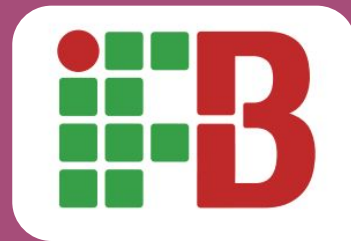


# Aula 1 - Introdução a Sistemas Operacionais

Sistemas Operacionais  
Ciência da Computação  
IFB - Campus Taguatinga

Professor João Victor de A. Oliveira



# Hoje

- Introdução
- O que é um sistema operacional e quais suas principais funções?
- História dos Sistemas Operacionais

# Introdução

- **Computador moderno**

- Consiste em um ou mais processadores, alguma memória principal, discos, impressoras, um teclado, um mouse, um monitor, interfaces de rede e vários outros dispositivos de entrada e saída.
  - Como um todo, trata-se de um **sistema complexo**
- Gerenciar todos esses componentes e usá-los de maneira otimizada é um trabalho extremamente desafiador.
  - Por essa razão, computadores são equipados com um dispositivo de software chamado de **sistema operacional**

# Introdução

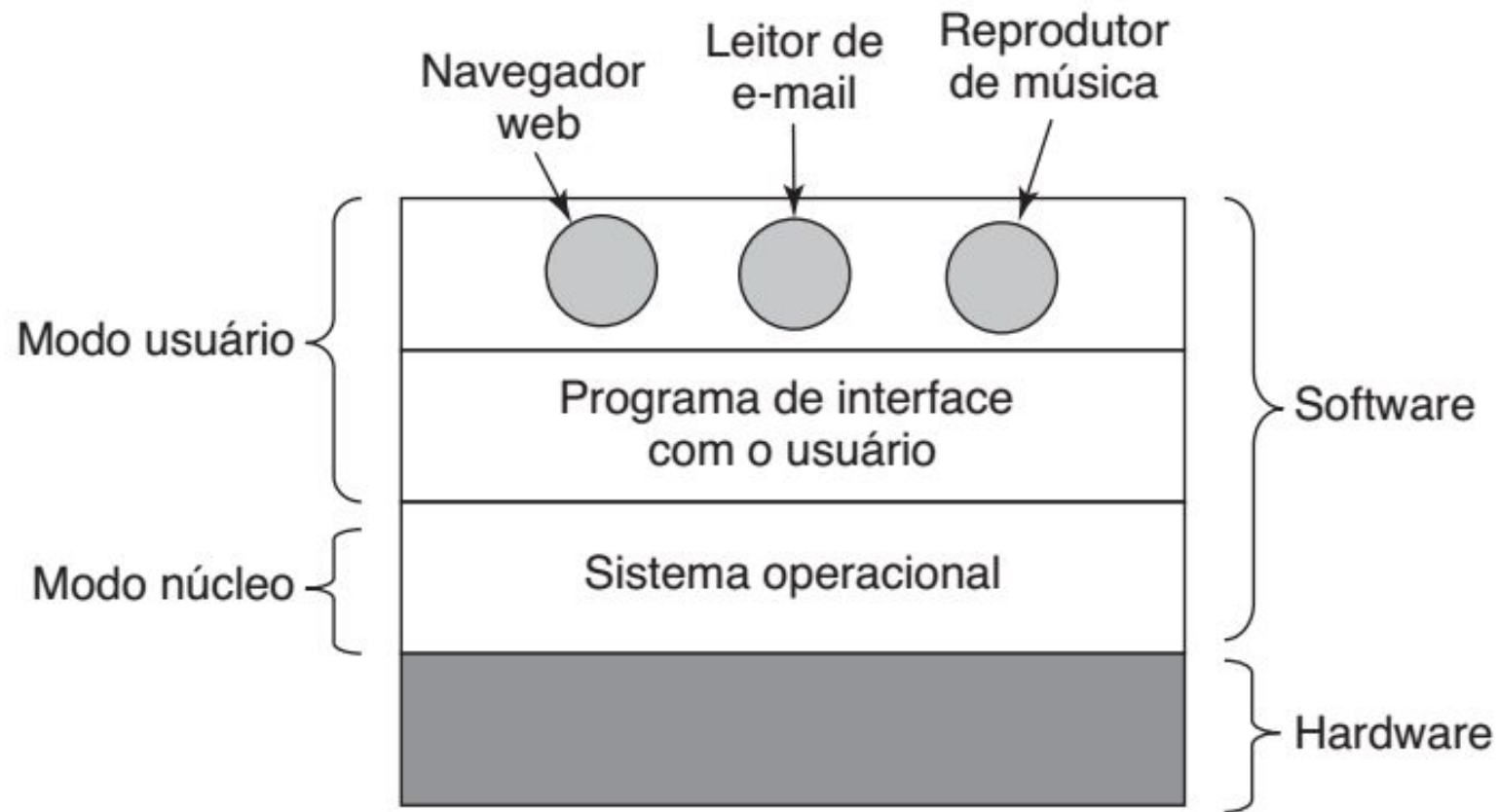
- **Sistema operacional**

- **Função:** fornecer aos programas do usuário um **modelo do computador melhor**, mais simples e mais limpo, assim como lidar com o **gerenciamento** de todos os recursos mencionados.
- **Exemplos:** Windows, Linux, FreeBSD, OS X, dentre outros
- **Programa usado para interagir com um sistema operacional:**
  - Shell (ou interpretador de comandos) ou
  - GUI (*Graphical User Interface*)

# Modos de Operação

- Maioria dos computadores possui dois modos de operação
  - **Modo Núcleo (supervisor)**
    - Acesso completo a todo o hardware
    - Pode executar qualquer instrução que a máquina for capaz de executar
  - **Modo Usuário**
    - Apenas um subconjunto das instruções da máquina está disponível.
      - Instruções que afetam o **controle da máquina ou realizam E/S** (Entrada/Saída) são **proibidas** para programas de modo usuário.
- **Sistema Operacional** opera em modo núcleo e proporciona a base para os outros softwares

## Visão simplificada de onde o sistema operacional se encaixa em um computador



## Distinções entre SO e softwares normais

- Se um usuário não gosta de um leitor de e-mail em particular, ele é livre para conseguir um leitor diferente ou escrever o seu próprio, se assim quiser
  - Contudo, ele não é livre para escrever seu próprio **tratador de interrupção de relógio**
    - Faz parte do sistema operacional e é protegido por hardware contra tentativas dos usuários de modificá-lo.
- SO's são enormes, complexos e têm vida longa
  - Códigos fontes do coração de SOs costumam ter cerca de 5 milhões de linhas
  - Por serem complexos, tendem a ter uma longa vida, sendo apenas atualizados
    - Ex.: Windows NT/2000/XP/Vista/Windows 7

# O que é um sistema operacional?

- **Difícil definir...**

- Software que opera em modo núcleo — nem sempre é verdade.
- Parte do problema é que os sistemas operacionais realizam **duas funções essencialmente não relacionadas**:
  - Fornecer a programadores um conjunto de recursos abstratos limpos em vez de recursos confusos de hardware (SO como uma máquina estendida)
  - Gerenciar esses recursos de hardware.
- Dependendo de quem fala, você poderá ouvir mais a respeito de uma função do que de outra
  - Vamos examinar essas duas definições



# Sistema operacional como uma máquina estendida

- **Arquitetura de Computadores**

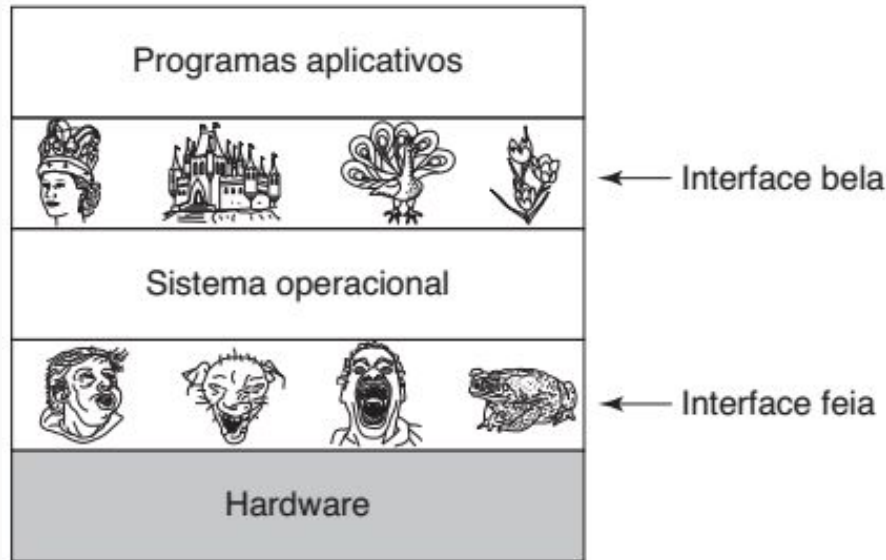
- Conjunto de instruções, organização de memória, E/S e estrutura de barramento
- **Primitiva e complicada de programar, especialmente para entrada/saída.**
  - **Exemplo:** Um livro (ANDERSON, 2007) descrevendo a versão inicial da interface do disco tinha mais de 450 páginas!
    - Nenhum programador iria **querer/poder** entender 450 páginas para lidar com esse disco específico
    - Em vez disso, um software chamado **driver de disco**, lida com o hardware e fornece uma **interface** para ler e escrever blocos de dados, sem entrar em muitos detalhes de programação

# Sistema operacional como uma máquina estendida

- A interface dos drivers de disco ainda é muito complexa para a maioria das aplicações
  - Por essa razão, todos os sistemas operacionais fornecem mais um **nível de abstração** para se utilizarem discos: **arquivos**
  - Usando essa abstração, os programas podem criar, escrever e ler arquivos, sem ter de lidar com os detalhes complexos de como o hardware realmente funciona
- **Função dos sistemas operacionais:** criar boas abstrações e então implementar e gerenciar os objetos abstratos criados desse modo.

# Sistema operacional como uma máquina estendida

- Uma das principais tarefas dos sistemas operacionais é **esconder o hardware** e em vez disso **apresentar programas** (e seus programadores) com abstrações de qualidade, limpas, elegantes e consistentes com as quais trabalhar.
- “Sistemas operacionais transformam o feio em belo”. Tanenbaum e Bos



# SO como um gerenciador de recursos

- **Função:** fornecer uma alocação ordenada e controlada dos processadores, memórias e dispositivos de E/S entre os vários programas competindo por eles
  - Exemplo: **Três programas sejam executados ao mesmo tempo**
    - O que aconteceria se os três programas tentassem imprimir sua saída ao mesmo tempo?
    - **Papel do SO:** cuidar para que uma impressão não interfira em outra

# Sistema Operacional como um gerenciador de recursos

- **Exemplo:** Computador com mais de um usuários
  - Necessidade de gerenciar e proteger a memória, dispositivos de E/S e outros recursos, tendo em vista que os usuários poderiam interferir um com o outro.
  - Usuários muitas vezes precisam compartilhar não apenas o hardware, mas a informação (arquivos, bancos de dados etc.)
- **Principal função de um SO:**
  - Manter um controle sobre quais programas estão usando qual recurso, conceder recursos requisitados, contabilizar o seu uso, assim como mediar requisições conflitantes de diferentes programas e usuários.
  - Exemplo: tabela de processos, htop

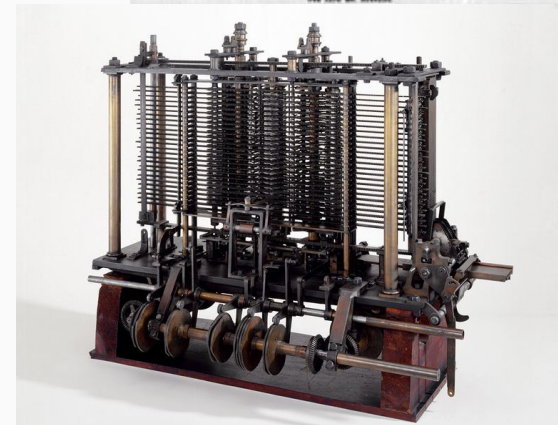
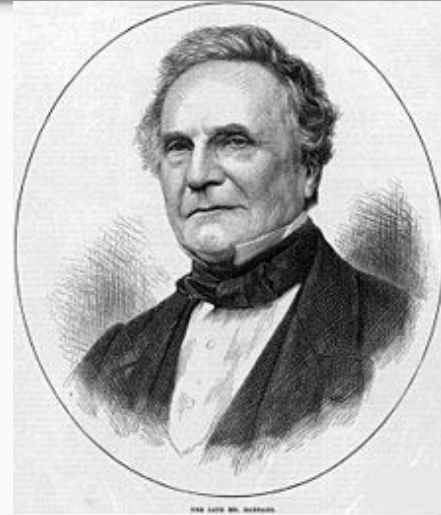
# Sistema Operacional como um gerenciador de recursos

- **Multiplexação (compartilhamento) de recursos é feita de dois modos:**
  - **Tempo:** diferentes programas ou usuários se **revezam**. Primeiro, um deles usa o recurso, então outro e assim por diante.
    - Ex.: Uso da CPU e fila de impressão
  - **Espaço:** Em vez de os clientes se revezarem, **cada um tem direito a uma parte do recurso**
    - Ex.: Memória principal dividida entre vários programas sendo executados
    - Mais eficiente manter vários programas na memória ao mesmo tempo do que dar a um deles toda memória
  - Isso gera questões de **justiça, proteção e outros critérios** que o SO deve resolver

# História dos Sistemas operacionais

# História dos sistemas operacionais

- História dos SOs está intimamente ligada às arquiteturas nas quais eles foram executados
  - Novos desenvolvimentos não esperaram que os anteriores tivessem terminado adequadamente antes de começarem.
- Primeiro Computador digital foi projetado pelo matemático inglês **Charles Babbage (1792-1871)**
  - Nunca conseguiu colocá-la para funcionar, devido a limitações tecnológicas da época
  - Sua máquina analítica, obviamente ainda não possuía SO



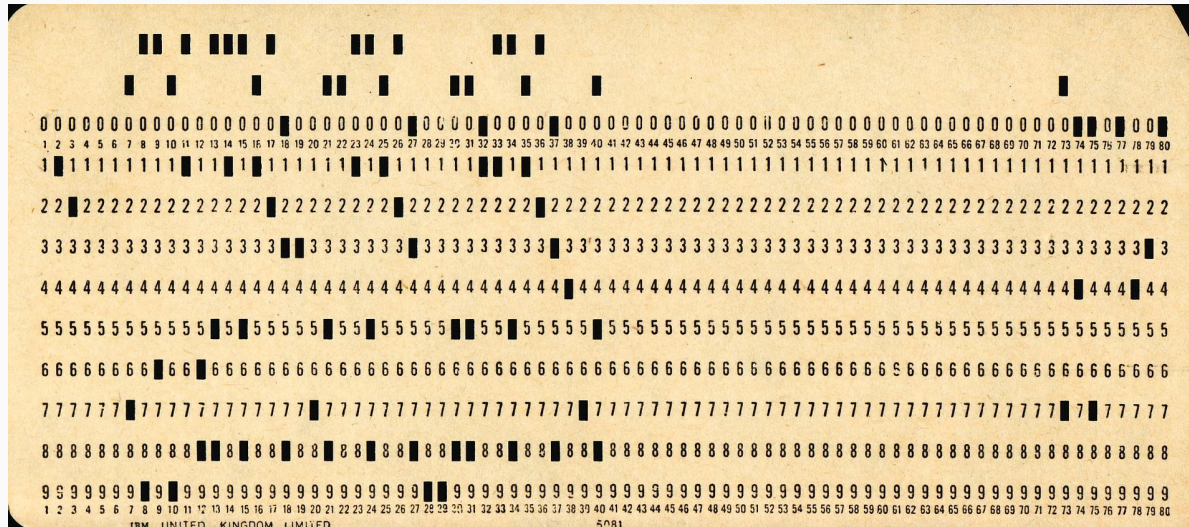


# A primeira geração (1945-1955): válvulas

- Computadores usavam, **válvulas, painéis** e alguns outros eram programáveis
  - Todos eram muito primitivos e levavam segundos para realizar cálculos simples
- A programação era feita em **código de máquina absoluto, ou, pior ainda, ligando circuitos elétricos** através da conexão de milhares de cabos a painéis de ligações para controlar as funções básicas da máquina.
  - **Linguagens de programação eram desconhecidas** (mesmo a linguagem de montagem era desconhecida). Ninguém tinha ouvido falar ainda de sistemas operacionais.

# A primeira geração (1945-1955): válvulas

- Na década de 50, houve uma melhoria na rotina de operação com a introdução dos **cartões perfurados**
  - Era possível agora escrever programas em cartões e lê-los em vez de se usarem painéis de programação;



# A segunda geração (1955-1965): transistores e sistemas em lote (batch)

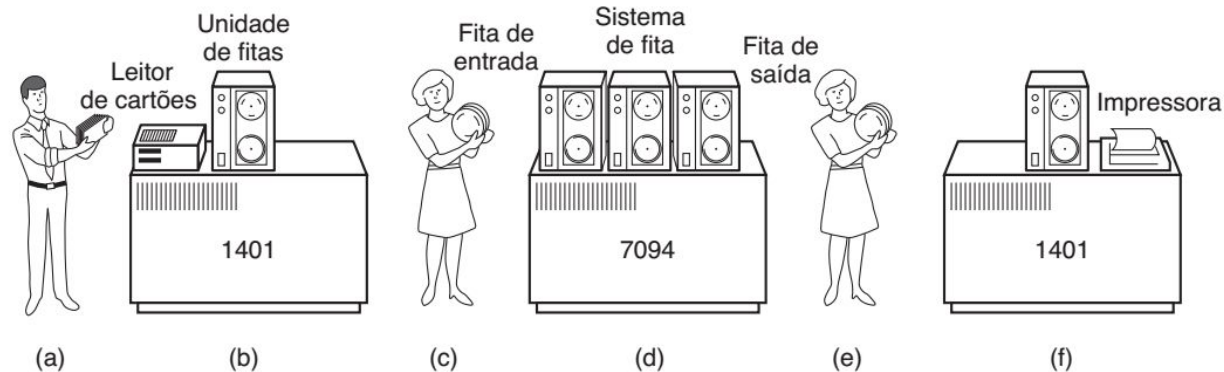
- Surgimento dos **transistores** em meados dos anos 1950
  - Aumento na confiança de que o computador iria funcionar por mais tempo (**viabilidade comercial**)
- Clara separação entre projetistas, construtores, operadores, programadores e pessoal de manutenção (**sistema operacional formado por pessoas?**).
- Essas máquinas eram chamadas de **Computadores de grande porte (Mainframes)**
  - Ficavam isoladas em salas grandes e climatizadas
  - Apenas grandes corporações ou importantes agências do governo ou universidades conseguiam pagar o valor alto para tê-las



# A segunda geração (1955-1965): transistores e sistemas em lote (batch)

- **Sistema em lote (batch)**

- Reunir um **lote de tarefas na sala de entradas** e então passá-lo para uma **fita magnética** usando um computador pequeno e (relativamente) barato, como um IBM 1401, que era muito bom na leitura de cartões, cópia de fitas e impressão de saídas, mas ruim em cálculos numéricos.



- (a) Programadores levavam cartões para o 1401.
- (b) O 1401 lia o lote de tarefas em uma 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

- 2 Linhas de Produto incompatíveis na década de 60
  - **Computadores científicos** de grande escala, orientados por palavras, como o 7094, usados para cálculos numéricos complexos na ciência e engenharia.
  - **Computadores comerciais**, orientados por caracteres, como o 1401, que eram amplamente usados para ordenação e impressão de fitas por bancos e companhias de seguro.
    - Desenvolver e manter duas linhas de produtos completamente diferentes era uma proposição cara para os fabricantes
    - Muitos clientes novos de computadores inicialmente precisavam de uma máquina pequena, no entanto mais tarde a sobreutilizavam e queriam uma máquina maior que executassem todos os seus programas antigos, porém mais rápidos
    - IBM tentou solucionar ambos os problemas com uma única tacada, introduzindo o **System/360**.

# A terceira geração (1965-1980): CIs e multiprogramação

- **System/360**

- Série de máquinas com softwares compatíveis, desde modelos do porte do 1401 a modelos muito maiores e potentes que o poderoso 7094
- As máquinas diferiam apenas em preço e desempenho
- Tendo em vista que **todos tinham a mesma arquitetura e conjunto de instruções**, programas escritos para uma máquina podiam operar em todas as outras — pelo menos na teoria.
  - Uma única máquina satisfazia os interesses de de ambas linhas de produto
    - Científica e comercial



## A terceira geração (1965-1980): CIs e multiprogramação

- **IBM 360 foi a primeira linha importante de computadores a usar CIs (Circuitos Integrados) de pequena escala**
  - **Vantagem significativa na relação preço/desempenho** sobre as máquinas de segunda geração
  - **O forte da ideia da “família única”** foi ao mesmo tempo seu maior ponto fraco
    - Ele tinha de funcionar em sistemas pequenos (como os que rodavam no 1401) quanto em sistemas muito grandes (como os que rodavam no 7094)
  - **Resultado: SO extremamente grande e complexo**
  - milhões de linhas de linguagem de montagem escritas por milhares de programadores
    - Continha dezenas de milhares de erros
    - Cada nova versão corrigia alguns erros e introduzia novos, de maneira que o número de erros provavelmente seguiu constante através do tempo.

# A terceira geração (1965-1980): CIs e multiprogramação

- **No 7094 (segunda geração)**

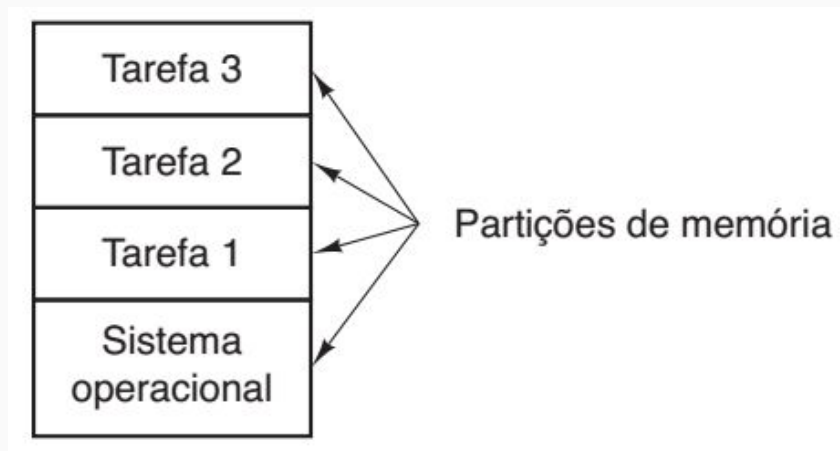
- Quando a tarefa atual fazia uma pausa para esperar por uma fita ou outra operação de E/S terminar, a CPU simplesmente ficava ociosa até o término da E/S
  - Para cálculos científicos com uso intenso da CPU, a E/S é esporádica, de maneira que o tempo ocioso não é significativo
  - Para o processamento de dados comercial, o tempo de espera de E/S pode muitas vezes representar de 80 a 90% do tempo total de maneira que algo tem de ser feito para evitar que a CPU (cara) fique ociosa tanto tempo.

- O que podemos fazer para evitar que a CPU fique ociosa?



## A terceira geração (1965-1980): CIs e multiprogramação

- Solução para o problema de ociosidade da CPU - **Multiprogramação**
  - Dividir a memória em várias partes, com uma tarefa diferente em cada partição
  - Enquanto uma tarefa ficava esperando pelo término da E/S, outra podia usar a CPU
    - **CPU podia se manter ocupada quase 100% do tempo**
  - Exige um hardware especial para proteger cada tarefa contra interferências e transgressões por parte das outras



## A terceira geração (1965-1980): CIs e multiprogramação

- Desejo por um tempo de resposta mais rápido levou o desenvolvimento do **Timesharing (Compartilhamento de tempo)**
  - Uma variante de multiprogramação, na qual **cada usuário tem um terminal on-line**.
  - Se 20 usuários estão conectados e 17 deles estão pensando, falando ou tomando café, a CPU pode ser alocada por sua vez para as três tarefas que demandam serviço
  - O primeiro sistema de compartilhamento de tempo para fins diversos, foi chamado de **CTSS (Compatible Time Sharing System — Sistema compatível de tempo compartilhado)** do M.I.T

## A terceira geração (1965-1980): CIs e multiprogramação

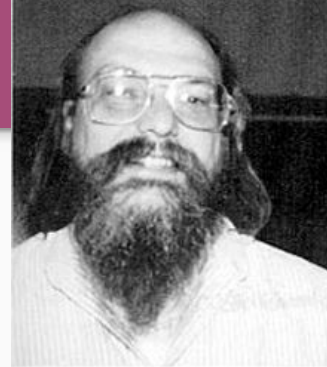
- Após o CTSS, o M.I.T, a Bell Labs, e a General Eletric decidiram embarcar no desenvolvimento de um **Computador utilitário**
  - Capaz de fornecer suporte a centenas de usuários simultâneos com compartilhamento de tempo
  - **Usava o modelo de eletricidade**
    - Quando você precisa de energia elétrica, basta conectar um pino na tomada e terá tanta energia quanto necessário
  - **MULTICS** (MULTiplexed Information and Computing Service – Serviço de Computação e Informação Multiplexada)
    - Sonhavam com um computador fornecendo energia computacional para todas as pessoas na área de Boston
    - **Problemas:** Linguagem de programação PL/I (Compilador precário)
    - **Prós:** Introduziu diversas ideias importantíssimas para a evolução da computação

# A terceira geração (1965-1980): CIs e multiprogramação

- MULTICS

- Instalado por mais ou menos 80 empresas e universidades importantes mundo afora
- Embora seus números fossem pequenos, os usuários do MULTICS eram muito leais
  - A General Motors, a Ford e a Agência de Segurança Nacional Norte-Americana, por exemplo, abandonaram os seus sistemas MULTICS apenas no fim da década de 1990
- **No fim do século XX**, o conceito de um computador utilitário havia perdido força, mas ele pode voltar para valer na forma da **computação na nuvem (cloud computing)**
  - Computadores pequenos estão conectados a servidores em vastos e distantes centros de processamento de dados
  - Toda a computação é feita pelo servidor com o computador local apenas executando a interface com o usuário.
  - **Motivação:** a maioria das pessoas não quer administrar um sistema computacional
    - preferem que esse trabalho seja realizado por uma equipe de profissionais

## A terceira geração (1965-1980): CIs e multiprogramação



- **Crescimento dos minicomputadores**

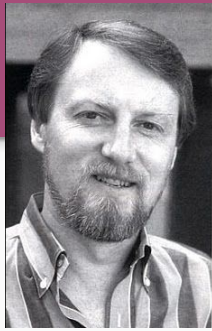
- **PDP-1 (1961)** tinha apenas 4K de palavras de 18 bits, mas a US\$ 120.000 por máquina (menos de 5% do preço de um 7094), sendo sucesso de vendas
- Um dos cientistas de computação no Bell Labs que havia trabalhado no projeto **MULTICS**, **Ken Thompson**, descobriu subsequentemente um minicomputador pequeno PDP-7 que ninguém estava usando e decidiu escrever uma versão despojada e para um usuário do **MULTICS**
- Mais tarde esse trabalho desenvolveu-se no sistema operacional **UNIX**
  - Versões importantes do UNIX na época: **System V da AT&T** e **BSD** da universidade de Califórnia
  - Para tornar possível escrever programas que pudessem ser executados em qualquer sistema UNIX, o IEEE desenvolveu um padrão para o UNIX, chamado **POSIX (Portable Operating System Interface** — interface portátil para sistemas operacionais), ao qual a maioria das versões do UNIX dá suporte hoje em dia

## A terceira geração (1965-1980): CIs e multiprogramação

- Em 1987, Andrew S. Tanenbaum, lançou um clone do UNIX, chamado **MINIX**, para fins educacionais
  - Muito similar ao UNIX, incluindo o suporte ao POSIX, hoje está na sua terceira versão
    - Capacidade de detectar e substituir módulos defeituosos ou mesmo danificados em funcionamento, sem reinicializá-lo e sem perturbar os programas em execução
  - Desejo de produzir uma **versão gratuita do MINIX** levou um estudante finlandês, **Linus Torvalds**, a escrever o **Linux**
    - Diretamente inspirado pelo MINIX, desenvolvido sobre ele e originalmente fornecendo suporte a vários aspectos do MINIX.
    - Desde então, foi ampliado de muitas maneiras por muitas pessoas, mas ainda mantém algumas estruturas subjacentes comuns ao MINIX e ao UNIX.



## A quarta geração (1980-presente): computadores pessoais



- Desenvolvimento dos **circuitos integrados em larga escala - LSI**
  - Chips contendo milhares de transistores em um centímetro quadrado de silicone
  - **Surgiu então a era dos computadores pessoais, também chamados de microcomputadores**
- **Em 1974 a intel lança o 8080**
  - Primeira CPU de 8 bits de uso geral
  - **Gary Kildall**, consultor da intel fica encarregado de construir um SO para o 8080
    - Construiu, primeiramente, um controlador para um disco flexível de 8 polegadas e o inseriram no 8080 (**Gerando o primeiro microcomputador com disco**)
    - Após isso escreveu um SO baseado em disco chamado **CP/M (Control Program for Microcomputers – programa de controle para microcomputadores)**
    - **Intel não achava que microcomputadores baseados em disco tinha muito futuro**
      - Kildall forma então a **Digital Research**, para desenvolver o **CP/M** e vendê-lo

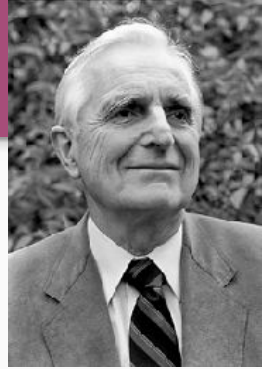
## A quarta geração (1980-presente): computadores pessoais

- Início da década de 80, a IBM projeta o **IBM PC**
  - A IBM então contrata **Bill Gates** que indica a **Digital Research** para elaborar um sistema operacional para sua máquina e para trabalhar com o interpretador **BASIC**
  - **Kildall recusa a proposta da IBM**
  - Gates procura uma fabricante de computadores local, **Seattle Computer Products**, e compra o sistema operacional **DOS (Disk Operating System)** por apenas 75.000 dólares
  - Gates oferece à IBM o pacote **DOS/BASIC**, e após algumas modificações feitas por um dos escritores do DOS, **Tim Patterson**, surge o **MS-DOS (MicroSoft Disk Operating System)**
    - **Fator chave de sucesso de Gates: Vender o SO em conjunto com o hardware**
      - Kildall vendia apenas o CP/M diretamente aos usuários finais
  - Mais tarde o MS-DOS foi amplamente usado nos **80386 e no 80486**

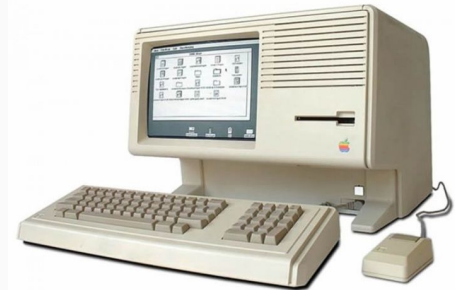




## A quarta geração (1980-presente): computadores pessoais



- O CP/M, MS-DOS e outros sistemas operacionais para os primeiros microcomputadores eram todos baseados na **digitação de comandos no teclado pelos usuários**
  - Anos mais tarde, **Engelbart** inventou a **Graphical User Interface (GUI – Interface Gráfica do Usuário)**
    - Possuía janelas, ícones, menus e mouse. Essas ideias foram adotadas por pesquisadores na Xerox PARC e incorporadas nas máquinas que eles produziram.
  - **Steve Jobs**, que coinventou o computador Apple em sua garagem, visitou a PARC, viu uma GUI e no mesmo instante percebeu o seu valor potencial
    - Produção do primeiro Apple com o GUI: o chamado **LISA**
      - Fracasso Comercial
    - Na segunda tentativa, criou o **Apple Macintosh**, sucesso comercial



# A quarta geração (1980-presente): computadores pessoais

- **Apple Macintosh**

- **Amigável ao usuário**

- Dirigido a usuários que não apenas não sabiam nada sobre computadores como não tinham intenção alguma de aprender sobre eles

- Em 1999, a Apple adotou um núcleo derivado do **micronúcleo Mach** da Universidade Carnegie Mellon que foi originalmente desenvolvido para substituir o núcleo do **BSD UNIX**.

- Desse modo, o **MAC OS X** é um sistema operacional baseado no **UNIX**, embora com uma interface bastante distinta

- Fortemente influenciou a microsoft a criar um SO com GUI, o chamado **Microsoft Windows**

- Um ambiente gráfico sobre o MS-DOS



# A quarta geração (1980-presente): computadores pessoais

- **Windows 95**

- Versão independente Incorporando muitos aspectos de SO
- Usa o MS-DOS subjacente apenas para sua inicialização e para executar velhos programas do MS-DOS

- **Windows 98**

- Evolução do windows 95, mas ainda possuindo uma grande quantidade de linguagem de montagem de 16 bits da intel

- **Windows NT**

- Compatível com windows 95 até um certo nível, foi totalmente reescrito em um sistema de 32 bits
- Projetado para substituir completamente o MS-DOS, só conseguiu atingir seu objetivo no Windows NT 4.0
- Na sua versão 5.0 foi renomeado para Windows 2000, que deveria ser o sucessor do Windows 98



# A quarta geração (1980-presente): computadores pessoais

- **Windows ME (Millenium Edition)**

- Após o fracasso do Windows 2000, o Windows produziu uma nova versão do Windows 98, o Windows ME

- **Windows XP**

- Em 2001, foi lançada uma versão ligeiramente atualizada do Windows 2000, o Windows XP

- Após isso a microsoft dividiu a família windows em **duas famílias**: a de **clientes** e a de **Servidores**

- **Clientes**: Windows XP e seus sucessores
- **Servidores**: Windows 2003 e Windows 2008

- Essas versões aumentaram suas variações na forma de **Pacotes de Serviço (Service Packs)**



# A quarta geração (1980-presente): computadores pessoais

- **Windows Vista =\**

- Lançado em 2007 como o sucessor do Windows XP
- Nova interface gráfica, segurança mais firme e muitos programas para os usuários novos ou atualizados
- **Nunca conseguiu substituir completamente o Windows XP**
- Em vez disso, ele recebeu muitas críticas e uma cobertura negativa da imprensa, sobretudo por causa das exigências elevadas do sistema, **termos de licenciamento restritivos e suporte para o Digital Rights Management**, técnicas que tornaram mais difícil para os usuários copiarem material protegido



# A quarta geração (1980-presente): computadores pessoais

- **Windows 7**

- O Windows 7 não introduziu muitos aspectos novos, mas era relativamente pequeno e bastante estável
- Em menos de três semanas, o Windows 7 havia conquistado um mercado maior do que o Vista em sete meses

- **Windows 8**

- Em 2012, a Microsoft lançou o sucessor, Windows 8, um sistema operacional com visual e sensação completamente diferentes, voltado para telas de toque
- Embora com bastante novidades, o windows 7 permaneceu forte no mercado, com uma comunidade resistente em atualizá-lo para versões mais novas

- **Windows 10**

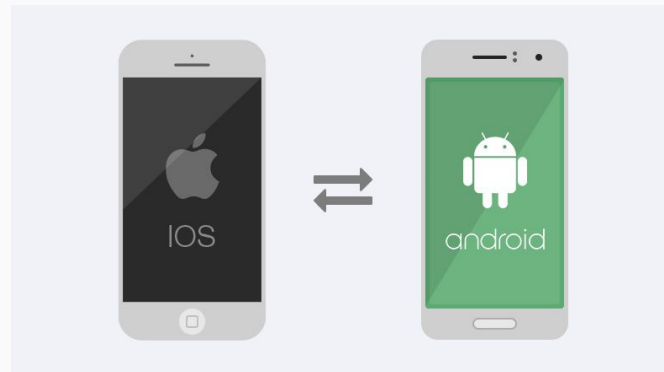
- Windows “atual” :)



## A quarta geração (1980-presente): computadores pessoais

- **Competidor da Microsoft: UNIX (e seus derivativos)**

- Forte em servidores de rede e de empresas
- Também presente em computadores, notebooks, tablets e smartphones
- Bastante populares nas arquiteturas x86 (32 bits) e x64 (64 bits)
- **OS X da apple usa uma versão modificada do FreeBSD, que é derivado do antigo BSD**
- UNIX também é padrão em estações de trabalho equipadas com chips RISC de alto desempenho
- Também extensamente usados em dispositivos móveis, como no iOS 7 e no Android



# A quarta geração (1980-presente): computadores pessoais

- **GUI UNIX**

- Sistemas UNIX dão suporte a um sistema de janelas chamado de **X Window System (também conhecido como X11)** produzido no M.I.T
  - Cuida do gerenciamento básico de janelas, permitindo que os usuários criem, removam, movam e redimensionem as janelas usando o mouse
  - Muitas vezes uma GUI completa, como **Gnome ou KDE**, está disponível para ser executada em cima do X11, dando ao UNIX uma aparência e sensação semelhantes ao Macintosh ou Microsoft Windows, para aqueles usuários do UNIX que buscam isso



# A quarta geração (1980-presente): computadores pessoais

- **SO de Rede**

- Usuários estão conscientes da existência de múltiplos computadores e podem conectar-se a máquinas remotas e copiar arquivos de uma máquina para outra
- Cada máquina executa seu próprio sistema operacional e tem seu próprio usuário local (ou usuários)

- **SO Distribuído**

- Aparece para os seus usuários como um sistema monoprocessador tradicional, embora seja na realidade composto de múltiplos processadores
- Os usuários não precisam saber onde os programas estão sendo executados ou onde estão localizados os seus arquivos; isso tudo deve ser cuidado automática e eficientemente pelo sistema operacional

## A quinta geração (1990-presente): computadores móveis

- Telefone móvel está presente em quase 90% da população global.
- Primeiro Smartphone: **Nokia N9000**
  - Combinava um telefone e um **PDA (Personal Digital Assistant)**
- Em 1997 a Ericksson cunhou o termo smartphone para o **'Penelope' GS88**
- Atualmente os dois SOs dominantes no mercado são:
  - Android
  - iOS



# Referências

- TANENBAUM, Andrew S e Bos, Herbert. **Sistemas Operacionais Modernos. 4.ed.** Pearson/Prentice-Hall. 2016.

