

Sistemas de Informação
Disciplina: Arquitetura e Organização de Computadores - 1º Período
Professor: José Maurício S. Pinheiro

AULA 7: Sistema Operacional

Um Sistema Operacional, ou simplesmente S.O, pode ser definido como um conjunto de programas especialmente feitos para a execução de várias tarefas, entre as quais servir de intermediário entre o usuário e o computador.

1. Processo Computacional

Um processo computacional ou simplesmente processo pode ser entendido como uma atividade que ocorre em meio computacional, usualmente possuindo um objetivo definido, tendo duração finita e que utiliza uma quantidade limitada de recursos computacionais. Esta definição traz algumas implicações:

- Apenas as atividades que acontecem num sistema computacional são compreendidas como sendo processos computacionais;
- A duração finita implica que um processo computacional, por mais rápido ou curto que possa ser tem sempre uma duração maior que zero, ou seja, não existem processos instantâneos;
- Um processo utiliza ao menos um dos recursos computacionais existentes para caracterizar seu estado.

Um processo pode ser entendido como um programa em execução, o que envolve o código do programa, os dados em uso, os registradores do processador, sua pilha (*stack*) e o contador de programa, além de outras informações relacionadas a sua execução. Desta forma, temos que a impressão de um documento é um processo computacional assim como a cópia de um arquivo, a compilação de um programa ou a execução de uma rotina qualquer. Todas as atividades, manuais ou automáticas, que ocorrem dentro de um computador podem ser descritas como processos computacionais.

Atualmente os computadores são capazes de realizar diversas tarefas ao mesmo tempo (multitarefa), onde cada uma destas tarefas pode representar um ou mesmo mais processos. Para funcionarem desta forma tais computadores são multiprogramados, ou seja, o processador é chaveado de processo em processo, em pequenos intervalos de tempo, isto é, o processador executa um programa durante um pequeno intervalo de tempo, para depois executar outro programa por outro pequeno intervalo de tempo e assim sucessivamente.

Num instante de tempo qualquer, o processador estará executando apenas um dado programa, mas durante um intervalo de tempo maior ele poderá ter executado trechos de muitos programas criando a ilusão de paralelismo. Este comportamento é, algumas vezes, chamado de paralelismo virtual

ou pseudoparalelismo. Em computadores com dois ou mais processadores é possível a existência de paralelismo verdadeiro pois cada processador pode executar um processo independentemente.

A administração de vários diferentes programas em execução concomitante é o que permite o funcionamento eficiente dos computadores modernos, ao mesmo tempo conferindo-lhe complexa organização e estrutura pois tal administração não é simples e requer a consideração de muitos fatores e situações diferentes, mesmo que improváveis.

1.1. Sistema Computacional

Um sistema computacional pode ser entendido como um conjunto de hardware no qual uma série de funções (aplicações) podem ser executadas. Uma aplicação é um conjunto de funções que permite a realização de uma tarefa para um usuário. Essas funções podem ser executadas se o hardware subjacente for projetado para atendê-las.

Para que os sistemas computacionais fossem difundidos era necessário, então, a produção em larga escala de arquiteturas padronizadas que atendessem um conjunto básico de funções. As aplicações podem assim rodar em um sistema desde que suas funções possam ser traduzidas em funções básicas da máquina (hardware). Várias aplicações podem, por sua vez, serem executadas em uma mesma máquina. Os recursos de hardware, porém, terão que ser divididos entre os processos.

Essa divisão de recursos é feita por um software no nível abaixo das aplicações, os sistemas operacionais. Sistemas operacionais são camadas de software inseridas entre o hardware e as aplicações, capaz de orientar a utilização desses recursos físicos. O objetivo é otimizar essa utilização. Ele é responsável por organizar o uso da máquina criando abstrações como, por exemplo, um arquivo. Um arquivo não é algo real, e sim apenas uma abstração no nível do usuário que corresponde a um grupo específico de bits armazenados em um local físico determinado. A Figura 1 apresenta o exemplo de sistema computacional onde o Assembler é o componente que traduz as funções exigidas pelo software em uma linguagem que o sistema físico compreenda.

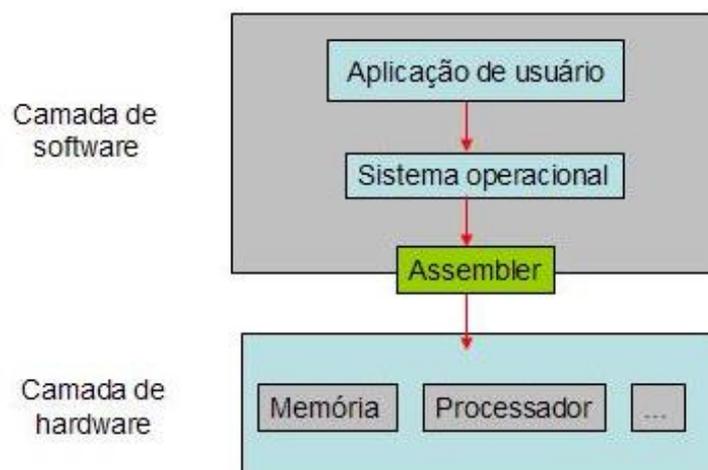


Figura 1 - Exemplo de sistema computacional

Essa abstração de recursos para que o usuário possa usar a máquina através de uma interface amigável é chamada virtualização de recursos. Com a utilização dessa técnica, cada programa tem a confortável ilusão de estar sendo executado sozinho, enquanto na verdade está dividindo os recursos da máquina com outras aplicações simultâneas.

2. Pipeline

Pipeline é uma técnica de implementação de processadores que permite a sobreposição temporal das diversas fases de execução das instruções. Em outras palavras, Pipeline é uma técnica de projeto onde o hardware processa mais de uma instrução de cada vez, sem esperar que uma instrução termine antes de começar a próxima, permitindo que os processadores rodem mais rapidamente.

Essa técnica consiste em executar as instruções de forma sequencial, ou seja, ao término do primeiro estágio de uma instrução o primeiro estágio da próxima instrução é inicializado. O Pipeline aumenta o número de instruções executadas simultaneamente e a taxa de instruções iniciadas e terminadas por unidade de tempo. O segredo do Pipeline é que todos os recursos envolvidos operam em paralelo. Notar que a técnica não melhora o tempo dos estágios das instruções (cada instrução continua necessitando da mesma quantidade de tempo para ser terminada), mas melhora o processo como um todo, visto que se fosse necessário esperar o término de uma instrução para iniciar outra, o tempo seria muito maior. Um exemplo prático de pipeline:

2.1. Lavagem de roupa sem Pipeline

A Figura 2 ilustra um processo de lavagem de roupas sem uso da técnica de Pipeline.

1. Coloca uma carga de roupa suja na lavadora;
2. Quando a lavadora termina o seu trabalho, coloca a roupa molhada na secadora;
3. Quando a secadora termina o seu trabalho, dobra as roupas;
4. Pede a empregada para guardar a roupa no armário;



Figura 2 – Lavagem sem pipeline

2.2. Lavagem com Pipeline

A Figura 3 ilustra um processo de lavagem de roupas com uso da técnica de Pipeline.

1. Tão logo termine de colocar a primeira carga de roupa na secadora, introduz a segunda carga de roupa na lavadora;
2. Quando a primeira carga de roupa estiver seca, coloque-a sobre a mesa, e ponha a roupa limpa molhada na secadora e, na lavadora, coloque a terceira carga de roupa suja;
3. Após 4 intervalos de tempo, todos os passos do processo de lavagem – chamados estágios do pipeline – estão operando de forma concorrente;

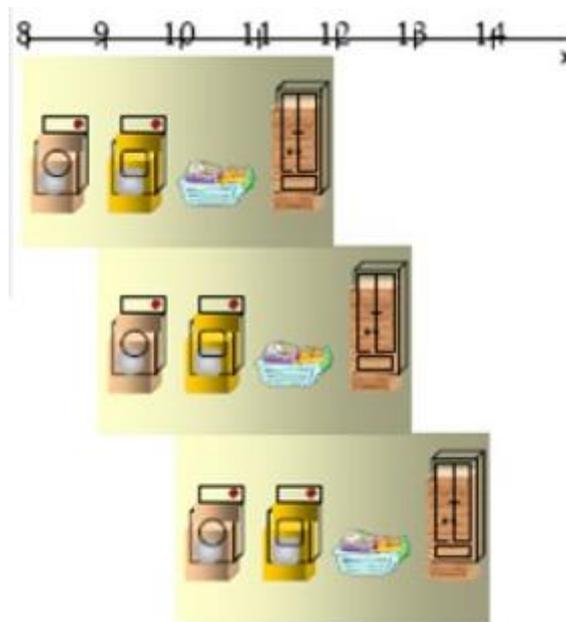


Figura 3 – Lavagem com pipeline

3. Sistema Operacional

Trata-se de um programa que controla a execução dos programas de aplicação e atua como uma interface entre o usuário do computador e o hardware do computador.

Um sistema operacional apresenta dois objetivos ou funções: conveniência, pois faz o sistema computacional mais conveniente de usar; e eficiência, pois permite que os recursos do sistema computacional sejam usados de maneira eficiente.

Um sistema operacional, tem também como função, gerir todos os periféricos de um computador, constituindo-se numa interface entre o usuário/programa e o hardware, de forma a facilitar o acesso aos recursos do mesmo (Figura 4).

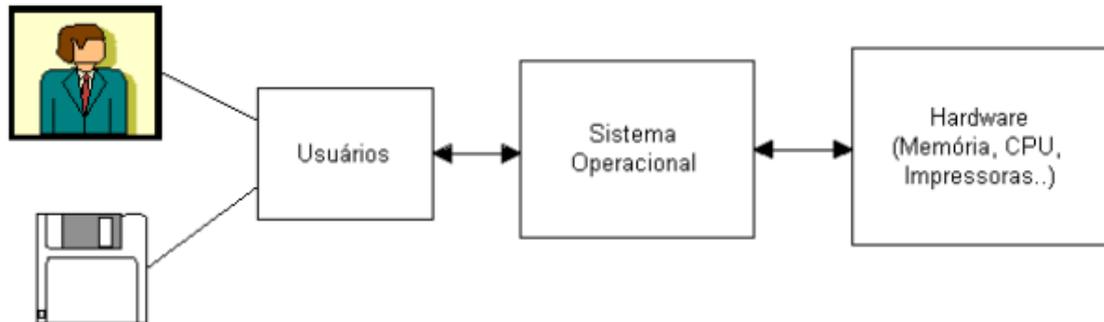


Figura 4 - Sistema Operacional como Interface entre usuários e recursos

Este tipo de "interface" que simplifica o acesso aos dispositivos (e os padroniza, em alguns casos) é também chamado de "virtualização de dispositivos". O caso mais comum deste tipo de virtualização são as operações "read" e "write" que, em geral, servem para ler e escrever em qualquer dispositivo, independentemente de seu funcionamento interno. Esta função de "virtualização" faz com que muitos usuários enxerguem o sistema operacional como uma "extensão da máquina" ou como uma máquina estendida.

Um sistema operacional deve atender aos seguintes princípios:

1. Oferecer os recursos do sistema de forma simples e transparente;
2. Gerenciar a utilização dos recursos existentes buscando seu uso eficiente em termos do sistema;
3. Garantir a integridade e a segurança dos dados armazenados e processados no sistema e também de seus recursos físicos.

Uma das responsabilidades do sistema operacional é controlar o acesso a dispositivos de forma organizada, de maneira que os usuários deste sistema (no caso, softwares) possam compartilhar recursos. Assim, se um software está usando a impressora e um segundo deseja utilizá-la, o sistema operacional deve agir de uma de duas formas:

1. Fazer com que o programa aguarde ou
2. Receber os dados a serem impressos e liberá-los para a impressora assim que ela esteja livre (no caso de impressoras, é muito comum que a segunda alternativa seja implementada, e o nome deste recurso é *spool*).

Uma das atribuições do sistema operacional é carregar na memória e providenciar a execução dos programas que o usuário solicita. Mesmo quando um programa qualquer está em execução, o sistema operacional pode continuar trabalhando. Por exemplo, muitos programas precisam realizar acesso ao teclado, vídeo e impressora, assim como acessos ao disco para ler e gravar arquivos. Todos esses acessos são realizados pelo sistema operacional, que fica o tempo todo ativo, prestando serviços aos programas que estão sendo executados.

O S.O. também faz um gerenciamento dos recursos do computador, para evitar que os programas entrem em conflito. Por exemplo, evita que dois programas

simultaneamente acessem a mesma área da memória, o que poderia causar grandes problemas.

4. Recursos e Ambiente Operacional

O hardware do computador determina suas capacidades brutas, ou seja, seus verdadeiros limites. Todos os elementos funcionais do computador são considerados recursos do sistema computacional e são representados pelos dispositivos que o compõe e que podem ser utilizados pelos usuários, ou seja: monitores de vídeo, teclado, mouse, mesas digitalizadoras, portas de comunicação serial e paralela, placas de rede ou comunicação, impressoras, scanners, unidades de disco flexível ou rígido, unidades de fita, unidades leitoras/gravadoras de CD, DVDs etc.

O sistema operacional aparece como uma camada sobre o hardware e firmware, mas simultaneamente envoltória deste. O sistema operacional está sobre o hardware e firmware pois deles depende para sua própria execução. Ao mesmo tempo é uma camada envoltória pois pretende oferecer os recursos do computador ao usuário do sistema minimizando os aspectos de como são tais dispositivos ou como serão feitas as operações que os utilizam. Desta forma o sistema operacional, através de sua interface, define uma nova máquina que é a combinação de um certo hardware com este sistema operacional.

O conjunto de hardware e sistema operacional, usualmente chamado de plataforma ou ambiente operacional será capaz de realizar tarefas de um modo específico ditado pela própria interface. Note que o ambiente operacional é distinto do hardware, pois o hardware do computador, por si só, não é capaz de copiar um determinado arquivo de uma unidade de disco rígido para uma unidade transportável (pen drive, por exemplo). Para realizar esta cópia, uma série de procedimentos devem ser executados, indo desde o acionamento das unidades utilizadas, passando pela localização das partes do arquivo origem e das áreas disponíveis no disco de destino, até a transferência efetiva dos dados.

4.1. Sistema Operacional como Gerenciador de Recursos

Como um gerenciador de recursos, a função do S.O. é controlar (ou gerenciar) a utilização de todos os recursos fornecidos pelo hardware e a sua distribuição entre os diversos programas que competem por ele de forma a garantir:

- A execução correta dos diversos programas;
- Alta eficiência na utilização dos recursos, ou seja, ao S.O. compete tratar de quem está usando qual recurso, aceitar (ordenadamente) requisições de um dado recurso, realizar a contagem de utilização de recursos e mediar conflitos nos pedidos de recursos por vários programas.

4.2. Máquina Real, Emuladores e Máquinas Virtuais

Uma máquina real é um conjunto de componentes físicos que fornecem operações para os níveis de abstração. Seu núcleo é o processador e o chip-set da placa-mãe, e inclui todos os recursos como o áudio, o vídeo, a memória, as portas, etc.

Um emulador é um programa que funciona para enganar uma aplicação. Seja uma aplicação que funciona em certa máquina real. Caso ela tenha que funcionar em outra máquina que não oferece os mesmos recursos, um emulador pode ser usado. Ele será capaz de traduzir instruções definidas para a máquina de origem em instruções para a máquina emulada, de forma a obter os mesmos resultados que obteria se executasse na máquina original. A Figura 5 ilustra a emulação de um sistema operacional, técnica que permite rodar uma aplicação projetada para um sistema operacional em outro incompatível.

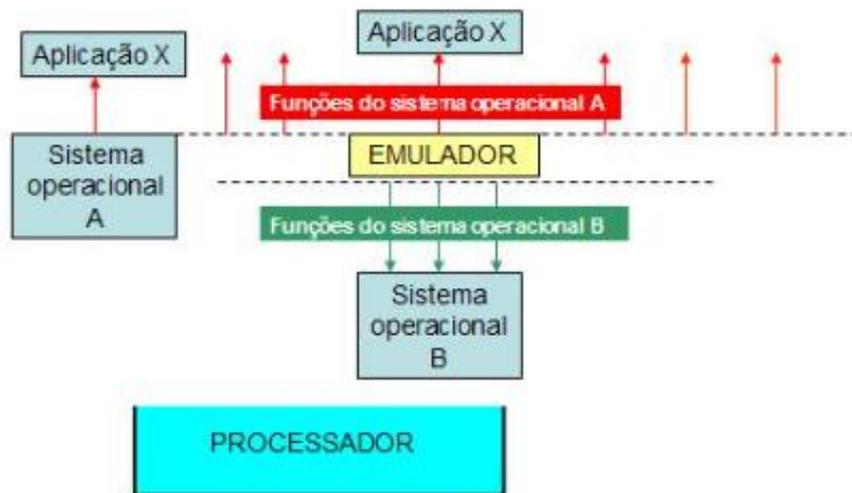


Figura 5 - Emulação de SO

4.3. Sistema Operacional como Máquina Virtual

Uma máquina virtual é a imitação, por software, de uma máquina real. Sendo assim, sistemas operacionais podem rodar em máquinas virtuais, e terão a ilusão de rodarem em uma máquina real. Essas imitações podem ser criadas no nível de aplicação ou no nível de sistema operacional.

Quando criadas no nível de aplicação, uma máquina virtual é executada em cima de um sistema operacional, chamado anfitrião. Como o nível acima vai acreditar que abaixo dele há uma máquina real, pode-se executar outro sistema operacional possivelmente diferente do que está abaixo da imitação. Isso é extremamente útil quando se deseja rodar uma aplicação projetada para um sistema operacional em outro incompatível.

Quando criadas no nível de sistema operacional, um Monitor de Máquina Virtual (MMV ou *Virtual Machine Monitor – VMM*) é implementado. Ele controla o hardware e cria várias máquinas virtuais, ou seja, várias imitações do mesmo. Em cada “imitação” pode-se rodar um sistema operacional, que

acreditará estar usando uma máquina real e não uma imitação. O resultado vantajoso é poder rodar diversos sistemas operacionais, cada um com suas aplicações específicas, em uma mesma máquina.

É função do monitor de máquinas virtuais gerenciar o uso dos dispositivos, multiplexando no tempo as tentativas dos diversos programas em usar a máquina real. Outros recursos como uso de memória e de dispositivos de entrada e saída também devem ser administrados uma vez que a virtualização não resolve o problema de um hardware não poder ser duplicado fisicamente. Dessa forma, a virtualização faz com que todos os recursos da máquina sejam compartilhados entre os usuários sem que eles saibam disso. Continuarão acreditando estar rodando direto em uma máquina real. A Figura 6 mostra os dois tipos de máquina virtual e a comparação com um sistema não virtualizado.

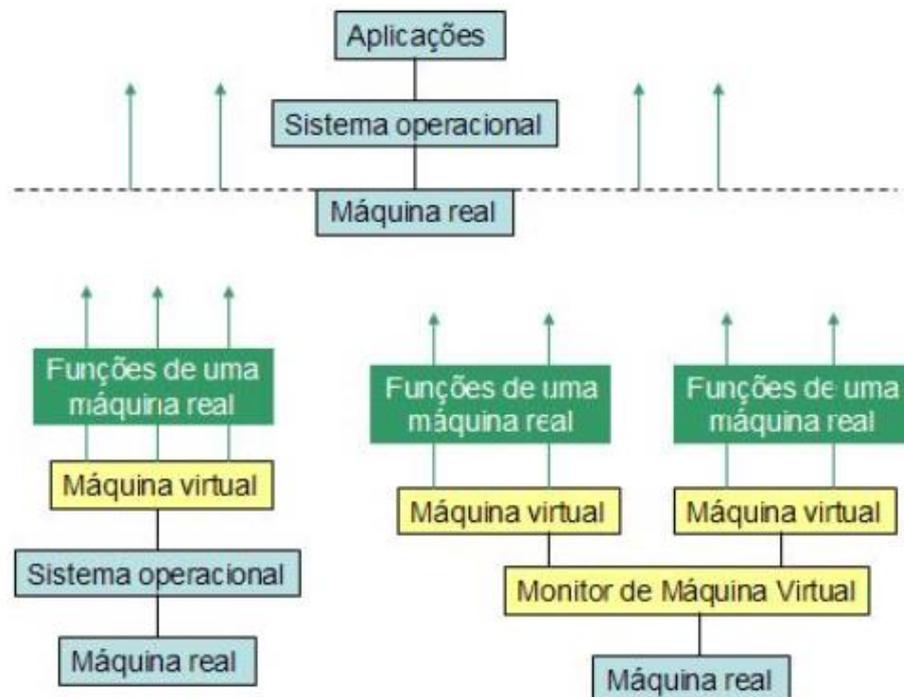


Figura 6 - Máquina real e máquina virtual

Como formador de uma máquina virtual, a função do S.O. é apresentar ao usuário uma máquina com as seguintes características:

- Facilidade de operação, isto é, o S.O. deve fornecer uma interface entre o usuário e o hardware que apresenta maior facilidade de programação de a presente originalmente no hardware.

Um exemplo típico é a escrita em disco, onde um controlador precisa das seguintes instruções: recalibração - corresponde a um ajuste da cabeça de leitura na primeira trilha, pois cada movimento posterior da cabeça é sempre relativo, de forma que se a mesma se apresenta inicialmente mal posicionada, todos os futuros posicionamentos serão errados; movimento da cabeça - o deslocamento da mesma para a trilha requerida; espera do setor - que representa uma espera até que a cabeça de leitura se posiciona sobre o setor;

escrita dos dados – verificação - para garantir que os dados foram escritos e sem nenhum erro. Além destes passos podem existir outros, dependendo do controlador utilizado, do tipo de acesso (leitura, escrita, formatação, etc.) e se ocorrem erros ou não. Atualmente existem diversos controladores de disco que cuidam eles mesmo de tarefas como posicionamento da cabeça, leitura ou escrita, verificação de erros.

- Extensão das capacidades da máquina: o S.O. pode fornecer algumas capacidades não presentes no computador original, como, por exemplo, múltiplos usuários e sistemas de proteção de acesso.

5. Computadores Pessoais e Redes

Com a integração em larga escala e o surgimento dos microcomputadores, surge também o conceito de "*user-friendly*" para S.O., que corresponde ao desenvolvimento de sistemas operacionais para serem utilizados por pessoas sem nenhum conhecimento de computação e que, provavelmente, não têm nenhum interesse em vir a conhecer algo.

Um outro desenvolvimento interessante que foi bastante impulsionado pelos microcomputadores (apesar de não depender dos mesmos) é o de sistemas operacionais para redes de computadores, que consistem em computadores distintos interligados por elementos de comunicação.

Os sistemas operacionais para redes são divididos em duas categorias:

- **Sistemas operacionais de rede:** no qual cada usuário tem conhecimento de seu próprio computador e pode acessar dados em outros computadores;
- **Sistemas operacionais distribuídos:** em que o sistema operacional faz com que todos os computadores da rede formem uma unidade, de forma que nenhum usuário tenha conhecimento de quantos computadores há na rede ou de em qual (ou quais) computador o seu específico programa está executando.

As modificações do hardware em favor das redes de comunicação implicaram em ajustes nos Sistemas Operacionais, adaptando-os para o novo ambiente de processamento. Os computadores pessoais, que antes funcionavam isolados, já possuíam seus respectivos Sistemas Operacionais Locais (SOL). Posteriormente surgiram os Sistemas Operacionais de Rede (SOR), como extensão dos sistemas locais, complementando-os com o conjunto de funções básicas e de uso geral, necessárias à operação das estações de trabalho, de forma a tornar transparente o uso dos recursos compartilhados no sistema computacional.

5.1. Transparência

A transparência é um dos requisitos fundamentais dos Sistemas Operacionais de Rede. Nesse sentido, os SOR's devem atuar de forma que os usuários utilizem os recursos da rede como se estivessem operando localmente. A solução encontrada para estender o Sistema Operacional das estações da

rede, sem modificar sua operação local, foi a introdução de um módulo redirecionador, como mostra a Figura 7.

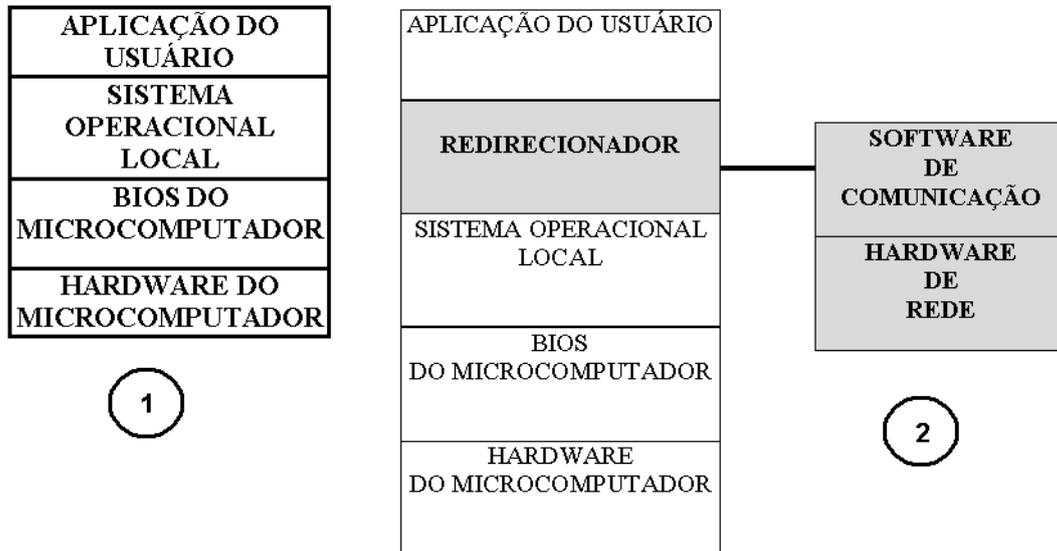


Figura 7 – Sistema Operacional Local sem Redirecionador (1) e com Redirecionador (2)

O redirecionador funciona interceptando as chamadas feitas pelas aplicações ao Sistema Operacional Local, desviando aquelas que dizem respeito a recursos remotos para o módulo do Sistema Operacional em Rede, responsável pelos serviços de comunicação que providenciam conexão ao dispositivo remoto.

Para as aplicações dos usuários, a instalação do Sistema Operacional de Rede é percebida apenas pela adição de novos recursos (chamados recursos verticais) aos que elas possuíam anteriormente. O redirecionador, como apresentado, foi o mecanismo sobre o qual foram desenvolvidos os Sistemas Operacionais de Rede atuais.

5.2. Arquiteturas Cliente-Servidor e Peer-to-Peer

A interface entre as aplicações de usuário e o Sistema Operacional baseia-se usualmente, em interações solicitação/resposta, onde a aplicação solicita um serviço (abertura de um arquivo, impressão de bloco de dados, alocação de uma área de memória etc.) através de uma chamada ao sistema operacional. O sistema operacional, em resposta, executa o serviço solicitado e responde, informando o status da operação (se foi executado com sucesso ou não) e transferindo os dados resultantes da execução para a aplicação, quando for o caso. No modo de interação Cliente-Servidor, a entidade que solicita um serviço é chamada cliente e a que presta o serviço é o servidor.

A interação cliente-servidor constitui-se no modo básico de interação dos sistemas operacionais de redes atuais. As estações que disponibilizam a outras estações o acesso aos seus recursos através da rede devem possuir a entidade (ou módulo) servidor. As estações que permitem que suas aplicações utilizem recursos compartilhados com outras estações devem possuir a entidade (ou módulo) cliente.

Nas estações que possuem o módulo cliente, o Sistema Operacional de Rede ao receber um pedido de acesso a um recurso localizado em outra estação da rede, monta uma mensagem contendo o pedido e a envia ao módulo servidor da estação onde será executado o serviço. Na estação remota, o SOR recebe a mensagem, providencia a execução (nos casos onde o pedido envolve a devolução para o SOR na estação requerente). Quando o SOR na estação que requisitou o serviço recebe a mensagem transportando a resposta, ele faz sua entrega à aplicação local.

5.3. Módulos Cliente e Servidor

As funções necessárias do SOR nos módulos clientes são diferentes das funções nos módulos servidores. No módulo cliente, o SOR restringe-se praticamente a fornecer serviços de comunicação de pedidos para o servidor e a entregar as respostas às aplicações. No módulo servidor além das funções de comunicação, vários outros serviços são executados. Um desses serviços é o controle do acesso aos recursos compartilhados por vários usuários através da rede, para evitar, por exemplo, que um usuário não autorizado apague arquivos que não lhe pertencem.

Portanto, podemos classificar os módulos de um Sistema Operacional de Rede, instalados nas estações, em módulo cliente e módulo servidor do sistema operacional. Na arquitetura Peer-to-Peer, em todas as estações o sistema operacional de rede possui o módulo cliente (SORC) e módulo servidor (SORS), conforme mostra a Figura 8.

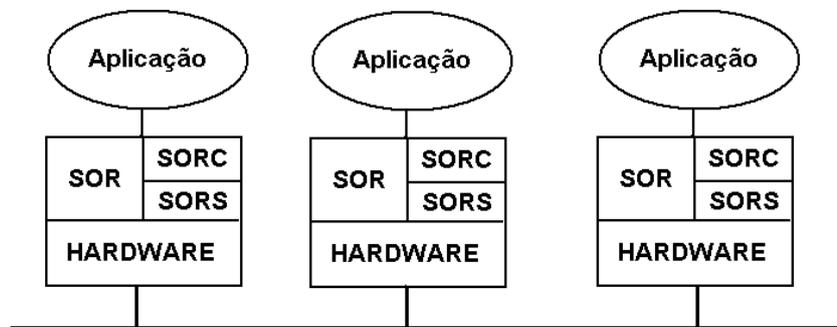


Figura 8 - Arquitetura Peer-to-Peer

Na arquitetura Cliente-Servidor, as estações da rede se dividem em estações clientes, que só possuem as funções do módulo cliente acopladas ao seu sistema operacional local, e em estações servidoras.

As estações servidoras necessariamente possuem as funções do módulo servidor e podem, opcionalmente, possuir também as funções do módulo cliente (possibilitando, por exemplo, que um servidor seja cliente de outro, caso típico da relação entre servidores de impressão de arquivos). Nessa arquitetura, usualmente, as estações servidoras não permitem usuários locais. Elas são integralmente dedicadas ao atendimento de pedidos enviados pelas estações clientes através da rede.

Na arquitetura Cliente-Servidor com servidor não dedicado, as estações servidoras possuem sistema operacional local (SOL) que é estendido por um módulo servidor (SORS) e um módulo cliente (SORC). O módulo cliente pode ser usado tanto pelo servidor, quanto pelas aplicações dos usuários locais da estação servidora. Assim, os recursos locais das estações servidoras são compartilhados tanto pelos usuários atendidos pelo sistema operacional local (que também podem ter acesso a serviços de outros servidores) quanto pelos usuários remotos que fazem pedidos ao Sistema Operacional de Rede através da rede (Figura 9).

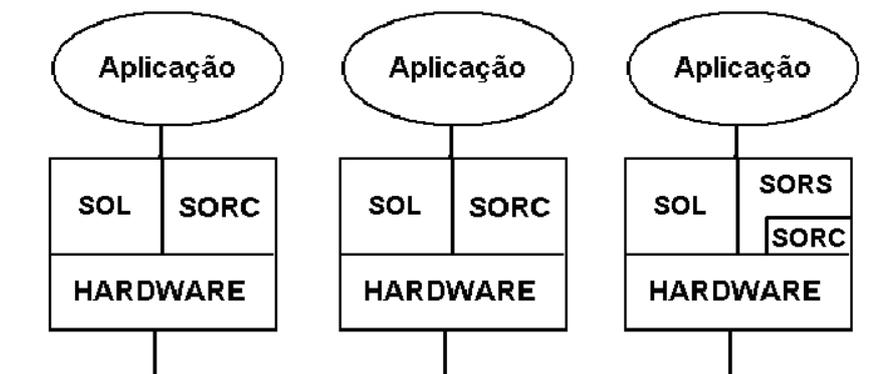


Figura 9 - Arquitetura Cliente-Servidor com servidor não dedicado

É importante observar que, como uma estação servidora possui um módulo cliente, seu módulo servidor pode ser cliente de outra estação servidora, como em alguns servidores dedicados.

6. Processos

Um processo é a representação de um programa. Trata-se basicamente de uma abstração que reúna os dados a serem manipulados pelo programa e as funções usadas pelo programa. É implementado pela representação de um espaço de endereçamento lógico divididos em regiões que guardam informações específicas.

A execução de um processo é feita através de dois registradores: PC (*Program Counter* - Contador de Programa), que indica a função a ser executada, e SP (*Stack Pointer* - Apontador de Pilha), que indica onde devem ser guardados, no processo, os dados de retorno dessa função, seus parâmetros e suas variáveis locais. O esquema da Figura 10 mostra as funções de cada região de um processo e a atuação dos registradores.

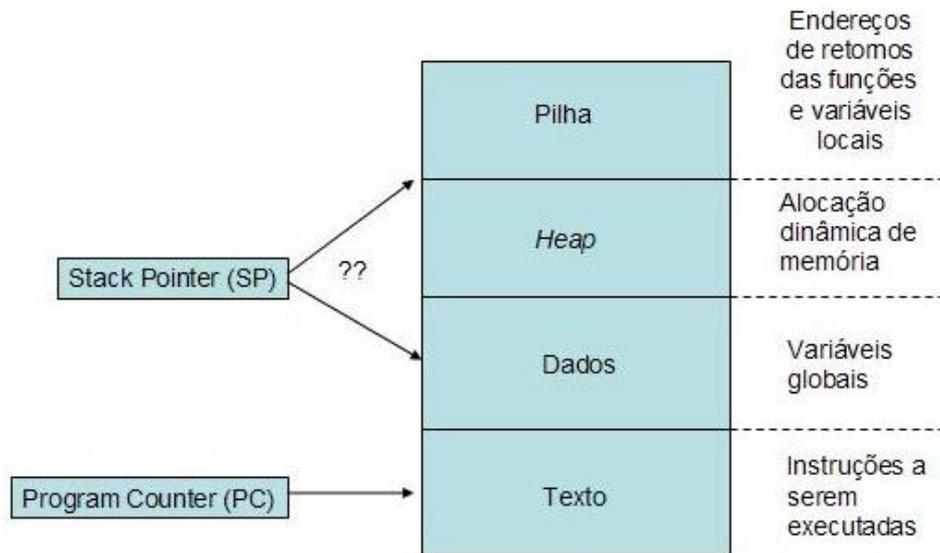


Figura 10 - Representação de um processo

No método tradicional de S.O., considera-se todo o software como organizado em um número de processos sequenciais ou, mais simplesmente, processos. Define-se um processo como um programa em execução, sendo que para sua especificação completa, deve-se incluir tanto o programa propriamente dito como os valores de variáveis, registradores, contador de programa (*Program Counter* - PC), e outros dados necessários à definição completa de seu estado. Cada processo trabalha como se possuísse para si uma CPU própria, chamada CPU virtual. Na realidade, na grande maioria dos casos uma única CPU é compartilhada entre todos os processos. Isto é, existe apenas uma CPU real, mas tantas CPU virtuais quantos forem os processos.

Alguns sistemas de multiprocessamento apresentam diversas CPU reais, entretanto, mesmo nestes casos, ocorre a necessidade de compartilhamento de várias ou todas CPU pelos diversos processos. Os processos, em uma mesma CPU real executam um por vez, mas essa execução é realizada de forma a criar a ilusão de que os mesmos estão executando em paralelo.

Um fator derivado do método como a implementação das CPU virtuais é realizado, e que apresenta fundamental importância no desenvolvimento de sistemas operacionais, é o fato de que a taxa de execução de cada processo não é uniforme nem reproduzível. Isto quer dizer que não é possível assumir nada com relação à taxa com que cada processo será executado, nem em relação a outros processos, nem em relação a diversas execuções de um mesmo processo. Uma das implicações disto é que os programas não podem ser feitos levando em consideração as temporizações de execução das instruções (por exemplo, não se pode fazer um programa que execute certo número de repetições de um "loop" e com isso espere conseguir uma demora fixa de tempo, como um segundo). Isto ocorre porque não se sabe o momento em que a CPU será chaveada para outro processo, fazendo com que o atual tenha sua continuação retardada.

6.1. Estados dos Processos

Durante a sua existência, os processos podem se apresentar, do ponto de vista do sistema, em diferentes estados. O primeiro estado a considerar consiste naquele em que um processo está efetivamente executando, isto é, está rodando. Durante a sua existência, um processo pode necessitar interagir com outros processos.

O segundo estado ocorre quando durante a execução de dois processos, como não se sabe a que taxa cada um será executado, pode ocorrer de que um processo queira ler um dado de entrada que ainda não foi gerado pelo outro. Quando isto ocorre, o processo que solicita a leitura deve ser bloqueado até haver entradas.

O terceiro estado a considerar é quando o processo tem todas as condições de executar, mas isso não acontece porque a CPU foi alocada para a execução de um outro processo. Neste caso o processo está pronto.

Os três estados e suas inter-relações podem ser apresentados na Figura 11.

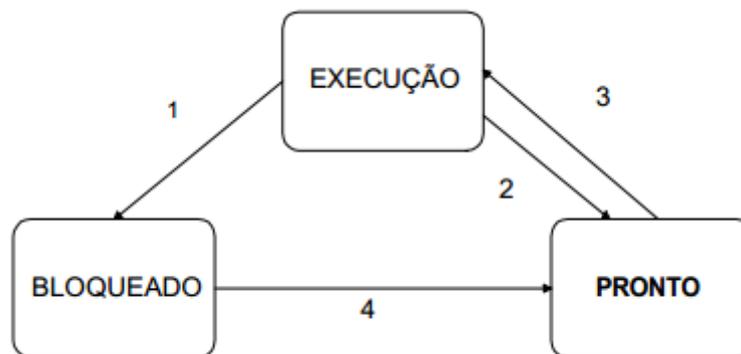


Figura 11 - Estados de um Processo

1. bloqueamento por falta de entrada;
2. escalonador selecionou um outro processo;
3. escalonador selecionou este processo;
4. entrada ficou disponível.

Em alguns sistemas, o processo deve requisitar o bloqueamento quando notar que não dispõe de entradas. Em outros sistemas, o bloqueamento ocorre automaticamente pelo próprio sistema, sem que o usuário precise se ocupar disso durante a programação.

O escalonador (*scheduler*) é parte do S.O. responsável pelo chaveamento da CPU entre os diversos processos, tendo como principal objetivo a conciliação da necessidade de eficiência do sistema como um todo e de justiça para com os processos individuais.

Para a implementação de processos, além dos programas à serem executados, devem ser guardadas algumas informações, necessárias ao escalonador, para permitir que um processo seja reassumido exatamente no ponto em que foi interrompido. Esses dados são armazenados na chamada tabela de processos, que consiste em um vetor de estruturas com uma entrada por processo. Esta estrutura contém dados como: contador de programa (PC),

ponteiro de pilha (SP), estado dos arquivos abertos, registradores, além de diversas outras informações.

6.2. Níveis de Abstração e Interface

A divisão de um sistema computacional em diversos níveis de abstração é um princípio importante. Cada nível de abstração troca informações com os níveis vizinhos sem precisar saber como essas informações são tratadas pelos outros níveis. Para que essa troca de informações seja feita, é necessária a interposição de uma interface entre os níveis. Os níveis de abstração podem ser divididos em aplicação, bibliotecas, sistema operacional e processador.

O nível de aplicação é responsável por executar a aplicação desejada pelo usuário, possivelmente trocando informações com o mesmo. As bibliotecas são constituídas por grupos de funções usadas pelo programador para fazer o programa acessar recursos protegidos da máquina. Isso permite que bibliotecas sejam criadas com o objetivo de fornecer aos programadores funções que gerem instruções privilegiadas. As instruções não-privilegiadas geradas pela aplicação atuam diretamente no processador, enquanto as privilegiadas são tratadas pelas bibliotecas. Ao tentar usar recursos protegidos do sistema, a aplicação se comunica com a biblioteca pela Interface Aplicativa de Programação (*Application Programming Interface* - API), conforme mostra a Figura 12.

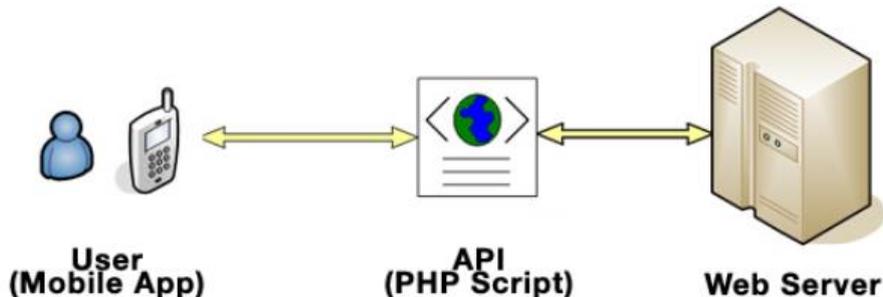


Figura 12 - Utilização de API

O terceiro nível, o sistema operacional, se encarrega de executar as instruções não privilegiadas de forma a otimizar a utilização de recursos da máquina. A biblioteca, encarregada de fazer o primeiro tratamento desse tipo de tarefa, usa as “chamadas de sistema” para pedir auxílio ao sistema operacional e usar recursos de hardware. As chamadas de sistema são, assim, a interface entre Bibliotecas e Sistema Operacional.

O nível de processador é o principal, onde o computador realmente “pensa”. Tudo o que acontece nos outros níveis serve para simplificar o uso do sistema através de instruções, que o processador entende. O conjunto de instruções não-privilegiadas pode ser usado diretamente por todos os níveis acima. O conjunto de instruções privilegiadas, porém, só pode ser usado pelo sistema operacional. Esses dois conjuntos são as interfaces entre o processador e os outros níveis.

O conjunto de instruções (ou ISA) é a interface entre o nível de abstração de hardware e o S.O. Essa interface é composta por todos os códigos de máquina

aceitos pelo processador, que correspondem cada um a uma instrução. Um processador possui, tipicamente, pelo menos dois modos de operação, o não privilegiado e o privilegiado.

A Figura 13 mostra um modelo de S.O. dividido em níveis de abstração (Aplicação, Bibliotecas, SO, Processador) e interfaces (API, Camadas de sistema, ISA privilegiadas, ISA não-privilegiadas). As aplicações somente acionam o sistema operacional quando querem executar instruções privilegiadas.



Figura 13 - Níveis de abstração