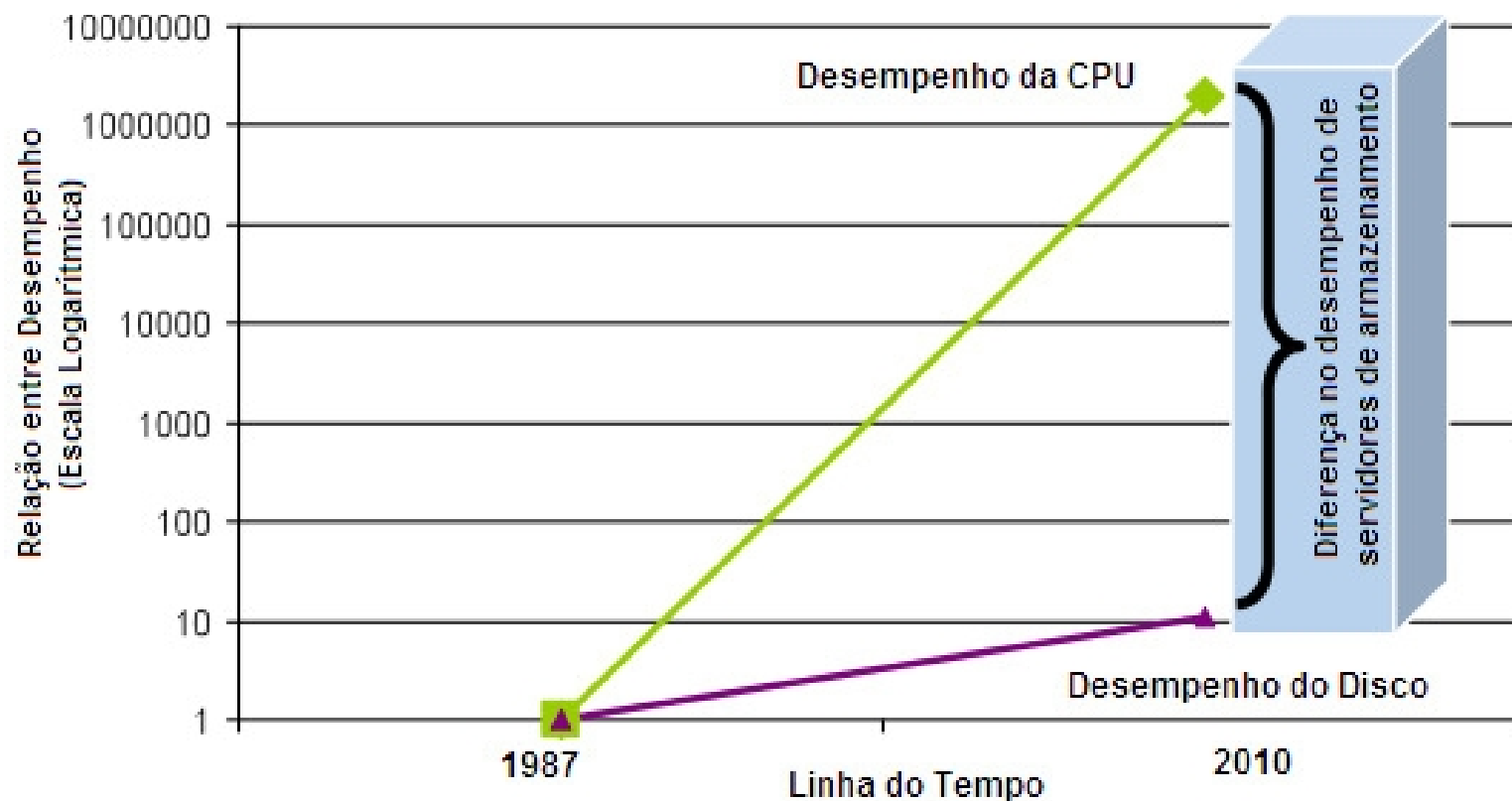


Pirâmide: Custo, Velocidade e Capacidade





Melhoria do desempenho entre CPUs e HDs



Fonte: Adaptada de Violin Memory Inc. (2010)



Velocidade de *Hardware*

Item	Tempo	Escala do Tempo Humano (100 milhões vezes + devagar)
ciclo CPU	10 ns (100MHz)	1 segundo
acesso a cache	30 ns	3 segundos
Acesso a memória	60 ns	6 segundos
Troca contexto	10,000 ns (100 μ s)	166 minutos
Acesso a disco	10,000,000 ns (10 ms)	11 dias

Fonte: Adaptado de UCUP/DCC



Proteção de Hardware

O compartilhamento de recursos requer que o SO garanta que um programa não prejudique a execução dos demais

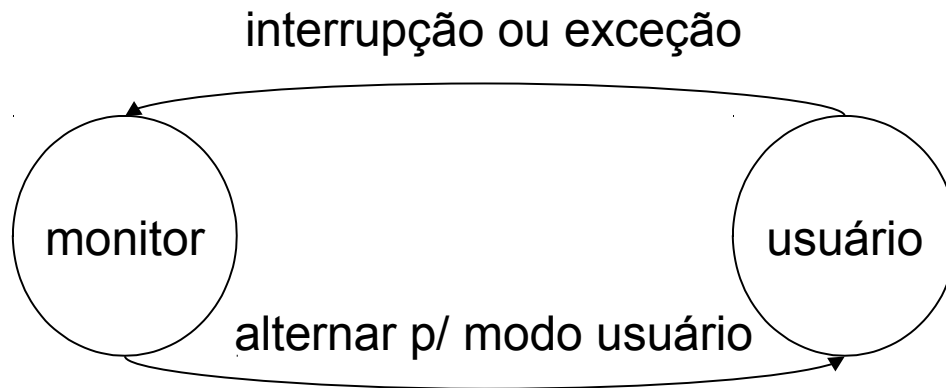
Um ***bit-mode*** acrescentado ao *hardware* permite diferenciar dois modos de operação:

1. ***Modo Usuário*** – execução a favor do usuário
2. ***Modo Monitor*** (*Modo supervisor ou Modo Sistema*) – execução a favor do SO.

Operação em Modo Dual

Bit de Modo indica o modo corrente: monitor (0) ou usuário (1)

Quando uma interrupção ou exceção acontece, o hardware muda para o modo monitor (***modo kernel***)



Instruções *privilegiadas* só podem ser executadas em modo monitor

Todas as instruções de I/O são instruções privilegiadas

Devem garantir que um programa usuário não tenha controle do computador em modo monitor. Operações de I/O devem ser solicitadas ao SO.

Ou seja, os programas aplicativos não acessam diretamente um periférico.

Proteção de Memória

A proteção de memória deve estar garantida pelo menos para o vetor de interrupções e suas rotinas

A proteção pode ser feita com o auxílio de 2 registradores que determinam a faixa legal de memória que pode ser acessada pelo programa usuário (espaço de endereçamento):

Registrador base (RB) – menor endereço físico legal

Registrador de limite (RL) – tamanho da faixa legal (variação)

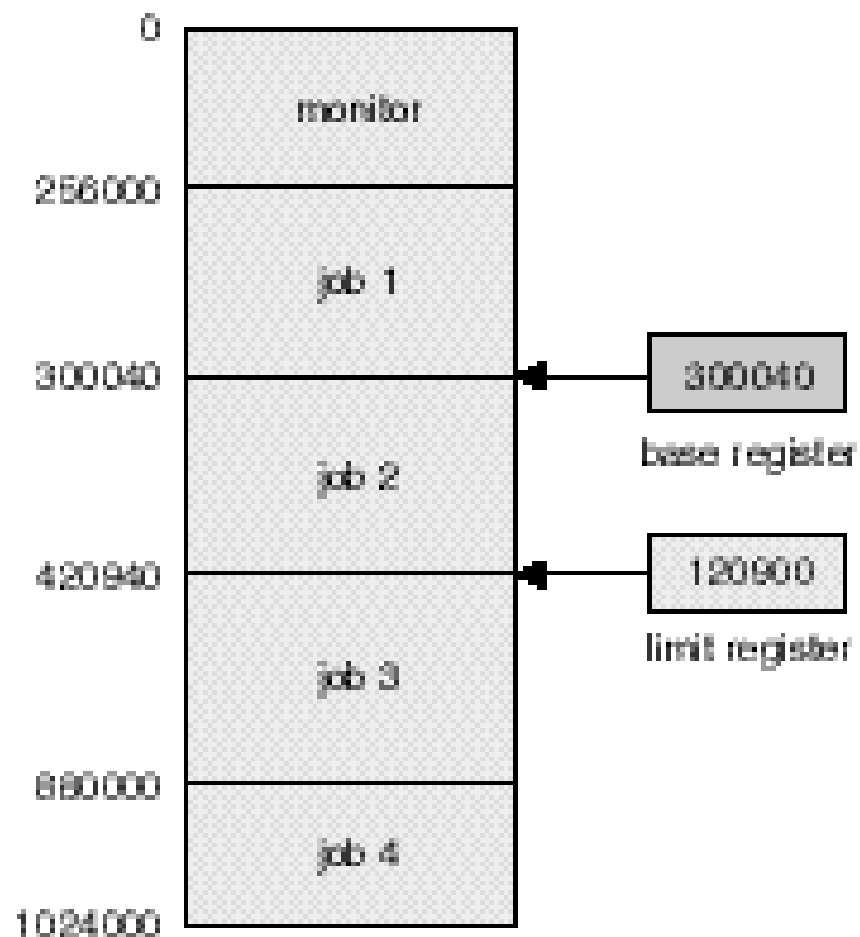
A memória fora do espaço de endereçamento fica protegida. Assim temos que acessos a endereços:

$$x \text{ tal que } x < RB \text{ ou } x > (RB + RL)$$

geram uma armadilha (**trap**) conhecida como **segmentation-fault**

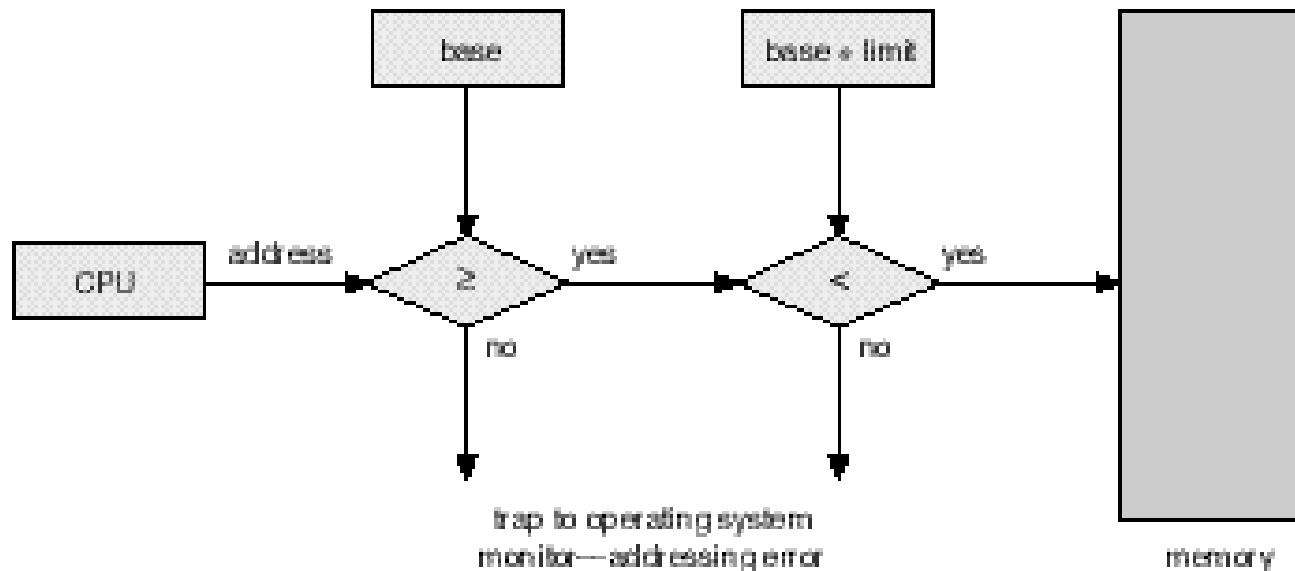


Definição do espaço de endereçamento lógico





Hardware de proteção



- Quando executando em modo monitor, o SO. tem acesso irrestrito à memória
- As instruções de *load* para os registradores de base e limite são instruções privilegiadas

Proteção de CPU

É necessário evitar que caso um programa do usuário entre em **deadlock** (ciclo infinito) o S.O. perca controle e fique inoperante.

Timer (*temporizador*) – interrompe o computador após um período de tempo específico, para garantir o controle do S.O.

O *Timer* é decrementado a cada pulso de clock
Quando o timer atinge o valor 0, uma interrupção ocorre

O Timer é também utilizado para implementar o compartilhamento da CPU (***Time-sharing***).

Carregamento de **Timer** é uma instrução privilegiada (***modo-kernel***).



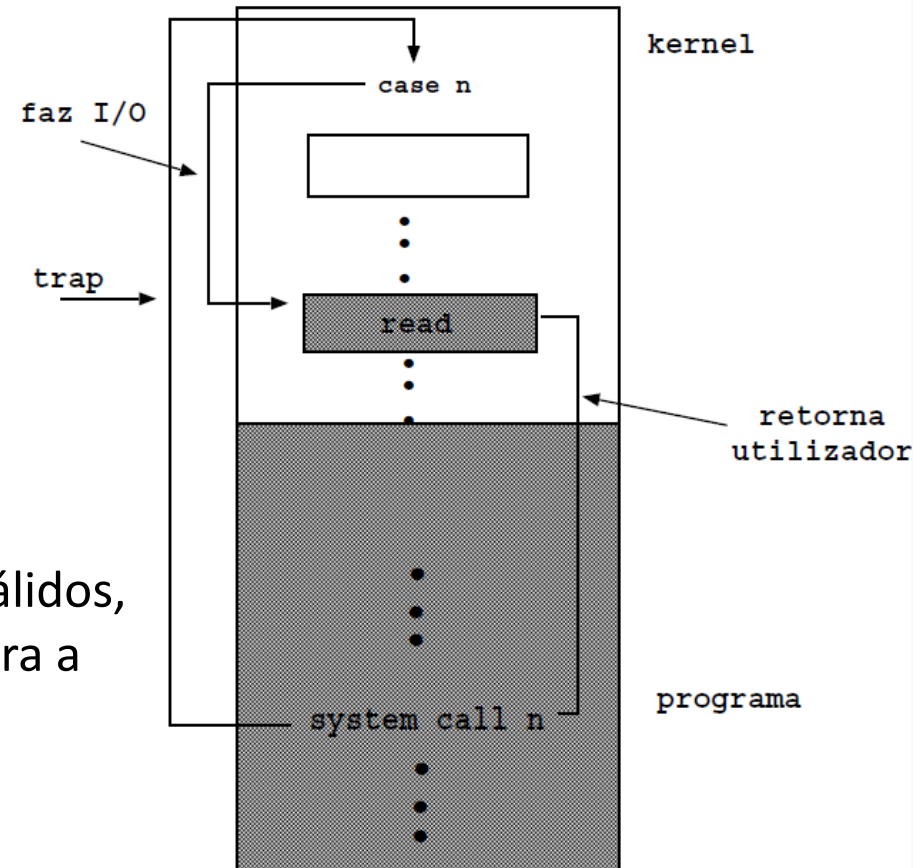
Arquitetura Geral do Sistema

PUC Minas

Visto que as instruções de I/O são privilegiadas, como o usuário executa I/O?

Chamadas ao Sistema (*System calls*) – método usado por um processo para requisitar uma determinada ação ao SO.

- Usualmente é na forma de exceção para uma posição específica do vetor de interrupções
- O **controle** passa a uma rotina de serviço no SO e o **bit de modo** é trocado para modo monitor (*kernel*)
- O **Kernel** verifica se os parâmetros estão válidos, executa a requisição e retorna o controle para a instrução seguinte à chamada ao sistema





Referências

Tanenbaum A. S; Woodhull A. S. **Sistemas Operacionais:** projeto e implementação. 2 Ed, Porto Alegre:Bookman, 2000.

SILVA, Fernando. Sistemas Operacionais: Notas de Aula – UCUP/DCC – 2012.