



INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO

AULA 9: SISTEMAS
OPERACIONAIS

PROF^a: LEONARA BRAZ
LEONARABRAZ@CRATEUS.UFC.COM



SEÇÃO 1

INTRODUÇÃO AOS SISTEMAS OPERACIONAIS

DEFINIÇÃO

- Apenas com o hardware o computador seria de difícil utilização
 - Comandos executados em linguagem de máquina (Assembly)
 - Cada computador possui arquitetura e hardware diferente
 - Seria uma grande dificuldade para os programadores desenvolver um programa de computador complexo
- Necessidade de uma camada intermediária entre hardware e programadores

ESTRUTURA DE UM SISTEMA COMPUTACIONAL

- Usuários
- Aplicativos (programas)
- Sistema Operacional
- Hardware

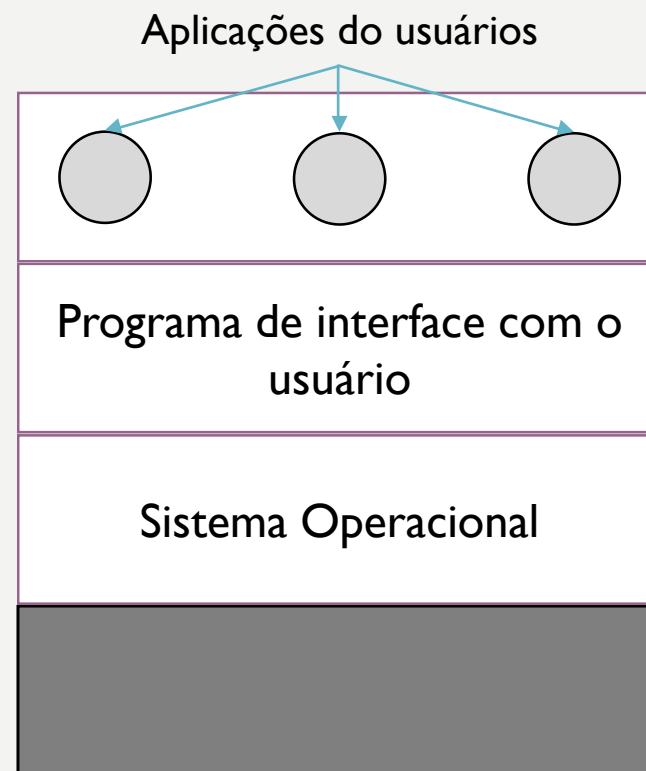


Figura: Onde o SO se encaixa

ESTRUTURA DE UM SISTEMA COMPUTACIONAL

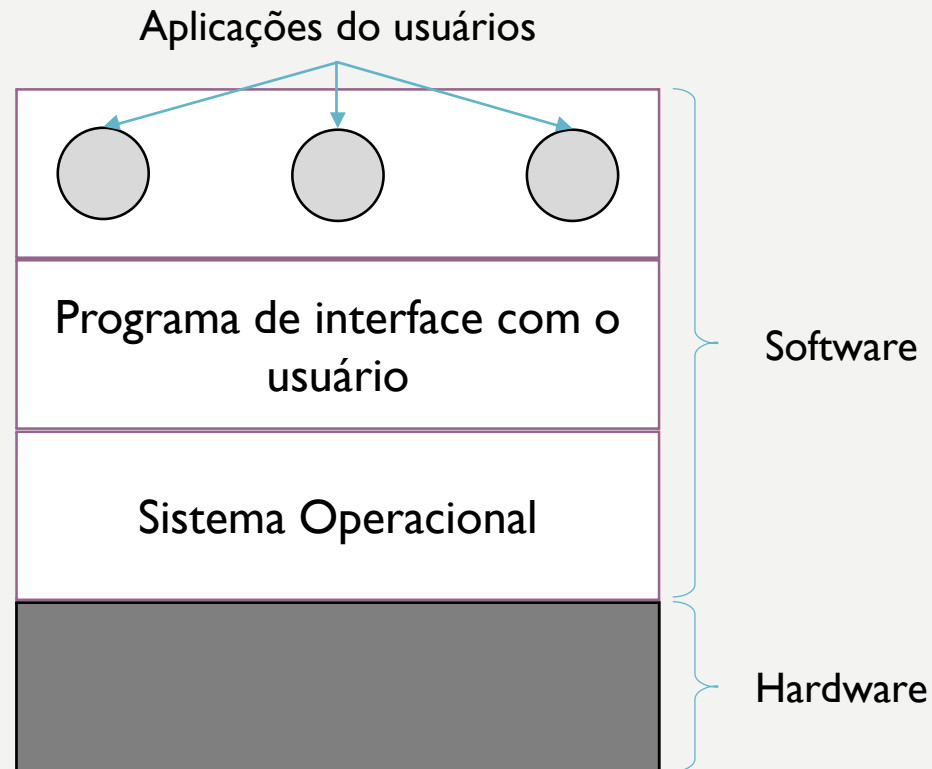


Figura: Onde o SO se encaixa

- **Hardware**
 - Chips, placas, discos, teclado – Objetos físicos
- **Software**
 - O programa de interface com o usuário é o nível mais inferior
 - Permite ao usuário iniciar programas como navegador web, leitor de e-mails, etc
- **Sistema Operacional**
 - Opera diretamente no hardware e fornece a base para os outros programas

PROGRAMAS E APLICATIVOS

- Os programas exigem diferentes recursos do Hardware
- Os programadores nem sempre são cuidadosos:
 - Especialmente em relação ao fechamento e liberação de recursos, tais como (regiões de memória, sockets, arquivos, etc).
 - **Isso pode prejudicar o funcionamento geral do sistema**

HARDWARE

- Elevada pluralidade no hardware de um computador:
 - Um ou mais processadores, memória principal, discos, impressoras, vários dispositivos de entrada e saída, etc.
 - *Cada componente tem diferentes fabricantes*
- **Como criar um programa que fosse capaz de conversar com todos esses tipos de componentes e fabricantes?**

SISTEMA OPERACIONAL

“Computadores têm um componente de software conhecido como **Sistema Operacional**, cujo trabalho é **fornecer** aos programas do usuário **um modelo de computador** melhor, mais simples e mais limpo de **lidar com o gerenciamento de todos os recursos de hardware** que formam um sistema computacional.”

TANEMBAUM (2010)

FUNÇÕES BÁSICAS DO SISTEMA OPERACIONAL

I. Fornecer aos programadores de aplicativos (e aos programas) um conjunto de recursos abstratos claros

2. Gerenciar os recursos do Hardware

OBJETIVOS GERAIS DO SISTEMA OPERACIONAL

1. Executar programas do usuário de forma a ajudá-lo a resolver seus problemas de forma mais simples
2. Facilitar o uso de um sistema computacional
3. Utilizar o hardware disponível de forma eficiente

OBJETIVOS ESPECÍFICOS DO SISTEMA OPERACIONAL

1. Organiza a execução do hardware e do software do computador
2. Gerencia a execução de múltiplos programas executando simultaneamente
3. Controla os recursos do computador
4. Evita violações de segurança, como controle de acesso aos dados de cada usuário
5. Otimização dos recursos
6. Facilidade para desenvolvedores com a transparência de comunicação com o hardware do computador

SO COMO MÁQUINA ESTENDIDA

- A **arquitetura** da maioria dos computadores em **nível de linguagem de máquina** é primitiva e de difícil programação
 - Especialmente a entrada/saída
- Exemplo: Chip controlador NEC-PD765

NEC-PD765

- Chip controlador de disquetes em computadores pessoais baseados em processadores Intel
- Possui 16 comandos para leitura e escrita de dados, para movimentação do braço e para formatação das trilhas
- Os comandos mais básicos são de leitura e escrita:
 - Cada um requer 13 parâmetros agrupados em 9 bytes
 - Os parâmetros especificam endereço de bloco de dados, número de setores, modo de gravação
- Ao completar a operação, o chip controlador retorna 23 campos de status e erros agrupados em 7 bytes

ABSTRAÇÃO

- O programador busca lidar com essas unidades de um modo mais abstrato e simples
- **Como podemos abstrair o procedimento das unidades de disco?**

ABSTRAÇÃO

Uma abstração típica seria aquela compreendida por um disco que contém uma **coleção de arquivos** com nomes. Cada arquivo pode ser aberto para leitura ou escrita e, então, **ser lido ou escrito** e, por fim, fechado.

Detalhes sobre o processo **não apareceriam** na abstração apresentada ao programador da aplicação.

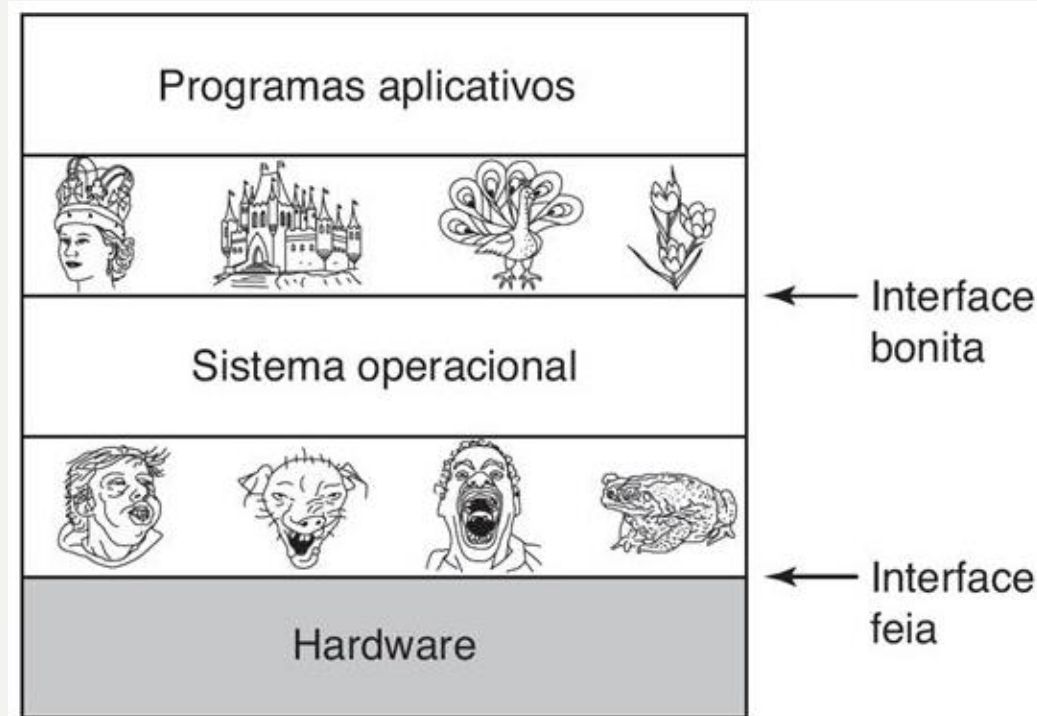
ABSTRAÇÃO

- **Abstração é o elemento-chave para gerenciar complexidade**
 - Boas abstrações transformam uma tarefa quase impossível em duas manejáveis
 1. Definição e implementação da abstração
 2. Uso da abstração para resolução de problema
- A tarefa principal do SO é **criar boas abstrações!**
- Exemplo de abstração: Arquivo
 - Fragmento de informação útil

ABSTRAÇÃO

O hardware é feio

- Uma das principais tarefas do sistema operacional é ocultar o hardware
 - Oferecendo abstrações precisas, claras e coerentes
- O SO transforma o feio em bonito



IMPORTANTE

Os clientes reais do Sistema Operacional são os programas



Os usuários finais lidam com abstrações fornecidas pela interface do usuário

```
cmd: Administrador: C:\Windows\System32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>systeminfo

Nome do host: PC
Nome do sistema operacional: Microsoft Windows 7 Ultimate
Versão do sistema operacional: 6.1.7601 Service Pack 1 compilação 7601
Fabricante do sistema operacional: Microsoft Corporation
Configuração do SO: Estação de trabalho autônoma
Tipo de compilação do sistema operacional: Multiprocessor Free
Proprietário registrado: edivaldo
Organização registrada:
Identificação do produto: 00406 0FM 0000600 000001
Data da instalação original: 03/02/2013, 10:23:50
Tempo de Inicialização do Sistema: 25/11/2013, 08:31:25
Fabricante do sistema: Positivo Informatica SA
Modelo do sistema: POS-ECIG41BS
Tipo de sistema: x64-based PC
Processador(es): 1 processador(es) instalado(s).
[01]: Intel64 Family 6 Model 23 Stepp
ing 10 GenuineIntel ~2793 Mhz
Versão do BIOS: Desenvolvida para Positivo Informatic
a SA 080015 , 14/05/2010
```

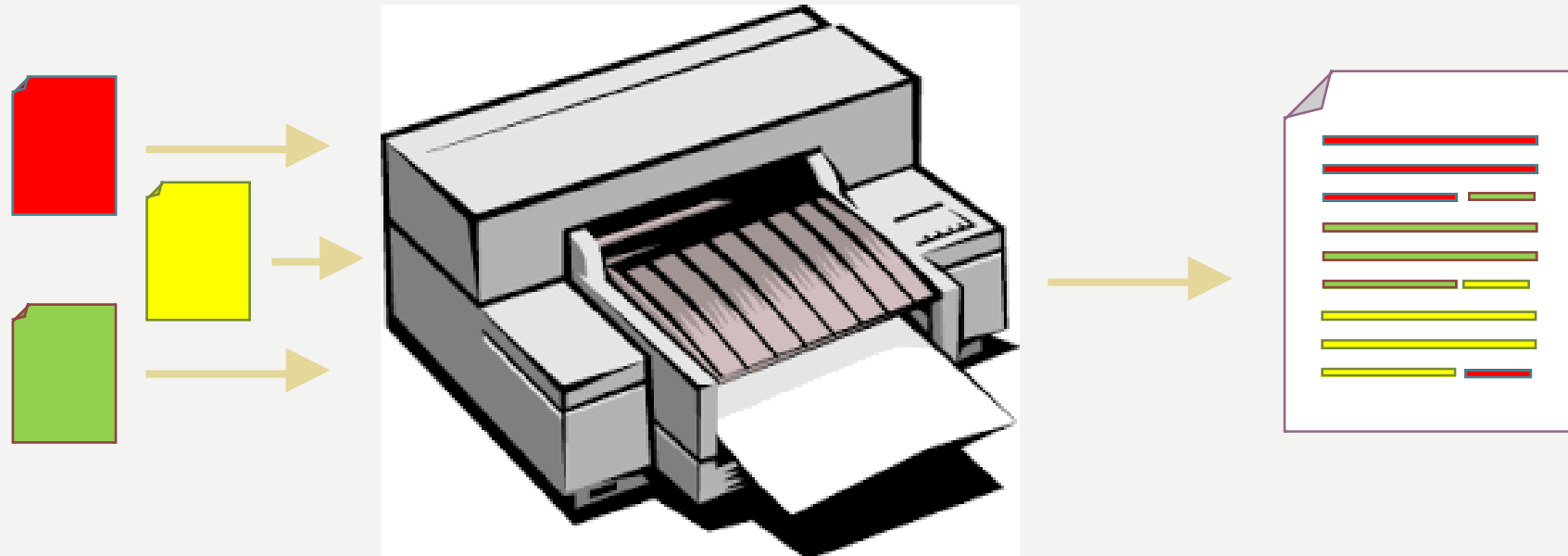
SO COMO GERENCIADOR DE RECURSO

Um sistema operacional **gerencia** todas **as partes** de um sistema complexo.

O trabalho de um sistema operacional é **fornecer** uma **alocação ordenada e controlada** de processadores, memórias e dispositivos de E/S **entre** vários **programas** que competem por eles

SO COMO GERENCIADOR DE RECURSO

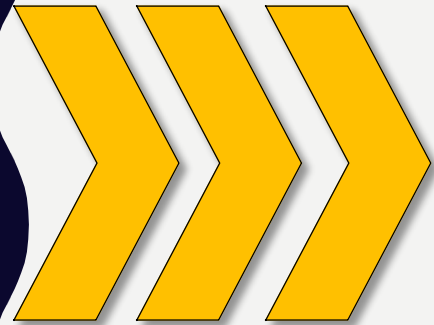
- Imagine o que aconteceria se três programas em execução tentassem imprimir suas saídas simultaneamente na mesma impressora



SO COMO GERENCIADOR DE RECURSO

- Algumas vezes, se faz necessário que usuários e programas compartilhem hardware e informações
 - O SO também deve estar preparado para tal ocasião
- O compartilhamento de recursos pode ser realizado de duas maneiras: **tempo e espaço**
 - **No tempo:** Diferentes programas esperam sua vez para utilizar o recurso
 - **No espaço:** Em lugar de esperar sua vez, cada programa ocupa uma parte do recurso compartilhado

TIPOS DE COMPARTILHAMENTO

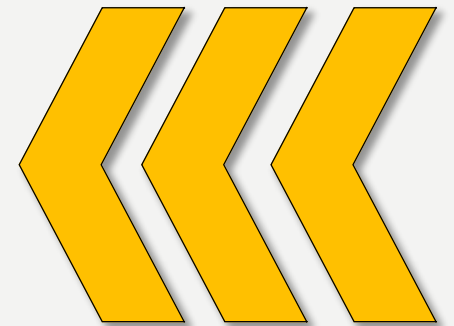


COMPARTILHAMENTO NO TEMPO

O processador somente recebe um programa por vez para ser executado, os demais esperam

COMPARTILHAMENTO NO ESPAÇO

A memória principal é normalmente segmentada entre vários programas em execução



FUNÇÕES DO SO NO COMPARTILHAMENTO

- Determinar como o recurso é compartilhado no tempo:
 - Qual processo vai utilizar o recurso e por quanto tempo será
- Determinar como o recurso é compartilhado no espaço:
 - Tamanho do espaço para cada processo; justiça; proteção; etc
- Tudo é feito através de componentes do SO:
 - **Escalonador de processos** para as questões temporais
 - **Gerenciador de memória** para as questões espaciais

ESCALONADOR DE PROCESSOS

- Quando um computador é multiprogramado, ele muitas vezes possui diversos processos que competem pela CPU, surgindo assim a necessidade de alguma entidade para escalonar a CPU entre os processos
 - Essa entidade é conhecida como escalonador
 - O algoritmo que é usado é chamado de algoritmo de escalonamento.
- O **objetivo do escalonador** é atribuir processos para serem executados pelos processadores de modo a atingir parâmetros de performance, tais como *tempo de resposta, vazão e eficiência do processador*.

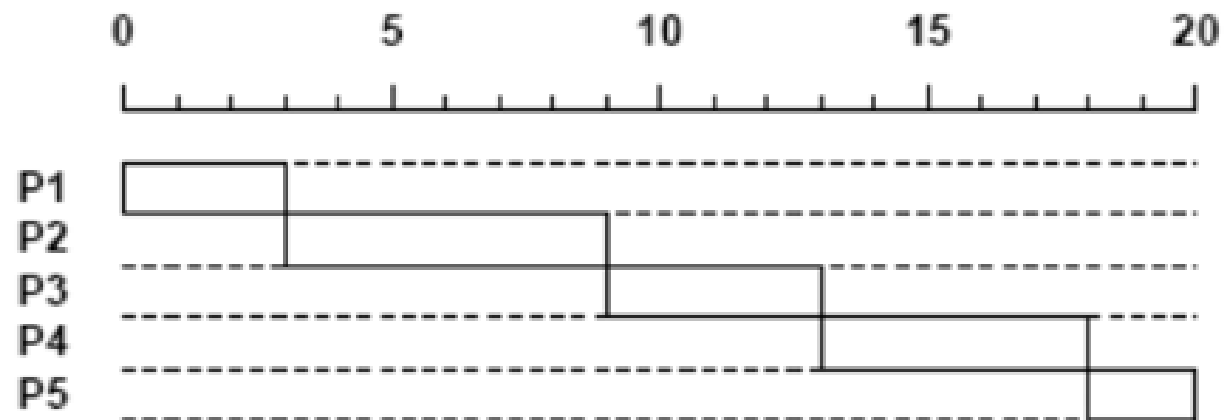
ESCALONADOR DE PROCESSOS

- Um escalonador deve possuir um algoritmo de escalonamento que se preocupe com 5 regras:
 1. **Justiça** – Todos processos devem ter acesso a CPU
 2. **Eficiência** – A meta do escalonador é chegar mais perto dos 100% de utilização da CPU
 3. **Minimizar o Tempo de Resposta**
 4. **Turnaround** – Minimiza os usuários batch
 5. **Throughput** – Maximizar o número de jobs processados

ALGORITMOS DE ESCALONAMENTO

- **Primeiro a chegar, primeiro a sair:**
 - A CPU é atribuída aos processos na ordem em que eles a requisitam. Basicamente há uma fila única de processos prontos.
 - Exemplo:

Processo	Tempo de chegada	Tempo de serviço
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

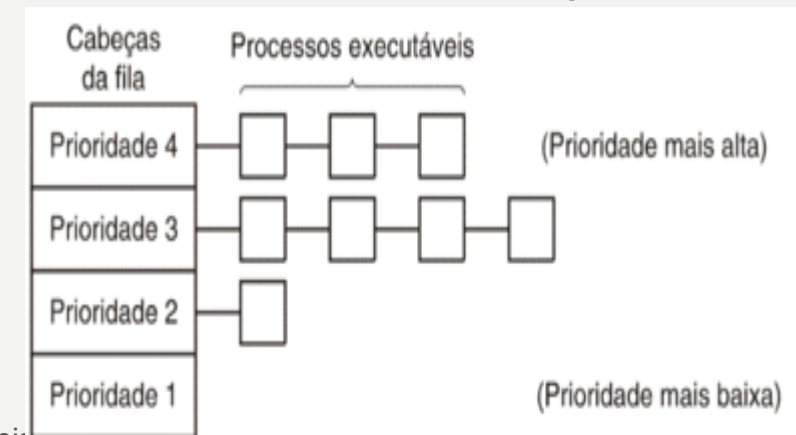


ALGORITMOS DE ESCALONAMENTO

- **Escalonamento por prioridade:**

- A cada processo é atribuído uma prioridade, e o processo com maior prioridade é executado primeiro.
- Para evitar que processos de alta prioridade executem infinitamente, o escalonador pode reduzir a prioridade do processo a cada execução do mesmo.

- As prioridades são atribuídas pelo S.O.



MULTIPROGRAMAÇÃO

- Um único programa não pode, em geral, manter a CPU ou os dispositivos de E/S ocupados por muito tempo – Justiça
 - Isso impossibilitaria a execução simultânea de programas
- O SO mantém várias tarefas na memória simultaneamente
- O SO escolhe uma das tarefas na memória para executar na CPU e pode trocar a tarefa atual por outra da memória

EXERCÍCIO 1

Com base no conhecimento adquirido na aula de hoje, elabore a sua própria definição de Sistema Operacional.

Não esqueça de, na sua definição, incluir as abordagens top-down (sentido usuários/hardware) e bottom-up (sentido hardware/usuários).

EXERCÍCIO 2

Em casa, leia as seções **1.3 Revisão sobre hardware de Computadores** e **1.5 Conceitos sobre Sistemas Operacionais** de (TANEMBAUM, 2010).

É importante a leitura dessas duas seções, pois o conhecimento nelas presentes serão muito úteis na próxima aula.



SEÇÃO 2

HISTÓRICO DOS SISTEMAS OPERACIONAIS

HISTÓRICO DOS SISTEMAS OPERACIONAIS

1940 - 1950

- Execução de programas requeria grande preparação com fitas, cartões perfurados e **usuário especializado**
- Computador executava apenas um programa por vez
- Programa era chamado de job
- Após execução do programa, fitas, cartões e outros componentes eram recolhidos para liberar a sala para o próximo usuário

HISTÓRICO DOS SISTEMAS OPERACIONAIS

1940 - 1950

- Os primeiros sistemas operacionais (SOs) tinham a função simples de suportar a separação de usuários e equipamentos
- Facilitavam a transição de um usuário para outro
- Um operador levava os equipamentos necessários para execução do job (programa) de um usuário e após a execução do job, os resultados eram recolhidos e levados pelo operador para o usuário

HISTÓRICO DOS SISTEMAS OPERACIONAIS

1960

- Computadores executam mais rapidamente, permitindo interatividade nos programas
- SOs passaram a suportar programas em tempo real
- Operadores passaram a ser um problema
 - Pois poderiam não conhecer detalhes e especificidades importantes para execução de um programa, como o que fazer de acordo com a mensagem de erro apresentada pelo programa, pois normalmente só o usuário sabia

HISTÓRICO DOS SISTEMAS OPERACIONAIS

1970

- Computadores muito caros e mais rápidos
 - Executar apenas um programa por vez não é suficiente
- SOs permitem execução de mais que um programa por vez
- O tempo de execução do processador é dividido igualmente entre os programas executados no computador (multiprogramação)

HISTÓRICO DOS SISTEMAS OPERACIONAIS

2000 em diante

- Computadores com processador de vários núcleos
- SO deve distribuir a execução de tarefas entre os núcleos para aumentar o desempenho do computador (balanceamento)
- SO pode dividir uma programa em partes menores para cada parte ser executada em um núcleo (escalonamento)

HISTÓRICO DOS SISTEMAS OPERACIONAIS

2000 em diante

- Computador realiza grande comunicação com outros computadores através das redes de computadores
 - SO gerencia a comunicação e os recursos que podem estar disponíveis através da Internet
- Surgimento de computadores mais simples para tarefas específicas (**sistemas embarcados ou embedded systems**)
 - SO de sistemas embarcados deve economizar bateria, executar programas com pouca ou sem interação humana, etc



SEÇÃO 3

EXEMPLOS DE SISTEMAS OPERACIONAIS

SISTEMA OPERACIONAL WINDOWS

- Mais usado em computadores pessoais
- Desenvolvido pela Microsoft
- SO comercial, ou seja, deve-se pagar para usá-lo
- Grande variedade e quantidade de software para as mais diversas atividades
- Devido a grande quantidade de usuários é muito visado por hackers
- Problemas de instabilidade e mal funcionamento

SISTEMA OPERACIONAL UNIX

- Sistema operacional desenvolvido na década de 1970
- Utilizado em equipamentos de médio porte e em equipamentos de arquitetura RISC
- Sistema multiusuário e multiprogramável

SISTEMA OPERACIONAL LINUX

- Sistema operacional similar ao UNIX
 - Criado por hackers como uma alternativa barata e funcional para quem não está disposto a pagar o alto preço de um sistema UNIX
- Suas vantagens:
 - Sistema multitarefa
 - Suporte a diversas linguagens
 - Estabilidade
 - Permissão de arquivo

SISTEMA OPERACIONAL LINUX

- SO gratuito sob a licença GNU GPL (software livre)
- Código fonte aberto
 - Ou seja, disponível para acesso, modificação e distribuição por qualquer um sob a licença GNU GPL
- Diversas distribuições
 - Como: Ubuntu, Debian, Mint, Slackware, Gentoo, etc
- Uma distribuição Linux é um SO que usa o kernel do Linux mas que possui softwares e configurações próprias

SISTEMA OPERACIONAL LINUX

- Estável: frequência baixa de problemas de execução
- Portátil: usado em vários tipos de dispositivos
- Boa segurança: problemas descobertos e solucionados pela comunidade de desenvolvedores
- Pouco visado por hackers
- Pequena variedade de software para algumas tarefas

SISTEMA OPERACIONAL FREEBSD

- Baseado no UNIX
- Kernel desenvolvido pela Universidade da Califórnia na cidade de Berkeley
- Mantido até hoje principalmente pela Universidade
- Difícil aprendizado para usuários pouco experientes com computadores

SISTEMA OPERACIONAL FREEBSD

- Usado principalmente em servidores
- Grande estabilidade
- Compatibilidade binária com o Linux,
 - Ou seja, programas executáveis do Linux funcionam com baixíssima perda de desempenho
- Variedade muito pequena de softwares

SISTEMA OPERACIONAL OS X

- Usado em computadores da Apple
- Desenvolvido pela Apple
- Segundo SO a usar GUI com janelas
- Primeiro a usar o conceito de ícones e desktop (área de trabalho)

SISTEMA OPERACIONAL OS X

- Baseado no FreeBSD
- Objetivo de simplificar e facilitar os comandos e a interação do usuário
- Baixa variedade de software para algumas tarefas, embora vários softwares importantes possuam versão para OS X

SISTEMA OPERACIONAL iOS

- Desenvolvido pela Apple
- SO móvel dos smartphones da Apple
- Suporte aos vários tipos de interação de dispositivos móveis (toque, acelerômetro, gestos, etc)
- Loja de aplicativos: App Store
 - Lucro com vendas de aplicativos, além do controle de qualidade e segurança dos aplicativos

SISTEMA OPERACIONAL iOS

- **Siri:** assistente pessoal
- Executa comandos feitos por voz como ligação de telefone, pesquisas web, envio de mensagens, etc
- Integração com o OS X e outros dispositivos Apple
- Variações para carros e smartwatches

SISTEMA OPERACIONAL ANDROID

- SO móvel mais utilizado no mundo
- Desenvolvido pela Google
- Mantido por algumas empresas lideradas pela Google
- Baseado no Linux
- Gratuito e de código fonte aberto
- **Mascote:** BugDroid

SISTEMA OPERACIONAL ANDROID

- Fabricantes de dispositivos podem mudar o Android para melhor adaptar-se ao seu produto
- Variações para smartTVs, carros, smartwatch e video games
- Loja própria: Google Play
 - Menor controle na qualidade dos aplicativos vendidos, mas com grande controle na segurança
- Grande maioria dos aplicativos gratuitos

SISTEMA OPERACIONAL ANDROID

- Grande variedade de aplicativos para diversos tipos de tarefas
- Muito visado por hackers
- Constantes problemas de segurança
 - Instalação de aplicativos a partir de lojas piratas de aplicativos

SISTEMA OPERACIONAL WINDOWS PHONE

- Desenvolvido pela Microsoft
- SO comercial
- Loja própria: Windows Phone Store
- Média variedade de aplicativos

SISTEMA OPERACIONAL WINDOWS PHONE

- Alguns aplicativos famosos não possuem ou fornecem baixo suporte ao Windows Phone
- **Assistente:** Cortana
- Recebe comando de voz para realizar várias tarefas
- Fadado ao fim
 - Já que a Nokia usará Android nos seus smartphones



SEÇÃO 4

COMPONENTES DO SISTEMA OPERACIONAL

SHELL

- Responsável pela comunicação do usuário com o sistema operacional
- Verifica se o comando do usuário está correto.
 - Quando existem erros no comando, mostra uma mensagem de erro
- Tipos mais comuns:
 - **Linha de comando:** terminal ou console, onde todos os comandos são textuais
 - **GUI (Graphical User Interface):** composta por botões, janelas, menus e outros elementos gráficos

KERNEL

- Núcleo, core ou kernel é a parte do SO responsável por todas as operações de controle do SO
- Executa os comandos enviados pelo shell
- Mais importante e mais utilizada
- Implementada em assembly
- Gerencia a execução dos programas, sistema de arquivos e a memória do computador



SEÇÃO 5

SEGURANÇA

SEGURANÇA

- SO responsável de várias formas pelo sistema computacional
 - Evitar que um arquivo se corrompa
 - Um arquivo está corrompido quando não é possível acessar os dados armazenados nele
 - Caso um arquivo seja corrompido, devem existir maneiras de recuperá-lo
 - Impossibilitar a perda de dados vitais para uma pessoa ou empresa

SEGURANÇA

- SO responsável de várias formas pelo sistema computacional
 - Impedir acesso não autorizado para arquivos e contas de usuários
 - Principal usuário de um sistema computacional: administrador ou super usuário
 - Administrador tem total controle e acesso para arquivos e configurações do sistema computacional
 - *Falhas de segurança do SO pode permitir que usuários não autorizados (hackers) acessem o sistema como administrador*

SEGURANÇA

- SO deve ser **constantemente atualizado para correção de bugs** (falhas de execução) ou problemas de segurança
- Software como **antivírus e firewalls** são importantes para a segurança do computador
 - Antivírus monitora os softwares em execução e arquivos copiados para o computador buscando códigos maliciosos (prejudiciais)
 - **Códigos maliciosos** são instruções que buscam roubar informações do computador ou ter acesso de administrador
 - Firewalls são softwares que monitoram a entrada e saída de informações do computador pela rede, barrando a entrada ou a saída de dados não autorizadas

SEGURANÇA

- A principal falha de segurança dos computadores são os próprios usuários
- Usuários por falta de conhecimento e realização de atividades ilegais instalam softwares com códigos maliciosos
- Criação de senhas previsíveis, simples e fáceis
 - **Exemplo de senhas frágeis:** datas pessoais importantes, nome de entes queridos, 123456, etc

EXERCÍCIO 3

1. Quais as principais características de um sistema operacional?
2. Cite exemplos de sistemas operacionais e apresente suas características
3. Quais as principais medidas de segurança adotada pelo SO?