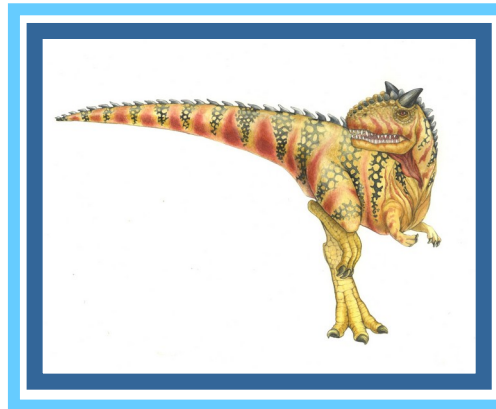


Capítulo 1: Introdução





Sobre a apresentação (About the slides)



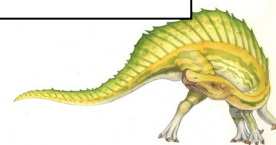
Os slides e figuras dessa apresentação foram criados por Silberschatz, Galvin e Gagne em 2009. Essa apresentação foi modificada por Cristiano Costa (cac@unisinis.br). Basicamente, os slides originais foram traduzidos para o Português do Brasil. Alguns slides foram adicionados por Osvaldo Vargas Jaques, para melhor entendimento/ É possível acessar os slides originais em <http://www.os-book.com>
Essa versão podia ser obtida em <http://www.inf.unisinis.br/~cac>



The slides and figures in this presentation are copyright Silberschatz, Galvin and Gagne, 2009. This presentation has been modified by Cristiano Costa (cac@unisinis.br). Basically it was translated to Brazilian Portuguese. Some slides were added by Osvaldo Vargas Jaques, for better understanding

You can access the original slides at <http://www.os-book.com>

This version could be obtained from downloaded at <http://www.inf.unisinis.br/~cac>





Capítulo 1: Introdução

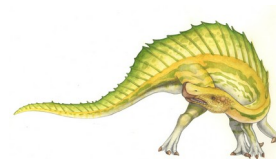
- O que faz um Sistema Operacional
- Organização de um Sistema Computacional
- Arquitetura de um Sistema Computacional
- Estrutura de um Sistema Operacional
- Operações de Sistemas Operacionais
- Gerência de Processos
- Gerência de Memória
- Gerência de Armazenamento
- Proteção e Segurança
- Sistemas Distribuídos
- Sistemas de Propósitos Específicos
- Ambientes Computacionais
- Sistemas Operacionais *Open-source*





Objetivos

- Proporcionar uma visão geral dos principais componentes de sistemas operacionais
- Cobrir os conceitos básicos de organização de sistemas computacionais





O que é um Sistema Operacional?

- Um programa que age como um intermediário entre o usuário e o hardware de um computador.

- Objetivos de um Sistema Operacional:
 - Executar programas do usuário e tornar a solução de problemas mais fácil.
 - Tornar os sistemas computacionais convenientes ao uso.
 - Usar o hardware do computador de maneira eficiente.

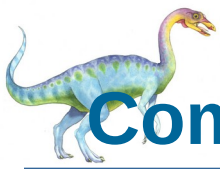




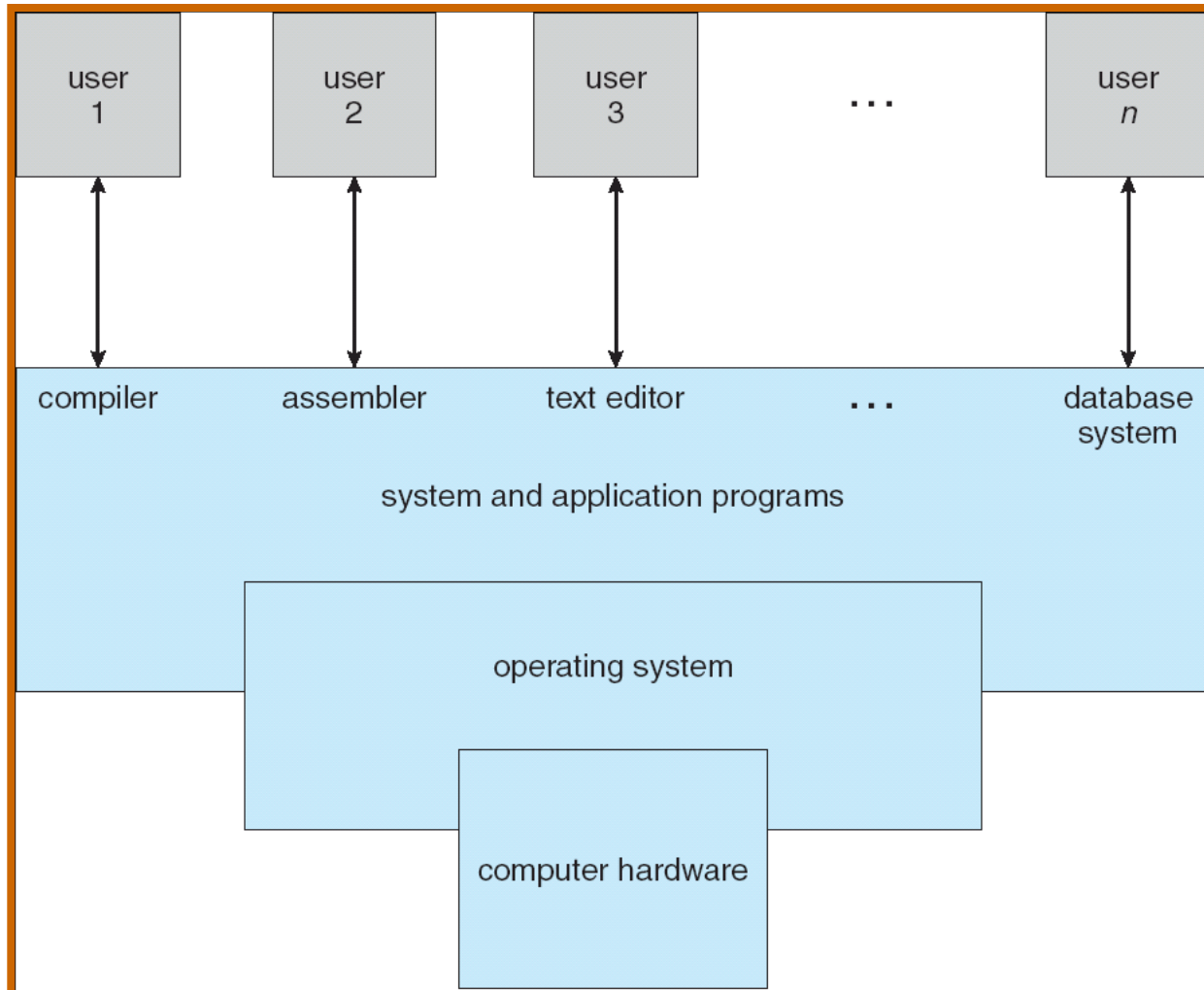
Estrutura de um Sistema Computacional

- Sistemas Computacionais podem ser divididos em quatro componentes
 - Hardware – provê os recursos básicos de computação
 - ▶ CPU, memória, dispositivos de E/S
 - Sistema Operacional
 - ▶ Controla e coordena o uso do hardware entre vários aplicativos e usuários
 - Programas Aplicativos – definem as formas em que os recursos computacionais são usados para resolver problemas computacionais dos usuários
 - ▶ Processadores de Textos, compiladores, navegadores web, bancos de dados, jogos
 - Usuários
 - ▶ Pessoas, máquinas, outros computadores





Componentes de um Sistema Computacional





Definição de Sistema Operacional

- SO é um **alocador de recursos**
 - Gerencia todos os recursos
 - Decide entre requisições conflitantes para uso eficiente e justo dos recursos
- SO é um **programa de controle**
 - Controla execução de programas para prevenir erros e usos indevidos do computador





Definição de Sistema Operacional(Cont.)

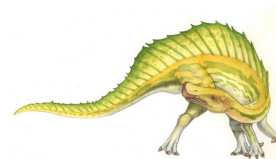
- Não existe definição aceita universalmente
- “Tudo aquilo que um vendedor entrega quando você pede um sistema operacional” é uma boa aproximação
 - Muita variação
- “O único programa executando todo o tempo em um computador” é o **kernel** (núcleo). Todo o resto é um programa de sistemas (vendido com o sistema operacional) ou um programa aplicativo





Inicialização do Computador

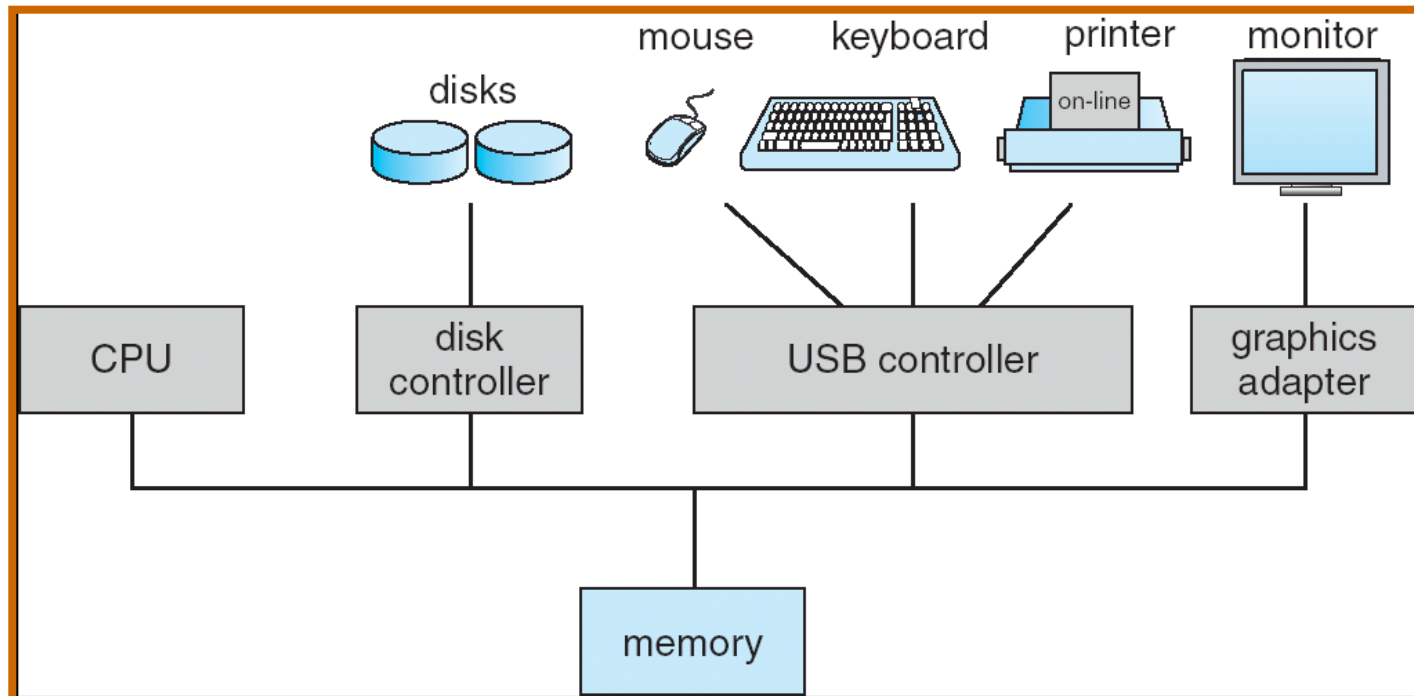
- **programa de bootstrap** é carregado ao ligar ou reiniciar o computador
 - Tipicamente armazenado em ROM ou EEPROM, geralmente conhecido como **firmware**
 - Inicializa todos os componentes do sistema
 - Carrega o kernel do sistema operacional e inicia execução





Organização do Sistema Computacional

- Operação de um Sistema Computacional
 - Uma ou mais CPUs, controladores de dispositivos conectados através de um barramento comum provendo acesso à memória compartilhada
 - Execução concorrente das CPUs e dos dispositivos competindo por ciclos de memória





Operação do Sistema Computacional

- dispositivos de E/S e CPU podem executar concorrentemente.
- Cada controlador de dispositivo é encarregado de um tipo particular de dispositivo.
- Cada controlador de dispositivo tem um buffer local.
- CPU move dados de/para memória principal para/de buffers locais.
- E/S é de um dispositivo para o buffer local da controladora.
- Controladora de dispositivos informam a CPU que terminaram uma operação gerando uma interrupção.





Funções Comuns de Interrupções

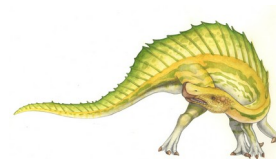
- Transferir o controle para a rotina de serviço da interrupção, geralmente através do vetor de interrupção, o qual contém os endereços de todas as rotinas de serviço.
- Arquitetura da interrupção deve salvar o endereço da instrução interrompida.
- A geração de novas interrupções é desabilitada enquanto uma outra interrupção está sendo processada para evitar a *perda de uma interrupção*.
- Uma *trap* é uma interrupção gerada por software causada por uma requisição do usuário ou por um erro.
- Um sistema operacional é acionado por interrupções.

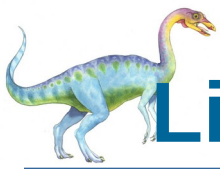




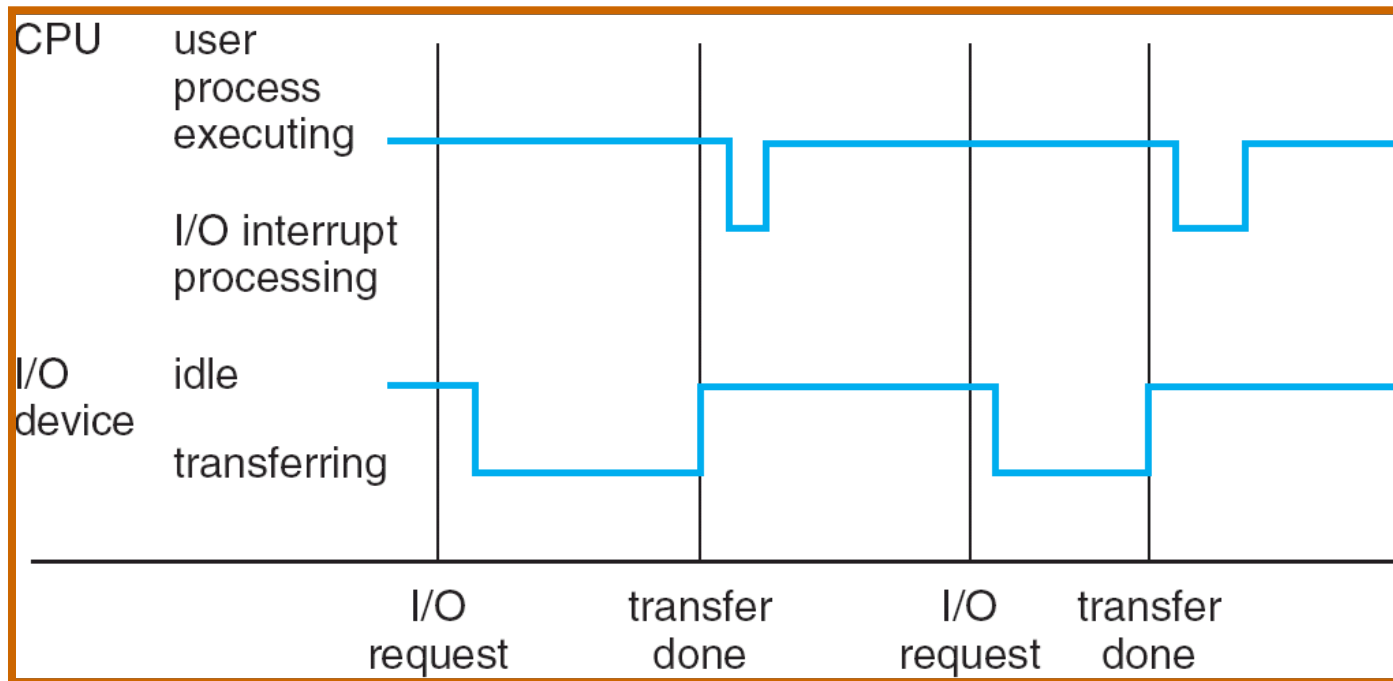
Tratamento de Interrupção

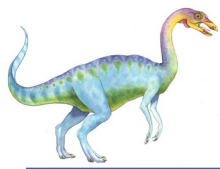
- O sistema operacional preserva o estado da CPU armazenando registradores e o PC (contador de programas).
- Determina que tipo de interrupção ocorreu:
 - baseadas em *polling* (sondagem)
 - através do *vetor de interrupções*
- Segmentos de código separados determinam qual ação deve ser realizada para cada tipo de interrupção





Linha do Tempo de uma Interrupção





Estrutura de E/S

- Após o início de uma E/S, o controle retorna ao programa do usuário somente quando a requisição de E/S terminar.
 - Instruções de espera liberam a CPU até a próxima interrupção
 - Laço de espera (contenção para acesso a memória).
 - No máximo uma requisição de E/S está pendente de cada vez, não ocorre processamento de E/S simultâneo.
- Após o início de uma E/S, o controle retorna ao programa do usuário sem aguardar o tratamento da requisição de E/S.
 - *Chamada de Sistemas (System call)* – requisição ao sistema operacional para permitir ao usuário aguardar o término da E/S.
 - *Tabela de Estados de Dispositivos* contém uma entrada para cada dispositivo de E/S indicando seu tipo, endereço e estado.
 - Sistema operacional acessa a tabela de dispositivos de E/S para determinar a situação do dispositivo e modificar a entrada na tabela para incluir a interrupção.





Dois métodos de E/S

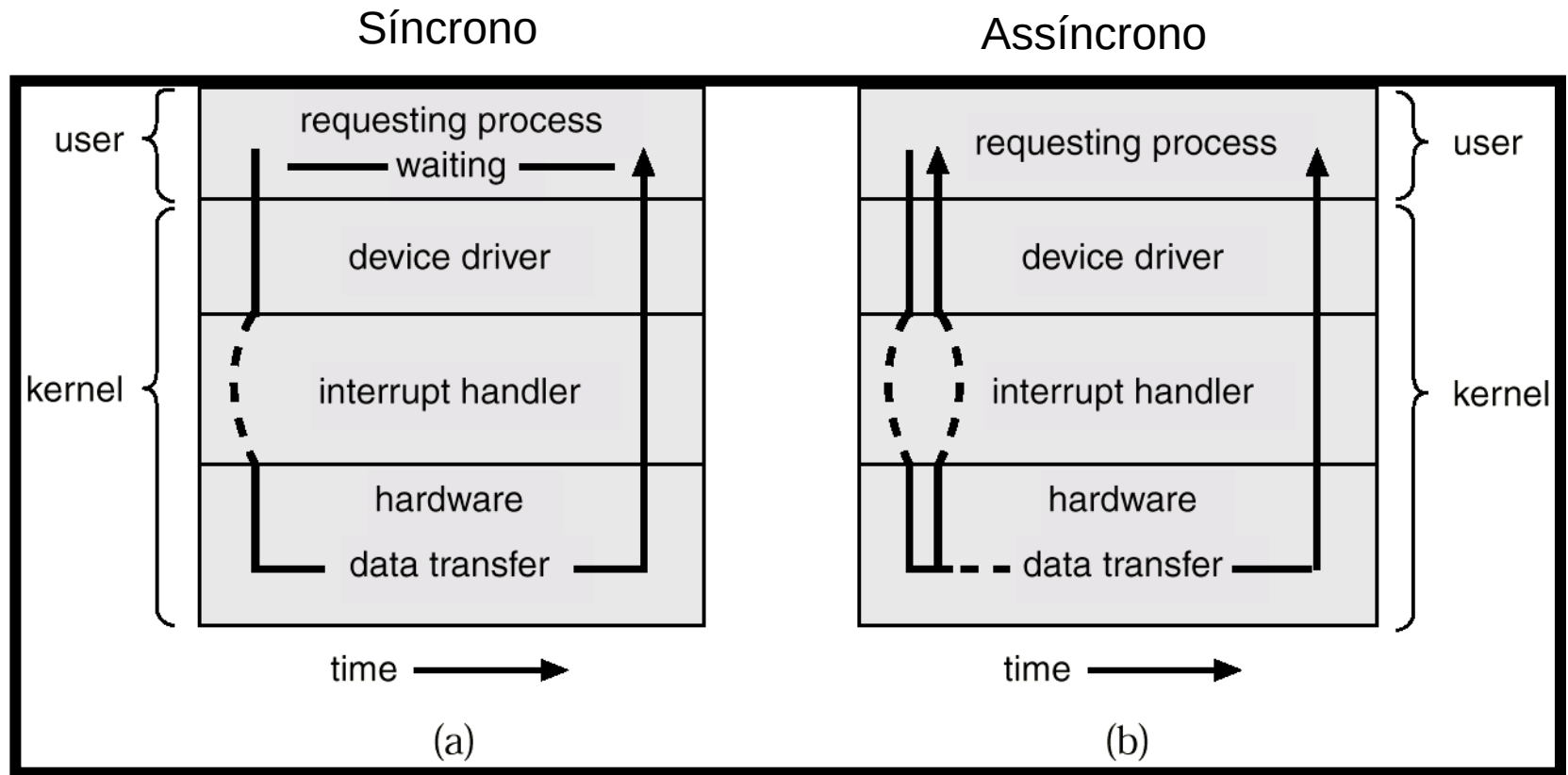
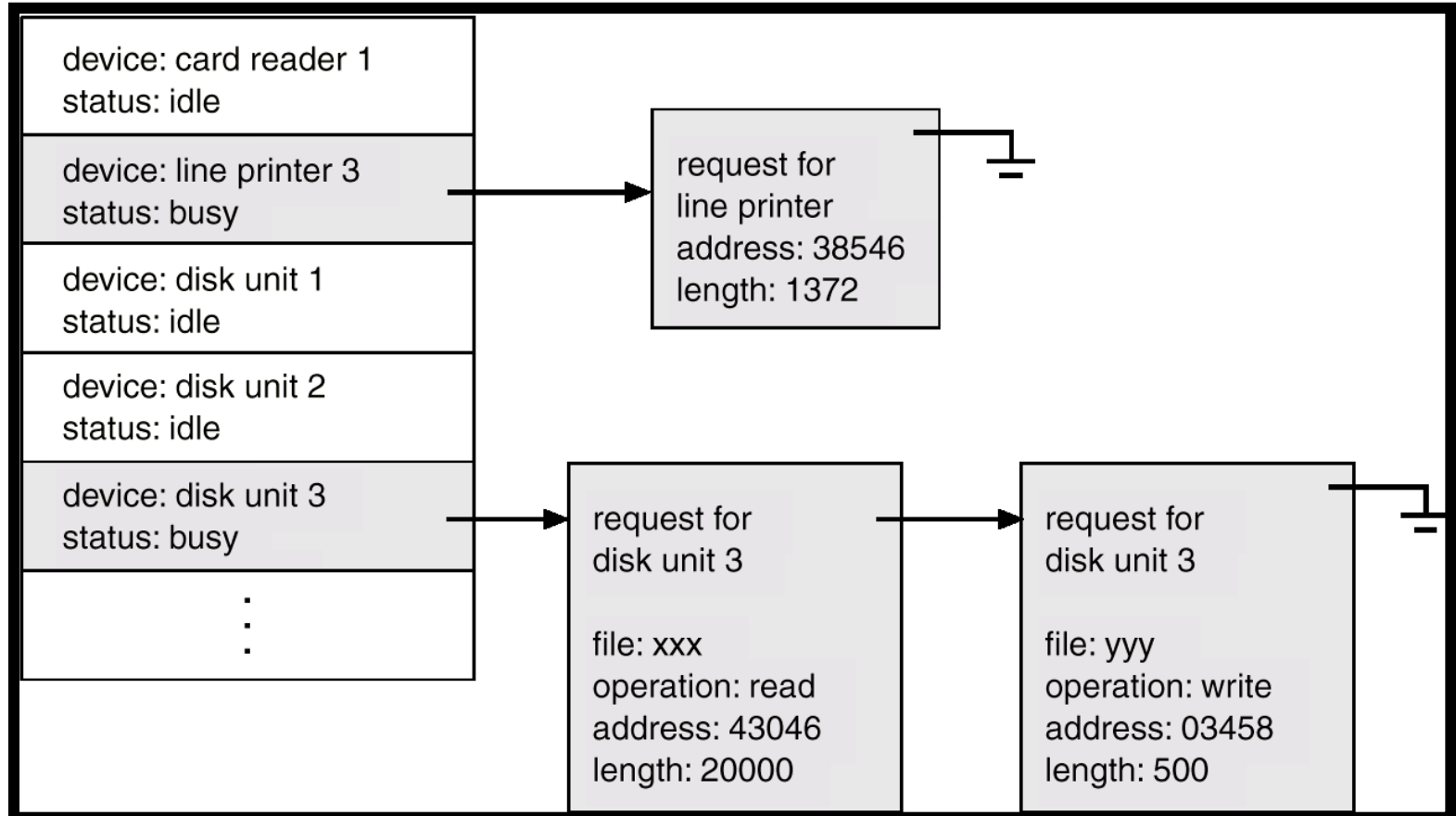




Tabela de Estados de Dispositivos





Estrutura de Acesso Direto à Memória

- Usado para dispositivos de E/S de alta velocidade capazes de transmitir informação a uma taxa próxima da memória.
- Controladores de dispositivo transferem blocos de dados de um buffer de armazenamento diretamente para a memória principal, sem intervenção da CPU.
- Somente uma interrupção é gerada por bloco, ao invés de uma interrupção por byte.





Estruturas de Armazenamento

- Memória Principal – única grande área de armazenamento que a CPU pode acessar diretamente.

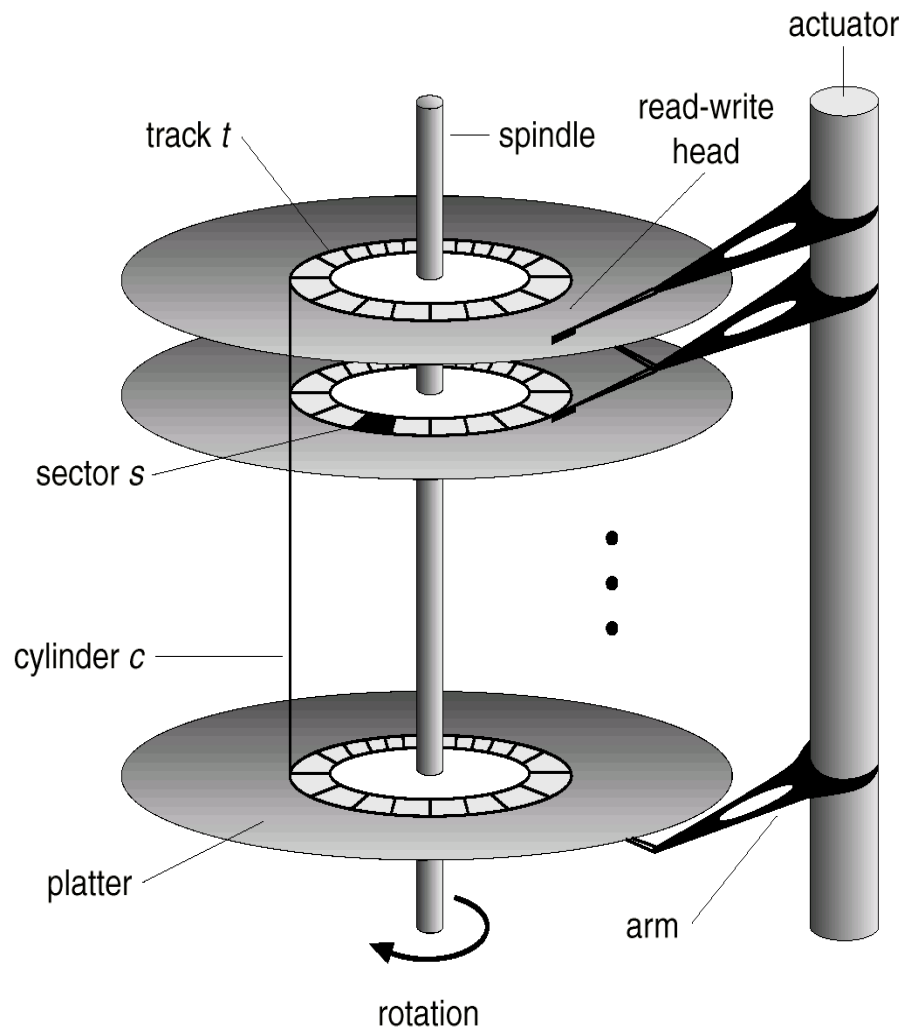
- Armazenamento Secundário – extensão da memória principal que fornece grande capacidade de armazenamento persistente (não volátil).

- Discos Magnéticos – prato de metal rígido ou vidro coberto com material magnético para armazenamento de dados
 - Superfície do disco é dividida logicamente em *trilhas*, que são subdivididas em *setores*.
 - A *controladora de disco* determina a interação lógica entre o dispositivo e o computador.





Mecanismo de Movimento do Disco





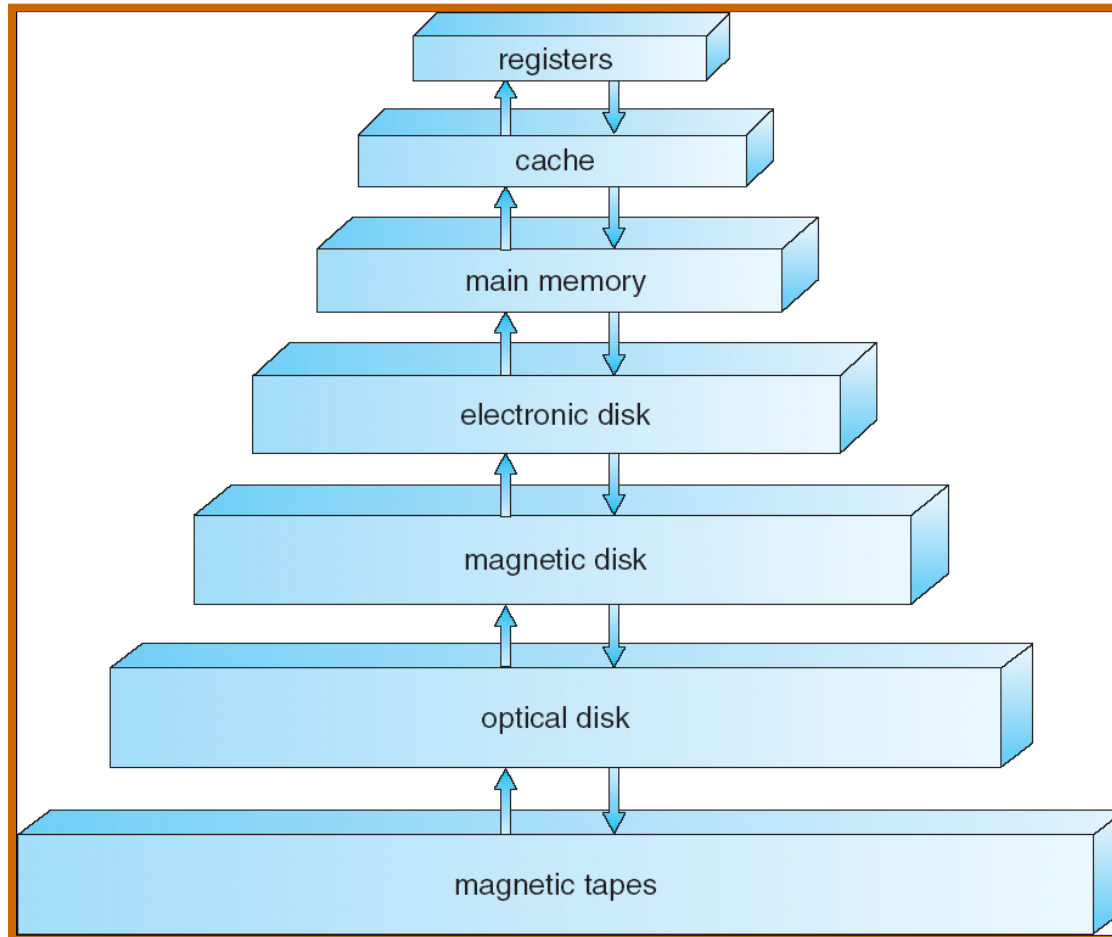
Hierarquia de Armazenamento

- Sistemas de Armazenamento são organizados em hierarquia.
 - Velocidade
 - Custo
 - Volatilidade
- *Caching* – copiar informação em um sistema mais rápido de armazenamento; memória principal pode ser vista como um último *cache* para o armazenamento secundário





Hierarquia de Dispositivos de Armazenamento





Caching

- Princípio importante, realizado em muitos níveis no computador (no hardware, sistema operacional, software)
- Informação em uso é copiada de um armazenamento mais lento para um mais rápido temporariamente
- Armazenamento mais rápido (cache) é verificado primeiro para determinar se a informação necessária está lá
 - Se sim, informação é usada diretamente do cache (rápido)
 - Se não, dados são copiados para o cache e usados lá
- Cache é menor que o armazenamento que está sendo usado
 - Gerência de cache é um problema de projeto importante
 - Tamanho do cache e política de substituição





Arquitetura de Sistemas Computacionais

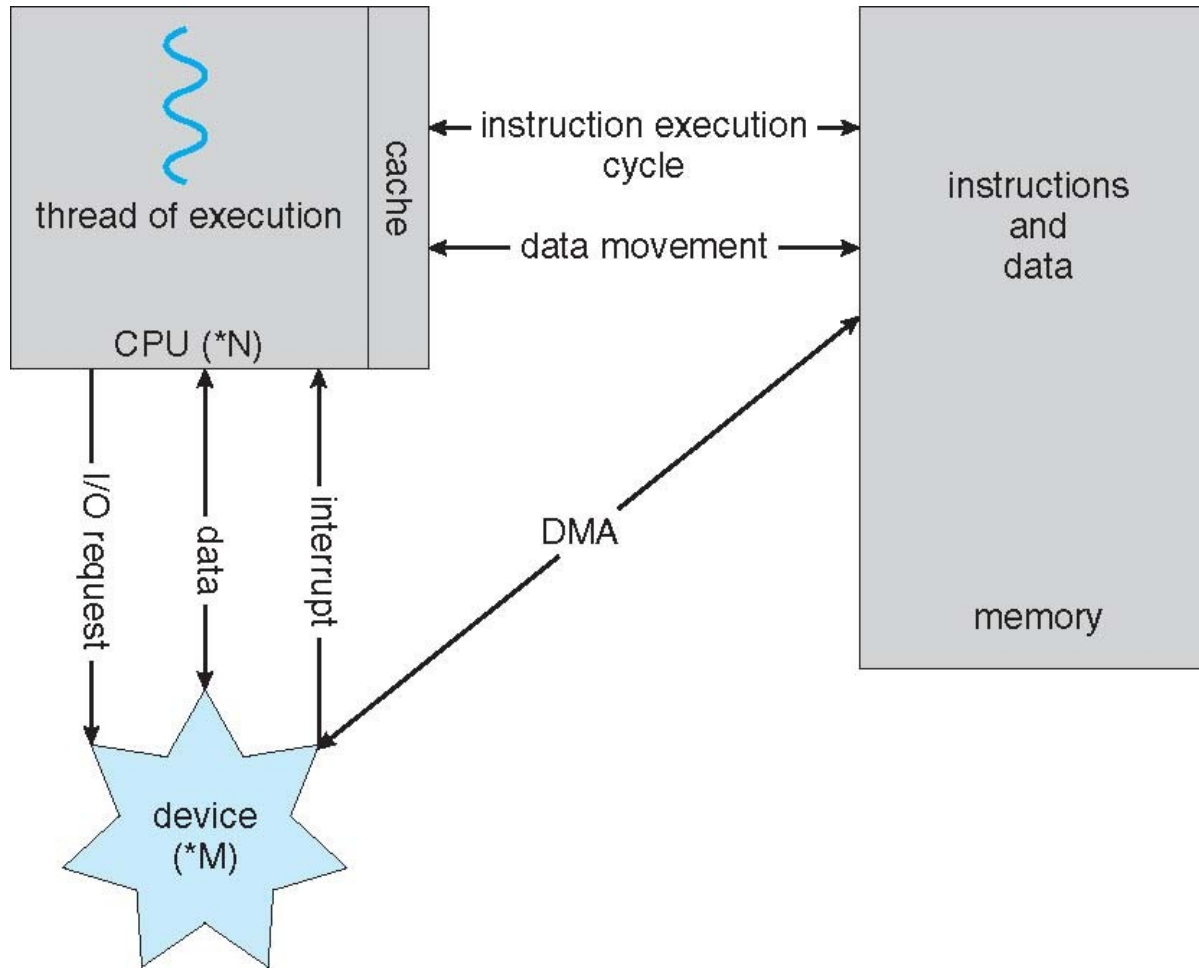
- A maioria dos sistemas usa um único processador de propósito geral (de PDAs a mainframes).
 - Maioria dos sistemas têm também processadores de propósito específico.

- sistemas com **Multiprocessadores** crescem em uso e importância
 - Também chamados de **sistemas paralelos**, **sistemas fortemente acoplados**
 - Vantagens incluem:
 - ◆ **Vazão (throughput) aumentada**
 - ◆ **Economia de escala**
 - ◆ **Confiabilidade aumentada – degradação limpa ou tolerância a falhas**
 - Dois tipos:
 - ◆ **Multiprocessamento assimétrico**
 - ◆ **Multiprocessamento simétrico**





Como computadores modernos funcionam





Simétrico x Assimétrico

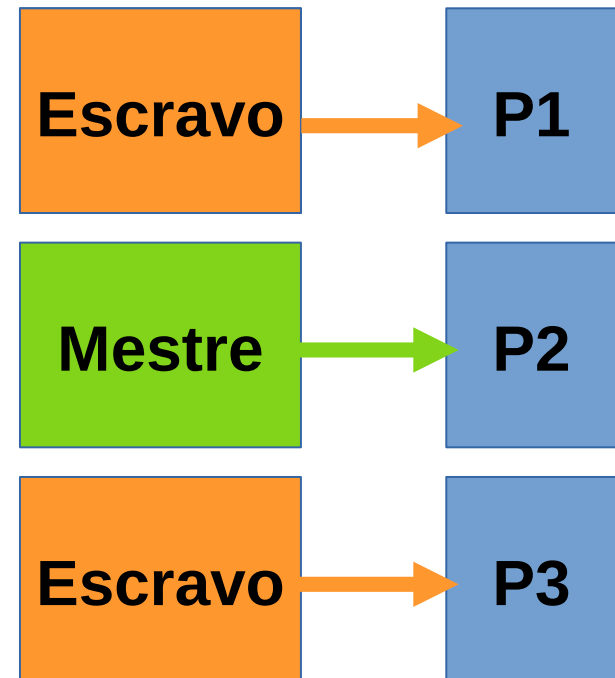
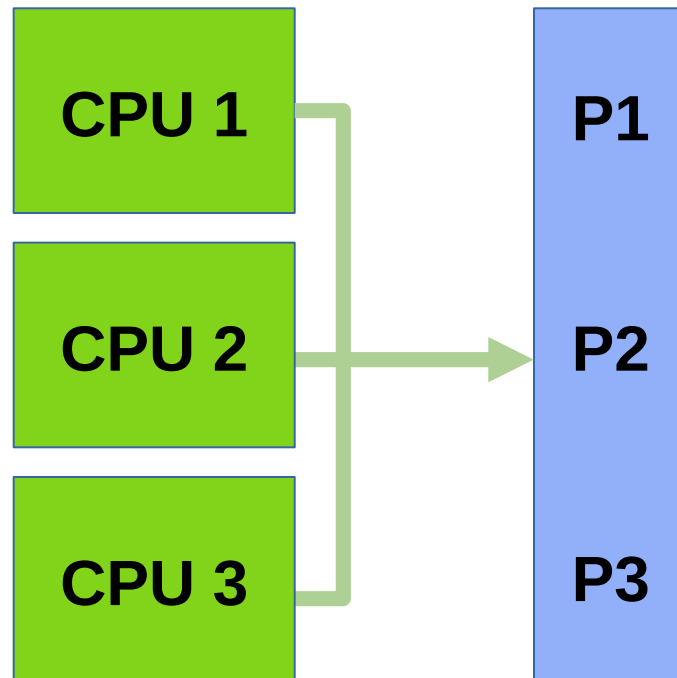
Multiprocessamento

Multiprocessamento

X

Simétrico

Assimétrico





Simétrico x Assimétrico

Base para Comparação	Multiprocessamento Simétrico	Multiprocessamento Assimétrico
Basic	Cada processador executa as tarefas no sistema operacional.	Apenas o processador mestre executa as tarefas do sistema operacional.
Processo	O processador usa processos de uma fila pronta comum ou pode haver uma fila privada pronta para cada processador.	O processador mestre atribui processos aos processadores escravos, ou eles têm alguns processos predefinidos.
Arquitetura	Todo processador em Multiprocessamento Simétrico possui a mesma arquitetura.	Todo processador em Multiprocessamento Assimétrico pode ter arquitetura igual ou diferente.





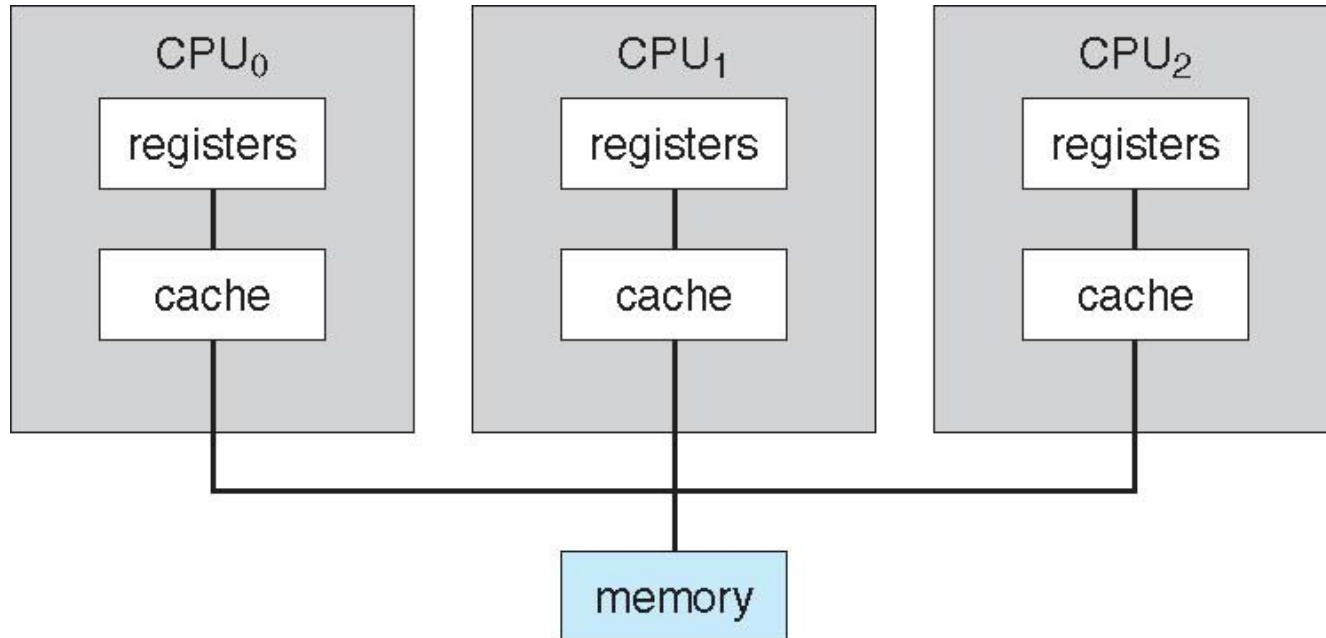
Simétrico x Assimétrico

Base para Comparação	Multiprocessamento Simétrico	Multiprocessamento Assimétrico
Comunicação	Todos os processadores se comunicam com outro processador por uma memória compartilhada.	Os processadores não precisam se comunicar, pois são controlados pelo processador mestre.
Falha	Se um processador falhar, a capacidade de computação do sistema será reduzida.	Se um processador mestre falhar, um escravo é ligado ao processador mestre para continuar a execução. Se um processador escravo falhar, sua tarefa será alternada para outros processadores.
Facilidade	O Multiprocessador Simétrico é complexo, pois todos os processadores precisam ser sincronizados para manter o equilíbrio de carga.	O multiprocessador assimétrico é simples, pois o processador mestre acessa a estrutura de dados.



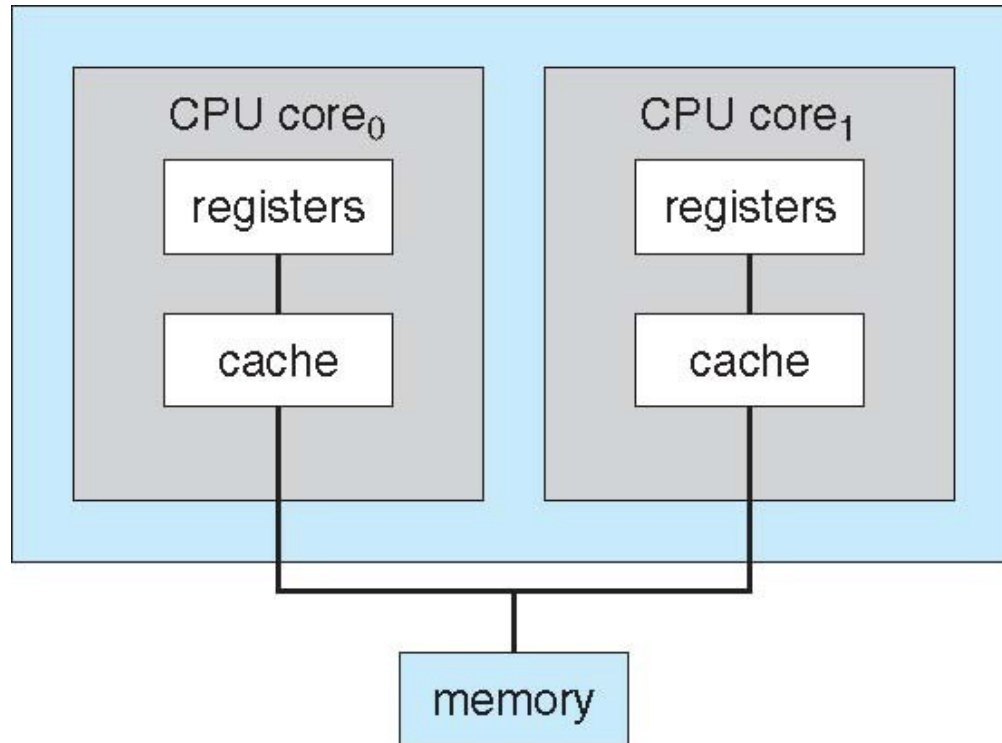


Arquitetura de Multiprocessamento Simétrico





Um Projeto Dual-core





Sistemas Agrupados (clusters)

- Como sistemas multiprocessados, mas vários sistemas trabalhando juntos
 - Usualmente compartilhando armazenamento via uma rede de armazenamento ou **storage-area network (SAN)**
 - Fornecem serviços de **alta disponibilidade** que sobrevivem as falhas
 - ▶ **Cluster Assimétrico** tem uma máquina que permanece em *hot-standby mode*
 - ▶ **Cluster Simétrico** tem vários nodos executando aplicações, monitorando uns aos outros
 - Alguns clusters são para computação de alto desempenho - **high-performance computing (HPC)**
 - ▶ Aplicações devem ser escritas para usar a **paralelização**





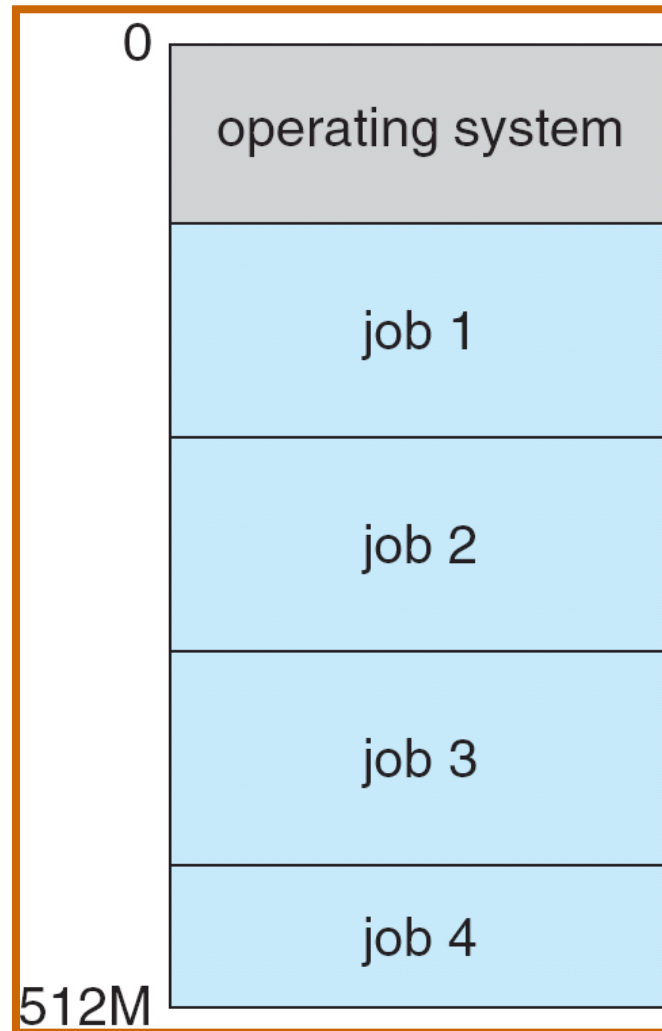
Estrutura de Sistema Operacional

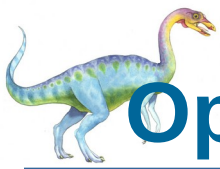
- **Multiprogramação** é necessária para eficiência
 - Um único usuário não pode manter CPU e dispositivos de E/S ocupados todo o tempo
 - Multiprogramação organiza jobs (código e dados) de forma que a CPU sempre tenha um para ser executado
 - Um subconjunto do total de jobs no sistema é mantido na memória
 - Um job é selecionado e executa via **escalonamento de job**
 - Quando ele deve esperar (E/S por exemplo), SO executa outro job
- **Tempo Compartilhado ou Timesharing (multitarefa)** é uma extensão lógica na qual a CPU altera jobs tão frequentemente que os usuários podem interagir com cada job durante sua execução, criando computação interativa
 - **Tempo de Resposta** deve ser < 1 segundo
 - Cada usuário tem no mínimo um programa executando na memória □ **processo**
 - Se vários jobs estão prontos para executar ao mesmo tempo □ **escalonamento de CPU**
 - Se o processo não cabe na memória, **swapping** move alguns processos para o armazenamento secundário para carregá-lo e executá-lo
 - **Memória Virtual** permite execução de processos que não estejam completamente na memória





Memória em Sistemas Multiprogramados





Operações de Sistemas Operacionais

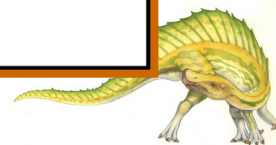
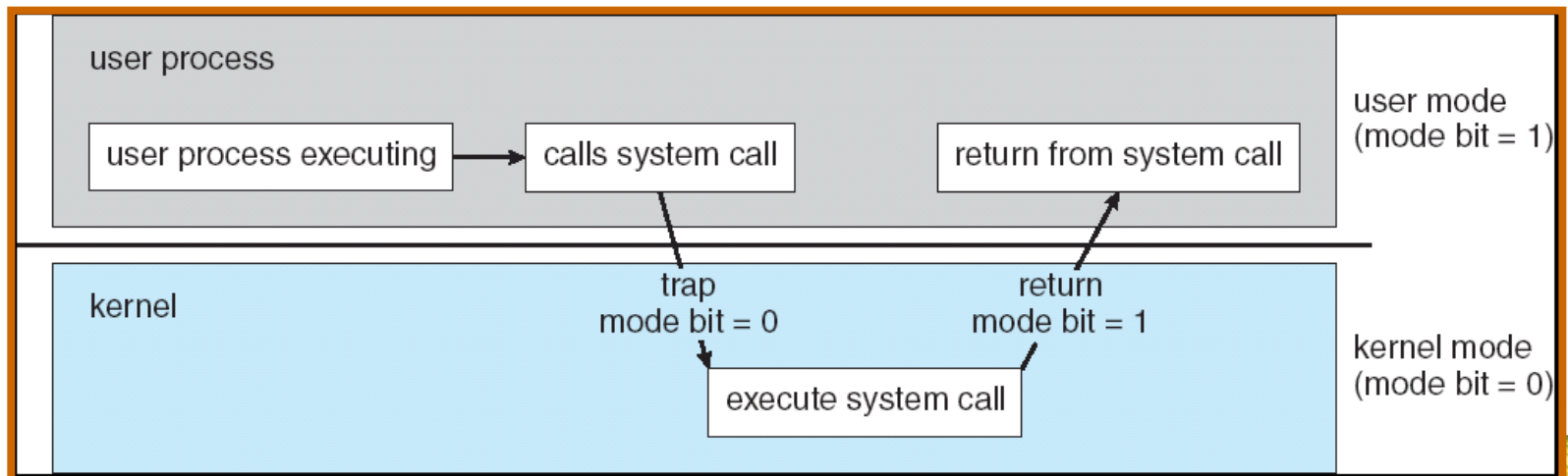
- Acionado por Interrupções pelo hardware
- Erro de Software ou requisições criam **exceções** ou **trap**
 - Divisão por zero, requisição por serviços do sistema operacional
- Outros problemas de processos incluem loops (laços) infinitos, processos que modificam outros ou o sistema operacional
- operação em **Dois Modos** possibilita que o SO se proteja e proteja outros componentes do sistemas
 - **modo Usuário e modo Kernel**
 - **Mode bit** fornecido pelo hardware
 - ▶ Possibilita determinar quando o sistema está executando código do usuário ou do kernel
 - ▶ Algumas instruções são projetadas como **privilegiadas**, sendo somente executáveis em modo kernel
 - ▶ System call (Chamada de sistema) troca o modo para kernel, no retorno da chamada volta para modo usuário





Transição do Modo Usuário para Modo Kernel

- Timer (Temporizador) para prevenir laços infinitos / processos monopolizando recursos
 - Gera interrupção após um período específico
 - Sistema Operacional decrementa contador
 - Quando um contador atinge zero é gerada uma interrupção
 - Inicializado antes de escalonar o processo para ganhar novamente o controle ou terminar o programa que excede o tempo alocado





Gerência de Processos

- Um processo é um programa em execução. É uma unidade de trabalho no sistema. Programa é uma *entidade passiva*, processo é uma *entidade ativa*.
- Processo necessita de recursos para realizar sua tarefa
 - CPU, memória, E/S, arquivos
 - Dados de inicialização
- terminação de processos necessita reaproveitar quaisquer recursos reusáveis
- processos com uma única thread (*Single-threaded*) têm um **contador de programas (PC)** especificando a localização da próxima instrução a ser executada
 - Processos executam instruções sequencialmente, uma por vez, até acabar
- processos com várias threads (*Multi-threaded*) possuem um contador de programas por thread
- Tipicamente sistemas possuem vários processos, alguns usuários, algum sistema operacional executando concorrentemente em uma ou mais CPUs
 - A Concorrência é obtida multiplexando as CPUs entre os processos / threads





Atividades da Gerência de Processos

O sistema operacional é responsável pelas seguintes atividades relacionadas com gerência de processos:

- Criar e deletar processos do usuário e do sistema
- Suspende e retomar processos
- Fornecer mecanismos para sincronização de processos
- Fornecer mecanismos para comunicação entre processos
- Fornecer mecanismos para tratamento de deadlocks





Gerência de Memória

- Todos dados estão na memória antes e depois do processamento
- Todas instruções estão na memória para serem executadas
- Gerência de Memória determina “o que está na memória e quando”
 - Otimizando a utilização da CPU e a resposta do computador aos usuários
- Atividades da Gerência de Memória
 - Controlar as partes da memória que estão sendo utilizadas e por quem
 - Decidir quais processos (ou partes deles) e dados colocar e retirar da memória
 - Alocação e desalocação de espaços de memória quando necessário





Gerência de Armazenamento

- SO fornece uma visão lógica e uniforme do armazenamento de informações
 - Propriedades físicas são abstraídas para unidade de armazenamento lógico - **arquivo**
 - Cada mídia é controlada por um dispositivo (ex., drive de disquete, drive de fita)
 - ▶ Propriedades variam: velocidade de acesso, capacidade, taxa de transferência de dados, método de acesso (seqüencial ou aleatório)
- Gerência de Sistemas de Arquivos
 - Arquivos são normalmente organizados em diretórios
 - Existe controle de acesso na maioria dos sistemas para determinar quem acessa o que
 - Atividades do SO incluem
 - ▶ Criação e deleção de arquivos e diretórios
 - ▶ Primitivas para manipular arquivos e diretórios
 - ▶ Mapeamento de arquivos em armazenamento secundário
 - ▶ Cópia de Segurança de arquivos em mídias de armazenamento estáveis (não voláteis)





Gerência de Armazenamento em Massa

- Tipicamente discos são usados para armazenar dados que não cabem na memória principal ou dados que devem ser guardados por um “longo” período de tempo.
- Gerência adequada é de vital importância
- A velocidade das operações do computador dependem fortemente dos subsistemas de disco e seus algoritmos
- Atividades do SO
 - Gerência do espaço livre
 - Alocação de armazenamento
 - Escalonamento de disco
- Alguns armazenamentos não precisam ser rápidos
 - Armazenamento terciário inclui armazenamento óptico, fitas magnéticas
 - Ainda devem ser gerenciados
 - Varia entre WORM (write-once, read-many-times – escreve uma vez, lê muitas vezes) e RW (read-write – leitura-escrita)





Desempenho de Vários Níveis de Armazenamento

- Movimento entre níveis da hierarquia de armazenamento pode ser explícito ou implícito.

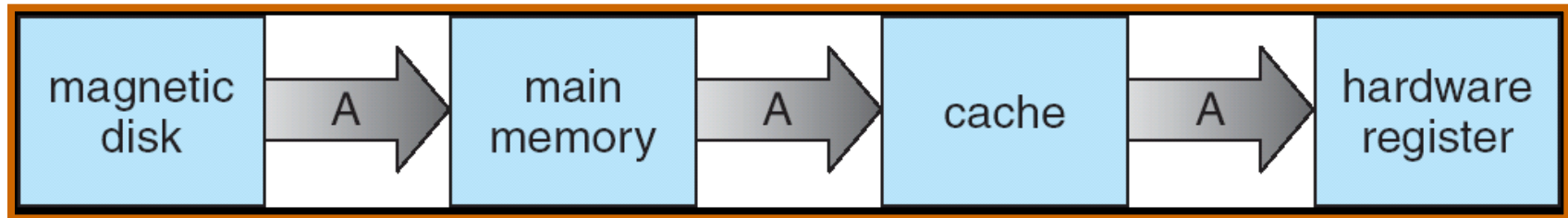
Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape





Migração de um Inteiro A do Disco para Registrador

- Em ambientes multitarefa é preciso tomar cuidado para usar o valor mais recente, não importa onde ele esteja armazenado na hierarquia



- Ambientes multiprocessados devem fornecer coerência de cache em hardware possibilitando que todas as CPUs tenha o valor mais recente em seu cache
- Em ambientes distribuídos a situação é ainda mais complexa
 - Várias cópias dos dados podem existir
 - Várias soluções são apresentadas no Capítulo 17





Subsistema de E/S

- Um dos objetivos do SO é esconder peculiaridades dos dispositivos de hardware do usuário
- Subsistema de E/S é responsável por
 - Gerência de Memória do E/S incluindo:
 - ▶ bufferização - armazenando dados temporariamente enquanto estão sendo transferidos
 - ▶ *caching* - armazenando partes dos dados em armazenamento mais rápido para aumentar o desempenho
 - ▶ *spooling* - sobreposição da saída de um *job* com a entrada de outros *jobs*
 - Interface geral para *drivers* de dispositivos
 - *Drivers* para dispositivos de hardware específicos





Proteção e Segurança

- **Proteção** – qualquer mecanismo para controlar acesso de processos ou usuários aos recursos definidos pelo SO
- **Segurança** – defesa do sistema contra ataques internos e externos
 - Conceito muito amplo, incluindo negação de serviços (DOS), worms, vírus, roubo de identidade, roubo de serviço
- Sistema geralmente primeiro identifica os usuários, para determinar quem pode fazer o que
 - Identidades de Usuários (**user IDs**, **security IDs**) incluem nomes e associam números, um por usuário
 - *user ID* é associado com todos os arquivos, processos daquele usuário para determinar o controle de acesso
 - Identificador de Grupo (**group ID**) permite que um conjunto de usuários seja definido e seu controle gerenciado, também sendo associado com cada processo e arquivo
 - aumento de privilégio (**privilege escalation**) permite que usuários alterem para uma identificação com mais direitos





Ambientes Computacionais

■ Computadores Tradicionais

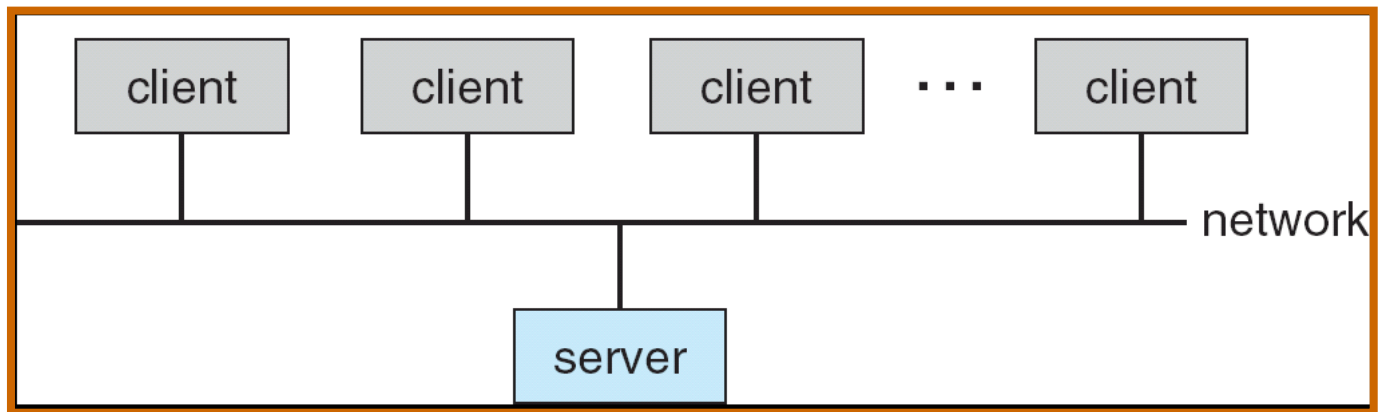
- Ficam obsoletos com o tempo
- Ambiente de Escritório
 - ▶ PCs conectados a um rede, terminais ligados ao mainframe ou minicomputadores fornecendo *batch* (execução em lote) e tempo compartilhado
 - ▶ Hoje existem portais permitindo acesso a redes e sistemas remotos e aos mesmos recursos
- Redes Caseiras
 - ▶ Costumava ser um sistema isolado, com modem
 - ▶ Hoje conectado em rede e com *firewall*

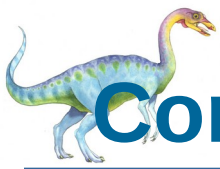




Ambientes Computacionais (Cont.)

- Computação Cliente-Servidor
 - Terminais burros suplantados por PCs mais poderosos
 - Muitos sistemas agora **servidores**, respondendo a requisições geradas pelos **clientes**
 - ▶ **Servidor** fornece uma interface ao cliente permitindo a requisição de serviços (ex. bancos de dados)
 - ▶ **Servidor de Arquivos** fornece interface para clientes armazenar e acessar arquivos





Computação entre Pares (*Peer-to-Peer*)

- Outro modelo de sistema distribuído
- P2P não faz distinção entre clientes e servidores
 - Ao contrário todos os nodos são considerados igualitários
 - Cada um pode agir como cliente, servidor ou ambos
 - Nodo deve se juntar a rede P2P
 - ▶ Registrar seus serviços com um serviço de procura central na rede, ou
 - ▶ Distribuir requisição por serviço e responder para requisições por serviço via protocolo de descoberta
 - Exemplos incluem *Napster* e *Gnutella*





Computação Baseada na Web

- Web se tornou ubíqua
- PCs são os dispositivos mais predominantes
- Mais dispositivos estão sendo conectados em redes permitindo acesso pela web
- Nova categoria de dispositivos para gerenciar tráfego na web entre servidores similares: **balanceamento de carga**
- Uso de sistemas operacionais como Windows 95, no lado do cliente, desenvolveram-se para Linux. Mac OS e Windows Vista, que podem ser clientes e servidores





Sistemas Operacionais Open-Source

- Sistemas operacionais que disponibilizam o código-fonte e não somente o código binário fechado
- Contra a **proteção de cópia** e o movimento **Digital Rights Management (DRM)**
- Iniciado pela **Free Software Foundation (FSF)**, que mantem “copyleft” **GNU Public License (GPL)**
- Exemplos incluem **GNU/Linux**, **BSD UNIX** (incluindo o principal componente do **Mac OS X**), e **Sun Solaris**



Fim do Capítulo 1

