

Unidade II

3 TIPOS DE SISTEMAS OPERACIONAIS

Podemos classificar os sistemas operacionais com base referencial em diversos parâmetros e perspectivas, velocidade, suporte a recursos específicos, acesso à rede etc.¹

A seguir, são apresentados alguns tipos de sistemas operacionais típicos (muitos sistemas operacionais estão enquadrados em mais de uma das categorias apresentadas).²

3.1 Batch (de lote)

Os sistemas operacionais mais antigos, principalmente os desenvolvidos para a segunda geração de computadores, trabalhavam "por lote", ou seja, todos os programas a serem executados eram colocados numa fila.³

O processador recebia um programa após o outro, para ser processado em sequência, o que permitia um alto grau de utilização do sistema. Atualmente, o termo "em lote" é usado para designar um conjunto de comandos que deve ser executado em sequência.⁴

OS/360 e VMS são alguns exemplos de sistemas operacionais de lote.

3.2 De rede

Os sistemas operacionais de rede devem suportar operações em rede, ou seja, a capacidade de oferecer às aplicações locais recursos que estejam localizados em outros computadores da rede LAN (*Local Area Network*) ou WAN (*Wide Area Network*), como serviços de autenticação remota, acesso a arquivos, acesso a banco de dados, impressoras etc. Deve também disponibilizar seus recursos locais aos demais computadores, de forma controlada.⁵

A maioria dos sistemas operacionais atuais está alinhada com esta classificação.

¹ Disponível em: <<http://www.computronixbras.com/cursos/SOP/Windows2000/ApostilaSOSite.pdf>>. Acesso em: 8 jun. 2011.

² Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

³ Disponível em: <<http://www.ebah.com.br/sistemas-operacionais-pdf-a52910.html>>. Acesso em: 8 jun. 2011.

⁴ Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

⁵ Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

3.3 Distribuído

No sistema operacional distribuído, os recursos utilizados por cada usuário em cada máquina estão disponíveis de forma transparente aos usuários. Ao acessar uma aplicação, o usuário interage com sua interface, entretanto não sabe onde está executando ou armazenando seus arquivos, sendo esta decisão de total responsabilidade do sistema e totalmente transparente para o usuário do sistema.⁶

3.4 Multiusuário e multitarefas

Para entendermos um sistema operacional, é muito importante definir o que é processo ou tarefa. Chamamos de processo um programa que está em execução. E onde está a importância em saber essa definição? Um programa em execução requer recursos, principalmente da CPU, e concorre com outros processos para ser realizado. Portanto, cabe ao sistema operacional, quando multiusuário e multitarefas, gerenciar os recursos disponíveis a todos os processos.

Um sistema operacional multiusuário deve suportar a autenticação e a autorização de cada usuário quando necessário o uso de algum recurso contido no sistema (arquivos, processos, conexões de rede) e garantir regras para o controle de acesso, objetivando impedir o uso desses recursos por usuários não autorizados. Essa funcionalidade é fundamental para a segurança dos sistemas operacionais de rede e distribuídos. Grande parte dos sistemas operacionais atuais é multiusuário.⁷

3.5 Desktop/Computador pessoal

Um sistema operacional *Desktop* é projetado para atender ao usuário doméstico e/ou corporativo que necessite realizar atividades triviais, como editar textos, elaborar planilhas e gráficos, navegar na internet etc. Suas principais características são a *interface* gráfica, o suporte à interatividade e a operação em rede. Exemplos de sistemas *Desktop* são o Windows XP, Mac OS X e Linux.

3.6 Servidor

O sistema operacional classificado como servidor possibilita a gestão eficiente de grandes quantidades de recursos físicos como: disco, memória e processadores, gerindo as prioridades e limites sobre o uso dos recursos. Tipicamente, um sistema operacional servidor também tem suporte a rede e multiusuários.⁸

⁶ Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

⁷ Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

⁸ Disponível em: <<http://www.fatecsbc.edu.br/Download/sistemasoperacionais/socap01.pdf>>. Acesso em: 8 jun. 2011.

Alguns exemplos de sistemas servidores são: HP-UX, Solaris, AIX, FreeBSD, Linux e Windows Server.

3.7 Embarcados

Um sistema operacional embarcado ou embutido (*embedded*) é construído para operar sobre um *hardware* com poucos recursos de processamento e armazenamento.⁹

Exemplos típicos são aparelhos de TV, reprodutores de MP3, aparelhos de DVD etc. Um ponto positivo destes equipamentos que dependem deste tipo de sistema é que dificilmente será possível instalar algum *software* que não seja extremamente compatível, não permitindo o uso de *software* não confiável. Exemplos de sistema embarcado: QNX e VxWorks.

3.8 Tempo real

Esse sistema operacional é caracterizado por ter o tempo como parâmetro principal. Outras duas subdivisões são: **sistemas de tempo real críticos**, voltados tipicamente ao controle de processos industriais e militares e os **sistemas de tempo real não críticos**, como os aplicados em sistemas de áudio digital ou multimídia. A grande diferença entre estas duas subcategorias é que, no caso de sistemas que dependem do **tempo real crítico**, eles não podem ter degradação de desempenho (como é o caso dos sistemas que controlam a linha de produção de veículos), já o **não crítico**, apesar de não desejado, se houver um pequeno atraso, não irá gerar tantos danos.

3.9 De computadores de grande porte

Sistema de grande porte é tipicamente utilizado por grandes corporações e, como características predominantes desses sistemas podem-se considerar a elevada capacidade de E/S, sistema em lote (*batch*), processamento de transações e tempo compartilhado. Exemplos de sistema de grande porte são: OS/390 e S/400.

3.10 Multiprocessadores

O sistema operacional desta categoria pode tratar múltiplas *CPUs* simultaneamente. Equipamento com multiprocessadores ou multinúcleos têm como objetivo principal melhorar a capacidade computacional dos equipamentos, trazendo melhor desempenho para o ambiente. Com o advento dos processadores multinúcleo, até sistemas operacionais voltados para computadores pessoais estão começando a lidar com multiprocessadores. Alguns exemplos de sistemas operacionais multiprocessados são: Windows, Linux, Solaris e AIX.

⁹ Disponível em: <<http://www.scribd.com/doc/13290978/socap01>>. Acesso em: 8 jun. 2011.

3.11 Portáteis

Voltados para computadores como os *PDA* (*Personal Digital Assistant*), telefones celulares e *smartphones*. Um ponto de destaque para os sistemas portáteis é que os *PDA*, celulares e *smartphones* não possuem disco rígido *multigigabyte*, fazendo grande diferença. Alguns exemplos de sistemas operacionais para portáteis são: Google Android, Apple IOS e Windows Phone.

4 VISÃO GERAL SOBRE HARDWARE DE COMPUTADORES

O *hardware* e o sistema operacional devem ser extremamente congruentes para que seja possível obter o melhor resultado desta combinação. Há pelo menos dois pontos de vista para analisarmos essa dupla: um seria a estratégia adotada pela Intel ao criar o processador 8080 e posteriormente solicitar para Gary Kildall que criasse um *software* para rodar neste *hardware*; já a outra forma seria a vontade de qualquer desenvolvedor imaginar que qualquer funcionalidade desejada seria prontamente atendida pelo *software*, porém sabemos que isso não é possível e, para seu melhor aproveitamento, faz-se necessária uma homogeneidade entre os desenvolvedores de *hardware* e *software*.

Independente das ideologias, turma do *hardware* ou do *software*, temos que ter conceitualmente um modelo pertinente à arquitetura do *hardware* de um computador pessoal para que possamos entender melhor os sistemas operacionais.

Conceitualmente, computadores simples podem ser abstraídos para um modelo esquemático representado na Figura 14. Temos o barramento que seria a nossa estrada possibilitando a comunicação entre os elementos, a memória e os dispositivos de E/S.

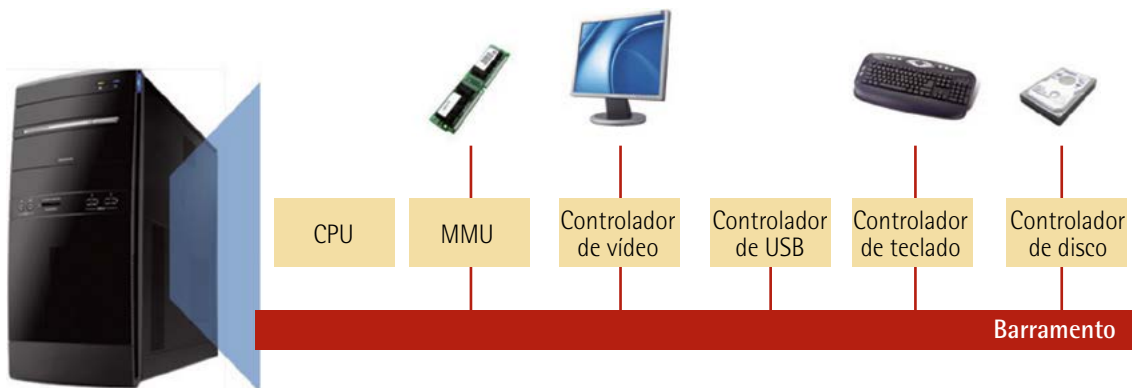


Figura 14 – Principais elementos de um computador pessoal simples

4.1 Processadores

A *CPU* traz das memórias instruções, decodifica, interpreta as instruções a serem executadas e as executa; a partir daí, busca as instruções subsequentes e processa o ciclo novamente até ter instruções a serem executadas.

Cada arquitetura de *CPU* tem um conjunto específico de instruções que pode executar. Portanto, uma *CPU RISC* não executa programas desenvolvidos para plataformas *CISC*, o oposto também é verdadeiro.

Todas as *CPUs* possuem registradores internos para armazenamento de variáveis importantes e de resultados temporários. Em adição aos registradores de propósito geral, usados para conter variáveis e resultados temporários, a maioria dos computadores possuem vários registradores especiais disponíveis de forma aparente para os programadores. O primeiro a se destacar é o contador de programa que contém o endereço de memória da próxima instrução a ser buscada, ou seja, ele é atualizado para apontar a próxima instrução.

Outro registrador especial é o ponteiro de pilha, que aponta para o topo da pilha da memória que contém uma estrutura para cada rotina chamada, mas que ainda não se finalizou. Uma estrutura de pilha da rotina contém os parâmetros de entrada, as variáveis locais e as variáveis temporárias que não são mantidas nos registradores.

Outro registrador especial é a *PSW (Program Status Word – palavra de estado do programa)*. Esse registrador contém os *bits* do código de condições, os quais são alterados pelas instruções de comparação, pelo nível de prioridade da *CPU*, pelo modo de execução e por vários outros *bits* de controle. Programas de usuários normalmente podem ler toda a *PSW*, mas em geral são capazes de alterar somente alguns de seus campos. Normalmente, o *PSW* conterà o ponteiro/endereço da próxima instrução para ser executada.

Toda vez que o sistema operacional compartilha o tempo de *CPU* e, por algum motivo, ele interrompe a execução de um programa, inicia ou reinicia outro, então o sistema operacional necessita salvar todos os registradores para que possam ser restaurados quando o programa for executado novamente.

CPUs modernas possuem recursos para executar mais de uma instrução em tempo concorrente, o que chamamos de *pipeline* (Figura 15). Elas podem executar uma busca, executar decodificação e, simultaneamente, a execução de instrução.



Figura 15 – Pipeline

Além do *pipeline*, temos o superescalar, esse tipo de processador possui múltiplas unidades de execução, conforme demonstrado na Figura 16. Portanto, duas ou mais instruções são buscadas, decodificadas e armazenadas temporariamente em um *buffer*, até que possam ser executadas.

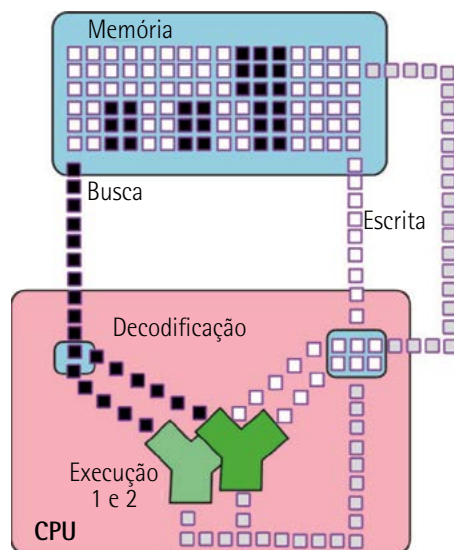


Figura 16 – Superescalar

4.1.1 Processadores *multithread* e multinúcleo

O presidente da Intel, Gordon E. Moore, fez uma prospecção que por muitos é considerada uma profecia, na qual o número de transistores dos *chips* teria um aumento de 100% a cada período de 18 meses. Como a prospecção se tornou um fato, essa teoria recebeu o nome de Lei de Moore. A Figura 17 demonstra graficamente essa lei.

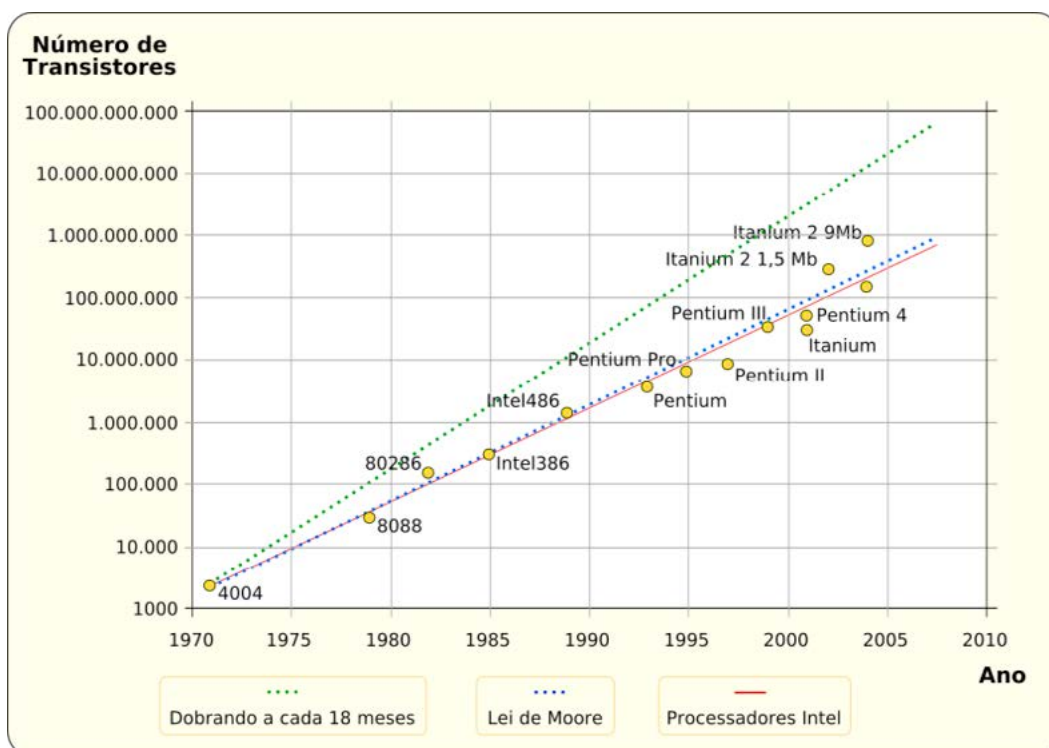


Figura 17 – Crescimento do número de transistores para processadores Intel (pontos) e Lei de Moore (Linha de cima = 18 meses, linha de baixo = 24 meses)

O Intel Pentium 4 e outros *chips* de processadores têm a propriedade chamada *multithreading* ou *hyperthreading*. No ambiente da ciência da computação, a execução de um *thread* é a menor unidade de processamento que pode ser atendida por um sistema operacional. A estrutura de *threads* e processos é diferente de um sistema operacional para outro, porém, na maior parte dos casos, um *thread*¹⁰ é contido num processo. Múltiplos *threads* podem existir num mesmo processo e compartilhar recursos como a memória, enquanto diferentes processos não compartilham esses recursos. Em particular, o *thread* de um processo compartilha as instruções e contextos. Analogicamente, múltiplo *thread* de um processo é o mesmo que múltiplos alunos lendo instruções em um mesmo livro, porém não necessariamente todos os alunos lendo a mesma página.

Se um dos processos precisa ler uma palavra a partir da memória demandando muitos ciclos de relógio, uma *CPU multithreading* não oferece paralelismo, então apenas um processo por vez é executado, mas o tempo de chaveamento é reduzido para a ordem de um nanossegundo.

O grande problema é que *multithreading* é compreendido pelo sistema operacional como uma *CPU*, portanto se um computador tiver, por exemplo, duas *CPUs*, cada uma com dois *threads*, o sistema operacional entenderá e terá que gerenciar como se existissem 4 *CPUs* no sistema.

Outra opção é a CPU com multinúcleo. Isso é o mesmo que fisicamente vemos um único *chip*, porém, internamente, temos múltiplos *chips* como se fossem várias CPUs. É importante destacar que sistemas com CPU multinúcleo requerem SO para multiprocessadores. A Figura 18 ilustra uma CPU com dois núcleos.

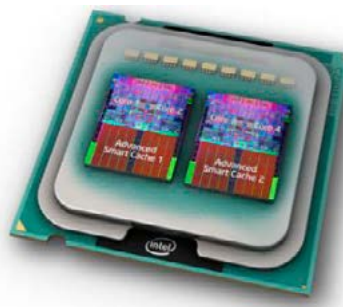


Figura 18 – CPU Dual Core (dois núcleos)



Saiba mais

<<http://www.training.com.br/lpmaia/multithread.pdf>>

4.2 Memória

Na teoria, a memória deveria ser mais performática do que a execução de uma instrução processada por uma CPU, pois, com isso, a CPU jamais teria que esperar pela resposta da memória; entretanto, na prática,

¹⁰ Thread = em cadeia; programa que consiste em várias seções menores independentes.

isso não é uma verdade. Para tentar resolver este problema, a abordagem contemporânea é construir o sistema de memória seguindo uma hierarquia de camadas conforme se demonstra na Figura 19.

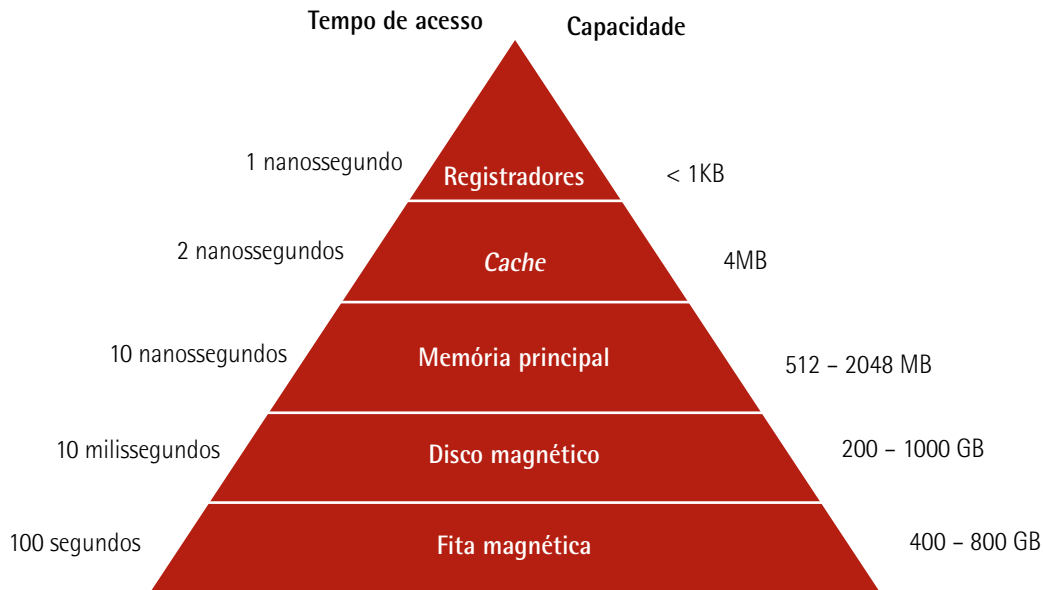


Figura 19 – Hierarquia de memória típica (números aproximados)

No topo da pirâmide, temos os registradores que estão contidos nas *CPUs*. Esses são feitos do mesmo material que as *CPUs* e são tão rápidos quanto elas. Desta forma, o tempo do registrador para a memória é desprezível e a capacidade de memória disponível nos registradores é de 32 x 32 *bits* para *CPU* de 32 *bits* e de 64 x 64 *bits* para *CPU* de 64 *bits*, sendo menos de 1 KB mesmo no caso das *CPUs* de 64 *bits*.

Seguindo, no segundo nível, vem a memória *cache*¹¹, que é controlada principalmente pelo *hardware* – conforme demonstrado na Figura 20. Atualmente, vários dispositivos como processadores, discos rígidos, placas-mãe, placas controladoras e outros possuem *cache*. A memória *cache* é mais usada para armazenar informações frequentemente mais usadas, porém memória *cache* é muito cara e não é tão abundante nos sistemas.

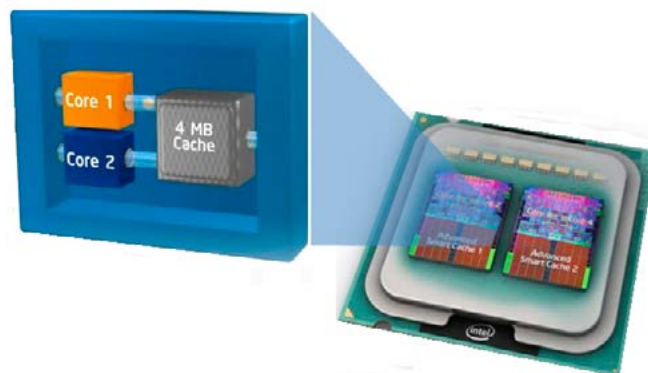


Figura 20 – Cache na CPU Dual Core

¹¹ Cache = pequena área de memória rápida, integrada à unidade central de processamento.

No próximo nível, é possível identificar a memória principal também conhecida como memória RAM (*Random Access memory* – *memória de acesso aleatório*), demonstrada na Figura 21. Todas as solicitações vindas da *CPU* e que não estão na memória *cache* são encaminhadas para a memória principal.

From Computer Desktop Encyclopedia
© 2007 The Computer Language Co. Inc.

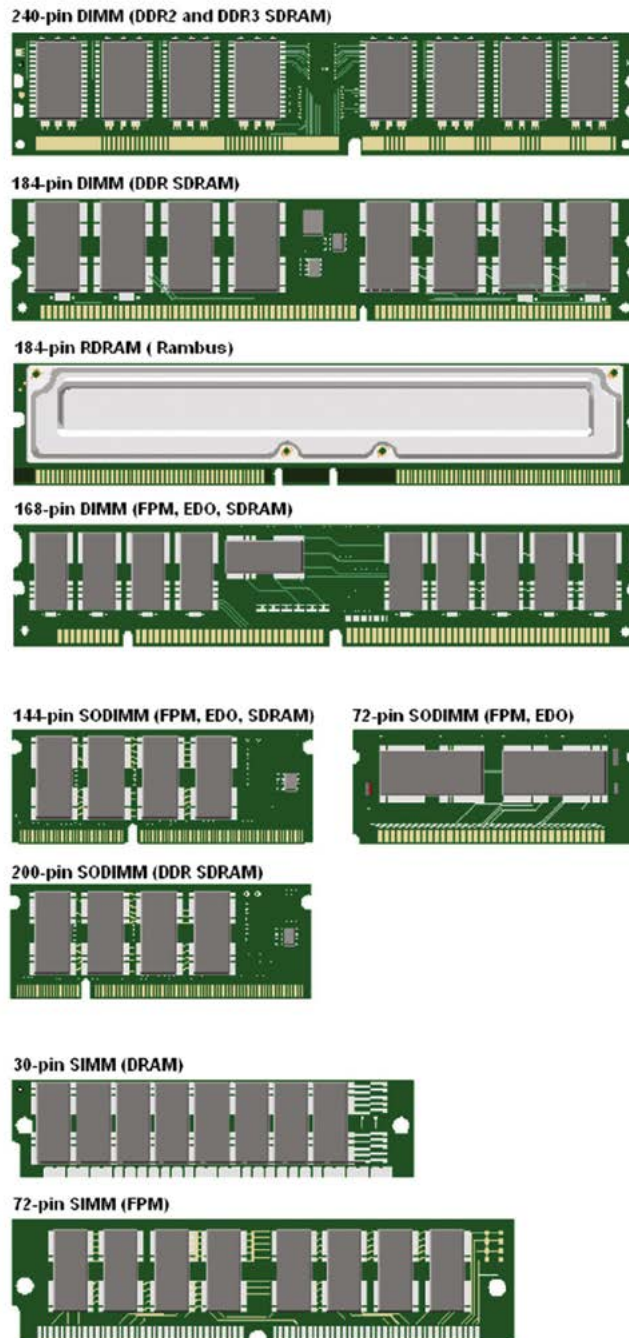


Figura 21 – Tipos de memória RAM

ROM (*Read Only Memory*) e CMOS são outros tipos de memórias também presentes nos sistemas computacionais. A memória ROM é normalmente usada pelos fabricantes para gravar códigos controladores do *hardware* e são previamente programadas em fábrica, não sendo possível sua alteração, a não ser que se usem equipamentos específicos para esta finalidade. A memória CMOS (Figura 22) é tipicamente usada para manter data e hora atualizadas e parâmetros de configuração do *hardware* como sequência de *boot* e outros mesmo que o computador seja desligado. A memória CMOS necessita de uma bateria (Figura 23) para manter seu conteúdo.



Figura 22 – ROM

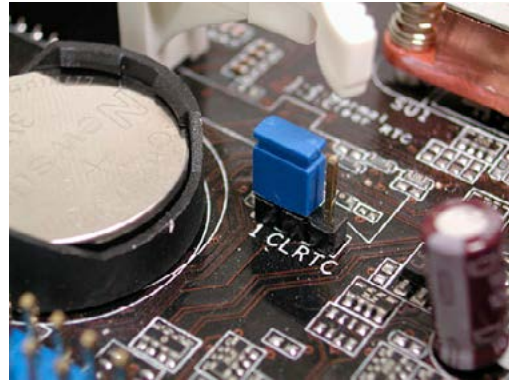


Figura 23 – Bateria da CMOS

4.3 Disco

Os discos magnéticos estão na camada logo abaixo da memória principal, sendo conhecidos como discos rígidos ou, em inglês, *hard disks* e também, de forma abreviada, como *HD*. O grande atrativo dos discos magnéticos em relação à memória é o preço bem menor se comparado R\$/GB, entretanto o contraponto é a velocidade de acesso que é muito mais lenta por ser um dispositivo mecânico, conforme ilustrado na Figura 24.



Figura 24 – Disco rígido aberto

Na estrutura de um disco magnético, temos como uma das principais partes o grupo de discos metálicos em que são gravadas as informações. Tipicamente, esses discos rodam a velocidades de 5400 a 10000 rpm (rotações por minuto) e têm uma média de transferência de 0.5 Gbit/s. Em ambientes projetados com servidores de rede e, principalmente, servidores de banco de dados transacional, normalmente, são instalados discos com velocidade de 15000 rpm, podendo-se atingir a média de transferência acima de 1.6Gbit/s. Os discos de 10000 ou 15000 rpm usam discos menores para mitigar grandes demandas de energia, entretanto isso acarreta que estes discos com maior capacidade de rotação por minuto possuam menos capacidade que os discos magnéticos de menos rpm.

As informações são escritas no disco em uma série de círculos que têm o mesmo centro. Cada cabeça pode ler e gravar uma região circular chamada trilha. Juntas, as trilhas de uma posição do braço formam um cilindro. Mover o braço entre cilindros próximos leva aproximadamente 1 ms e mover o braço de um determinado cilindro para outro distante leva em torno de 5 a 10 ms. A seguir, a Figura ilustrativa da unidade é a fita de *backup* LTO 5 que é a tecnologia mais atual durante a elaboração deste material.

4.4 Fitas

A fita magnética é o último tipo de memória na pirâmide ilustrada na Figura 25. Esse meio é muito utilizado como mídia de cópia de segurança (*backup*), transportando uma cópia daquilo que está nos discos magnéticos para fitas magnéticas. Com base nas normas de segurança ISO 27001, e até mesmo do Banco Central Brasileiro, é obrigatório o uso de sistemas de *backup* para garantir que a informação esteja disponível em caso de o sistema principal apresentar problema. Grandes sistemas utilizam robôs que controlam a troca de fitas, bem como *softwares* especiais para *backup* como o *Veritas Backup Exec*, *Data Protector*, *ARCserve*, *Tivoli Storage Management* e outros.



Figura 25 – LTO 5 (Capacidade 1.5 TB – 6 Gb/sec. SAS)

4.5 Dispositivos de E/S

Os dispositivos de E/S são geralmente os controladores e os dispositivos. O controlador é formado por um ou mais *chips* numa placa; estes controladores possuem a função de receber os comandos do sistema operacional e gerar as instruções mais adequadas para os dispositivos, fornecendo orientação exatamente de como o disco rígido armazena e acessa os dados, por exemplo. Os dispositivos possuem *interfaces* bastante padronizadas. Isso ajuda, porque a controladora IDE pode controlar qualquer disco IDE¹².

¹² IDE: *Integrated Drive Electronics*.

Entre o sistema operacional e o controlador, há outro *software* chamado *driver* de dispositivo. É função do *driver* de dispositivo se comunicar com o controlador emitindo comando e recebendo respostas. Os fabricantes de controladores devem fornecer *drivers* específicos para cada sistema operacional a que dão suporte.

Existem três maneiras diferentes para entrada e saída (E/S): o método mais simples é quando o programa de um usuário emite uma chamada de sistema, então o núcleo do sistema operacional a traduz em uma chamada ao *driver* relacionado, em seguida, o *driver* inicia a entrada e saída e fica em constante checagem se o dispositivo terminou a operação; quando a operação é finalizada, o *driver* coloca os dados onde são necessários; o sistema operacional, então, remete o controle para quem originou a chamada. Neste processo, a *CPU* fica ocupada durante a monitoração, se a operação de E/S terminou ou não.

No segundo método, o *driver* inicia o dispositivo e instrui que ele o informe quando terminar; durante este período de intervalo, o sistema operacional retoma o controle da *CPU* para executar outra tarefa. Assim que o controlador recebe a sinalização do final da transferência, ele gerará uma interrupção para sinalizar o término.

O DMA (*Direct Memory Access*) é o terceiro método para a implementação de entrada e saída. Neste cenário, é utilizado um *chip* especial de acesso direto à memória, controlando o fluxo de *bits* entre a memória e algum controlador sem intervenção constante da *CPU*. Nesse processo, a *CPU* configura o chip DMA informando a quantidade de *bytes* que devem ser transferidos, os endereços do dispositivo e de memória empregados, e a direção. Então, a execução fica a cargo do DMA. Assim que o DMA finalizar a tarefa, haverá uma interrupção.



Saiba mais

<<http://www.di.ufpb.br/raimundo/Hierarquia/Hierarquia>>

4.6 Barramento

O termo barramento é definido como elos de comunicação que consistem em um conjunto de vias.

Ao longo da evolução da arquitetura computacional, os barramentos foram tomando forma mais heterogênea e estruturada para as necessidades modernas. Os processadores e memórias foram ficando cada vez mais velozes, e o computador antigo, que inicialmente tinha somente um barramento, passou a não mais dar conta. Podemos fazer uma analogia com uma estreita rua de bairro onde temos somente casas e um tráfego predominante de pessoas, porém, com o passar do tempo, prédios são construídos e os moradores passam a se locomover usando carros. Em determinados horários, o fluxo de carros saindo do mesmo local e entrando no barramento, ou melhor, entrando na rua, irá gerar uma sobrecarga impossível de ser atendida por esta via; é então que surge a implantação de novas vias ou, pelo menos, o alargamento deste elo de comunicação.

Tipicamente, os sistemas modernos possuem oito barramentos, conforme a Figura 26, sendo: barramento de *cache*, local, memória, PCI¹³, SCSI¹⁴, USB¹⁵, IDE¹⁶ e ISA¹⁷, cada um com diferentes funções e taxas de transferência.

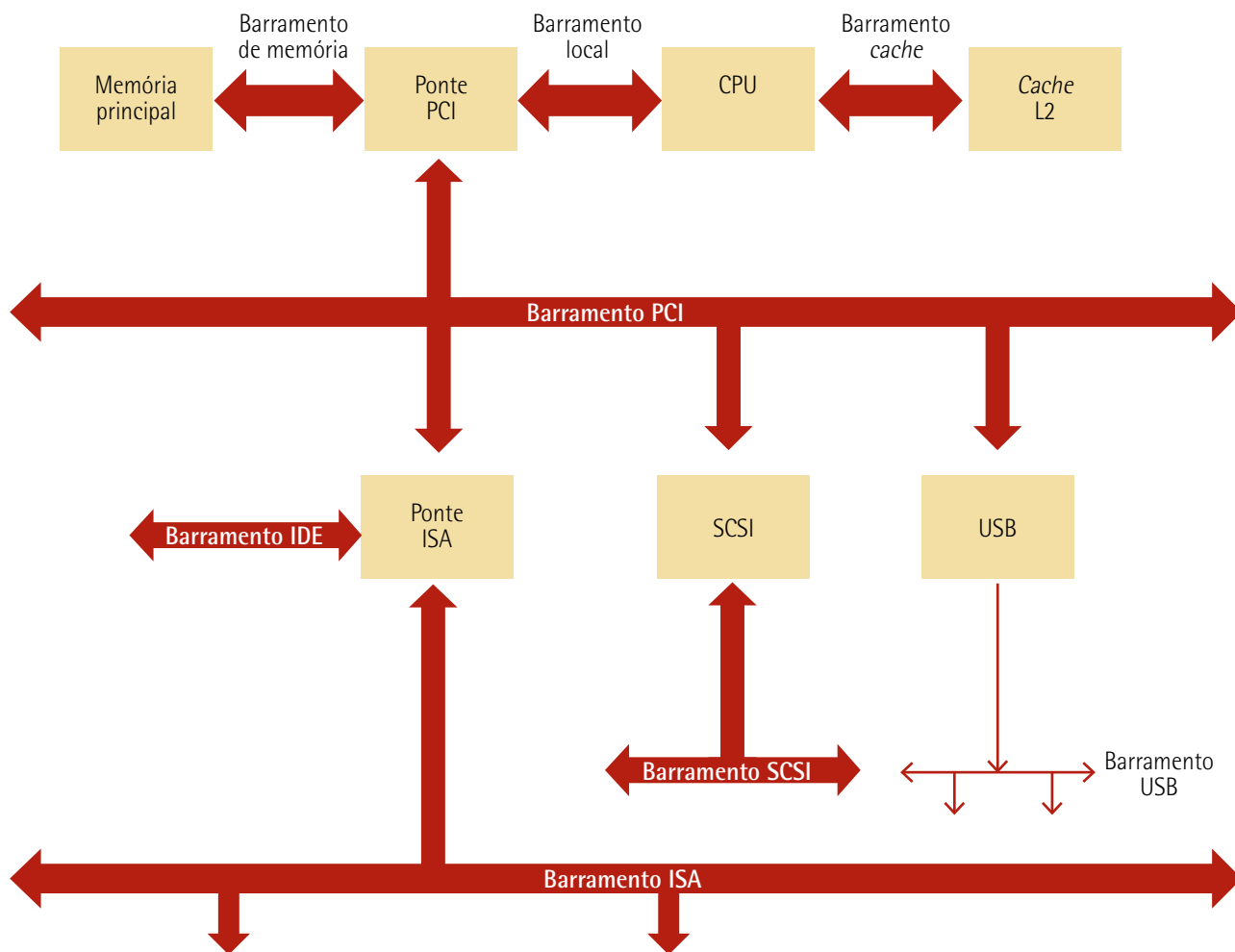


Figura 26 – Estrutura de barramentos de um computador típico



Saiba mais

<<http://www.pcisig.com/specifications/pciexpress/>>.

¹³ PCI: *Peripheral Component Interconnect* – Interconexão de Componentes Periféricos.

¹⁴ SCSI: *Small Computer System Interface* – Interface de Pequenos Sistemas de Computadores.

¹⁵ USB: *Universal Serial Bus* – Barramento Serial Universal.

¹⁶ IDE: *Integrated Drive Electronics* – Unidade Eletrônica Integrada.

¹⁷ ISA: *Industry Standard Architecture* – Arquitetura para Padrão Industrial.

PCI Express	
x1	BANDWIDTH Single direction: 2.5 Gbps/200 MBps Dual Directions: 5 Gbps/400 MBps
x4	BANDWIDTH Single direction: 10 Gbps/800 MBps Dual Directions: 20 Gbps/1.6 GBps
x8	BANDWIDTH Single direction: 20 Gbps/1.6 GBps Dual Directions: 40 Gbps/3.2 GBps
x16	BANDWIDTH Single direction: 40 Gbps/3.2 GBps Dual Directions: 80 Gbps/6.4 GBps

Source: IBM

Figura 27 – Tipos de conectores

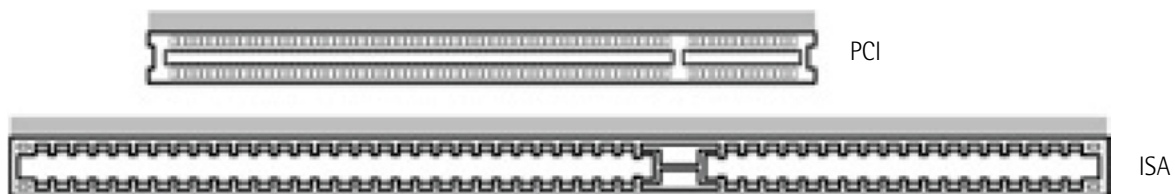


Figura 28 – Tipos de conectores

A *CPU* se comunica com o barramento PCI por meio do barramento local que, por sua vez, se comunica com a memória por intermédio de um barramento dedicado. Usando, por exemplo, um sistema Pentium com uma *cache* de nível 1 dentro do *chip* e uma *cache* de nível 2 muito maior que fica na parte externa do *chip* e é conectada à *CPU* pelo barramento *cache*.

Este sistema contém também três barramentos específicos: IDE, USB e SCSI. O barramento IDE, como descrito anteriormente, pode ser usado para conectar discos físicos e unidade de CD-ROM.

O padrão USB foi desenvolvido por um consórcio de empresas, entre as quais se destacam: Microsoft, Apple, Hewlett-Packard, NEC, Intel e Agere. O símbolo do USB é representado na Figura 29.



Figura 29 – Símbolo do USB – Tridente

A versão USB 1.0 suporta taxa de 1,5 Mbit/s a 12 Mbit/s (versão 1.1), a versão 2.0, velocidades na taxa de 480 Mbit/s (60MB/s), e a versão 3.0, de 4,8 Gbit/s (600MB/s).

Uma das características fundamentais para o sucesso do USB, principalmente pela demanda de agilidade de interoperabilidade, é que esta tecnologia compartilha o mesmo *driver* entre os seus dispositivos, tornando dispensável instalar um novo *driver* para cada novo dispositivo USB. Isso traz como benefício maior o fato de podermos instalar dispositivos USB no computador sem precisar reiniciá-lo, ou seja, *plug and play*¹⁸.

Como barramento de alta *performance*, temos o SCSI, destinado para discos de alto desempenho e demais dispositivos que demandam grande largura de banda.

O SCSI é baseado na *Shugart Associates System Interface* (SASI). A SASI foi desenvolvida em 1981 pela Shugart Associates em conjunto com a NCR Corporation. Em 1986, o Instituto Americano de Padrões Nacionais (ANSI – *American National Standards Institute*) aprovou o SCSI, uma versão modificada da SASI. O SCSI usa uma controladora para enviar e receber dados e energia de dispositivos SCSI, como HDs e impressoras.¹⁹

O SCSI é tipicamente usado para controlar discos com estrutura ordenada contendo elementos acessíveis individualmente referenciados por números, denominado RAID²⁰.

Outras tecnologias, como o serial-ATA (SATA), também podem ser usadas para este propósito. O RAID também aumenta a *fault tolerance* (tolerância a falhas) por meio de *mirroring* (espelhamento) e *parity* (paridade). O espelhamento faz uma cópia exata dos dados de um disco rígido em um segundo disco rígido. A paridade requer no mínimo três discos, e os dados são gravados sequencialmente para cada unidade, exceto o último. A última unidade armazena um número que representa a soma dos dados dos outros discos.²¹

¹⁸ *Plug and play*: termo inglês que significa que se pode conectar e jogar, ou seja, é rápido e fácil, sem necessidade de reiniciar o sistema para fazer o novo dispositivo ser reconhecido.

¹⁹ Disponível em: <<http://informatica.hsw.uol.com.br/scsi1.htm>>. Acesso em: 8 jun. 2011.

²⁰ RAID: *Redundant Array of Independent Disks*.

²¹ Disponível em: <<http://informatica.hsw.uol.com.br/scsi1.htm>>. Acesso em: 8 jun. 2011.

Na Planilha 1, são demonstrados diversos tipos de SCSI. Os mais lentos não estão em uso, mas foram incluídos nesta lista para efeito de comparação.

Nome	Especificação	Número de dispositivos	Largura de barramento	Velocidade de barramento	MB/s
SCSI –Assíncrono	SCSI-1	8	8 bits	5 MHz	4 MB/s
SCSI – Síncrono	SCSI-1	8	8 bits	5 MHz	5 MB/s
Wide	SCSI-2	16	16 bits	5 MHz	10 MB/s
Fast	SCSI-2	8	8 bits	10 MHz	10 MB/s
Fast/Wide	SCSI-2	16	16 bits	10 MHz	20 MB/s
Ultra	SCSI-3 – SPI	8	8 bits	20 MHz	20 MB/s
Ultra/Wide	SCSI-3 – SP	8	16 bits	20 MHz	40 MB/s
Ultra2	SCSI-3 SPI-2	8	8 bits	40 MHz	40 MB/s
Ultra2/Wide	SCSI-3 SPI-2	16	16 bits	40 MHz	80 MB/s
Ultra3	SCSI-3 SPI-3	16	16 bits	40 MHz	160 MB/s
Ultra320	SCSI-3 SPI-4	16	16 bits	80 MHz	320 MB/s

Planilha 1 – Tipos de SCSI

Objetivando aprimorar e possibilitar que o sistema operacional tivesse possibilidades reais de gerenciar de forma consolidada os diversos dispositivos de E/S, e baseado no conceito similar previamente implantado pela Apple no Macintosh, a tecnologia *plug and play* para computador pessoal foi projetada pela Intel e Microsoft.

O *plug and play* facilitou a vida dos profissionais e usuários caseiros que se aventuravam a tentar instalar placas no sistema. Naquela época, que não deixa saudades, uma simples instalação de uma nova placa de som, por exemplo, poderia fazer com que outro dispositivo parasse de funcionar de forma quase que inexplicável e isso ocorria por conflito de interrupção e endereços de registradores.

Naquela época, era comum encontrar placas controladoras com componentes chamados *jumpers*, conforme a Figura 26, que tinham a finalidade de possibilitar ao usuário configurar um nível de interrupção e endereço de dispositivos de E/S que não conflitassem com outros dispositivos já existentes no sistema.

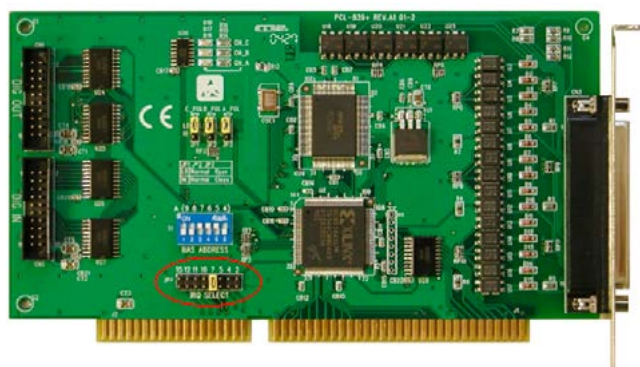


Figura 30 – Placa controladora com *jumpers* para configuração de IRQ



Saiba mais

<http://www.gta.ufrj.br/grad/01_1/barramento>



Resumo

No modelo abstrato de sistema operacional, temos que ter alguns entendimentos a respeito de processadores, memórias e dispositivos de entrada e saída que são conectadas por barramento.

Além da estrutura de *hardware* em que está montado o sistema operacional, os conceitos típicos sobre os quais todos os sistemas operacionais são construídos são:

- Processos.
- Gerenciamento de memória.
- Gerenciamento de E/S.
- Sistema de arquivos.
- Segurança.

Os sistemas operacionais podem ser vistos por dois prismas:

- Dispositivo gerenciador de recursos.
- Máquina estendida.

Todos os sistemas operacionais possuem, na sua essência, o conjunto de chamadas de sistema com que eles podem lidar. Dentre elas, destacamos:

- Criação e finalização de processos.
- Leitura e escrita de arquivos.
- Gerenciamento de diretórios.
- Chamadas diversas.

Os sistemas operacionais são, geralmente, entre outros, estruturados conforme as classificações a seguir:

- Sistema monolítico.
- Hierarquia de camadas.
- Micronúcleo.
- Sistema de máquina virtual.
- Exonúcleo: tem como finalidade permitir que uma aplicação solicite uma região específica da memória, simplesmente para assegurar que os recursos pedidos estão disponíveis e que o programa tem direito a acessá-los.
- Modelo cliente servidor.



Exercícios

Questão 1. (ENADE 2008) Uma alternativa para o aumento de desempenho de sistemas computacionais é o uso de processadores com múltiplos núcleos, chamados *multicores*. Nesses sistemas, cada núcleo, normalmente, tem as funcionalidades completas de um processador, já sendo comuns, atualmente, configurações com dois, quatro ou mais núcleos. Com relação ao uso de processadores multicores e sabendo que *threads* são estruturas de execução associadas a um processo, que compartilham suas áreas de código e dados, mas mantêm contextos independentes, analise as seguintes asserções.

- Ao dividirem suas atividades em múltiplas *threads*, que podem ser executadas paralelamente, aplicações podem se beneficiar mais efetivamente dos diversos núcleos dos processadores multicores.
- O sistema operacional nos processadores *multicores* pode alocar os núcleos existentes para executar simultaneamente diversas sequências de código, sobrepondo suas execuções e, normalmente, reduzindo o tempo de resposta das aplicações às quais estão associadas.

Acerca dessas asserções, assinale a opção correta.

- As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira.
- A primeira asserção é uma proposição verdadeira e a segunda, uma proposição falsa.

D) A primeira asserção é uma proposição falsa e a segunda, uma proposição verdadeira.

E) Tanto a primeira quanto a segunda asserções são proposições falsas.

Resposta correta: alternativa C.

Análise das alternativas

A) Alternativa incorreta.

Justificativa: ao afirmar que existe uma sobreposição de execuções, o autor induz o leitor ao erro, pois, apesar do processamento estar distribuído entre as CPUs, ainda existe um compartilhamento de recursos, que devem manter um sincronismo a fim de se evitar *Deadlocks*, portanto, a alternativa possui dois erros: afirmar que as duas asserções são verdadeiras e ainda afirmar que a segunda asserção justifica a primeira de forma correta.

B) Alternativa incorreta.

Justificativa: o erro encontra-se na falsidade da segunda asserção.

C) Alternativa correta.

Justificativa: justificável pelo entendimento do funcionamento básico de *Threads*.

D) Alternativa incorreta.

Justificativa: o erro está justamente em se questionar a veracidade da primeira assertiva.

E) Alternativa incorreta.

Justificativa: o erro está justamente em se questionar a veracidade da primeira assertiva.

Questão 2. (ENADE 2008) Com relação às diferentes tecnologias de armazenamento de dados, julgue os itens a seguir.

I. Quando a tensão de alimentação de uma memória ROM é desligada, os dados dessa memória são apagados. Por isso, esse tipo de memória é denominado volátil.

II. O tempo de acesso à memória RAM é maior que o tempo de acesso a um registrador da unidade central de processamento (UCP).

III. O tempo de acesso à memória *cache* da UCP é menor que o tempo de acesso a um disco magnético.

IV. O tempo de acesso à memória *cache* da UCP é maior que o tempo de acesso à memória RAM.

Estão certos apenas os itens:

- A) I e II.
B) I e III.
C) II e III.
D) II e IV.
E) III e IV.

Resolução desta questão na Plataforma.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.