

## **Capítulo 1**

### **Aspectos Gerais**

Sem software, um computador é basicamente um inútil. Com software, um computador pode armazenar, processar e recuperar informações, exibir documentos de multimídia, pesquisar na Internet , etc...

O software pode ser dividido, a grosso modo, em duas espécies:

- ❖ Programas de sistema, que gerenciam a operação do computador em si.
- ❖ Programas aplicativos, que executam o trabalho que o usuário realmente deseja.

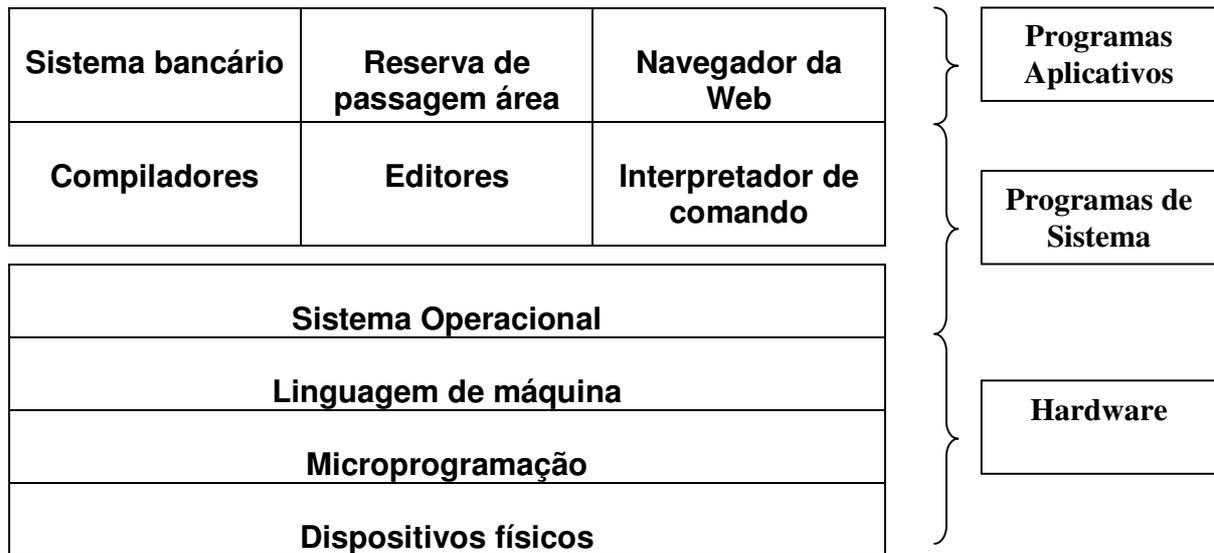
**Nota  
Importante**

**O programa de sistema mais fundamental é o sistema operacional, que controla todos os recursos do computador e fornece a base sobre a qual os programas aplicativos podem ser escritos.**

Um moderno sistema de computador consiste em um ou mais processadores, alguma memória principal (memória RAM), discos, impressoras, interfaces de rede e outros dispositivos de entrada/saída.

**Nota  
Importante**

- **Escrever os programas que controlam todos esses componentes e usá-los corretamente é um trabalho extremamente difícil. Se cada programador tivesse de preocupar-se com o modo como as unidades de disco funcionam e com todas as dúzias de coisas que poderiam dar errado ao ler um bloco de disco, seria provável que muitos programas sequer pudessem ser escritos.**
  - **A necessidade de se isolar os programadores da complexidade do hardware tornou-se evidente.**
  - **A maneira encontrada para se resolver este problema foi a seguinte: Colocar uma camada de software por cima do hardware básico para gerenciar todas as partes do sistema e oferecer ao usuário uma interface ou máquina virtual que é mais fácil de entender e de programar.**
  - **Essa camada de software é o sistema operacional.**



- **Hardware**

Em muitos casos, é composto de duas ou mais camadas. A camada baixa contém dispositivos físicos, consistindo em circuitos integrados, cabos, fonte alimentadora, tubos de raios catódicos e dispositivos físicos semelhantes.

- **Microprograma**

Camada de software primitivo que controla diretamente esses dispositivos e proporciona uma interface limpa para a próxima camada. Normalmente está localizado em memória somente para leitura. Ele é realmente um interpretador, buscando as instruções de linguagem de máquina como ADD, MOVE e JUMP, executando-as como uma série de pequenos passos. Para executar uma instrução ADD, por exemplo, o microprograma deve determinar onde os números a serem somados estão localizados, busca-los, adiciona-los e armazenar o resultado em algum lugar.

- **Linguagem de máquina**

Conjunto de instruções que o microprograma interpreta. Possui entre 50 e 300 instruções que, na sua maior parte, servem para mover dados pela máquina, para fazer aritmética e para comparar valores. Nessa camada, os dispositivos de entrada / saída são controlados carregando valores em registradores especiais de dispositivo.

- **Sistema Operacional**

**Uma importante função do sistema operacional é esconder toda essa complexidade e oferecer um conjunto mais conveniente de instruções para o programador trabalhar.**

- **Programas de sistema**

Por cima do sistema operacional está o resto do software de sistema. Encontramos:

- Interpretador de comandos (Shell).
- Sistemas de janelas.
- Compiladores.
- Editores.

Esses programas não são parte do sistema operacional, mesmo que eles tipicamente sejam fornecidos pelo fabricante do computador. **Esse é um ponto crucial.**

<b>Nota Importante</b>
----------------------------

<b>O sistema operacional é a porção do software que executa no modo kernel ou <u>no modo supervisor</u>. Ele é protegido do usuário pelo hardware.</b>
--

<b>Os compiladores e editores executam <u>no modo usuário</u>. Se um usuário não gosta de um compilador particular, ele é livre para escrever seu próprio compilador se preferir; mas ele não é livre para escrever seu próprio manipulador de interrupções de disco, que é parte do sistema operacional e normalmente é protegido por hardware contra tentativas dos usuários de modificá-lo.</b>
--

- **Programas aplicativos**

Programas responsáveis pela resolução de problemas particulares dos usuários, tais como, processadores de texto, planilhas eletrônicas, programas de cálculo de engenharia, jogos, etc...

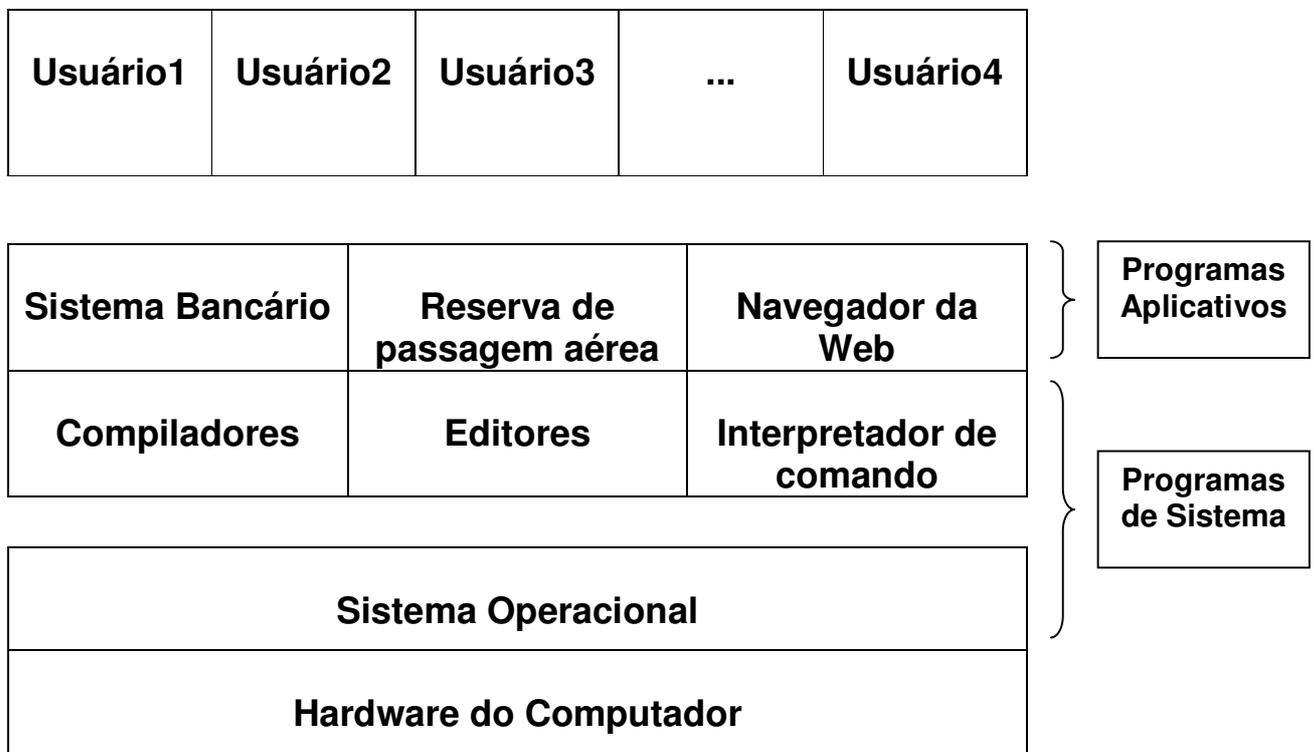
## **O que é um Sistema Operacional?**

### **1ª. Definição:**

Programa que atua como um intermediário entre um usuário e o hardware de um computador. Seu propósito é fornecer um ambiente no qual os usuários possam executar seus programas. O principal objetivo de um sistema operacional é tornar conveniente o uso de um sistema computacional. Um objetivo secundário é usar o hardware do computador de maneira eficiente.

### **2ª. Definição:**

Um sistema operacional é parte importante de quase todo sistema de computação. Um sistema computacional pode ser dividido de maneira geral em:



**Nota  
Importante**

**Um sistema operacional é semelhante a um governo. Os componentes de um sistema computacional são: hardware, software e dados.**

**Como um governo, o sistema operacional não desempenha função útil por si próprio. Ele simplesmente fornece um ambiente no qual outros programas podem realizar algum trabalho útil.**

**O sistema operacional atua como um gerenciador de recursos, que os aloca, quando necessário, a programas e usuários, a fim de que estes possam realizar suas tarefas.**

**Um sistema operacional é um programa de controle.**

**3ª. Definição:**

Um sistema operacional é o programa que permanece em execução no computador em todos os momentos – normalmente denominado núcleo (**kernel**) -, sendo todo o restante composto pelos programas aplicativos.

É mais fácil definir sistemas operacionais pelo que fazem do que pelo que são.

Do ponto de vista de computadores pessoais:

- **O objetivo principal de um sistema operacional é a conveniência para o usuário, ou seja, tornar mais fácil o uso do sistema computacional.**

Do ponto de vista de grandes sistemas compartilhados e de multiusuários:

- **Operação eficiente do sistema computacional.**

**Esses dois objetivos, conveniência e eficiência, são às vezes contraditórios. No passado, freqüentemente dava-se mais importância às considerações sobre eficiência do que às relativas a conveniência.**

## **Historia dos Sistemas Operacionais**

Os sistemas operacionais tiveram grande influência sobre a arquitetura de computadores e vice-versa. Os primeiros foram desenvolvidos para facilitar o uso do hardware. À medida que os sistemas operacionais eram desenvolvidos, tornava-se cada vez mais óbvio que mudanças no projeto do hardware poderiam simplifica-los.

O primeiro computador digital verdadeiro foi projetado pelo matemático inglês Charles Babbage ( 1792 – 1871 ). Embora Babbage tenha gasto a maior parte de sua vida e sua fortuna tentando construir seu “motor analítico”, ele nunca conseguiu fazê-lo funcionar adequadamente porque a coisa era puramente mecânica e a tecnologia do seu tempo não podia produzir as necessárias rodas e engrenagens de alta precisão de que ele precisava. O motor analítico não tinha um sistema operacional.

Curiosidade: Babbage sabia que precisaria de um software para seu motor analítico, assim contratou Ada Lovelace, como primeiro programador do mundo. O nome da linguagem de programação Ada foi dado em sua homenagem.

### **A Primeira Geração de Computadores – Válvulas e Painéis de Conectores ( 1945 – 55)**

- Essas máquinas eram enormes, ocupando salas inteiras com dezenas de milhares de válvulas, e eram muito mais lentas que as mais simples calculadoras de hoje.
- Um único grupo de pessoas projetava, construía, programava, operava e mantinha cada máquina. Toda programação era feita em linguagem de máquina. As linguagens de programação eram desconhecidas(nem mesmo a linguagem assembly) e ninguém tinha ouvido falar em sistemas operacionais.
- O modo normal de operação era o programador reservar um período de tempo na folha de reserva na parede, depois descer até o lugar da máquina, inserir suas conexões no computador e gastar algumas horas esperando que nenhuma das aproximadamente 20.000 válvulas queimasse durante a execução. Praticamente todos os problemas eram simples cálculos numéricos, tais como gerar tabelas de senos e co-senos.
- **No início da década de 50, a rotina havia melhorado um pouco com a introdução dos cartões perfurados. Agora era possível gravar programas em cartões e lê-los, em vez de usar cabos e conectores; fora isso, o procedimento era o mesmo.**

## Primeiro Sistema Operacional

- Os primeiros computadores eram máquinas fisicamente enormes, cuja execução era iniciada por meio de um console. O programador, que era também o operador do sistema de computação, escrevia um programa e controlava a execução desse programa diretamente, usando o console de operação.
- O tempo necessário para preparar a execução de programas poderia ser bastante longo. Cada tarefa consistia em vários passos distintos: carregar o compilador FORTRAN, executar o compilador, desmontar a fita do compilador, carregar o tradutor de código de montagem, executar esse tradutor, desmontar sua fita, montar a fita do programa objeto e, finalmente, executar esse programa. Se ocorresse algum erro em algum desses passos, todo esse procedimento poderia ter que ser iniciado novamente. Cada passo da tarefa podia envolver a montagem e desmontagem de fitas magnéticas, fitas de papel e cartões de perfuração.
- Esse tempo de preparação constituía um sério problema. Enquanto as fitas eram montadas, ou o programador operava o console, a CPU permanecia ociosa. Nessa época, existiam poucos computadores e eles eram extremamente caros. Além disso, havia custos operacionais de energia, refrigeração, programação e assim por diante. Portanto, o tempo de computação era extremamente valioso, e os proprietários dos computadores queriam que eles fossem utilizados tanto quanto possível. Exigia-se uma alta taxa de utilização.
- Melhoria do tempo de preparação através das seguintes medidas:
  - Contratação de um operador de computador, que liberava o programador da tarefa de operar a máquina. Quando uma tarefa era concluída, o operador podia, imediatamente, iniciar outra. Como resultado o tempo de preparo foi reduzido.
  - Tarefas similares deveriam ser agrupadas e executadas no computador em grupos. Por exemplo: suponha que o operador recebesse uma tarefa FORTRAN, uma tarefa COBOL e então outra tarefa FORTRAN novamente. O operador deverá executar primeiro as tarefas em FORTRAN.

Entretanto, existiam ainda muitos problemas. Por exemplo, se uma tarefa parasse, o operador tinha que perceber esse fato, observando o console, determinar por que o programa parou (terminação normal ou anormal), obter uma cópia da memória e dos registradores da máquina, se necessário, e então carregar a próxima tarefa, no dispositivo apropriado, e reiniciar o computador. Durante essa transição de uma tarefa para outra, a CPU permanecia ociosa.

**Nota  
Importante**

Para evitar esse tempo ocioso da CPU, foi desenvolvido um mecanismo automático de execução seqüencial de tarefas. Com essa técnica, foi criado o primeiro sistema operacional rudimentar. O que se buscava era um procedimento para a transferência automática de controle, de uma tarefa para a próxima. Para isso, foi criado um pequeno programa, denominado monitor residente. O monitor residente permanecia sempre armazenado na memória.

Quando o computador era ligado, o monitor residente era chamado e transferia o controle para um programa. Ao término desse programa, o controle retornava para o monitor, que podia então passar para o próximo programa. Portanto, o monitor residente prosseguia automaticamente de um programa para outro e de uma tarefa para outra.

### **COMO O MONITOR RESIDENTE SABERIA QUAL PROGRAMA ELE DEVERIA EXECUTAR?**

O operador recebia, previamente, uma breve descrição dos programas que deveriam ser executados e quais dados deveriam ser utilizados. Para que essa informação pudesse ser fornecida diretamente ao monitor, foram introduzidos cartões de controle.

Além do programa e dos dados de uma tabela, o programador incluía cartões especiais(cartões de controle), que continham diretivas para o monitor residente, indicando o programa a ser executado. Por exemplo, um programa comum de usuário poderia requerer a execução de três programas: o compilador FORTRAN (FNT), o tradutor de linguagem de montagem(ASM) e o programa do usuário(RUN).

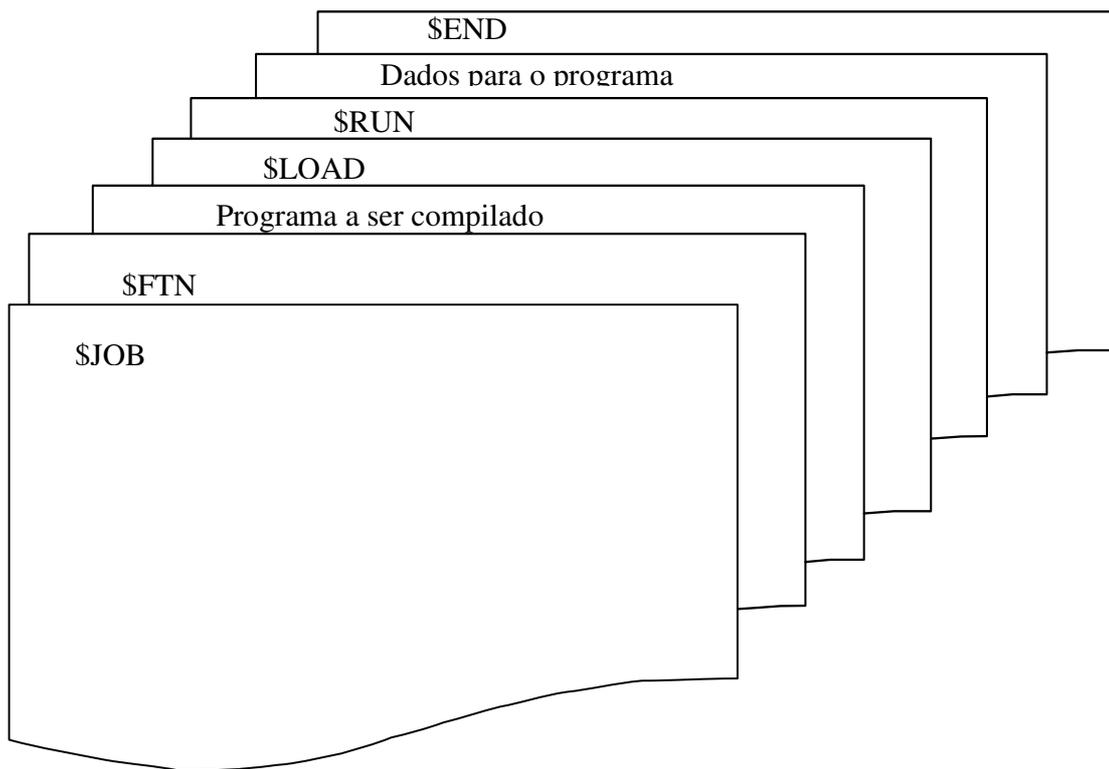
\$FNT – Executar o compilador FORTRAN.

\$ASM – Executar o tradutor de linguagem de montagem.

\$RUN – Executar o programa de usuário.

\$JOB – Primeiro cartão da tarefa.

\$END – Último cartão da tarefa.



**Cartões para um sistema de processamento de serviços em lotes.**

A mudança para sistemas de processamento de serviços em lotes, com ordenação automática dos lotes de serviços para execução, tinha como objetivo melhorar o desempenho do sistema de computação. O ser humano é extremamente lento, se comparado a um computador. Era desejável substituir a operação humana por um programa de operação automática do sistema. A ordenação automática dos programas para execução eliminou o tempo gasto pelo operador, para ordenar os programas em lotes e preparar os dispositivos para operação.

Mesmo com esse método, a CPU continuava ociosa. O problema residia na velocidade dos dispositivos mecânicos de E/S, que são intrinsecamente mais lentos que os dispositivos eletrônicos.

A maioria dos sistemas de computação, no final dos anos 50 e início dos anos 60, eram sistemas de processamento de serviços em lotes, que liam dados de leitoras de cartões e escreviam em impressoras de linhas ou perfuradoras de cartões. Entretanto, em vez de a CPU usar a leitora de cartões para realizar a operação de leitura, os cartões eram primeiramente copiados para uma fita magnética.

O ganho real devido à operação dos dispositivos de E/S separadamente do processador central residia na possibilidade de utilizar vários sistemas de

leitura ou gravação de múltiplas fitas, para uma única CPU. Se a CPU podia efetuar processamento duas vezes mais rápido que uma leitora podia ler cartões, então era possível usar duas leitoras, simultaneamente, para produzir uma quantidade de fita gravada suficiente para manter a CPU ocupada.

**Por outro lado, havia agora um maior atraso na execução de cada tarefa, pois cada tarefa precisava ser, primeiro, lida para a fita. Portanto, a tarefa era atrasada até que um número suficiente de tarefas fossem lidas para a fita e ela ficasse cheia.**

**Nota  
Importante**

O problema existente com os sistemas de fita era que a leitora de cartões não podia escrever em uma extremidade da fita, enquanto a CPU usava a outra extremidade em uma operação de leitura. Uma fita tinha que ser totalmente gravada e rebobinada, pois ela é, por natureza, um dispositivo de acesso seqüencial.

Os sistemas de disco eliminaram esse problema, uma vez que os discos são dispositivos de acesso aleatório. Como a cabeça do disco pode ser movida de uma área do disco para outra, o disco pode rapidamente mudar de uma área que está sendo usada para gravação dos cartões, lidos a partir da leitora de cartões, para a posição requerida pela CPU, para ler o próximo cartão

### Spooling

Em sistemas com discos, os cartões eram lidos diretamente da leitora de cartões para um disco. A localização das cópias dos cartões era armazenada em uma tabela, usada pelo sistema operacional. Quando uma tarefa era executada, o sistema operacional atendia às requisições de leitura de cartões, realizando a leitura dos dados em um disco. Analogamente, quando uma tarefa requisitava a impressão de uma linha, essa linha era copiada em um disco. Quando a tarefa fosse concluída, a saída seria realmente impressa. A esse procedimento damos o nome de Spooling.

**Nota  
Importante**

**Spooling – O disco é utilizado como uma grande área de armazenamento, para efetuar a leitura antecipada dos dispositivos de entrada e armazenar arquivos de saída, até que os dispositivos de saída estejam disponíveis para imprimi-los.**

**Nota  
Importante**

**Spooling – Esse mecanismo levou à multiprogramação, que constitui a base dos sistemas operacionais modernos.**

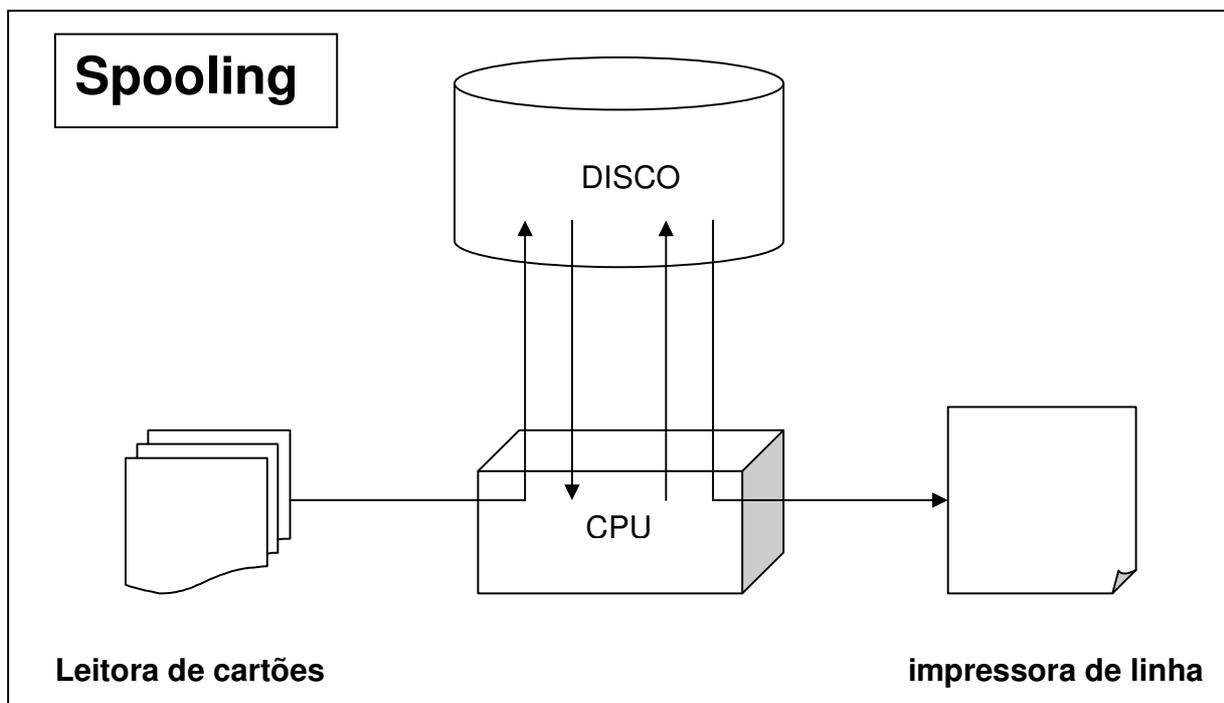
### **Sistemas batch simples**

Um sistema operacional batch lê uma seqüência de serviços separados, cada um com seus próprios cartões de controle que predefinem o que é feito em cada serviço. Quando o serviço termina, a saída é impressa.

**A característica marcante de um sistema de batch é a ausência de interação entre o usuário e o serviço enquanto este último está sendo processado. O serviço é preparado e submetido à execução; algum tempo mais tarde obtém-se o resultado.**

**Neste ambiente de execução, a CPU fica quase sempre ociosa. Por que?**

**Porque, as velocidades dos dispositivos mecânicos de E/S são intrinsecamente inferiores às dos dispositivos eletrônicos. Mesmo uma CPU lenta opera na faixa de microssegundos, executando milhares de instruções por segundo. Uma leitora de cartões rápida, por outro lado, poderia ler 1200 cartões por minuto. A diferença entre a velocidade da CPU e seus dispositivos de E/S poderia ser de três ordens de grandeza ou mais.**



**A introdução da tecnologia de discos apresentou uma grande contribuição.** Em vez de os dados contidos em cartões serem lidos da leitora de cartões diretamente para a memória, para que um serviço pudesse ser então processado, eles passaram a ser lidos antes pela leitora de cartões e então gravados em um disco.

**Nota  
Importante**

A posição do disco no qual os dados são armazenados fica gravada em uma tabela mantida pelo sistema operacional.

- O mecanismo de spooling é também usado para processar dados em locais remotos. A CPU envia os dados, por meio de canais apropriados para transferência de sinais, para a impressora remota. O processamento remoto é feito então separadamente, sem nenhuma intervenção da CPU. A CPU necessita apenas ser notificada quando o processamento termina, para que ela possa fazer o armazenamento correspondente ao próximo lote de dados.
- O mecanismo de spooling sobrepõe à E/S de um serviço a computação de outros serviços.
- Mesmo em um sistema simples, o spooler pode estar lendo a entrada de um serviço enquanto está imprimindo a saída de outro. Enquanto isso ocorre, outros serviços podem estar em execução, lendo seus “cartões” do disco ou imprimindo suas linhas de saída em disco.
- O mecanismo de spooling tem um efeito benéfico direto sobre o desempenho do sistema. Pelo custo de algum espaço em disco e de algumas tabelas, a computação de um serviço pode se sobrepor à E/S de outros serviços.
- O mecanismo de spooling pode manter tanto a CPU quanto os dispositivos de E/S trabalhando a taxas de ocupação muito maiores.

### **Sistemas batch multiprogramados**

O spooling oferece uma estrutura de dados importante: o pool de serviços. Possibilita que diversos serviços que já foram lidos fiquem armazenados em disco esperando para ser processados. **Uma fila de serviços em disco permite que o sistema operacional selecione qual serviço deve ser executado em seguida, a fim de aumentar a utilização da CPU.**

<b>Nota Importante</b>
------------------------

- |  |
|--|
| <ul style="list-style-type: none"><li>➤ Quando os serviços chegam diretamente de leitoras de cartões ou fitas magnéticas, não é possível executá-los em uma ordem diferente. Os serviços devem ser executados seqüencialmente, na forma, o primeiro que chega é o primeiro a ser atendido.</li><li>➤ Quando diversos serviços estão prontos para serem executados, em um dispositivo de acesso direto, tal como um disco, um escalonamento de serviços passa a ser possível.</li></ul> |
|--|

<b>O tópico mais relevante do escalonamento de serviços é a capacidade de multiprogramação</b>
--

<b>Observação</b>
-------------------

- |   |
|---|
| <ul style="list-style-type: none"><li>❖ Um único usuário não pode, em geral, manter a CPU ou os dispositivos de E/S ocupados a todo instante.</li><li>❖ A multiprogramação aumenta a utilização da CPU pela organização de serviços de maneira a que esta sempre tenha um serviço a executar.</li></ul> |
|---|

<b>Sistema Operacional</b>
<b>Serviço1</b>
<b>Serviço2</b>
<b>...</b>
<b>Serviço<sub>n</sub></b>

<b>O sistema operacional mantém diversos na memória em um dado instante. Esse conjunto de serviços é um subconjunto dos serviços mantidos no pool de serviços (pois a quantidade de serviços que pode ser mantida simultaneamente na memória é normalmente muito menor do que a quantidade que pode estar no pool de serviços). O sistema operacional seleciona e começa a executar um dos serviços que estão na memória.</b>
---

Ocasionalmente, o serviço pode ter de esperar por alguma tarefa, tal como a montagem de uma fita ou a conclusão de uma operação de E/S. Em um sistema não multiprogramado, a CPU ficaria ociosa, sem realizar tarefa alguma. Em um sistema multiprogramado, o sistema operacional simplesmente libera a CPU para a execução de um outro serviço. Quando esse serviço precisa esperar, a CPU é trocada para outro serviço e assim por diante. Enquanto houver algum serviço a ser executado, a CPU ficará ocupada.

Esse conceito é comum em diversas situações. Um advogado não atende a apenas um cliente de cada vez. Ao contrário, diversos clientes podem estar sendo atendidos ao mesmo tempo. Enquanto um caso está esperando para entrar em julgamento, ou é necessário esperar até que documentos sejam digitados, o advogado pode trabalhar em outro caso. Se o advogado tiver um número suficiente de clientes, ele nunca ficará desocupado.

**A multiprogramação é o primeiro exemplo no qual o sistema operacional tem de tomar decisões para os usuários.**

Todos os serviços que passam a fazer parte do sistema são mantidos no pool de serviços.

- ✓ Esse **pool** contém todos os **processos** que residem na **memória secundária** esperando para serem alocados na **memória principal**.
- ✓ Se diversos processos estão prontos para serem carregados na memória principal e não há espaço suficiente para todos eles, o sistema tem de escolher entre eles (**escalonamento de processos**).
- ✓ Quando o sistema operacional seleciona um serviço do pool de memória ao mesmo tempo, é necessário haver alguma forma de **gerenciamento de memória**.
- ✓ Além disso, se diversos serviços estão prontos para serem executados em um dado instante, o sistema deve escolher entre eles. Esse processo de decisão é chamado de **alocação de CPU**
- ✓ Finalmente, quando vários serviços estão sendo executados simultaneamente, a capacidade de cada um de afetar o processo de execução de outro precisa ser controlada em todas as fases de um sistema operacional, incluindo o escalonamento de serviços, o armazenamento em disco e o gerenciamento de memória.

## **Sistemas de Compartilhamento de Tempo**

O grande problema apresentado pelos sistemas batch multiprogramados foi a falta de interação do usuário com os processos que estavam sendo executados. Um programador, por exemplo, não tinha acesso ao seu programa a fim de modificá-lo durante a sua execução, de modo a estudar o seu comportamento. O que representa uma grande restrição.

**Compartilhamento de tempo, ou processamento multitarefa, é uma extensão lógica de multiprogramação. Diversos serviços são executados, com a CPU alternando-se entre eles, mas a mudança ocorre tão freqüentemente que os usuários podem interagir com cada programa enquanto ele está sendo executado.**

**Um sistema computacional interativo fornece uma comunicação dinâmica entre o usuário e o sistema.** O usuário fornece instruções ao sistema operacional ou a um programa diretamente e obtém uma resposta imediata.

- ❑ O usuário dá um comando, espera pela resposta e pode decidir qual será o próximo comando com base no resultado do comando anterior. O usuário pode facilmente experimentar e pode ver os resultados imediatamente.
- ❑ Para que os usuários possam ter acesso tanto ao dados quanto ao código de programas de modo conveniente, deve existir um sistema de arquivos que permita ao usuário uma interação dinâmica.

**Nota importante**

- ❖ **Sistemas batch são apropriados para a execução de grandes serviços que precisam de pouca interação.**
- ❖ **O usuário pode submeter serviços para execução e retornar mais tarde para obter os resultados.**
- ❖ **Não é necessário que ele espere o serviço ser processado.**
- ❖ **Serviços interativos tendem a ser compostos por muitas ações de pequena duração, em que os resultados do próximo comando podem não ser previsíveis. O usuário submete o comando e então espera pelos resultados.**
- ❖ **O uso de um sistema interativo é adequado nos casos em que se necessita de um tempo de resposta pequeno.**

**Conclusões**

- ✓ Os computadores mais antigos com um único usuário eram sistemas interativos. Ou seja, o sistema todo estava à disposição do programador/operador. Porém resultava em uma quantidade significativa de tempo ocioso da CPU.
- ✓ Devido aos elevados custos desses primeiros computadores, não era desejável que a CPU ficasse ociosa.
- ✓ Para resolver esse problema foram desenvolvidos sistemas operacionais do tipo batch.
- ✓ Oferecem um aumento na taxa de utilização desses sistemas.
- ✓ Sistemas de compartilhamento de tempo foram desenvolvidos para viabilizar o uso interativo de um sistema computacional a um preço razoável.

Sistemas operacionais de tempo compartilhado usam os mecanismos de alocação e multiprogramação da CPU para fornecer a cada usuário uma pequena porção de um computador. Para cada usuário existe pelo menos um programa separado na memória. **Um programa que é carregado na memória e está sendo executado é chamado de processo.** Quando um processo está em execução, ele normalmente permanece assim por um período curto de tempo antes que termine ou necessite realizar alguma operação e E/S. Essa operação pode ser interativa; isto é, a saída é enviada para a tela do usuário, enquanto a entrada vem do teclado do usuário. Como a E/S interativa é, via de regra, realizada em velocidades comuns para os seres humanos, pode levar muito tempo para ser concluída. A entrada pode estar limitada a velocidade de digitação do usuário, que é incrivelmente lento para os computadores. Em vez de deixar a CPU ficar ociosa enquanto essa entrada interativa acontece, o sistema operacional pode rapidamente permitir que a CPU passe a executar um programa de algum outro usuário.

Um sistema operacional de tempo compartilhado permite aos muitos usuários compartilhar simultaneamente o computador. Como cada ação em um sistema de tempo compartilhado em geral é realizada em pouco tempo, apenas um pouco de tempo da CPU é necessário para cada usuário.

A idéia de compartilhamento de tempo já era usada em 1960, mas, como sistemas de tempo compartilhado são caros e difíceis de implementar, eles não se tornaram comuns até o início da década de 70.

Hoje em dia, a maioria dos sistemas oferece tanto processamento em batch quanto com compartilhamento de tempo, embora o projeto e uso básico desses sistemas tendam a ser de um tipo ou de outro.

**Nota**

❖ **Sistemas operacionais de compartilhamento de tempo.**

- **O disco atua como uma memória secundária para armazenar e acessar dados da memória principal.**
- **A memória virtual é um método comum de gerenciamento e proteção da memória que se caracteriza por permitir a execução de um serviço que pode não estar completamente armazenado na memória principal.**
- **Principal vantagem desse esquema : os programas podem ser maiores que a memória física.**

### **Sistemas de Computadores Pessoais**

Com a diminuição dos custos do hardware, tornou-se viável o uso de um computador por um único usuário.

#### **Finalmente chegamos aos Personal Computers - 1970.**

Os dispositivos de E/S mudaram, tendo os painéis de chaves e as leitoras de cartões sido substituídos por teclados similares aos de máquinas de escrever e mouses. Impressoras de linha e perfuradoras de cartões desapareceram, dando lugar às telas e às impressoras pequenas e rápidas.

Os microcomputadores consideravelmente menores e mais baratos do que os computadores de grande porte. Durante sua primeira década de existência, as CPUs não tinham ainda as características necessárias para proteger o sistema operacional dos programas dos usuários.

#### **Os sistemas operacionais dos PCs não eram portanto nem multiusuários nem multitarefas.**

#### **Entretanto, os seus objetivos mudaram com o tempo;**

- Em vez de maximizar a utilização da CPU e dos periféricos, os sistemas optaram por maximizar a conveniência e a rapidez no atendimento aos usuários. (Windows, Macintosh, OS/2,..)
- O MS-DOS foi substituído pelo Windows

- A IBM substituiu o MS-DOS pelo sistema multitarefa OS/2.
  - O Macintosh foi transportado para um hardware mais avançado e agora apresenta **novas características, tais como memória virtual.**
  - Os sistemas operacionais para esses computadores beneficiaram-se em diversos aspectos do desenvolvimento dos sistemas operacionais maiores.
- 
- **Os custos do hardware dos microcomputadores são suficientemente baixos para permitir que o uso de um computador possa ser reservado a uma única pessoa, e para que a utilização da CPU não seja mais a preocupação principal.**
  - Algumas das escolhas feitas em sistemas operacionais para computadores de grande porte podem não ser apropriadas para sistemas menores. Por exemplo, a proteção de arquivos não parece necessária em uma máquina pessoal.
  - **Embora a proteção de arquivos pareça não ser necessária no caso de computadores pessoais isolados, estes são geralmente ligados a outros computadores por meio de linhas telefônicas ou redes locais de computadores. Quando outros computadores e outros usuários podem ter acesso a arquivos de um computador pessoal, a proteção de arquivos volta a ser uma característica necessária.**

**Nota importante**

❖ Uma análise dos sistemas operacionais de microcomputadores e de computadores grande porte mostra que as características que eram, em um dado momento, exclusividade de computadores de grande porte foram adotadas nos microcomputadores. Os mesmos conceitos são apropriados para as diversas classes de computadores: computadores de grande porte, minicomputadores e microcomputadores.

## **Migração dos conceitos e características de SO.**

- **1950**
  - ✓ **Grande Porte**
    - Nenhum software
- **1950-1960**
  - ✓ **Grande Porte**
    - Compiladores
    - Batch
    - Monitores residentes
- **1960**
  - ✓ **Grande Porte**
    - Tempo compartilhado
  - ✓ **Mini**
    - Nenhum software
    - Monitores Residentes
- **1960-1970**
  - ✓ **Grande Porte**
    - MULTICS
    - multiusuário
  - ✓ **Mini**
    - Compiladores
  - ✓ **Micro**
    - Nenhum software
- **1970**
  - ✓ **Grande Porte**
    - multiusuario
  - ✓ **Mini**
    - Tempo compartilhado
  - ✓ **Micro**
    - Compiladores
    - Monitores Redentes
- **1970-1980**
  - ✓ **Grande Porte**
    - UNIX
  - ✓ **Mini**
    - Multiusuário
  - ✓ **Micro**
    - Compiladores
- **1980**
  - ✓ **Grande Porte**
    - Sistemas distribuídos
    - Multiprocessador
  - ✓ **Micro**
    - Interativo

- **1980-1990**
    - ✓ **Grande Porte**
      - Tolerante a falhas
    - ✓ **Mini**
      - Multiprocessador
      - Tolerante a falhas
      - UNIX
    - ✓ **Micro**
      - Multiusuário
    - ✓ **Rede**
      - Nenhum software
  - **1990**
    - ✓ **Grande Porte**
      - Tolerante a falhas
    - ✓ **Mini**
      - Multiprocessador
      - Tolerante a falhas
    - ✓ **Micro**
      - Multiprocessador
    - ✓ **Rede**
      - Compiladores
- 
- Um bom exemplo desse movimento ocorreu com o sistema operacional MULTICS. O MULTICS foi desenvolvido entre 1965 e 1970 no Instituto de Tecnologia de Massachusetts(MIT) como um utilitário e era executado em um complexo computador de grande porte. Muitas das idéias desenvolvidas para o MULIICS foram posteriormente utilizadas na Bell Laboratories no projeto do UNIX.
  - Por volta de 1980, os recursos e as características do UNIX tornaram-se a base dos sistemas operacionais semelhantes ao UNIX, para uso em microcomputadores, e estão sendo incluídos em sistemas operacionais mais recentes como o Windows NT, o OS/2 e o Macintosh.
  - Recursos desenvolvidos para um sistema de computadores de grande porte foram com o tempo sendo transportados para os microcomputadores.
  - Ao mesmo tempo que recursos de sistemas operacionais de grande porte estavam sendo reduzidos para que pudessem ser usados em computadores pessoais, sistemas de hardware mais potentes, rápidos e complexos eram desenvolvidos.
  - A medida que os PCs passam a ter hardware e software mais complexos, a linha divisória que separa as características de cada uma dessas duas categorias fica menos clara.

### **Sistemas paralelos**

A maioria dos sistemas atuais é de um único processador; uma CPU principal.

Entretanto, existe uma tendência na direção de sistemas multiprocessadores. Esses sistemas têm mais de um processador se comunicando, localizados próximos um do outro, e compartilhando um barramento, por onde são transmitidos os dados entre os componentes do sistema computacional, assim como o relógio, e algumas vezes também a memória e os dispositivos periféricos. **Esses sistemas são ditos fortemente aclopados.**

**Razões que justificam o desenvolvimento desses sistemas:**

- ✓ **Aumento da quantidade de trabalho realizado por unidade de tempo ( throughput ).**
- ✓ **Proporcionam uma economia em comparação a vários sistemas com um único processador, uma vez que os processadores podem compartilhar periféricos, gabinetes e fontes de energia.**
- ✓ **Se diversos programas têm de operar sobre um mesmo conjunto de dados, fica mais barato armazenar esses dados em um disco compartilhado por todos os processadores, em vez de ter muitos computadores com discos locais e muitas cópias de dados.**
- ✓ **Maior confiabilidade.**
- ✓ **Se as funções podem ser distribuídas de maneira apropriada pelos diversos processadores, uma falha em um processador não provocará uma parada do sistema, mas fará apenas com que a velocidade de processamento seja reduzida. Se tivermos dez processadores e um deles deixar de funcionar, todos os nove processadores restantes deverão assumir uma parte do trabalho do processador avariado.**
- ✓ **Essa capacidade de continuar provendo um serviço com desempenho proporcional ao número de componentes em operação é chamada de DEGRADAÇÃO GRACIOSA.**
- ✓ **TOLERANTE A FALHAS.**

## **Sistemas distribuídos**

Ao contrário dos sistemas fortemente acoplados (Paralelos), os processadores não compartilham a mesma memória ou o mesmo relógio. Em vez disso, cada processador tem sua própria memória local. Os processadores comunicam-se entre si por meio de várias linhas de comunicação, como barramentos de alta velocidade ou linhas telefônicas. **Chamados de sistemas fracamente acoplados.**

### **Permite:**

- **Compartilhamento de Recursos.**
- **Aumento na velocidade de computação.**
- **Confiabilidade.**
- **Comunicação.**

### **Redes de Computadores**

## **Sistemas de tempo real**

Forma de sistema operacional com propósito específico.

Utilizado quando existem requisitos rígidos relativos ao tempo sobre a operação de um processador ou sobre o fluxo de dados; usado como um dispositivo de controle em uma aplicação dedicada a um propósito específico.

- ❖ Sistemas que controlam experimentações científicas.
- ❖ Sistemas de geração de imagens para medicina.
- ❖ Sistemas de controle industrial.
- ❖ Sistemas de injeção de combustível em motores de automóveis.
- ❖ Sistemas existentes em armamentos.

Um sistema operacional de tempo real tem restrições fixas e bem definidas com relação ao tempo. O processamento precisa ser feito dentro das restrições definidas ou o sistema não funciona. Por exemplo, um braço de um robô não poderia ser instruído a parar depois que já tivesse espatifado o carro que estava construindo. Considera-se que um sistema de tempo real funciona de maneira correta somente se ele fornecer o resultado correto dentro das restrições de tempo estabelecidas.