

# Sistemas Operacionais



Introdução

# Introdução

---



# Introdução

---

- Sistema Operacional
  - é um programa colocado entre o hardware do computador e os programas do usuário
  - torna o uso do computador **conveniente**
    - Ex.: Computadores Pessoais → GUI
  - uso **eficiente** do hardware
    - No passado apenas este aspecto
    - Teoria dos SO → uso ideal dos sistemas de computação
  - o sistema operacional controla e coordena o uso do hardware entre os vários programas aplicativos para os vários usuários
- Visão do usuário → facilidade de uso e desempenho
- Visão do sistema → ...

# Introdução

---

- SO é um
  - alocador de recursos
    - Tempo de CPU, memória, espaço de armazenamento de arquivos, dispositivos de I/O
    - Aloca esses recursos aos programas do usuário
    - Tomada de decisões para alocação
  - programa de controle
    - Controla a execução de programas do usuário
    - Evitar erros e uso indevido do hardware

# Introdução

---

- Proporcionar a execução de programas do usuário e facilitar a resolução de problemas do usuário
- SO é um programa que está sempre executando – *kernel*, o restante são programas aplicativos
- Os SO e a arquitetura dos computadores tiveram influência mútua
  - Desenvolvimento nos últimos 35 anos



# Histórico

---

## 1. Sistemas de Grande Porte

- Usados nos *mainframes*
  - ▣ *primeiras aplicações comerciais e científicas*
- Sistemas em lote
- Sistemas multiprogramados
- Sistemas de Tempo Compartilhado

# Histórico

---

## ❑ Sistemas em lote (batch)

- Máquinas grandes, operadas a partir de um console
- Dispositivos de entrada
  - ❑ Leitoras de cartões e unidades de fita
- Dispositivos de saída
  - ❑ Impressoras, unidades de fita e perfuradoras de cartões
- O usuário não interagia diretamente com a máquina
  - ❑ Operadores profissionais agrupavam as tarefas
- O SO era simples
  - ❑ Transferia o controle de um job para outro

# Histórico

---

## □ Sistemas em lote (batch) – continuação

### ■ Job (tarefa)

- Criado pelo usuário – dados e programa; uso de cartões
- Organizados pelo operador em lote, conforme as características/necessidades
- Passagem entre diferentes jobs é manual
- Saída aparecia algum tempo depois

### ■ CPU ociosa

- Velocidade dos dispositivos de entrada/saída x velocidade CPU



# Histórico

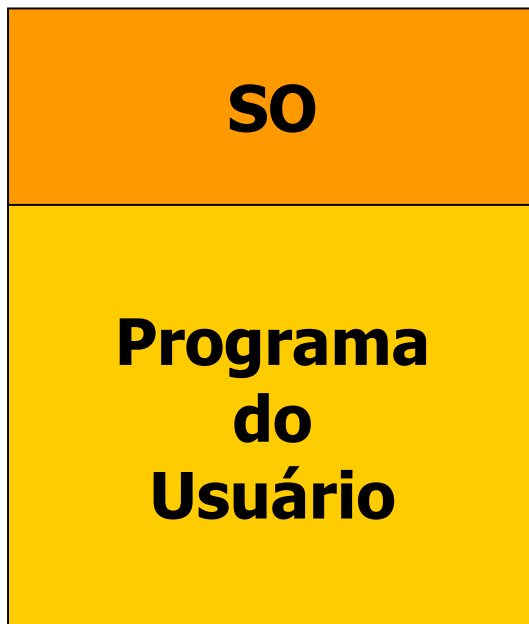
---

- **Sistemas em lote (batch) – continuação**
  - SO sempre residente na memória – Monitor residente
    - Fazia o sequenciamento automático dos jobs
    - Quando o job termina, controle retorna ao monitor
    - Centraliza acesso a periféricos
      - Tempo ocioso da CPU (trabalhava na faixa dos microssegundos: milhares de instruções por segundo; periféricos 20 cartões por segundo)
    - Execução de um programa de cada vez

# Histórico

---

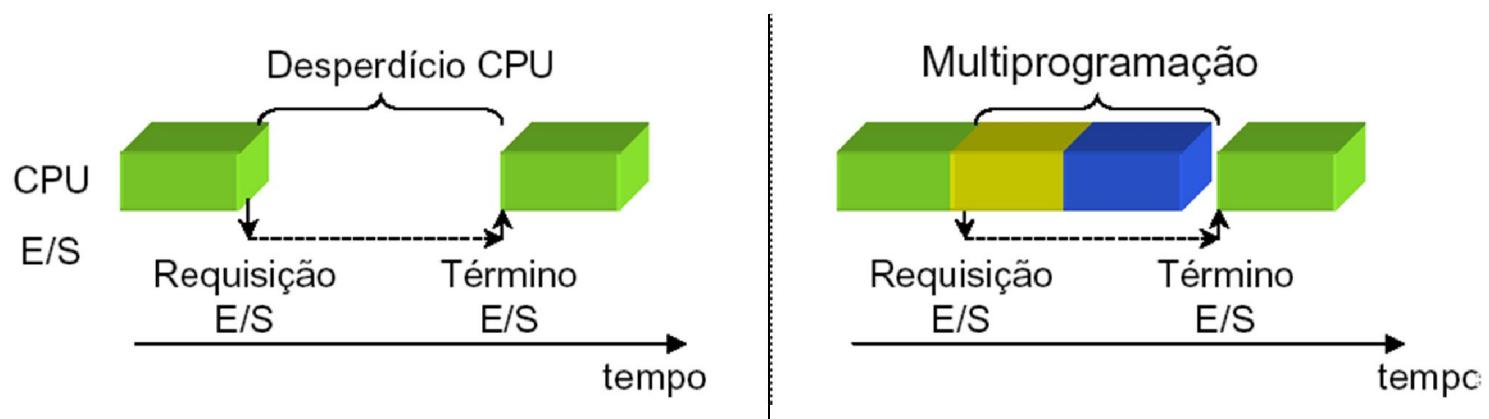
- Sistemas em lote (batch) – continuação
  - Memória em um sistema batch simples



# Histórico

## ❑ Sistemas multiprogramados

- Sistemas *batch* com os discos
  - ❑ Todas as tarefas no disco (ao invés de leitora serial)
- E com as interrupções → escalonamento de job → multiprogramação
- Tarefa pode esperar por evento ou E/S
- Vários jobs na memória
  - ❑ Sobrepor operações de E/S com processamento



# Histórico

---

- Sistemas multiprogramados – continuação
  - Memória



# Histórico

---

## □ Sistemas multiprogramados – continuação

- SO precisa tomar decisões
- *Pool* de Jobs (no disco)
- Escalonamento de Jobs
- Escalonamento de CPU
- Gerência de Memória
- Gerência de I/O
- ...

# Histórico

---

- Sistemas *Timesharing* ou Tempo Compartilhado
  - CPU executa vários jobs alternando entre eles
    - As trocas ocorrem com frequência
    - Os usuários podem interagir com o programa em execução → possuem terminal
    - Tarefas curtas = muita E/S
  - Há uma divisão do tempo de processamento entre os vários usuários → tempo de resposta
    - Permite aos usuários compartilharem o computador ao mesmo tempo

# Histórico

---

- Sistemas *Timesharing* ou Tempo Compartilhado
  - Multiprogramação
    - Cada usuário tem um programa na memória (pelo menos)
    - Compartilhamento de tempo
    - Sistema interativo → uso CPU x uso I/O
    - Escalonamento de CPU
    - Gerência de Memória
    - Memória virtual
    - Sistemas de Arquivos
    - ...
  - Tornaram-se comuns no início da década de 70

# Histórico

---

## 2. Sistemas de Computadores Pessoais (ou *desktop*)

- Inicialmente, não possuíam recursos para proteger o SO; não eram multiusuário, nem multitarefa
- MULTICS
  - Foi desenvolvido de 1965 a 1970 no MIT – *Massachusetts Institute of Technology* como utilitário
  - Executava em um mainframe GE645
- Idéias usadas no *Bell Laboratories* para criação do Unix (1970), em um minicomputador PDP-11
- 1980-90 – recursos do Unix → sistemas de microcomputador



# Histórico

---

## □ Sistemas monousuário

- Ex.: MS-DOS, Windows 3.x

## □ Sistemas multiusuário

- Suportam várias sessões de um usuário em um computador
- Ex.: Win NT, Unix

## □ Sistemas monotarefa

- Ex.: MS-DOS

## □ Sistemas multitarefa

- Não-Preemptivo: Ex.: Windows 3.x, 9x (16 bits)
- Preemptivo: Ex.: Win NT, OS/2, Unix, Windows 9x (32 bits)
- Beneficiaram-se de vários conceitos de *mainframes*

# Histórico

---

## 3. Sistemas Paralelos

- Vários processadores (multiprocessados) compartilhando barramento, *clock*, memória...
- Fortemente acoplados
- Comunicação através da memória
- Motivos
  - Maior *throughput*
  - Mais trabalho em menos tempo
  - Maior confiabilidade
- Multiprocessamento simétrico (mais comuns)
  - Todos processadores são iguais
  - Um processo mestre controla o sistema (SMP – *Symmetric Multi-Processing*)
  - A maioria dos SO suporta SMP através de *multithreading*
  - Sistemas Operacionais como Win NT, Solaris, Unix, OS/2, Linux possuem suporte a SMP

# Histórico

---

## 4. Sistemas Distribuídos

- 1990 – crescimento das redes de computadores
- Coleção de computadores que não compartilham memória ou *clock*, interconectados e trabalham de forma a disponibilizar um serviço
- Sistemas fracamente acoplados
- Comunicação acontece por trocas de mensagens
  - ▣ Linhas de comunicação: barramentos, linhas telefônicas

### 4.1 Sistemas Cliente-Servidor

### 4.2 Sistemas peer-to-peer

### 4.3 Sistemas Operacionais Distribuídos

# Histórico

---

## □ Sistemas de Tempo Real

- Sistemas com propósito específico
- Aplicação dedicada
- Processamento precisa ser feito dentro dos limites de tempo definidos
- Quase não é encontrada memória virtual nestes sistemas.
- Ex.: experimentos científicos, imagens médicas, controle industrial, sistemas de injeção de combustíveis, controladores de eletrodomésticos

# Considerações Finais

---

## □ SO

- Oferece ao usuário um ambiente conveniente para o desenvolvimento e execução de programas
- Gerencia os recursos de um sistema de computação de forma a garantir a correta e eficiente utilização dos mesmos

## □ Multiprogramação



# Bibliografia

---

- ❑ SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas Operacionais com Java**. Rio de Janeiro: Campus, 2004. (capítulo 1)
- ❑ OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão. **Sistemas Operacionais**. Porto Alegre: Sagra-Luzzato, 2001. (capítulo 1)