

Histórico sobre SO e conceitos básicos

Professor Wagner Gadêa Lorenz
wagnerglorenz@gmail.com

Disciplina: Sistemas Operacionais

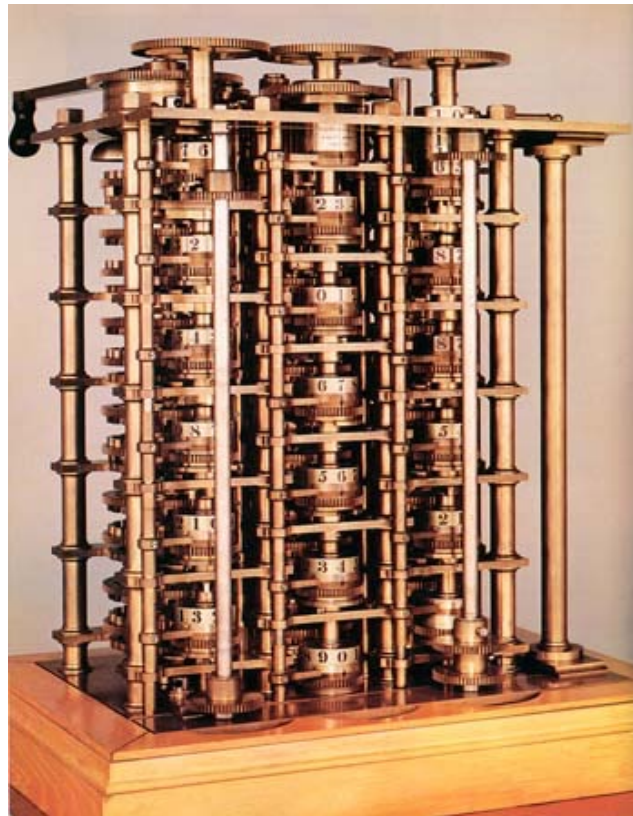
Cachoeira do Sul, 26 de Fevereiro de 2015

História dos sistemas operacionais

- O **primeiro computador** digital foi projetado pelo matemático inglês **Charles Babbage** (1792-1871).
- Embora Babbage tenha empregado a maior parte de sua vida e fortuna para construir sua “máquina analítica”, ele nunca conseguiu vê-la funcionando de modo apropriado, 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.
- A **máquina analítica não** possuía um **sistema operacional**.

História dos sistemas operacionais

- Em 1822, **Babbage** apresentou a Sociedade Real de Astronomia o primeiro modelo de uma máquina de “diferença”, capaz de efetuar cálculos necessários para elaborar uma tabela de logaritmos.



História dos sistemas operacionais



História dos sistemas operacionais

- Outro aspecto histórico interessante foi que 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.

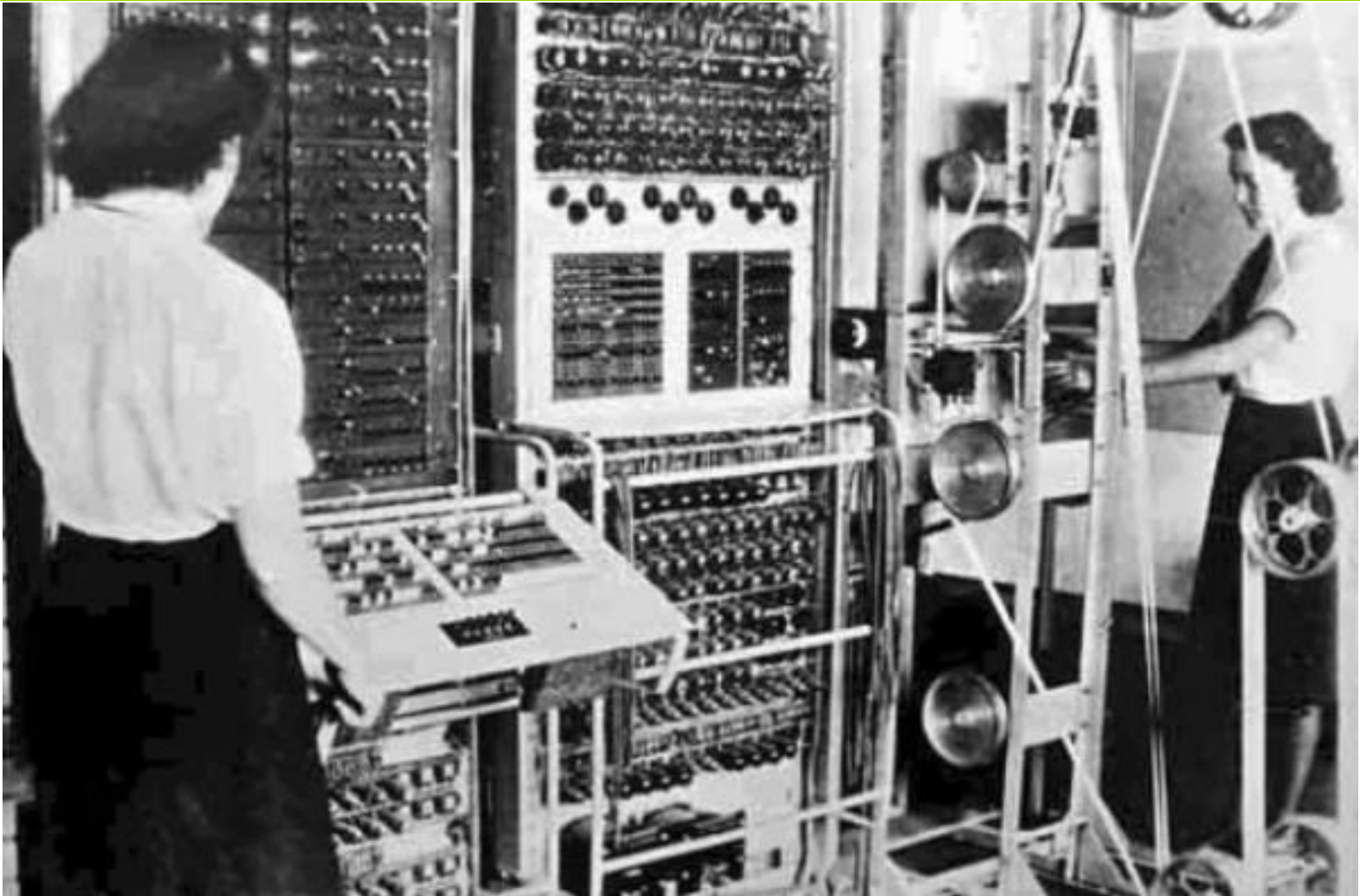
A primeira geração (1945 - 1955) - válvulas

- Depois dos vários esforços de Babbage, seguiram-se poucos progressos na construção de computadores digitais até a **Segunda Guerra Mundial**, que estimulou uma explosão de atividades.
- O professor John Atanasoff e seu aluno de graduação Clifford Berry construíram o que consideramos o primeiro computador digital em funcionamento, na Universidade do Estado de Iowa. Ele utilizava 300 válvulas.

A primeira geração (1945 - 1955) - válvulas

- Em 1944, o **Colossus** foi desenvolvido por um grupo em Bletchley Park, Inglaterra.
- O **Mark** foi construído por Howard Aiken em Harvard.
- **ENIAC** foi construído por William Mauchley e seu aluno de graduação J. Presper Eckert na Universidade da Pensilvânia.

A primeira geração (1945 - 1955) - válvulas



Colossus - Fonte da imagem: Reprodução/[Colossus Computer](#)

A primeira geração (1945 - 1955) - válvulas



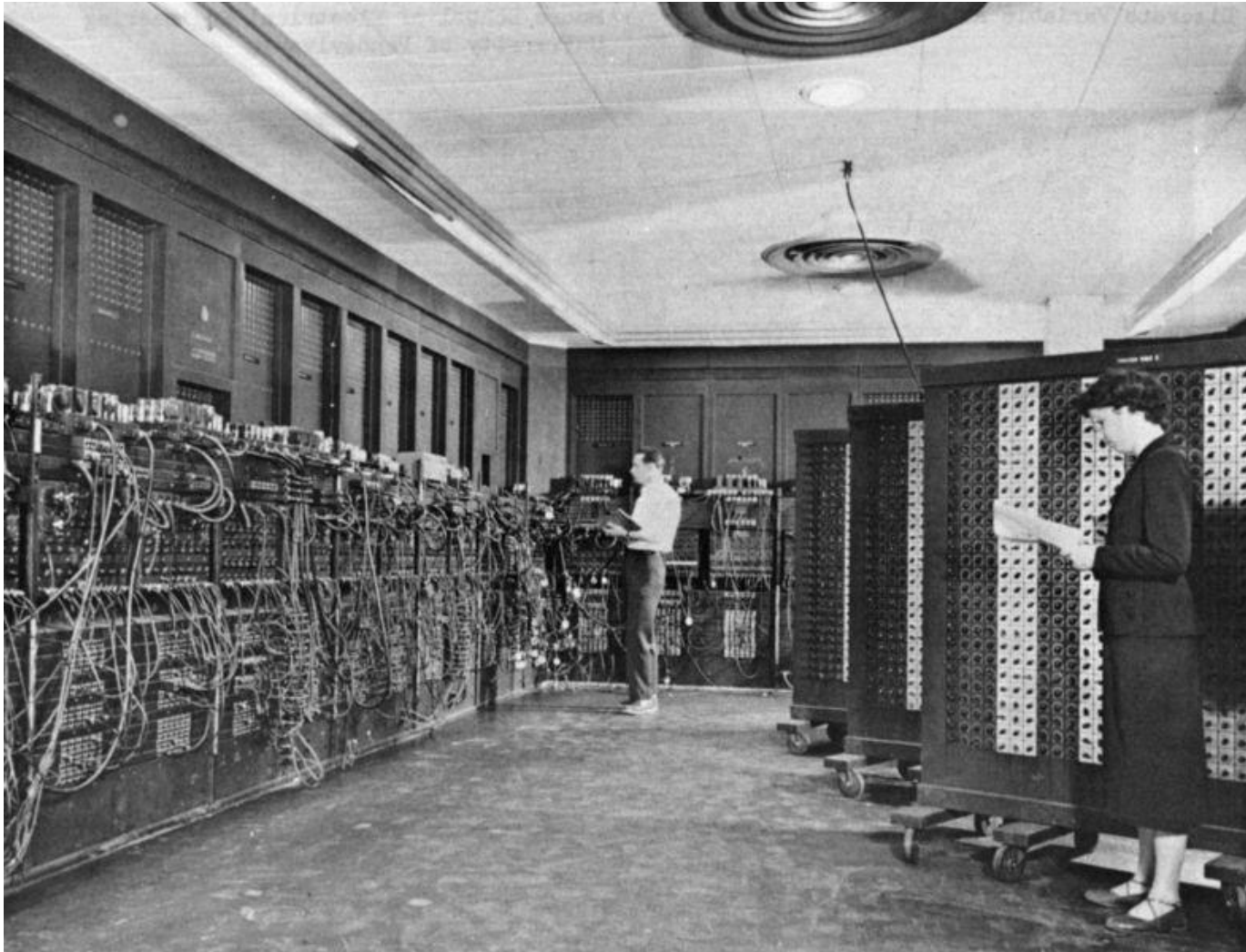
Algumas das válvulas dessa réplica são as originais, da década de 1940. Fonte [IEEE Computer Society](https://www.ieee.org/cs).

A primeira geração (1945 - 1955) - válvulas



The Harvard Mark I

A primeira geração (1945 - 1955) - válvulas



ENIAC. Fonte <http://www.hardware.com.br/guias/historia-informatica/eniac.html>

A primeira geração (1945 - 1955) - válvulas

- Alguns eram **binários**, alguns usavam **válvulas**, alguns eram **programáveis**, mas todos eram muito primitivos e levavam segundos para executar até **cálculos** mais **simples**.

A primeira geração (1945 - 1955) - válvulas

- Naquela época, um mesmo grupo de pessoas projetava, construía, programava, operava e realizava a manutenção de cada máquina.
- Toda a programação era feita em **código de máquina absoluto** e muitas vezes conectando plugs em painéis para controlar as funções básicas da máquina.
- **Não havia linguagens de programação** (nem mesmo a linguagem assembly existia).
- Os **sistemas operacionais** também **não** haviam sido **inventados**.

A primeira geração (1945 - 1955) - válvulas

- A **rotina de funcionamento** era a seguinte:
 - O programador reservava antecipadamente tempo na máquina em uma planilha, ia para a sala de máquina, inseria seu painel de programação no computador e passava algumas horas torcendo para que nenhuma das 20 mil válvulas queimasse durante a execução.
- Praticamente todos os problemas eram cálculos numéricos diretos, como determinar tabelas de senos, cossenos e logaritmos.

A primeira geração (1945 - 1955) - válvulas

- Apenas no início da **década de 1950**, essa rotina havia sido aprimorada com a introdução das **perfuradoras de cartões**.
- Era possível, então, escrever programas em cartões e lê-los em lugar de painéis de programação; de outra maneira, o procedimento seria o mesmo.

Segunda geração (1955 - 1965) - transistores e sistemas em lote (batch)

- Com 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.
- Pela primeira vez, havia uma clara separação entre projetistas, fabricantes, programadores e técnicos da manutenção.

Segunda geração (1955 - 1965) - transistores e sistemas em lote (batch)

- Essas máquinas eram denominadas **computadores de grande porte** (*mainframes*).
- Ficavam isoladas em salas especiais com ar-condicionado, operadas por equipes profissionais.
- Somente grandes corporações, agências governamentais ou universidades podiam pagar vários milhões de dólares para tê-las.

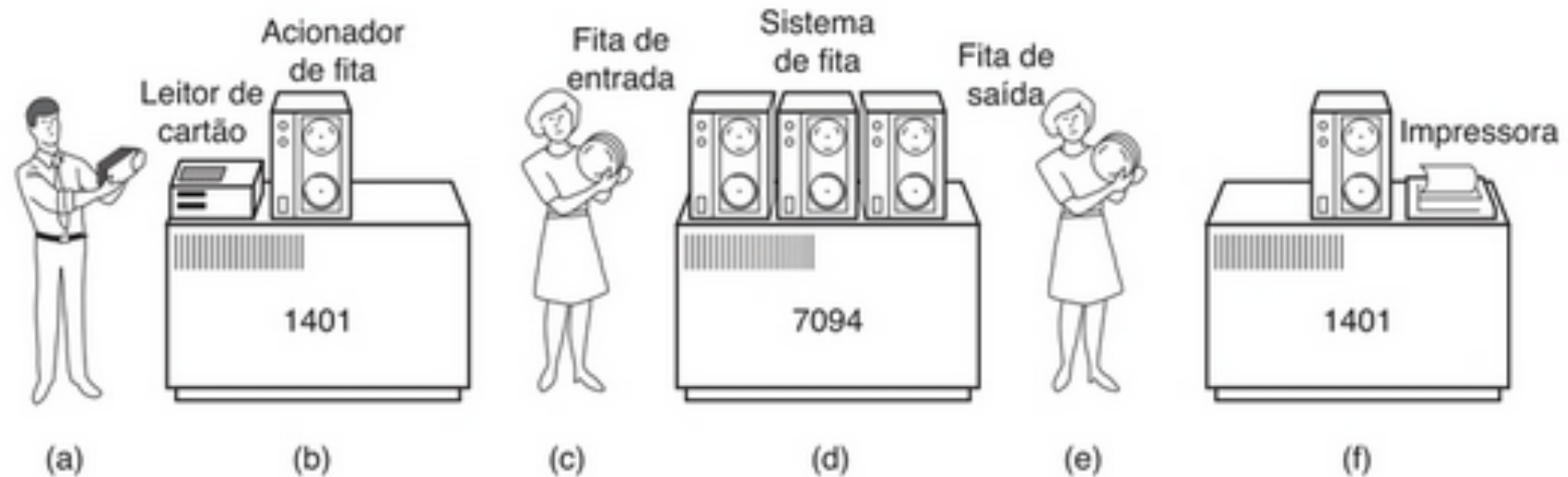
Segunda geração (1955 - 1965) - transistores e sistemas em lote (batch)

- 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 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.
- Ao fim da execução de uma tarefa pelo computador, um operador ia até a impressora, retirava sua saída e a levava para a sala de saídas, de modo que o programador pudesse retirá-la mais tarde.
- Ele então apanhava um dos maços de cartões que foram traduzidos para a sala de entradas e colocava na leitora de cartões.
- Se fosse necessário um compilador Fortran, o operador precisava retirar do armário o maço de cartões correspondentes e lê-lo.
- Muito tempo de computador era desperdiçado enquanto os operadores andavam pela sala das máquinas.

Segunda geração (1955 - 1965) - transistores e sistemas em lote (batch)

- Por causa do alto custo do equipamento, buscou-se maneiras de reduzir o desperdício de tempo no uso da máquina.
- A solução geralmente adotada era a do **sistema em lote** (*batch*).
- A ideia era gravar várias tarefas em fita magnética usando um computador relativamente mais barato, como o IBM 1401, que era muito bom para ler cartões, copiar fitas e imprimir saídas, mas não tão eficiente em cálculos numéricos.

Segunda geração (1955 - 1965) - transistores e sistemas 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 na 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.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- No início da década de **1960**, a maioria dos fabricantes de computadores oferecia duas linhas de produtos distintas e totalmente incompatíveis.
- De um lado havia os **computadores científicos de grande escala e orientados a palavras**, como o 7094, usados para cálculos numéricos na ciência e na engenharia.
- De outro, existiam os **computadores comerciais orientados a caracteres**, como o 1401, amplamente usados por bancos e companhias de seguros para ordenação e impressão em fitas.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- A IBM tentou resolver esses problemas de uma única vez introduzindo o **System/360**. O 360 era uma série de máquinas - desde máquinas do porte do 1401 até mais potentes que o 7094 - cujos softwares eram compatíveis.
- Como todas as máquinas tinham a b **arquitetura** e o mesmo **conjunto** de **instruções**, os programas escritos para uma máquina podiam ser **executados** em todas as outras, pelo menos teoricamente.

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**.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- A forte ideia de “família de máquinas” era sua maior fraqueza.
- A intenção era que qualquer software, inclusive o sistema operacional **OS/360**, pudesse ser executado em qualquer dos modelos.
- O software precisa ser executado em **sistemas pequenos** e em **sistemas muito grandes** (previsão do tempo).
- Deveria ser eficiente em sistemas com **poucos periféricos** e com **muitos periféricos**.
- Deveria funcionar bem em **ambientes comerciais** e em **ambientes científicos**.

Terceira geração (1965 - 1980) - CIs e multiprogramação

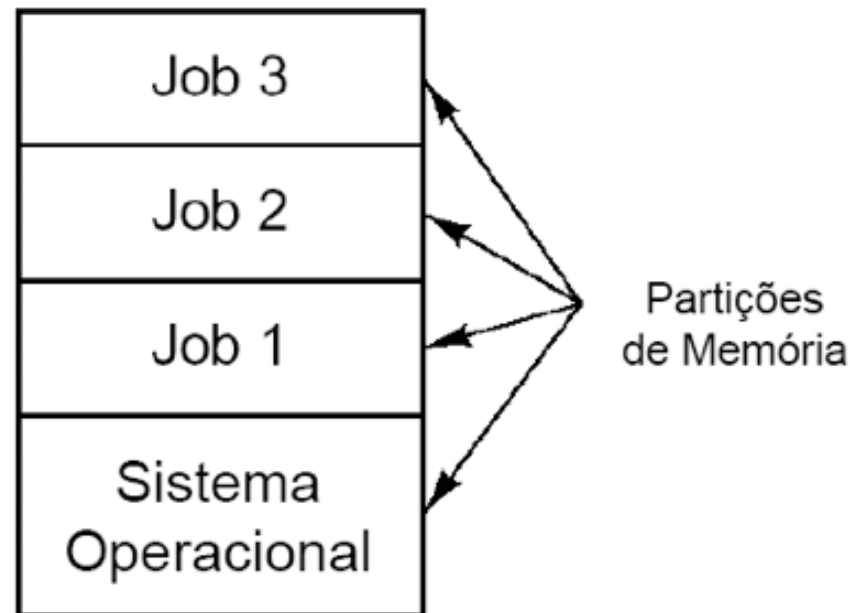
- O resultado disso foi um **sistema operacional enorme e extraordinariamente complexo**.
- Eram milhões de linhas escritas em linguagem assembly por milhares de programadores, contendo milhares de erros, que precisavam de um fluxo contínuo de novas versões par tentar corrigi-los.
- Cada nova versão corrigia alguns erros, mas introduzia outros, fazendo com que, provavelmente, o número de erros permanecesse constante ao longo do tempo.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- A técnica mais importante presente no sistema operacional era a **técnica de multiprogramação**.
- Essa técnica dividia a memória em várias partes, com uma tarefa diferente em cada partição.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- Enquanto uma tarefa esperava que uma operação de E/S se completasse, outra poderia usar a CPU. Se o número suficiente de tarefas pudesse ser mantido na memória ao mesmo tempo, a CPU poderia permanecer ocupada por quase 100 por cento do tempo.



Terceira geração (1965 - 1980) - CIs e multiprogramação

- Outro aspecto importante nos sistemas operacionais de terceira geração era a capacidade de **transferir tarefas de cartões perfurados para discos magnéticos** logo que esses chegassem à sala do computador.
- Dessa forma, assim que uma **tarefa fosse completada**, o sistema operacional poderia **carregar uma nova tarefa** a partir do disco na partição que acabou de ser liberada e, então, processá-la.
- Essa **técnica** é denominada **spooling**.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- Outro aspecto da terceira geração de sistemas operacionais é o **timesharing** ou **tempo compartilhado**, uma variante da multiprogramação na qual cada usuário se conectava por meio de um terminal on-line.
- Em um sistema de tempo compartilhado, se 20 usuários estivessem conectados e 17 deles estivessem pensando, falando ou tomando café, a CPU podia ser **ciclicamente alocada** a cada **uma** das três tarefas que estivessem requisitando a CPU.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- O primeiro sistema importante de tempo compartilhado, o **CTSS** (*compatible time sharing time* - sistema compatível de tempo compartilhado) foi desenvolvido no MIT em um 7094 modificado.
- Depois do sucesso desse sistema CTSS, o **MIT**, o **Beel Labs** e a **General Eletrics** decidiram desenvolver um “computador utilitário”, uma máquina que suportasse simultaneamente centenas de usuários compartilhando o tempo.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- O MULTICS foi projetado para suportar centenas de usuários em uma única máquina somente um pouco mais potente que um PC baseado no 386 da Intel, embora tendo muito mais capacidade de E/S.
- Alguns anos depois, **Ken Thompson**, um cientista da computação da Bell Labs que trabalhou no projeto MULTICS, achou um pequeno microcomputador 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.

Terceira geração (1965 - 1980) - CIs e multiprogramação

- Podemos dizer um clone do UNIX, TANENBAUM lançou o conhecido MINIX, com objetivo educacional, em 1987.
- O sistema MINIX 3 está disponível gratuitamente (com o código-fonte) pela Internet em www.minix3.org.
- Com o desejo de produzir uma versão gratuita do MINIX (diferente da educacional) levou um estudante finlandês, **Linus Torvalds**, a escrever o **Linux**.

Quarta geração (1980 até o 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 da classe PDP-11, mas no preço eram claramente diferentes.
- 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.
- **Kildall** escreveu um sistema operacional baseado em disco denominado **CP/M** (*control program for microcomputers* - programa de controle para microcomputador).

Quarta geração (1980 até o presente) - computadores pessoais

- Com o tempo surgiu o **MS-DOS** (MicroSoft disk operation system - sistema operacional de disco da Microsoft), uma junção do interpretador **Basic** de propriedade de **Bill Gates** e o **DOS** (*disk operation system* - sistema operacional de disco) de propriedade da Seattle Computers Products. Bill Gates comprou o DOS e ofereceu a IBM um pacote **DOS/Basic**.
- Com o tempo foram surgindo outras versões, conhecidas por todos como Windows.

Arquitetura de Sistemas Operacionais

- Em função de sua arquitetura os sistemas operacionais modernos podem ser classificados em três tipos:
 - Núcleo monolítico ou monobloco;
 - Micronúcleo ou microkernel;
 - Máquina virtual.

Núcleo monolítico ou monobloco

É um núcleo que **implementa** uma **interface** de **alto nível**, para **possibilitar chamadas de sistema específicas** para **gestão de processos, concorrência e gestão de memória** por parte de módulos dedicados, que são executados com privilégios especiais.

Mesmo que cada módulo de manutenção dessas operações seja separado, de uma forma geral, é muito difícil fazer o código de integração entre todos esses módulos.

E, uma vez que todos eles executam em um mesmo espaço de endereçamento, um erro em um módulo pode derrubar todo o sistema.

Núcleo monolítico ou monobloco

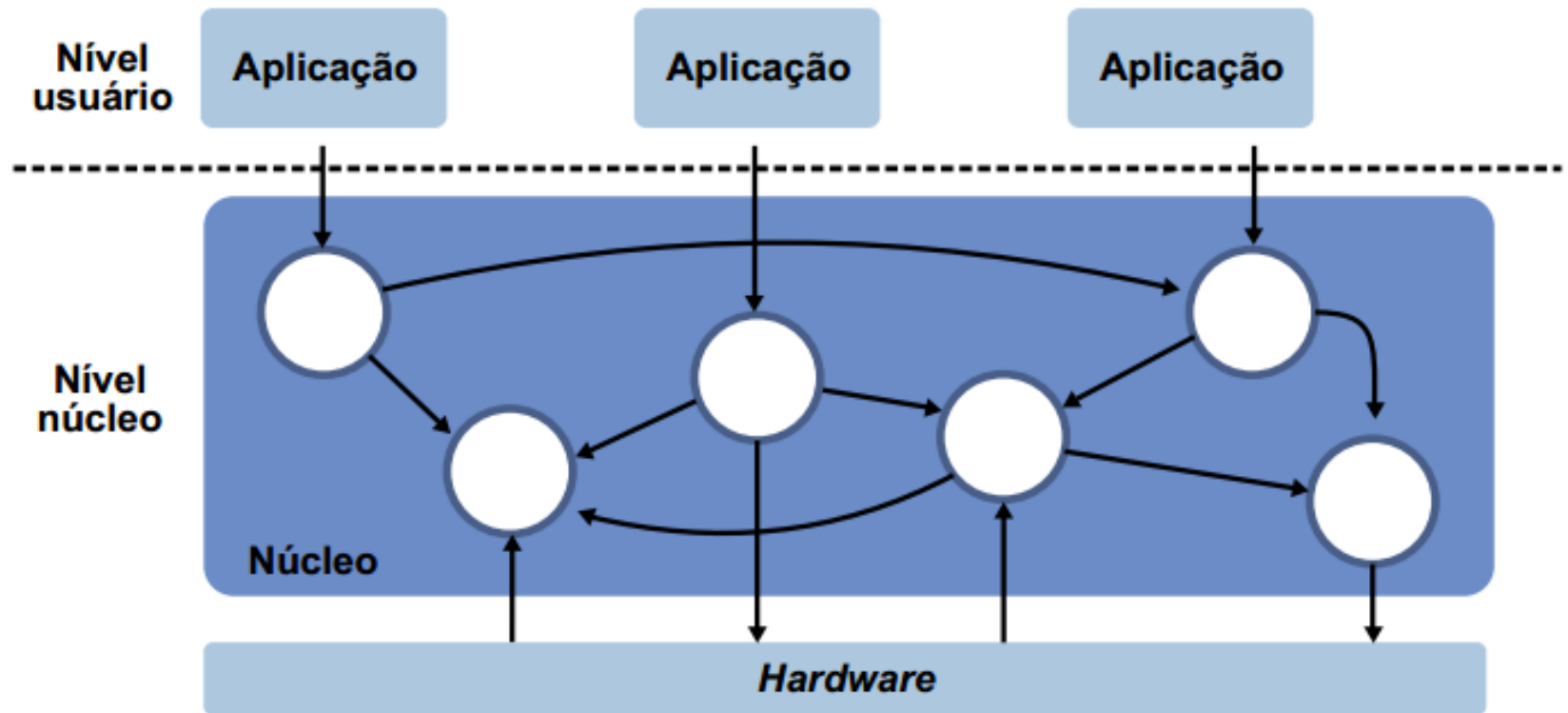


Figura. Núcleo monolítico ou monobloco.

Micronúcleo ou microkernel

É um termo usado para caracterizar o sistema cujas **funcionalidades saíram do núcleo** e foram para **servidores**, que se comunicam com um **núcleo mínimo**, usando o **mínimo possível** do “espaço do sistema” (nesse local, o programa tem acesso a todas as instruções e a todo o hardware) e **deixando o máximo de recursos** rodando no “espaço do usuário” (nesse espaço, o software sofre algumas restrições, não podendo acessar alguns hardwares e não tendo acesso a todas as instruções).

Micronúcleo ou microkernel

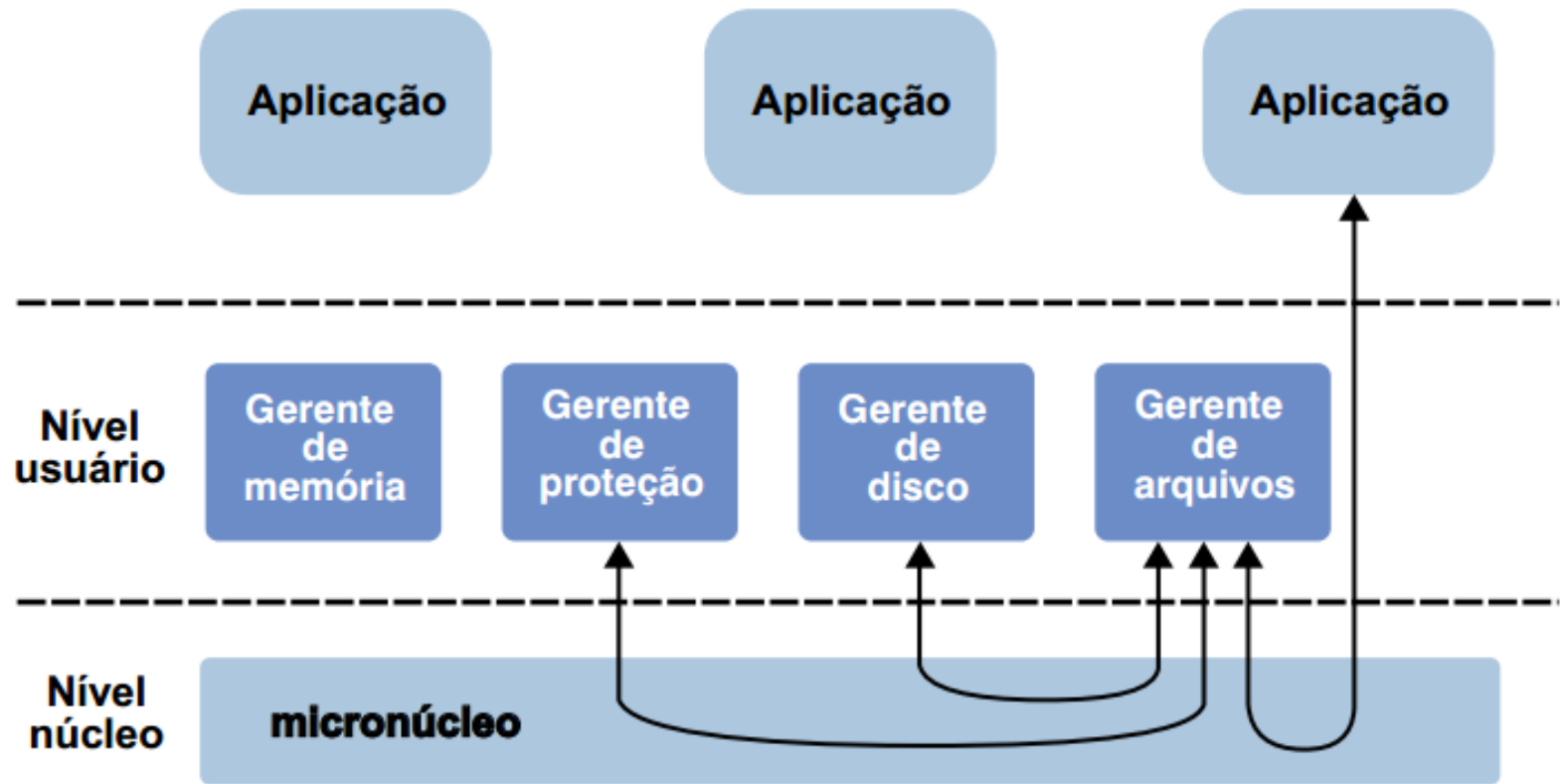


Figura. Micronúcleo ou microkernel.

Máquina Virtual

É uma **arquitetura** que permite que um **sistema operacional** suporte aplicações de outros sistemas (Linux executando sobre o Windows, por exemplo) ou mesmo outro sistema operacional completo.

Possibilita que sobre um mesmo hardware possam ser executados dois ou mais sistemas operacionais diferentes.

Máquina Virtual

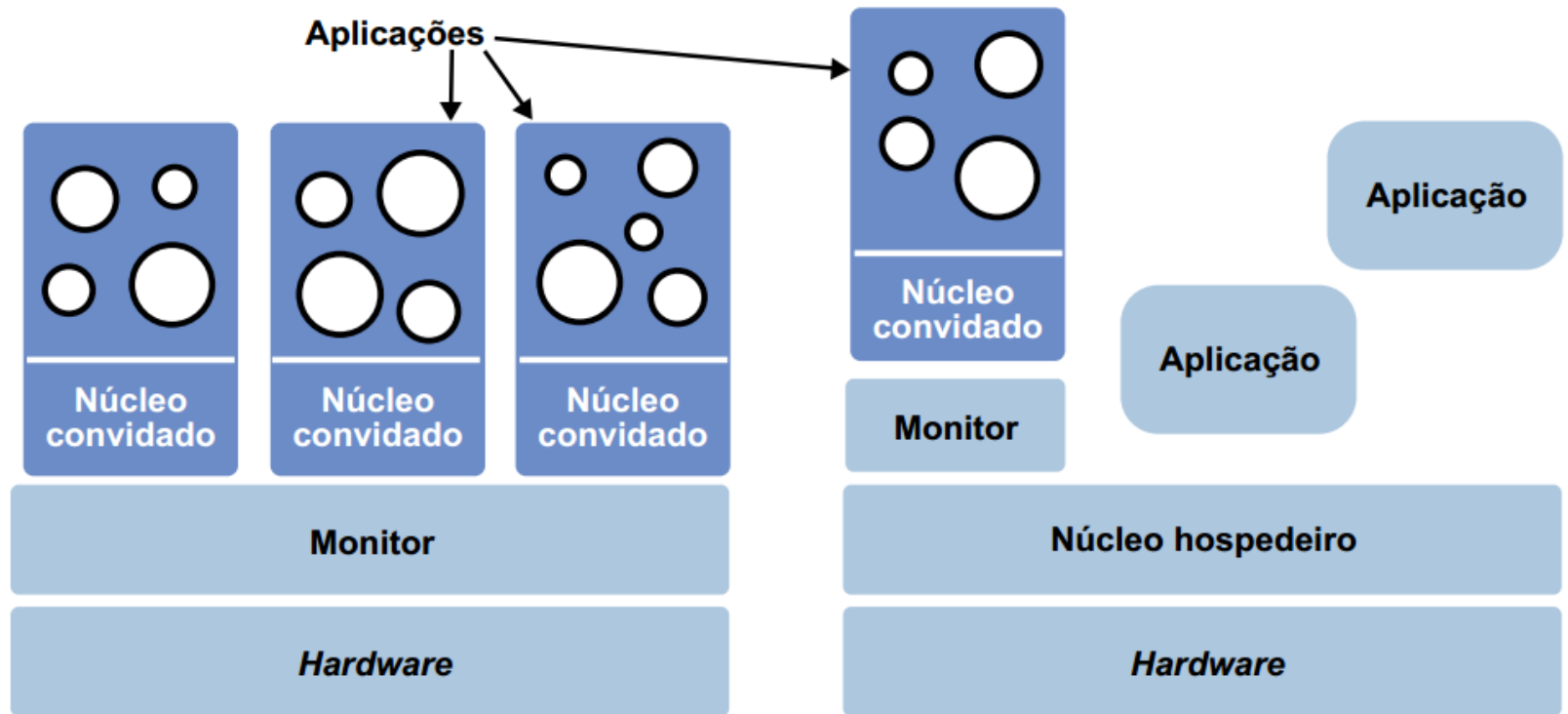
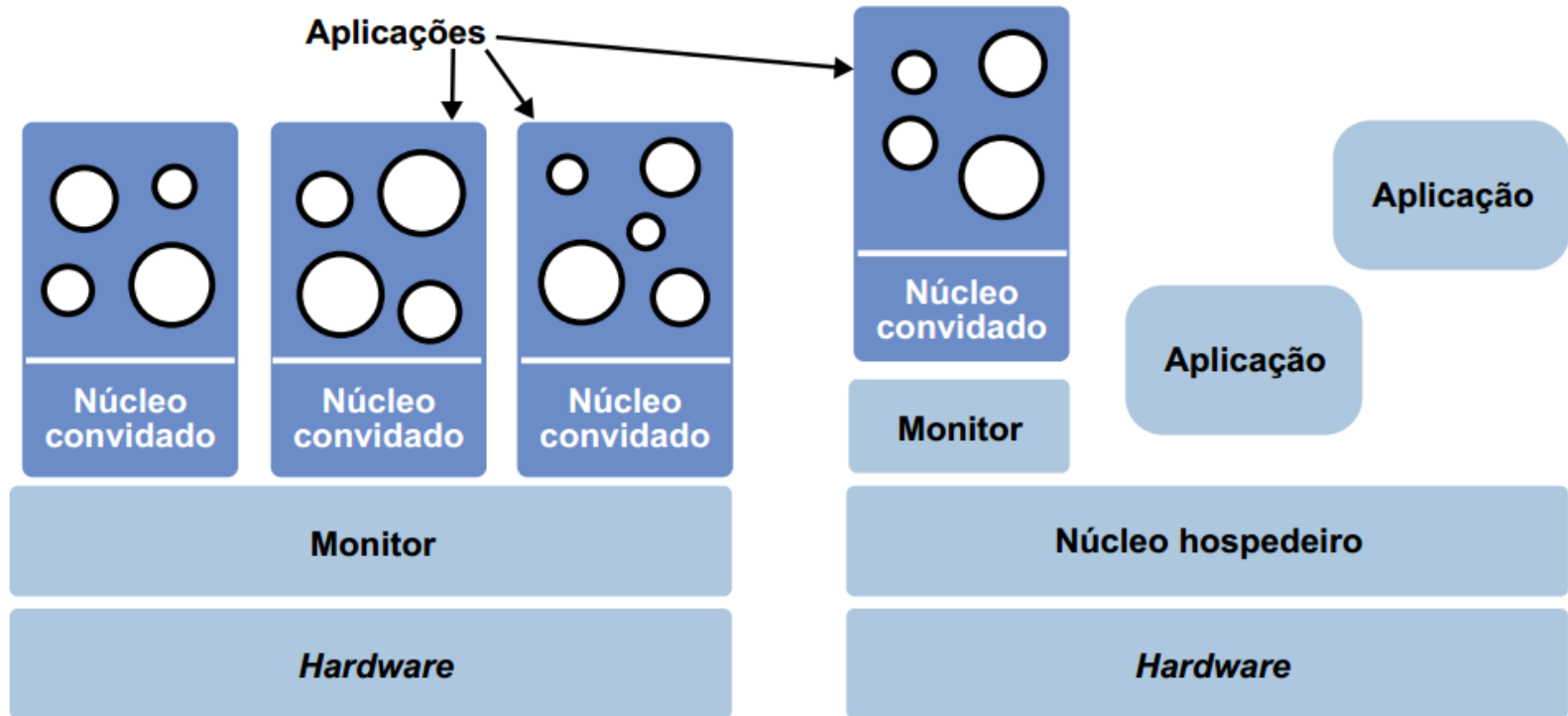


Figura. Máquina Virtual.

Máquina Virtual



No primeiro caso, o monitor executa diretamente sobre o hardware e gerencia um ou mais sistemas operacionais convidados. No segundo caso, o monitor possibilita a execução de um sistema operacional sobre o outro.

PRÓXIMA AULA

- Estrutura do SO;
- Cliente Servidor.



Conteúdo e Dúvidas

- ❑ Conteúdo
 - NetAula
- ❑ Dúvidas
 - wagnerglorenz@gmail.com.



Referências Bibliográficas

- TANENBAUM, A. Sistemas Operacionais Modernos.
São Paulo: Prentice Hall, 2003.

BIBLIOTECA PERSON:

Sistemas Operacionais Modernos - 3ª edição Tanenbaum,
Andrew S.

[http://ulbra.bv3.digitalpages.com.br/users/publications/
9788576052371](http://ulbra.bv3.digitalpages.com.br/users/publications/9788576052371)