

I. SISTEMAS OPERACIONAIS.....	2
1. INTRODUÇÃO	2
2. FUNCIONAMENTO	2
AUTOTESTE.....	3
BOOT.....	3
3. TIPOS DE SISTEMAS OPERACIONAIS.....	4
3.1 SISTEMAS MONOPROGRAMÁVEIS / MONOTAREFA	5
3.2 SISTEMAS MULTIPROGRAMÁVEIS/MULTITAREFA	5
3.2.1 SISTEMAS BATCH.....	6
3.2.2 SISTEMAS DE TEMPO COMPARTILHADO.....	6
3.2.3 SISTEMAS DE TEMPO REAL.....	7
3.3 SISTEMAS COM MÚLTIPLOS PROCESSADORES	7
3.3.1 SISTEMAS FORTEMENTE ACOPLADOS	8
3.3.1.1 SISTEMAS ASSIMÉTRICOS	9
3.3.1.2 SISTEMAS SIMÉTRICOS	9
3.3.2 SISTEMAS FRACAMENTE ACOPLADOS	9
3.3.2.1 SISTEMAS OPERACIONAIS DE REDE.....	10
3.3.2.2 SISTEMAS OPERACIONAIS DISTRIBUÍDOS	10
4. DOS - DISK OPERATING SYSTEM.....	11
MS-DOS	11
5. CP/M - CONTROL PROGRAM FOR MICROCOMPUTERS	12
6. UNIX.....	13

I. SISTEMAS OPERACIONAIS

1. INTRODUÇÃO

Originalmente, os sistemas operacionais visavam somente gerenciar uma das operações mais complexas de entrada/saída:

- a comunicação com várias unidades de disco

Mas o sistema operacional rapidamente transformou-se numa ponte completa entre o PC e o programa executado nele.

Sem um sistema operacional, cada programador teria que inventar, a partir de um esboço, como o programa faria certas funções, como por exemplo:

- apresentar o texto e a parte gráfica na tela
- enviar os dados à impressora
- ler e gravar arquivos no disco
- e uma série de outras funções que ligam o programa ao hardware.

Entretanto, um sistema operacional faz mais do que facilitar a vida dos programadores. O sistema operacional cria uma plataforma comum a todos os programas utilizados. Sem ele, você provavelmente não seria capaz de gravar arquivos criados em programas diferentes no mesmo disco, porque cada um teria sua própria maneira de armazenar tais arquivos. O sistema operacional também fornece ferramentas para gerenciar todas as tarefas que você deseja executar fora do aplicativo, como por exemplo:

- apagar e copiar arquivos
- ver a lista dos arquivos contidos no disco
- executar uma série de comandos em um arquivo de lote (batch file).

2. FUNCIONAMENTO

O sistema operacional não trabalha isolado. Ele depende não somente da cooperação de outros programas mas da combinação sem conflitos com o BIOS. Como será visto em seguida, quando algumas partes do sistema operacional são carregadas do disco, elas são acrescentadas ao BIOS, unidas através de drivers de dispositivos, e todas elas executam rotinas de funções de hardware.

O sistema operacional, na verdade, é formado por todos esses três componentes. Juntos, o BIOS, os drivers de dispositivos e o sistema operacional executam tantas funções que é impossível mostrar essa complexidade em algumas páginas de ilustrações. É simplista pensar no sistema operacional como sendo somente os arquivos contidos num certo disco que acompanha o PC.

A princípio para podermos carregar o sistema operacional devemos passar por algumas etapas, como visto a seguir:

Autoteste

Ao ligar o computador, nada de importante parece acontecer durante alguns segundos. Na verdade, ele está passando por um complexo conjunto de operações para verificar se todos os componentes estão funcionando de acordo e avisar se algo não está indo bem.

Esta operação é a primeira etapa de um processo um tanto complicado, chamado de *partida* (boot-up), ou simplesmente, *boot*. Num PC, este processo é necessário porque a máquina precisa ter uma forma de despertar todos os seus componentes e assim carregar o sistema operacional sem problemas. Em seguida, o sistema operacional encarrega-se de tarefas um pouco mais complicadas, que o código de boot não consegue gerenciar sozinho, inclusive a de fazer o equipamento interagir com os programas.

Mas mesmo antes de o PC tentar carregar um sistema operacional, ele precisa ter certeza de que todos os seus componentes estão rodando e que a CPU e a memória estão funcionando corretamente. Este é o trabalho do Autoteste inicial ou **POST**.

Boot

O computador pessoal não pode fazer nada de aproveitável se não possuir um sistema operacional (o programa que permite ao PC utilizar os demais programas). Mas antes de executar o sistema operacional, o PC precisa de uma forma de carregá-lo na memória de acesso aleatório (RAM). Esta forma é conhecida como bootstrap, ou apenas boot, um pequeno código que faz parte permanente do PC.

O bootstrap possui este nome porque permite que o PC execute algo por si só, sem o auxílio de um sistema operacional externo. Na verdade, a operação de boot não faz muita coisa. Ela possui somente duas funções:

- executar o POST ou Autoteste inicial (descrito anteriormente)
- pesquisar as unidades de disco para o sistema operacional

Ao terminar as duas funções, inicia-se o processo de leitura dos arquivos do sistema operacional e a cópia destes arquivos para a memória de acesso aleatório (RAM).

Por que os PCs usam esse tipo de método? Por que não integrar o sistema operacional no PC? Alguns computadores mais especializados ou mais básicos o fazem. Os primeiros modelos, utilizados principalmente para jogos, como Atari 400 e 800, e o recente palmtop LX95 da Hewlett-Packard possuem um sistema operacional permanente. O LX95 inclui até mesmo um aplicativo, o Lotus 1-2-3, num microchip especial. Mas, na maioria das vezes, o sistema operacional é carregado do disco por dois motivos.

É mais fácil atualizar a versão do sistema se este for carregado através de disco. Quando uma empresa como a Microsoft (fabricante do MS-DOS, WINDOWS-95 e 98, sistemas operacionais mais utilizado em PCs) deseja acrescentar novos recursos ou eliminar erros de programação (bugs), ela simplesmente lança um novo conjunto de discos com a nova versão. Às vezes, somente um único arquivo é inserido para corrigir uma falha no sistema operacional. É mais barato para a Microsoft distribuir disquetes ou CD-ROMs do que projetar um microchip que contenha o sistema operacional. E bem mais fácil para os usuários instalar um novo sistema operacional em disquetes do que trocar chips. Outra razão para se carregar o sistema operacional a partir de discos é permitir que o usuário possa escolher o sistema.

Embora a maioria dos PCs baseados em microprocessadores fabricados pela Intel utilizem o MS-DOS, há sistemas operacionais alternativos como o OS/2, o DR-DOS e o UNIX. Na configuração de alguns PCs, pode-se até definir qual sistema operacional será utilizado toda vez que se liga o computador.

OBS.: O nome do sistema operacional mais conhecido para PCs baseados no microprocessador INTEL : MS-DOS (**M**icro**S**oft **D**isk **O**perating **S**ystem), significa *disco de sistema operacional da MicroSoft* .

3. TIPOS DE SISTEMAS OPERACIONAIS

Os tipos de sistemas operacionais e sua evolução estão intimamente relacionados com a evolução do hardware e das aplicações por ele suportadas. Muitos termos inicialmente introduzidos para definir conceitos e técnicas foram substituídos por outros, na tentativa de refletir uma nova maneira de interação ou processamento. Isto fica muito claro quando tratamos da unidade de execução do processador. Inicialmente, os termos **programa** ou **job** eram os mais utilizados, depois surgiu o conceito de **processo** e **subprocesso** e, mais recentemente, os conceitos de **tarefa** e de **thread**.

A evolução dos sistemas operacionais para computadores pessoais e estações de trabalho popularizou vários conceitos e técnicas, antes só conhecidos em ambientes de grande porte. A nomenclatura, no entanto não se manteve a mesma. Surgiram novos termos para conceitos já conhecidos, que foram apenas adaptados para uma nova realidade.

Neste capítulo abordaremos os diversos tipos de sistemas operacionais, suas características, vantagens e desvantagens (Fig. 1).

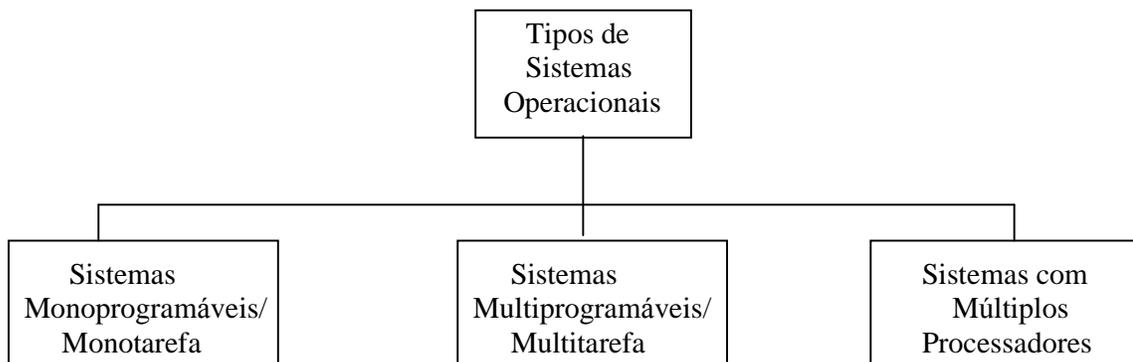


Fig. 1 - Tipos de sistemas operacionais.

3.1 *SISTEMAS MONOPROGRAMÁVEIS / MONOTAREFA*

Os primeiros sistemas operacionais eram tipicamente voltados para a execução de único programa (job). Qualquer outro programa para ser executado, deveria aguardar o término do programa corrente. Os sistemas monoprogramáveis, como vieram a ser conhecidos, se caracterizam por permitir que o processador, a memória e os periféricos permaneçam exclusivamente dedicados à execução de um único programa.

Os sistemas monoprogramáveis estão tipicamente relacionados ao surgimento dos mainframes. Posteriormente, com a introdução dos primeiros computadores pessoais e estações de trabalho, estes tipos de sistemas voltaram a ser desenvolvidos para atender máquinas que, na época, eram utilizadas por apenas um usuário. Os sistemas monotarefa, como também são chamados, se caracterizam por permitir que todos os recursos do sistema fiquem exclusivamente dedicados a uma única tarefa.

Neste tipo de sistema, enquanto um programa aguarda por um evento, como a digitação de um dado, o processador permanece ocioso, sem realizar qualquer tipo de processamento. A memória é subutilizada caso o programa não a preencha totalmente, e os periféricos, como discos e impressoras, estão dedicados a um único usuário, nem sempre utilizados de forma integral.

Comparados a outros sistemas, os sistemas monoprogramáveis/monotarefa são de simples implementação, não existindo muita preocupação com problemas de proteção.

3.2 *SISTEMAS MULTIPROGRAMÁVEIS/MULTITAREFA*

Os sistemas multiprogramáveis, que vieram a substituir os monoprogramáveis, são mais complexos e eficientes. Enquanto em sistemas monoprogramáveis existe apenas um programa utilizando seus diversos recursos, nos multiprogramáveis vários programas dividem esses mesmos recursos.

Por exemplo, enquanto um programa espera por uma operação de leitura ou gravação em disco, outros programas podem estar sendo processados neste mesmo intervalo de tempo. Nesse caso, podemos observar o compartilhamento da memória e do processador. O sistema operacional se preocupa em gerenciar o acesso concorrente aos seus diversos recursos, como memória, processador e periféricos, de forma ordenada e protegida, entre os diversos programas.

As vantagens do uso de sistemas multiprogramáveis são o aumento da produtividade dos seus usuários e a redução de custos, a partir do compartilhamento dos diversos recursos do sistema.

Nos sistemas monoprogramáveis, apenas um único usuário pode interagir com o sistema, enquanto nos multiprogramáveis é permitido que mais de um usuário o utilize.

A partir do número de usuários que interagem com o sistema, podemos classificar os sistemas multiprogramáveis como monousuário e multiusuário.

O conceito de sistema multiprogramável está tipicamente associado aos mainframes e minicomputadores, onde existe a idéia do sistema sendo utilizado por vários usuários (multiusuário). No mundo dos computadores pessoais e estações de trabalho, apesar de existir apenas um único usuário interagindo com o sistema (monousuário), é possível que ele execute diversas tarefas concorrentemente ou mesmo simultaneamente. Os sistemas multitarefa, como também são chamados, se caracterizam por permitir que o usuário edite um texto, imprima um arquivo, copie um arquivo pela rede e calcule uma planilha.

Os sistemas multiprogramáveis/multitarefa podem ser classificados pela forma com que suas aplicações são gerenciadas, podendo ser divididos em **sistemas batch**, **de tempo compartilhado** ou **de tempo real**. Um sistema operacional pode suportar um ou mais desses tipos de processamento.

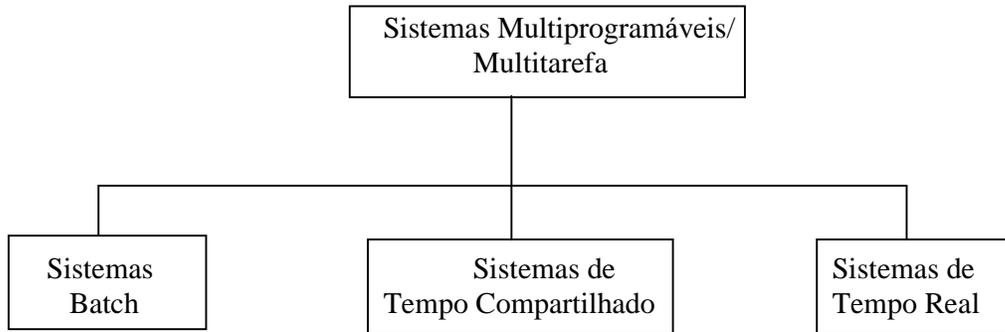


Fig. 2 - Tipos de sistemas multiprogramáveis/ multitarefa

3.2.1 SISTEMAS BATCH

Os sistemas batch (lote) foram os primeiros sistemas multiprogramáveis a serem implementados e caracterizam-se por terem seus programas, quando submetidos, armazenados em disco ou fita, onde esperam para ser executados seqüencialmente. Normalmente, os programas, também chamados de jobs, não exigem interação com os usuários, lendo e gravando dados em discos ou fitas. Alguns exemplos de aplicações originalmente processadas em batch são compilações, linkedições, sorts, backups e todas aquelas onde não é necessária a interação com o usuário.

Esses sistemas, quando bem projetados, podem ser bastante eficientes, devido à melhor utilização do processador. Entretanto, podem oferecer tempos de resposta longos, em face do processamento puramente seqüencial e com uma variação alta dos seus tempos de execução.

3.2.2 SISTEMAS DE TEMPO COMPARTILHADO

Os sistemas de tempo compartilhado (time-sharing) permitem a interação dos usuários com o sistema, basicamente através de terminais que incluem vídeo, teclado e mouse. Dessa forma, o usuário pode interagir diretamente com o sistema em cada fase do desenvolvimento de suas aplicações e, se preciso, modificá-las imediatamente. Devido a esse tipo de interação, os sistemas de tempo compartilhado também ficaram conhecidos como sistemas on-line.

Esses sistemas possuem uma linguagem de controle que permite ao usuário comunicar-se diretamente com o sistema operacional para obter informações, como, por exemplo, verificar os arquivos que possui armazenados em disco. O sistema, normalmente, responde em poucos segundos à maioria desses comandos.

Para cada usuário, o sistema operacional aloca uma fatia de tempo (time-slice) do processador. Caso o programa do usuário não esteja concluído nesse intervalo de tempo, ele é substituído por um de outro usuário, e fica esperando por uma nova fatia de tempo. Não só o processador é compartilhado nesse sistema, mas também a memória e os periféricos, como discos e impressoras. O sistema cria para o usuário um ambiente de trabalho próprio, dando a impressão de que todo o sistema está dedicado, exclusivamente, a ele.

Sistemas de tempo compartilhado são de implementação complexa, porém, se levado em consideração o tempo de desenvolvimento e depuração de uma aplicação, aumentam consideravelmente a produtividade dos seus usuários, reduzindo os custos de utilização do sistema.

3.2.3 SISTEMAS DE TEMPO REAL

Os sistemas de tempo real (real-time) são bem semelhantes em implementação aos sistemas de tempo compartilhado. A maior diferença é o tempo de resposta exigido no processamento das aplicações.

Enquanto em sistemas de tempo compartilhado o tempo de resposta pode variar sem comprometer as aplicações em execução, nos sistemas de tempo real os tempos de resposta devem estar dentro de limites rígidos, que devem ser obedecidos, caso contrário poderão ocorrer problemas irreparáveis.

Nos sistemas de tempo real não existe a idéia de fatia de tempo, utilizada pelos sistemas de tempo compartilhado. Um programa detém o processador o tempo que for necessário, ou até que apareça outro prioritário em função de sua importância no sistema. Esta importância ou prioridade de execução é controlada pela própria aplicação e não pelo sistema operacional, como nos sistemas de tempo compartilhado.

Esses sistemas, normalmente, estão presentes em controle de processos, como no monitoramento de refinarias de petróleo, controle de tráfego aéreo, de usinas termoeletricas e nucleares, ou em qualquer aplicação onde o tempo de resposta é fator fundamental.

3.3 SISTEMAS COM MÚLTIPLOS PROCESSADORES

Os sistemas com múltiplos processadores caracterizam-se por possuir duas ou mais CPUs interligadas, trabalhando em conjunto. Um fator-chave no desenvolvimento de sistemas operacionais com múltiplos processadores é a forma de comunicação entre as CPUs e o grau de compartilhamento da memória e dos dispositivos de entrada e saída. Em função desses fatores, podemos classificar os sistemas em **fortemente acoplados** ou **fracamente acoplados**.

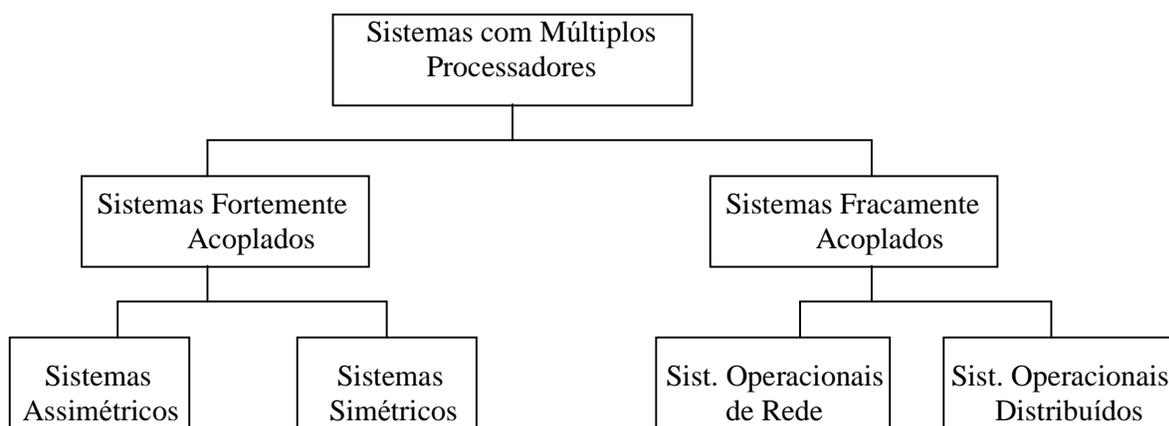


Fig. 3 - Sistemas com múltiplos processadores

Em sistemas fortemente acoplados existem dois ou mais processadores (multiprocessadores) compartilhando uma única memória e controlados por apenas um único sistema operacional. Tais sistemas são geralmente utilizados no processamento de aplicações que fazem uso intensivo da CPU, onde o processamento é voltado para a solução de um único problema.

Os sistemas fracamente acoplados caracterizam-se por possuir dois ou mais sistemas de computação (multicomputadores), conectados através de linhas de comunicação. Cada sistema funciona de forma independente, possuindo seu(s) próprio(s) processador(es), memória e dispositivos. A utilização de sistemas fracamente acoplados já é caracterizada pelo processamento distribuído entre os seus diversos processadores.

A grande diferença entre os dois tipos de sistemas é que em sistemas fortemente acoplados existe apenas um espaço de endereçamento (memória) compartilhado por todos os processadores, enquanto nos fracamente acoplados cada sistema tem sua própria memória individual. Além disso, a taxa de transferência entre CPUs e memória em sistemas fortemente acoplados é normalmente maior que nos fracamente acoplados.

Os conceitos utilizados no projeto de sistemas com múltiplos processadores incorporam os mesmos princípios apresentados para a multiprogramação, além de outras características e vantagens. Sistemas com múltiplos processadores possibilitam aumentar a capacidade computacional com menor custo (escalabilidade), além de permitir a reconfiguração e o balanceamento do sistema.

OBS.: A reconfiguração é a capacidade de um sistema poder continuar o processamento, mesmo se um dos processadores falhar ou parar de funcionar, embora com menor capacidade de computação. O balanceamento possibilita distribuir a carga de processamento entre os diversos sistemas da configuração, melhorando o desempenho como um todo.

3.3.1 SISTEMAS FORTEMENTE ACOPLADOS

Nos sistemas fortemente acoplados (tightly coupled) existem vários processadores compartilhando uma única memória e gerenciados por apenas um sistema operacional. Múltiplos processadores permitem que vários programas sejam executados ao mesmo tempo, ou que um programa seja dividido em subprogramas, para execução simultânea em mais de um processador. Dessa forma, é possível ampliar a capacidade de computação de um sistema, adicionando-se apenas novos processadores, com um custo muito inferior à aquisição de outros computadores.

A evolução desses sistemas se deve, em grande parte, ao elevado custo de desenvolvimento de processadores cada vez mais rápidos, utilizados nos sistemas com apenas um processador. Em função disso, passou-se a dar ênfase à arquitetura de computadores com múltiplos processadores, em vez de processadores de alta velocidade. Inicialmente, as configurações se limitavam a poucos processadores, mas, com o avanço tecnológico, já é possível desenvolver sistemas com milhares de processadores.

Com o multiprocessamento, novos problemas de concorrência foram introduzidos, pois vários processadores podem estar acessando as mesmas áreas de memória. Além disso, existe o problema de organizar de forma eficiente os processadores, a memória e os periféricos.

Uma conseqüência do multiprocessamento foi o surgimento dos computadores voltados, principalmente, para processamento científico, aplicado, por exemplo, ao desenvolvimento aeroespacial, prospecção de petróleo, simulações, processamento de imagens e CAD. A princípio qualquer aplicação que faça uso intensivo da CPU será beneficiada pelo acréscimo de processadores ao sistema.

Os sistemas fortemente acoplados podem ser divididos conforme a simetria existente entre seus processadores, ou seja, se todos os processadores podem executar ou não as mesmas funções.

Os sistemas fortemente acoplados assimétricos caracterizam-se por possuir um processador primário, responsável pelo controle dos demais processadores (secundários) e pela execução do sistema operacional. Os processadores secundários apenas processam programas de usuários e, sempre que necessitam de um serviço do sistema, solicitam ao processador primário. Já nos sistemas fortemente acoplados simétricos, todos os processadores têm as mesmas funções, podendo executar o sistema operacional independentemente.

3.3.1.1 SISTEMAS ASSIMÉTRICOS

Na organização assimétrica ou mestre / escravo (master/slave), somente um processador (mestre) pode executar serviços do sistema operacional, como, por exemplo, realizar operações de entrada/saída. Sempre que um processador do tipo escravo precisar realizar uma operação de entrada/saída, terá de requisitar o serviço ao processador mestre. Dependendo do volume de operações de entrada/saída destinadas aos processadores escravos, o sistema pode se tornar ineficiente, devido ao elevado número de interrupções que deverão ser tratadas pelo mestre.

Outra consequência dessa organização é que, se o processador mestre falhar, todo o sistema ficará incapaz de continuar o processamento. Neste caso, o sistema deve ser reconfigurado, fazendo um dos processadores escravos assumir o papel de mestre.

Mesmo sendo uma organização simples de implementar e quase uma extensão dos sistemas multiprogramáveis, esse tipo de sistema não utiliza eficientemente o hardware, devido à assimetria dos processadores, que não realizam as mesmas funções.

3.3.1.2 SISTEMAS SIMÉTRICOS

O multiprocessamento simétrico (Symmetric Multiprocessing - SMP), ao contrário da organização mestre/escravo, implementa a simetria dos processadores, ou seja, todos os processadores realizam as mesmas funções. Apenas algumas poucas funções ficam a cargo de um único processador, como, por exemplo, a inicialização (boot) do sistema.

Como vários processadores estão utilizando, independentemente, a mesma memória e o mesmo sistema operacional, é natural a ocorrência de acessos simultâneos às mesmas áreas de memória. A solução desses conflitos fica a cargo do hardware e do sistema operacional.

No processamento simétrico, um programa pode ser executado por qualquer processador, inclusive por vários processadores ao mesmo tempo (paralelismo). Além disso, quando um processador falha, o sistema continua em funcionamento sem nenhuma interferência manual, porém com menor capacidade de computação.

Inicialmente, os sistemas com múltiplos processadores estavam limitados aos sistemas de grande porte, restritos ao ambiente universitário e às grandes corporações. Com a evolução dos computadores pessoais e estações de trabalho os sistemas multitarefa evoluíram para permitir a existência de vários processadores no modelo simétrico.

Os sistemas simétricos são mais poderosos que os assimétricos, permitindo um melhor balanceamento do processamento e das operações de entrada/saída, apesar de sua implementação ser bastante complexa.

3.3.2 SISTEMAS FRACAMENTE ACOPLADOS

Os sistemas fracamente acoplados caracterizam-se por possuir dois ou mais sistemas de computação interligados, sendo que cada sistema possui o seu próprio sistema operacional, gerenciando os seus recursos, como processador, memória e dispositivos de entrada/saída.

Até meados da década de 80, os sistemas operacionais e as aplicações suportadas por eles eram tipicamente concentradas em sistemas de grande porte, com um ou mais processadores. Nos sistemas centralizados, os usuários utilizam terminais não inteligentes conectados a linhas seriais dedicadas ou linhas telefônicas públicas para a comunicação interativa com esses sistemas.

No modelo centralizado, os terminais não têm capacidade de processamento. Sempre que um usuário deseja alguma tarefa, o pedido é encaminhado ao sistema, que realiza o processamento e retorna uma resposta, utilizando as linhas de comunicação.

Com a evolução dos computadores pessoais e das estações de trabalho, juntamente com o avanço das telecomunicações e da tecnologia de redes, surgiu um novo modelo de computação, chamado modelo de *rede de computadores* (computer network).

Em uma rede existem dois ou mais sistemas independentes, também chamados de nós, hosts ou estações, com capacidade de processamento própria e interligados através de linhas de comunicação. Com base no grau de integração dos nós da rede, podemos dividir os sistemas fracamente acoplados entre sistemas operacionais de rede e sistemas operacionais distribuídos.

3.3.2.1 SISTEMAS OPERACIONAIS DE REDE

Em sistemas operacionais de rede (SOR), cada nó possui seu próprio sistema operacional, além de um hardware e software que possibilitam ao sistema ter acesso a outros componentes da rede, compartilhando seus recursos. O SOR permite entre outras funções:

- Cópia remota de arquivos.
- Emulação de terminal.
- Impressão remota.
- Gerência remota.
- Correio eletrônico.

Cada nó é totalmente independente do outro, podendo inclusive possuir sistemas operacionais diferentes. Caso a conexão entre os nós sofra qualquer problema, os sistemas podem continuar operando normalmente, apesar de alguns recursos se tornarem indisponíveis.

O melhor exemplo da utilização dos sistemas operacionais de rede são as redes locais (Local Area Network - LAN). Nesse ambiente, cada estação pode compartilhar seus recursos com o restante da rede. Caso uma estação sofra qualquer problema, os demais componentes da rede podem continuar o processamento, apenas não dispondo dos recursos oferecidos por ela.

3.3.2.2 SISTEMAS OPERACIONAIS DISTRIBUÍDOS

Em sistemas distribuídos, cada componente da rede também possui seu próprio sistema operacional, memória, processador e dispositivos. O que define um sistema distribuído é a existência de um relacionamento mais forte entre os seus componentes, onde geralmente os sistemas operacionais são os mesmos. Para o usuário e suas aplicações, é como se não existisse uma rede de computadores, mas sim um único sistema centralizado.

Este tipo de sistema distribuído é muitas vezes chamado de *cluster*. Uma grande vantagem destes sistemas é a capacidade de redundância do sistema. Como as aplicações estão distribuídas pelos diversos sistemas, caso ocorra algum problema com um dos componentes, é possível que um deles assuma o papel do sistema defeituoso. Em aplicações de missão crítica, como controle de tráfego aéreo, existem sistemas especialmente desenvolvidos para esta finalidade, conhecidos como sistemas de tolerância à falhas (fault tolerance).

Alguns fabricantes, juntamente com órgãos de padronização, definiram padrões para a implementação de sistemas distribuídos. O Distributed Computing Environment (DCE), o Common

Object Request Broker Architecture (COBRA) e o Object Linking and Embedding (OLE) são alguns dos padrões para o desenvolvimento de aplicações em ambientes distribuídos.

4. DOS - DISK OPERATING SYSTEM

O DOS é abreviação de “Disk Operating System “ , termo genérico para designar um **Sistema Operacional** baseado em disco .

Famílias padrões :

TRS - DOS
Apple - DOS ⇒ apple
Pro - DOS ⇒ apple
MS - DOS (máquina tipo IBM - PC)

MS-DOS

Representa a Microsoft Disk Operating System, sistema padrão para microcomputadores de 16 bits. O MS - DOS é um **Sistema Operacional** desenvolvido para computadores que utilizavam os microprocessadores 8086 e 8088.

A estrutura do DOS é formada pelos arquivos que contém os diferentes programas que compõe o **Sistema Operacional**. Esses arquivos estão sempre gravados no disco. Alguns são arquivos de comando , cujos nomes podem ser vistos no diretório, outros são arquivos ocultos, i.e., cujos nomes não podem ser vistos normalmente no diretório.

COMPOSIÇÃO DO MS-DOS

- **BOOT RECORD** - é um pequeno programa que carrega o resto do sistema. Realiza a carga dos arquivos **IBMBIOS.COM** e **IBMDOS.COM** que, junto com o **COMMAND.COM** constituem o Sistema Operacional.
 - programa residente no início do disquete (trilha ϕ e setor 1) .
 - é carregado na memória na inicialização do DOS .
 - carrega outra parte do DOS (IBMBIOS.COM e IBMDOS.COM) .
 - é gravado no disquete durante a formatação.
- **IBMBIOS.COM**
 - controla leitura / gravação nas unidades de armazenamento.
 - é gravado no disquete durante a formatação .
 - não aparece em lista de diretório .
- **IBMDOS.COM**
 - contém funções para controlar programas executando-os sob o DOS .
 - é gravado no disquete durante a formatação .
 - não aparece em lista de diretório .
- **COMMAND.COM** - que é o processador de comandos.

Estes são divididos em 2 tipos : residentes ou internos e não-residentes ou externos. O MS-DOS mantém na memória uma tabela com todos os nomes dos comandos para seu reconhecimento. No caso de um comando não ser encontrados nessa lista, ele é procurado como um programa do sistema (default) .

- processador de comandos digitados pelo operador .
- é gravado no disquete durante a formatação.
- aparece na lista do diretório .

5. CP/M - CONTROL PROGRAM FOR MICROCOMPUTERS

As iniciais CP/M são de Control Program for Microcomputers (controlador de programas para microcomputadores). O CP/M possui duas versões principais :

CP/M 80 - um **Sistema Operacional** para sistema monousuário, modular e independente do hardware. E executado em sistemas baseados nos microprocessadores Intel 8080 e 8085; Zilog Z-80 (todos de 8 bits) e permite endereçar até 64 Kbytes de memória principal .

CP/M 86 - utilizado pelos sistemas baseados nos microprocessadores de 16 bits, Intel 8086 e 8088. Permite endereçar até 1 Mbytes de memória. Mantém por razões de compatibilidade, a estrutura de arquivo do CP/M 80.

Entre compiladores e interpretadores que podem funcionar com CP/M encontram ADA , ALGOL, APL, BASIC, C, COBOL, FORTH, FORTRAN, LISP, LOGO, RPG, PASCAL e PL/1.

O CP/M divide-se em 4 módulos principais : **BIOS, BDOS, CCP e TPA**.

- **BIOS** (Basic Input / Output System) = funções de entrada / saída (com o console, discos, impressoras e outros periféricos) .
- **BDOS** (Basic Disk Operation System) = gerência discos e arquivos.
- **CCP** (Console Command Processor) = processador de comandos de console.
- **TPA** (Transiente Progran Area) = programas de dados.

Os comandos básicos do CP/M são divididos em 2 grupos principais :

- **Comando Intrínsecos** ou interno ou residentes = fazem parte do CCP e não precisam ser carregados na memória para a execução.
- **Comando Extrínsecos** ou externos ou transitórios = residem em disco. São carregados a partir do disquete e executados em módulo TPA. E podem ser definidos pelo usuário.

6. UNIX

O sistema UNIX foi originalmente projetado para rodar em minis, embora agora, haja versões para microcomputadores e para mainframes.

O UNIX é um sistema operacional moderadamente complexo. É muito mais simples do que os Sistemas Operacionais que rodam em mainframes, mas possui mais capacidade do que a maioria dos sistemas operacionais para microcomputadores. Por exemplo, o sistema UNIX permite rodar vários programas simultaneamente (Tempo Compartilhado) - técnica desenvolvida para compartilhar o computador com vários usuários.

No sistema UNIX todas as funções que são necessárias de imediato, são mantidas constantemente na memória. A parte residente na memória é chamada de núcleo. Enquanto um programa estiver sendo executado no sistema UNIX, ele é chamado de processo.

Usuários típicos do sistema UNIX, gastam a maioria de seu tempo usando dois programas: shell e o editor.

Um editor é um programa que lhe permite criar e modificar arquivos de texto. O editor padrão do sistema UNIX é um programa interativo. O controle do programa editor é feito por comandos. (ex. imprimir linhas, acrescentar textos).

O shell é um dos programas mais importantes do sistema UNIX. Como o editor, o shell também é um programa interativo. Pode-se controlar o shell através de comandos que ele interpreta (decodifica) e executa. Portanto o nome técnico é interpretador de comandos. A função de um interpretador de comandos é executar os comandos que forem inseridos. O shell também é uma linguagem de programação sofisticada, portanto ele é a chave para se coordenar e combinar programas no sistema UNIX.

O sistema UNIX é, na realidade, uma ferramenta para o gerenciamento de informações. O poder do UNIX provém de sua capacidade de permitir que programas rodem juntos para produzirem a informação desejada. No UNIX quase todos os programas são ferramentas simples que podem ser combinadas para produzirem ferramentas mais poderosas.

Obs.: Geralmente o interpretador de comandos é uma parte da estrutura interna do Sistema Operacional UNIX. → programa comum (Shell).