



Centro
Universitário
FMU

FMU

Professor Lima

Sistemas Operacionais



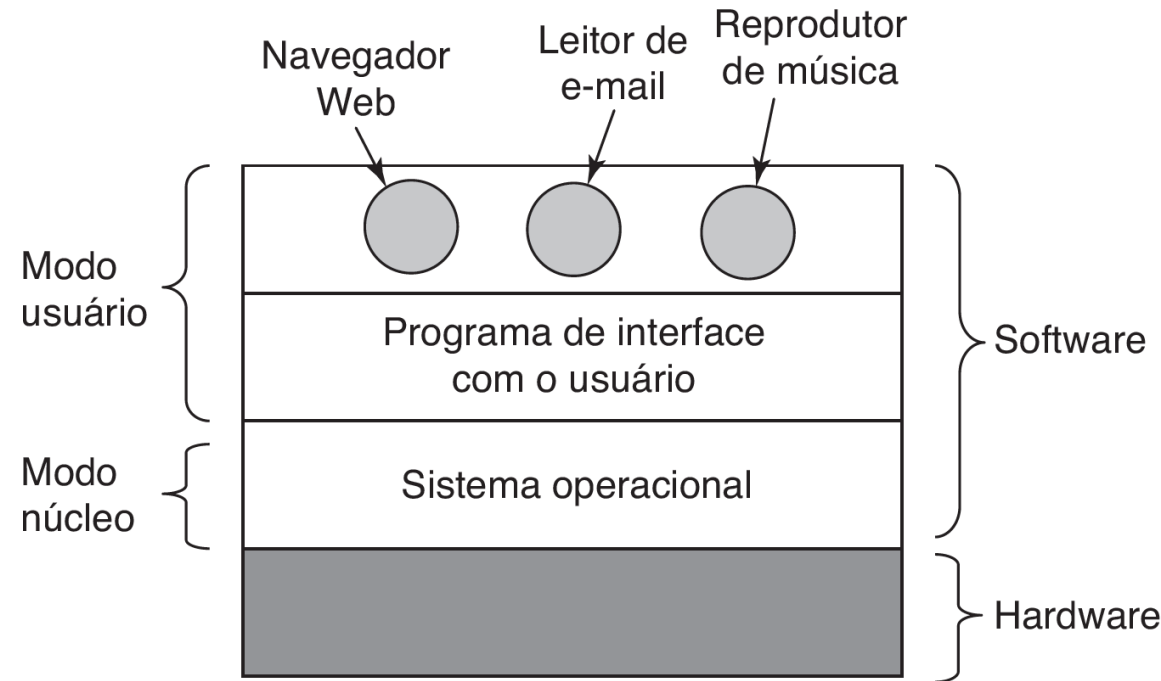
Identificar os conceitos sobre um Sistema Operacional.

Um sistema computacional moderno consiste em um ou mais processadores, memória principal, discos, impressoras, teclado, mouse, monitor, interfaces de rede e outros dispositivos de entrada e saída

- Se cada programador de aplicações tivesse de entender como tudo isso funciona em detalhes, nenhum código chegaria a ser escrito. Além disso, gerenciar todos esses componentes e usá-los de maneira otimizada é um trabalho extremamente difícil
- Por isso, os computadores têm um dispositivo de software denominado **sistema operacional**, cujo trabalho é fornecer aos programas do usuário um modelo de computador melhor, mais simples e mais limpo e lidar com o gerenciamento de todos os recursos mencionados

- A maioria dos usuarios deve ter alguma experiência com um sistema operacional como Windows, Linux, Free-BSD ou Mac OS X, mas as aparências podem enganar. O programa com o qual os usuários interagem, normalmente chamado de **shell** (ou interpretador de comandos) quando é baseado em texto e de **GUI** (*graphical user interface* — interface gráfica com o usuário) quando usa ícones, na realidade não é parte do sistema operacional embora o utilize para realizar seu trabalho.

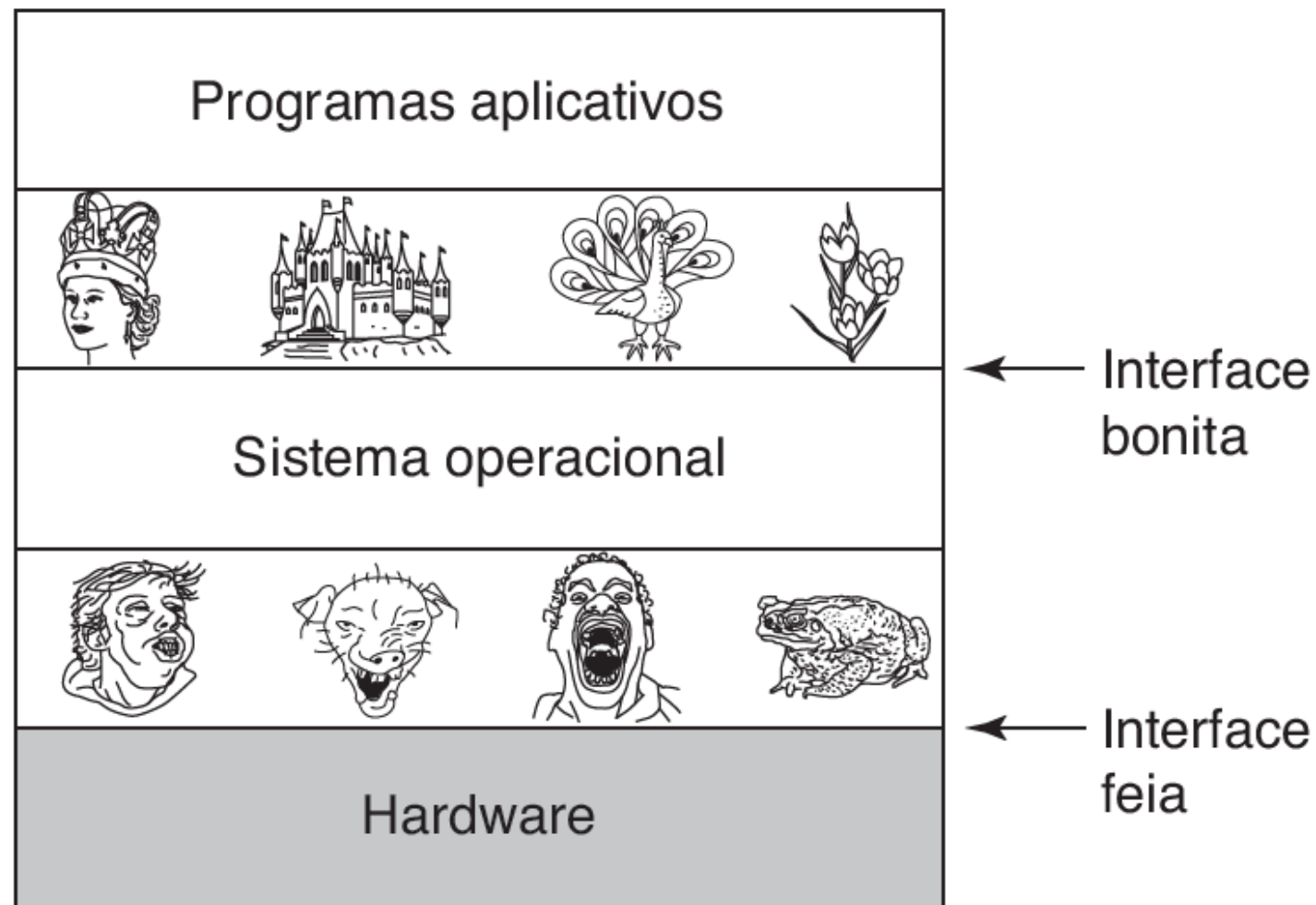
- Na parte inferior vemos o hardware. Ele consiste em chips, placas, discos, teclado, monitor e objetos físicos semelhantes. Na parte superior do hardware está o software. A maioria dos computadores tem dois níveis de operação: modo núcleo e modo usuário. O sistema operacional é a peça mais básica de software e opera em **modo núcleo** (também chamado **modo supervisor ou KERNEL**). Nesse modo ele tem acesso completo a todo o hardware e pode executar qualquer instrução que a máquina seja capaz de executar. O resto do software opera em **modo usuário**, no qual apenas um subconjunto de instruções da máquina está disponível.



- Em particular, as instruções que afetam o controle da máquina ou realizam E/S (Entrada/Saída) são proibidas para programas de modo usuário.
- O programa de interface com o usuário, shell ou GUI, é o nível mais inferior do software de modo usuário e permite que este inicie outros programas, como o navegador Web, leitor de e-mail ou reproduutor de música. Esses programas também usam muito o sistema operacional.
- Uma distinção importante entre o sistema operacional e o software normal (modo usuário) é que, se o usuário não gostar de um determinado leitor de e-mail, ele será livre para obter outro ou escrever seu próprio leitor de e-mail, se o quiser; mas não lhe é permitido escrever seu próprio manipulador de interrupção do relógio, que é parte do sistema operacional e normalmente está protegida pelo hardware contra tentativas de alterações pelo usuário.

- O código-fonte de um sistema operacional como o Linux ou o Windows tem cerca de cinco milhões de linhas de código
- Agora deve estar claro por que os sistemas operacionais têm vida longa — eles são muito difíceis de escrever e, uma vez tendo escrito um, o proprietário não se dispõe a descartá-lo e começar de novo
- A **arquitetura** (conjunto de instruções, organização de memória, E/S e estrutura de barramento) da maioria dos computadores em nível de linguagem de máquina é primitiva e de difícil programação, especialmente a entrada/saída.

- Uma das principais tarefas do sistema operacional é ocultar o hardware e oferecer aos programas (e seus programadores) abstrações precisas, claras, elegantes e coerentes com as quais trabalhar, os sistemas operacionais transformam o feio em bonito



- Deve-se observar que os clientes reais do sistema operacional são os programas aplicativos (via programadores de aplicativos, naturalmente). São eles que lidam diretamente com o sistema operacional e suas abstrações. Por outro lado, os usuários finais lidam com abstrações fornecidas pela interface do usuário, seja a linha de comandos **shell** ou uma **interface gráfica**.
- Para esclarecer esse ponto, considere a área de trabalho normal do Windows e o prompt de comando, orientado a linhas de comando. Ambos são programas executados no sistema operacional e usam as abstrações que o Windows fornece, mas oferecem interfaces de usuário muito diferentes.

O sistema operacional como um gerenciador de recursos

- O conceito de um sistema operacional como provedor de uma interface conveniente a seus usuários é uma visão top-down (abstração do todo para as partes). Em uma visão alternativa, bottom-up (abstração das partes para o todo), o sistema operacional gerencia todas as partes de um sistema complexo.
- Computadores modernos são constituídos de processadores, memórias, temporizadores, discos, dispositivos apontadores tipo mouse, interfaces de rede, impressoras e uma ampla variedade de outros dispositivos.

- Segundo essa visão, o trabalho do sistema operacional é fornecer uma alocação ordenada e controlada de processadores, memórias e dispositivos de E/S entre vários programas que competem por eles
- Sistemas operacionais modernos permitem que múltiplos programas sejam executados ao mesmo tempo. Imagine o que aconteceria se três programas em execução em algum computador tentassem imprimir suas saídas simultaneamente na mesma impressora

- O gerenciamento de recursos realiza o **compartilhamento** (ou multiplexação) desses recursos de duas maneiras diferentes: no tempo e no espaço
- Quando um recurso é compartilhado (multiplexado) no tempo, diferentes programas ou usuários aguardam sua vez de usá-lo
- Primeiro, um deles obtém o uso do recurso, daí um outro, e assim por diante
- Por exemplo, com somente uma CPU e múltiplos programas precisando ser executados nela, o sistema operacional aloca a CPU a um programa, e após este ser executado por tempo suficiente, outro programa obtém seu uso, então outro e, por fim, o primeiro programa novamente.

- Determinar como o recurso é compartilhado no tempo — quem vai depois de quem e por quanto tempo — é tarefa do sistema operacional.



História dos sistemas operacionais

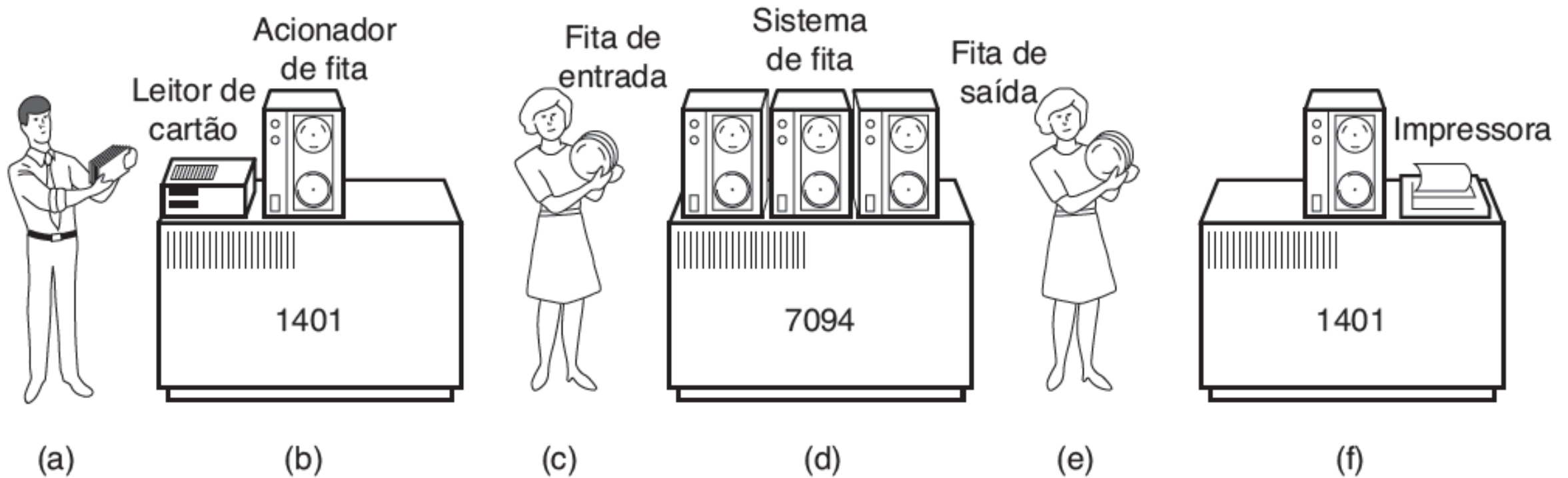
- A história dos sistemas operacionais é bastante ligada à arquitetura de computadores sobre a qual eles são executados
- O primeiro computador digital foi projetado pelo matemático inglês Charles Babbage (1792–1871) pois era inteiramente mecânica e a tecnologia de sua época não poderia produzir as rodas, as engrenagens e as correias de alta precisão que eram necessárias. É óbvio que a máquina analítica não possuía um sistema operacional.
- Babbage percebeu que seria preciso um software para sua máquina analítica. Para isso, ele contratou uma jovem chamada Ada Lovelace, filha do famoso poeta inglês Lord Byron, como a primeira programadora do mundo. A linguagem de programação Ada® foi assim denominada em sua homenagem

A primeira geração (1945–1955) — válvulas

- O professor John Atanasoff e seu então aluno de graduação Clifford Berry construíram o que consideramos o primeiro computador digital em funcionamento, na Universidade do Estado de Iowa. Ele usava 300 válvulas. Quase ao mesmo tempo, Konrad Zuse, em Berlim, construiu o computador Z3 de relés. Em 1944, o Colossus foi desenvolvido por um grupo em Bletchley Park, Inglaterra, o Mark foi construído por Howard Aiken em Harvard e o ENIAC foi construído por William Mauchley e seu aluno de graduação J. Presper Eckert na Universidade da Pensilvânia. Alguns eram binários, alguns usavam válvulas, alguns eram programáveis, mas todos eram muito primitivos e levavam segundos para executar até o cálculo mais simples.

A segunda geração (1955–1965) — transistores e sistemas em lote (batch)

- A introdução do transistor em meados da década de 1950 mudou o quadro radicalmente. Os computadores tornaram-se suficientemente confiáveis para que pudessem ser fabricados e comercializados com a expectativa de que continuariam a funcionar por tempo suficiente para executar algum trabalho útil.
- Essas máquinas — então denominadas **computadores de grande porte** (*mainframes*) — ficavam isoladas em salas especiais com ar-condicionado, operadas por equipes profissionais.
- Para uma **tarefa** (isto é, um programa ou um conjunto de programas) ser executada, o programador primeiro escrevia o programa no papel (em Fortran ou em linguagem assembly) e depois o perfurava em cartões. Ele então levava o maço de cartões para a sala de entradas, entregava-o a um dos operadores e ia tomar um café até que a saída impressa estivesse pronta.



- Por causa do alto custo do equipamento, era natural que se começasse a buscar maneiras de reduzir o desperdício de tempo no uso da máquina. A solução geralmente adotada era a do **sistema em lote** (*batch*)
- Um sistema em lote (*batch*) antigo. (a) Os programadores levavam os cartões para o 1401. (b) O 1401 gravava o lote de tarefas em fita. (c) O operador levava a fita de entrada para o 7094. (d) O 7094 executava o processamento. (e) O operador levava a fita de saída para o 1401. (f) O 1401 imprimia as saídas.

A terceira geração (1965–1980) — CIs e multiprogramação

- O IBM 360 foi a primeira linha de computadores a usar **circuitos integrados** (CIs) em pequena escala, propiciando, assim, uma melhor relação custo–benefício em comparação à segunda geração de máquinas, construídas com transistores individuais.
- O forte da ideia de ‘família de máquinas’ era simultaneamente sua maior fraqueza. A intenção era que qualquer software, inclusive o sistema operacional **OS/360**, pudesse ser executado em qualquer um dos modelos.
- Embora os sistemas operacionais de terceira geração fossem adequados para grandes cálculos científicos e processamento maciço de dados comerciais, eram basicamente sistemas em lote (*batch systems*)
- O anseio por respostas mais rápidas abriu caminho para o tempo compartilhado ou **timesharing**, uma variante da multiprogramação na qual cada usuário se conectava por meio de um terminal on-line.

- Ken Thompson, um dos cientistas da computação do Bell Labs que trabalharam no projeto MULTICS, achou um pequeno minicomputador PDP-7 que ninguém estava usando e aproveitou-o para escrever uma versão despojada e monousuário do MULTICS.
- Esse trabalho desenvolveu-se e deu origem ao sistema operacional **UNIX**[®], que se tornou muito popular no mundo acadêmico, em agências governamentais e em muitas empresas.
- Duas das principais versões desenvolvidas, o **System V**, da AT&T, e o **BSD** (*Berkeley software distribution* — distribuição de software de Berkeley), da Universidade da Califórnia em Berkeley,

- Para tornar possível escrever programas que pudessem ser executados em qualquer sistema UNIX, o IEEE desenvolveu um padrão para o UNIX denominado **POSIX** (*portable operating system interface* — interface portátil para sistemas operacionais, ao qual a maioria das versões UNIX agora dá suporte.
- O POSIX define uma interface mínima de chamada de sistema a que os sistemas em conformidade com o UNIX devem dar suporte
- Em 1987, Andrew S. Tanenbaum lançou um pequeno clone do UNIX, denominado **MINIX**, com objetivo educacional. Funcionalmente, o MINIX é muito similar ao UNIX, incluindo o suporte ao POSIX.

A quarta geração (1980–presente) — computadores pessoais

- Com o desenvolvimento de circuitos integrados em larga escala (*large scale integration* — LSI), que são chips contendo milhares de transistores em um centímetro quadrado de silício, surgiu a era dos computadores pessoais.
- Em termos de arquitetura, os computadores pessoais (inicialmente denominados **microcomputadores**) não eram muito diferentes dos minicomputadores
- Em 1974, a Intel lançou o 8080, a primeira CPU de 8 bits de uso geral, e buscava um sistema operacional para o 8080, em parte para testá-lo.
- Gary Kildall então escreveu para ele um sistema operacional baseado em disco denominado **CP/M** (*control program for microcomputers* — programa de controle para microcomputadores)

- No início dos anos 1980, a IBM projetou o IBM PC e buscou um software para ser executado nele. O pessoal da IBM entrou em contato com Bill Gates para licenciar seu interpretador Basic.
- Também lhe foi indagado se ele conhecia algum sistema operacional que pudesse ser executado no PC. Gates sugeriu que a IBM contatasse a Digital Research, a empresa que dominava o mundo dos sistemas operacionais naquela época.
- Então Gates percebeu que uma fabricante local de computadores, a Seattle Computer Products, possuía um sistema operacional adequado, o **DOS** (*disk operating system* — sistema operacional de disco). Entrou em contato com essa empresa e disse que queria comprá-la (supostamente por 75 mil dólares), o que foi prontamente aceito, Gates ofereceu à IBM um pacote DOS/Basic
- O sistema revisado teve seu nome mudado para **MS-DOS** (*MicroSoft disk operating system* — sistema operacional de disco da Microsoft)

O zoológico de sistemas operacionais

Sistemas operacionais de computadores de grande porte

- No topo estão os sistemas operacionais para computadores de grande porte — aqueles que ocupam uma sala inteira, ainda encontrados em centros de dados de grandes corporações. Esses computadores distinguem-se dos computadores pessoais em termos de capacidade de E/S. Um computador de grande porte com mil discos e milhares de gigabytes de dados não é incomum

Sistemas operacionais de servidores

- Um nível abaixo estão os sistemas operacionais de servidores. Eles são executados em servidores, que são computadores pessoais muito grandes, em estações de trabalho ou até mesmo em computadores de grande porte. Eles servem múltiplos usuários de uma vez em uma rede e permitem-lhes compartilhar recursos de hardware e de software.

Sistemas operacionais de multiprocessadores

- Um modo cada vez mais comum de obter potência computacional é conectar múltiplas CPUs em um único sistema. Dependendo precisamente de como elas estiverem conectadas e o que é compartilhado, esses sistemas são denominados computadores paralelos, multicomputadores ou multiprocessadores.

Sistemas operacionais de computadores pessoais

- A categoria seguinte é o sistema operacional de computadores pessoais. Os computadores modernos dão suporte a multiprogramação, muitas vezes com dezenas de programas iniciados. Seu trabalho é oferecer uma boa interface para um único usuário.

Sistemas operacionais de computadores portáteis

- Seguindo em direção a sistemas cada vez menores, chegamos aos computadores portáteis. Um computador portátil ou **assistente pessoal digital** (*personal digital assistant* — PDA) é um pequeno computador que cabe no bolso de uma camisa e executa um número pequeno de funções, como agenda de endereços e bloco de anotações.

Sistemas operacionais embarcados

- Sistemas embarcados são executados em computadores que controlam dispositivos que geralmente não são considerados computadores e que não aceitam softwares instalados por usuários. Exemplos típicos são fornos de micro-ondas, aparelhos de TV, carros, aparelhos de DVD, telefones celulares e reprodutores de MP3.

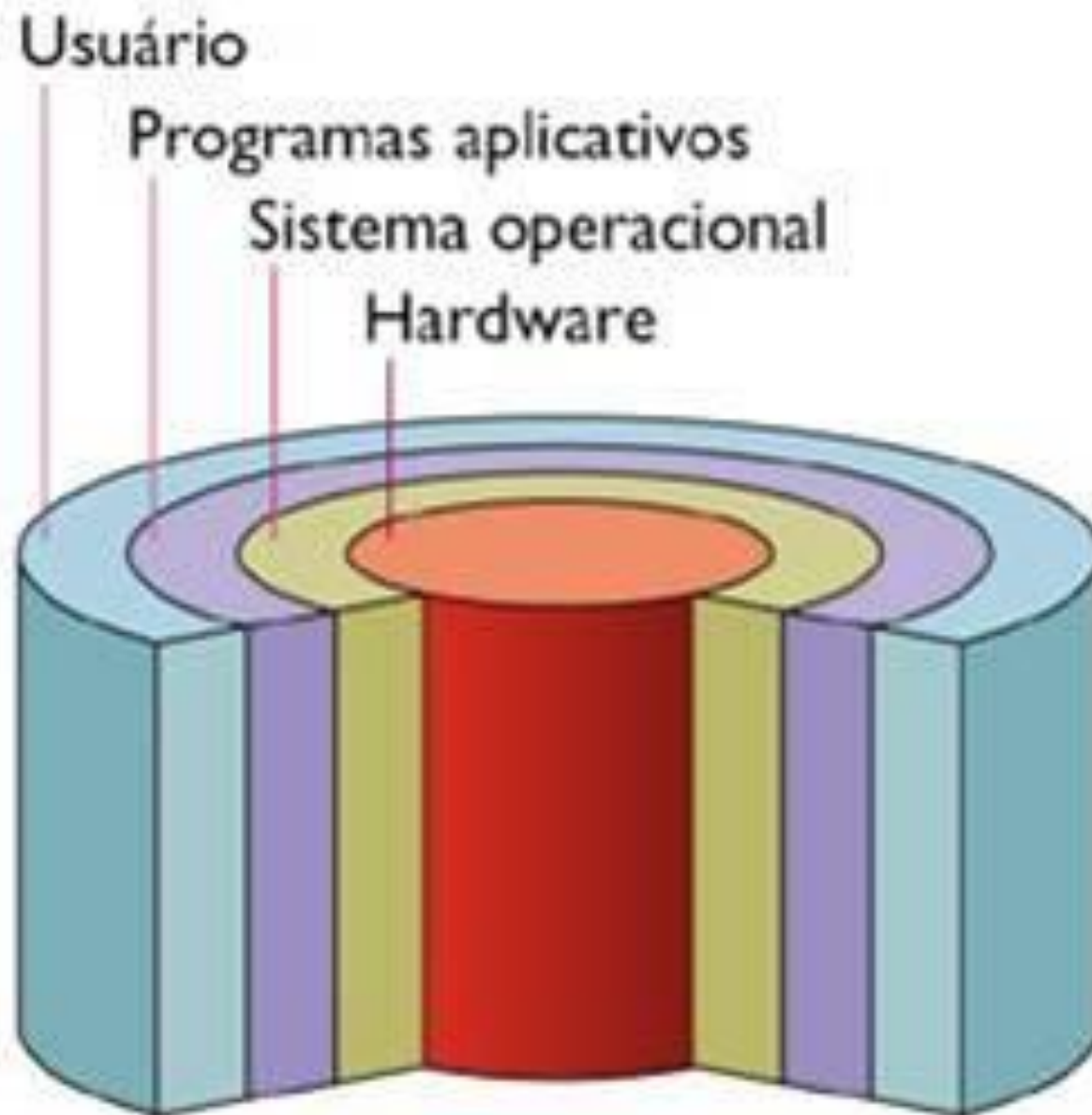
Sistemas operacionais de nós sensores (*sensor node*)

- Redes de nós sensores minúsculos estão sendo empregadas com inúmeras finalidades. Esses nós são computadores minúsculos que se comunicam entre si e com uma estação-base usando comunicação sem fio

- **Sistemas operacionais de tempo real**

Esses sistemas são caracterizados por terem o tempo como um parâmetro fundamental. Por exemplo, em sistemas de controle de processos industriais, computadores de tempo real devem coletar dados sobre o processo de produção e usá-los para controlar as máquinas na fábrica.

Virtualização





Uma definição genericamente usada para maquinas virtuais caracteriza virtualização como a criação de um ambiente de trabalho "falso" dentro de um ambiente "real". Entretanto apesar ser falso, o ambiente virtual imita bem as características do ambiente anfitrião, ou hospedeiro.



Geralmente a virtualização é assistida por hardware. Este é um recurso que existe desde a época dos mainframes, tendo despertado muita atenção como ferramenta de segurança, plataforma para maquinas virtuais em programação e para plataforma de sistemas operacionais.

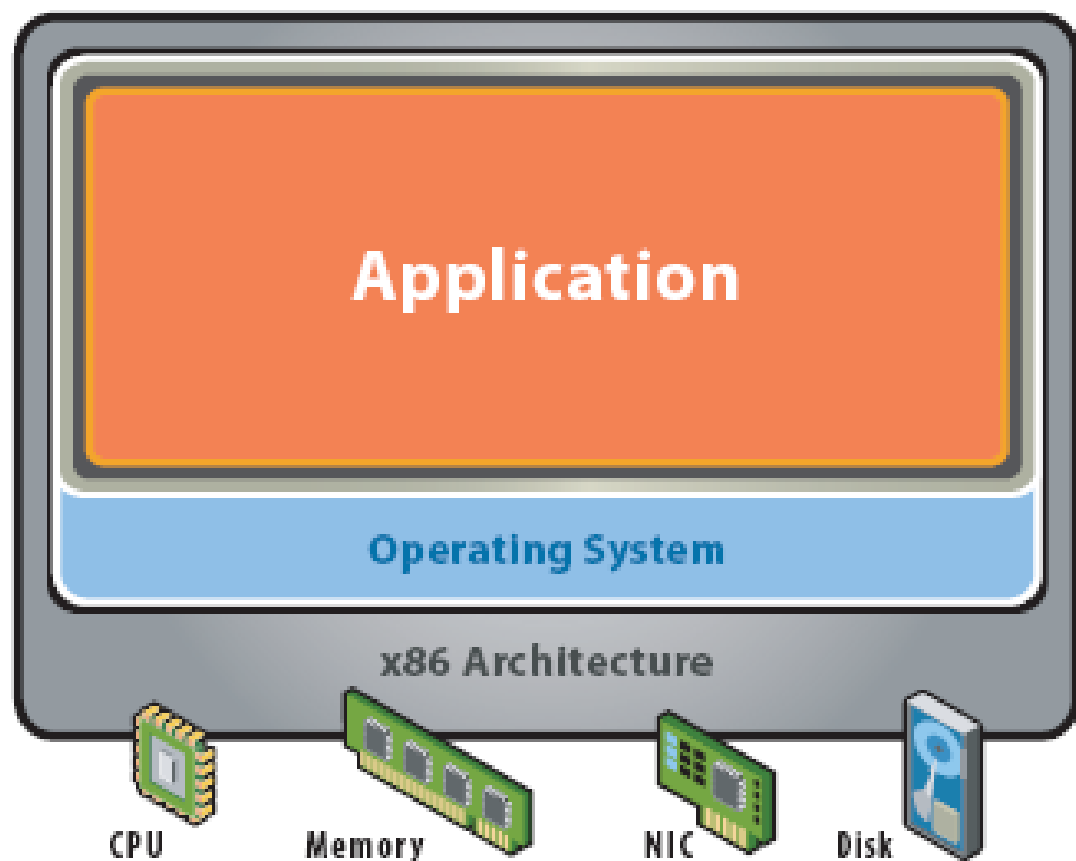


Atualmente, as maquinas virtuais estão ressurgindo e sendo utilizadas tanto em servidores como em desktops

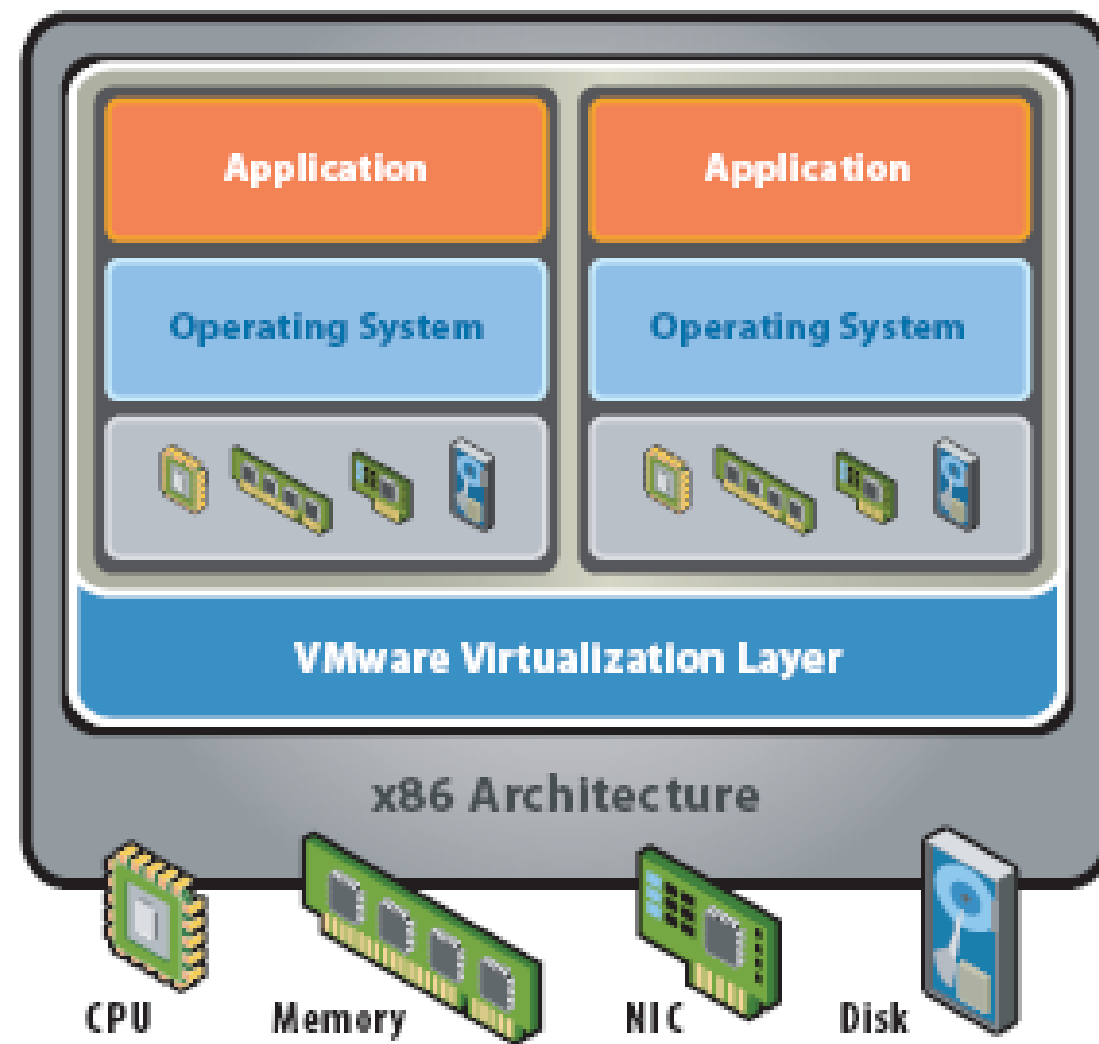
A Virtualização

Virtualização de Sistemas

- Tanto o Java Virtual Machine como o CLR do .NET, o VMWARE e o Virtual Box para Windows são considerados sistemas virtuais, entretanto ainda existe um sistema operacional chamado Virtual Machine (VM), que é executado em mainframes IBM. Nesse contexto um sistema virtual é uma máquina abstrata, o Java Virtual Machine, o .NET CLR são máquinas abstratas. Já o VMWARE virtualiza a CPU e emula o hardware, desde a BIOS até os dispositivos tradicionais como disquete. Sendo assim, uma sessão (virtualização) do VMWARE pode fazer tudo que um computador faria.



Arquitetura antes da Virtualização



Arquitetura depois da virtualização.



Obrigado

“O que move as grandes organizações é o desejo de serem significativas. Ter esse desejo não garante que você terá sucesso, mas também significa que, se você fracassar, pelo menos terá sido por tentar fazer algo que valeria a pena.”

Guy Kawasaki