

Software

Autora: Prof. Fernanda Vilas Bôas

Atualização: Prof. Vinicius C. Pinto

Software

- Parte lógica que permite utilizar os recursos computacionais para a resolução de problemas específicos.
 - Finalidades determinadas previamente.
 - Controlar o estoque da empresa
 - Monitorar a produção
 - Calcular os proventos/descontos da folha de pagamentos.
 - Criar um jogo de damas.

Software

- Um programa de computador é composto por uma seqüência de instruções, que é interpretada e executada pela UCP.

inteiro x, y , z

ler x

ler y

z = x + y

escrever z

fim

Software

- Tipos:
 - Aplicativo (Word, Oracle, DigiFolha)
 - Linguagem de Programação (C#, Java, Pascal, Basic)
 - De Sistema (Bios, S.O. [MS-Windows, Linux, Mac OS X], Drivers de dispositivos)

- Termos:
 - **User-friendly** – Fácil utilização pelo usuário.
 - **Freeware**: Gratuito, não é cobrada nenhuma taxa pela sua aquisição e utilização.
 - **Domínio público**: Não é protegido por direito autoral, pode ser usado ou alterado sem restrições.
 - **Open Source**: Software aberto, distribuição do código fonte, possibilitando alterações.
 - **Shareware**: software em que você pode copiá-lo, distribuí-lo sem restrições e usá-lo experimentalmente, por limitação de tempo ou funcionalidade.

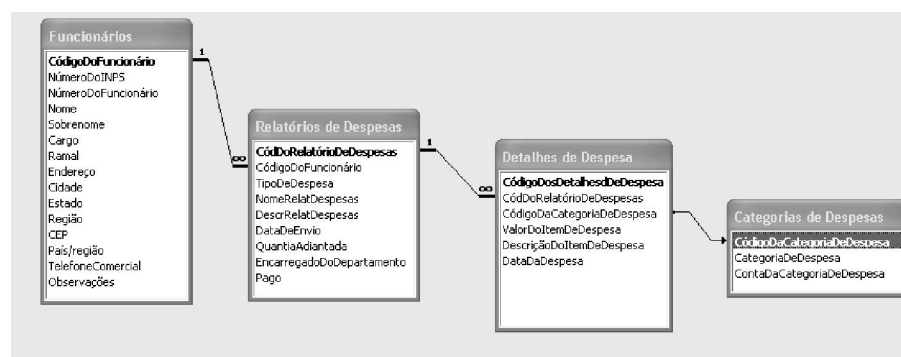
Aplicativo

- Aplicativos
 - Programa voltado para a solução de problemas do usuário.
- Categorias:
 - Uso (Aplicação) Geral
 - Podem ser utilizados em vários tipos de aplicações.
 - Ex: editores de texto, planilhas, gerenciadores de BD
 - Uso (Aplicação) Específico
 - Destinam exclusivamente a um único tipo de aplicação
 - Ex: folha de pagamento, imposto de renda

- Aplicativos:
 - Exemplos:
 - Editor de texto
 - Planilha eletrônica
 - Editor de slides
 - Editor de HTML
 - Navegador
 - Sistema gerenciador de banco de dados
 - **Controle de estoque**
 - **Folha de pagamento**
 - **Controle de matrícula**
 - **Identificador de código de barra**
 - ...

- Banco de dados
 - coleções organizadas e inter-relacionadas de dados armazenados em computadores digitais.
 - Modelos:
 - Modelo Hierárquico
 - Modelo em Redes
 - Modelo Relacional
 - Modelo Orientado a Objetos
 - *Modelo Relacional Estendido*

Banco de Dados Relacional



Fonte: Microsoft Access 2003

- **SGBD – Sistema Gerenciador de Banco de Dados**

- Programa de computador (*software*) que disponibiliza uma interface para que os usuários possam incluir, alterar ou consultar o banco de dados. O SGBD é um sistema complexo, responsável pela persistência, organização e recuperação dos dados.
- Exemplos:
 - *Firebird*, IBM DB2; MySQL, Oracle, PostgreSQL, SQL-Server, JADE, Microsoft Access, etc.

Utilitários

Utilitários

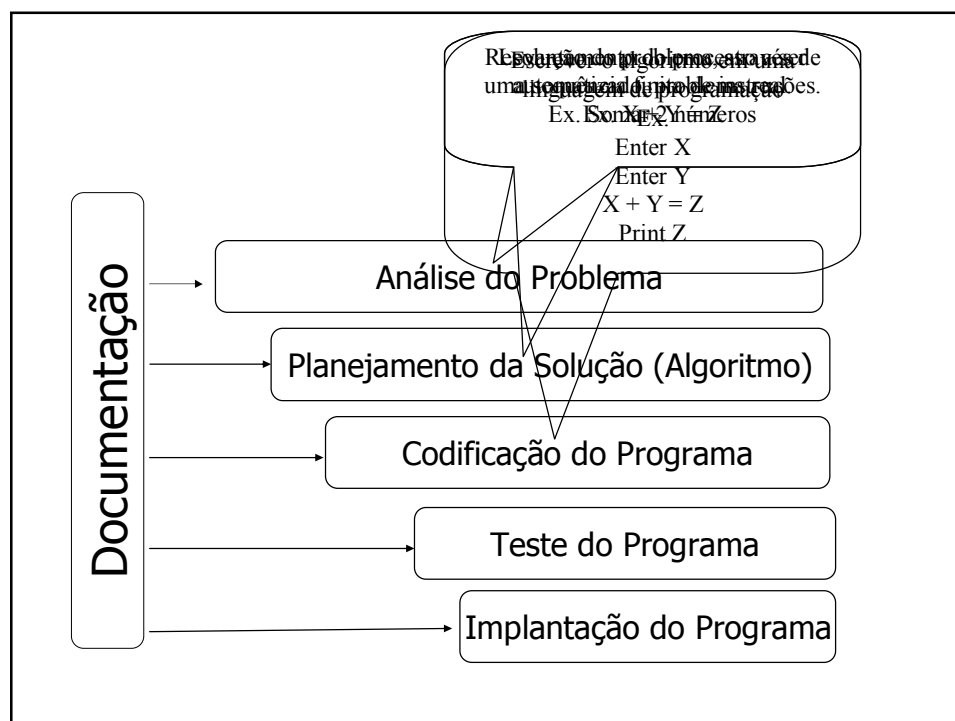


- Desfragmentador de disco
- Anti-vírus
- Compactador de arquivos
- Backup
- Restaurador do sistema
- Limpeza do sistema

Linguagem de Programação

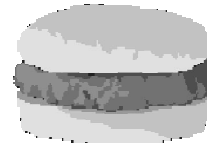
Linguagem de Programação

- Como pedir para o computador realizar alguma atividade e fazer com que ele entenda o que se deve realizar?
- Como podemos desenvolver programas?



Algoritmo: Criar um saboroso sanduíche

- 1º - Pegar um pão
- 2º - Cortar o pão ao meio
- 3º - Pegar a maionese
- 4º - Passar maionese nas fatias de pão
- 5º - Pegar alface e tomate
- 6º - Cortar alface e tomate
- 7º - Colocar alface e tomate no pão
- 8º - Pegar um hambúrguer
- 9º - Fritar o hambúrguer
- 10º - Colocar o hambúrguer no pão
- 11º - Juntar as fatias do pão cortado



Fluxograma

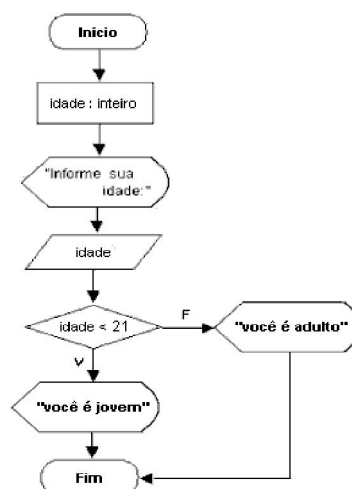
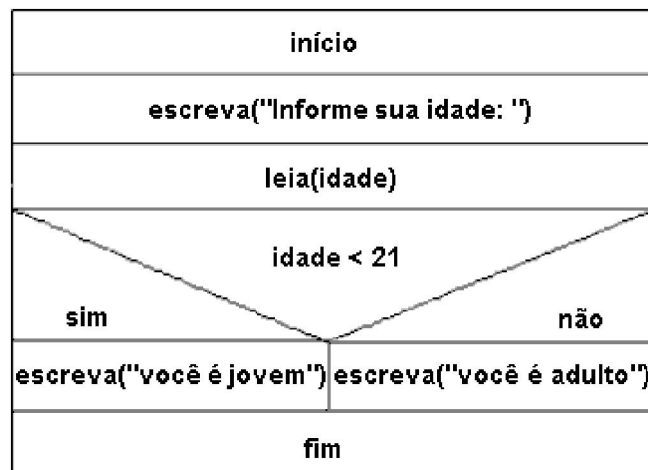
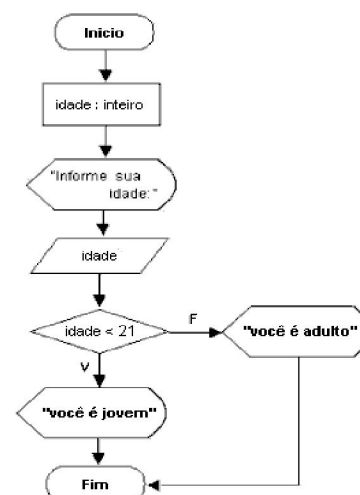


Diagrama de Chapin



Português Estruturado

```
algoritmo "avaliação médica"
// Síntese
  // Objetivo: analisar a idade de uma pessoa
  // Entrada: uma idade
  // Saída: indicação de jovem ou adulto
// Declarações
  var
    idade: inteiro
  inicio
    escreva ("Informe sua idade: ")
    leia (idade)
    se (idade < 21) entao
      escreva ("Você é jovem")
    senão
      escreva ("Você é adulto")
  fimse
finalgoritmo
```



Como podemos codificar um programa? Ou simplesmente, como programamos?



Através de uma linguagem de programação

Linguagem de programação

- O que é Linguagem de programação?
 - Método padronizado para expressar instruções para um computador.
 - Podemos desenvolver programas para informarmos ao computador o que queremos que seja feito através das LP.

Linguagem de programação

- Definições
 - Notação para escrever programas, através dos quais podemos nos comunicar com o hardware e dar as ordens adequadas para a realização de um determinado processo.
 - Conjunto de regras que providência a maneira de dizer a um computador que operações executar.
 - Conjunto de símbolos (vocabulário) e regras (gramática) que especifica como transmitir informações entre usuários e computador .

Linguagem de programação

- Como codificar (escrever) um programa para a execução através do SC, já que o “Hardware” só compreende binário?



Linguagem de programação

Linguagem de máquina	Significado
0010 0001 1110	carrega no registrador 1 o conteúdo da posição de memória 14
0010 0010 1111	carrega no registrador 2 o conteúdo da posição de memória 15
0001 0001 0010	soma o conteúdo do registrador 1 com o conteúdo do registrador 2 e coloca no registrador 1
0011 0001 1111	armazena o conteúdo do registrador 1 na posição de memória 15

Linguagem de programação

- Linguagem de máquina:
 - Única linguagem que o computador entende
 - As instruções desta linguagem estão sob a forma de 0s e 1s.
 - Quando um programa está nesta linguagem, ele está pronto para ser executado.
 - Principais desvantagens:
 - A programação é trabalhosa, complexa e exige um alto nível de detalhamento
 - Própria para cada processador

Linguagem de máquina

```
MZÀ$PÿvèŠÿ]Ë3ÀP,F ë fF, <
uè2Àëä Àt Ba
Àu C†à2Àüã ¬l,"t " < \u € < "u ¬l öÃ
é îY Ê.Ž t % " C Û Û ô ì + ërâ â %
.- OE ~ ã %v, vüÿv
ÿv è Å fÄ ÿvpÿvü è üêYY V F ë
Rÿvpÿvü è W í fÄ â ]ËU ì f iHVW < ~ F
%Fp Àu ' Í ! '3Àé • Š ~ Ø Š ‡ ĩn
```

Código de máquina visto através de um editor de texto

Linguagem de programação

•Linguagem de máquina:

```
C:\Utility>debug v.exe
-d 0 100
0E3D:0000 CD 20 FF 9F 00 9A F0 FE-1D F0 4F 03 F0 07 8A 03 . . . . .O. . . .
0E3D:0010 F0 07 17 03 F0 07 DF 07-01 01 01 00 02 FF FF FF . . . . .
0E3D:0020 FF FF FF FF FF FF FF FF-FF FF FF FF BD 0D 4C 01 . . . . .L.
0E3D:0030 D0 0C 14 00 18 00 3D 0E-FF FF FF FF 00 00 00 00 . . . . = . . . .
0E3D:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 . . . . .
0E3D:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 . ! . . . .
0E3D:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 . . . . .
0E3D:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 . . . . .
0E3D:0080 00 0D 76 2E 65 78 65 0D-44 4F 57 53 5C 73 79 73 ..v.exe.DOS\sys
0E3D:0090 74 65 6D 33 32 5C 64 6F-73 78 0D 0D 64 61 20 72 tem32\dosx..da r
0E3D:00A0 65 64 65 20 28 63 61 72-72 65 67 61 72 20 61 6E ede (carregar an
0E3D:00B0 74 65 73 20 64 6F 20 64-6F 73 78 2E 65 78 65 29 tes do dosx.exe)
```

Código de máquina visto através de um editor binário (apresentação em Hexadecimal).

Linguagem de programação

- Linguagem Intermediária ou Linguagem **Assembly**
- Simplificação da linguagem de máquina
 - Minimizar as dificuldades de programação em notação binária.
 - Manipulada com mais facilidade pelos programadores por que faz uso de códigos ao invés da notação binária.
- Principais desvantagens:
 - A programação continua sendo trabalhosa e exigindo um alto nível de detalhamento
 - Própria para cada processador

Linguagem de programação

Linguagem de Máquina	Significado
0010 0001 1110	carrega no registrador 1 o conteúdo da posição de memória 14
0010 0010 1111	carrega no registrador 2 o conteúdo da posição de memória 15
0001 0001 0010	soma o conteúdo do registrador 1 com o conteúdo do registrador 2 e coloca no registrador 1
0011 0001 1111	armazena o conteúdo do registrador 1 na posição de memória 15

Linguagem de Máquina	Linguagem de Baixo Nível
0010 0001 1110	LOAD R1, val1
0010 0010 1111	LOAD R2, val2
0001 0001 0010	ADD R1, R2
0011 0001 1111	STORE R1, val2

Linguagem de Alto Nível

- Linguagem simbólica
 - Utiliza combinação de um conjunto de símbolos de acordo com certas regras de sintaxe para expressar uma seqüência de operações de máquina
- Objetivos
 - Independência em relação ao processador
 - Afasta o programador da linguagem de máquina
 - Aproximação da linguagem humana
 - Redução de erros – utilização de palavras ao invés de caracteres
 - Definição de rotinas de uso freqüente
 - Reutilização de código
- O usuário pode dizer simplesmente o que deve ser feito sem ter que dizer como deve ser feito

Linguagem de programação

Linguagem de Máquina	Linguagem de Baixo Nível	Linguagem de Alto Nível
0010 0001 1110	LOAD R1, val1	val3 = val1 + val2
0010 0010 1111	LOAD R2, val2	
0001 0001 0010	ADD R1, R2	
0100 0001 1111	STORE R1, val3	

Linguagem de programação

- Fortran: Criada por volta de 1955
 - Desenvolvida para aplicações científicas
 - Potência em cálculos matemáticos
 - Limitada em manipulação de arquivos e elaboração de relatórios
- COBOL: Criada por volta de 1960
 - Desenvolvida para aplicações comerciais
 - Potência em manipulação de arquivos e elaboração de relatórios
 - Limitada em cálculos matemáticos

Linguagem de programação

- Pascal: Criada por volta de 1970
 - Desenvolvida para proporcionar uma linguagem adequada para o ensino dos conceitos e técnicas de programação
- **C: Criada por volta de 1972**
 - Desenvolvida para escrever o UNIX
 - Características de alto e de baixo nível
 - Permite um alto grau de eficiência sem exigir que o programador se preocupe com a arquitetura do equipamento

Linguagem de programação

- Visual Basic: Criada por volta de 1991
 - 1ª linguagem com características gráficas do Windows
 - As facilidades da GUI puderam ser aproveitadas
- Java: Criada por volta de 1995
 - Independente de plataforma
 - Permite a construção de Applets
 - Programas que rodam em navegadores internet
 - Aumentam a funcionalidade das páginas Web

Linguagem de programação

Mas, se os computadores trabalham internamente com a linguagem de máquina, como é que podemos fazer com que o computador entenda os programas desenvolvidos em linguagem de baixo ou de alto nível?



Utilizando TRADUTORES

Linguagem de programação

Tradutores

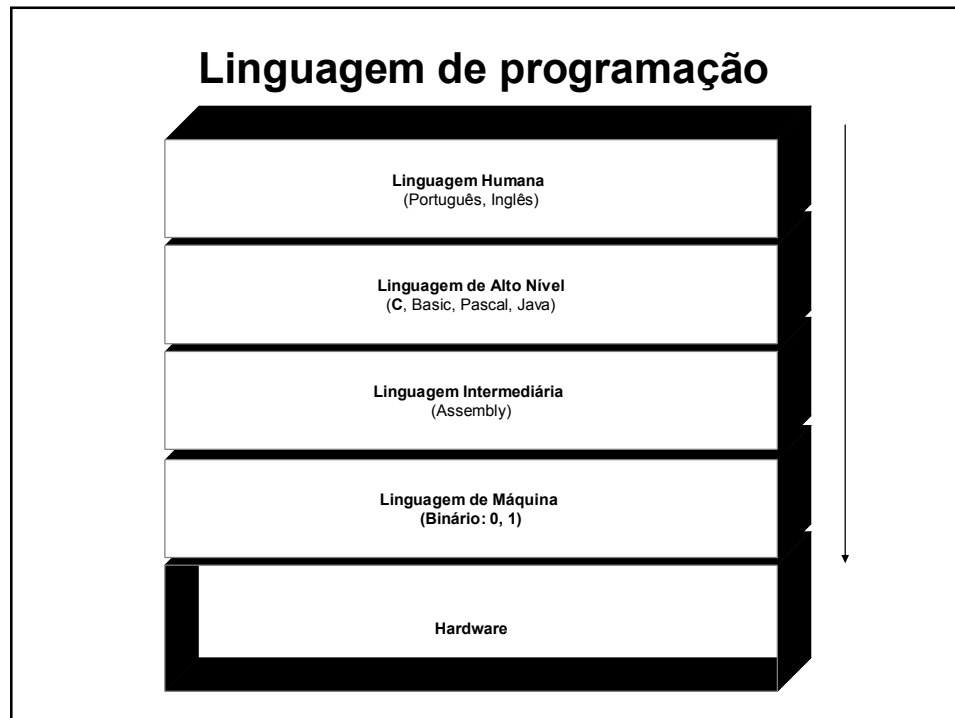
- Lêem um programa em uma linguagem em “Assembly” ou alto nível e o traduzem para a linguagem de máquina
- Tipos:
 - Compilador e Interpretador
 - Lê um programa em LAN e o traduz para LM; Cada instrução na LAN pode ser traduzida para um conjunto de instrução na LM.
 - Montador (ou Assembler)
 - Lê um programa em Assembly (LI) e o traduz para LM; Cada instrução na LI é traduzida para uma instrução na LM.

Linguagem de Programação

COMANDO	LINGUAGEM
writeln	Pascal
printf	C
System.out.println	Java

Conversão

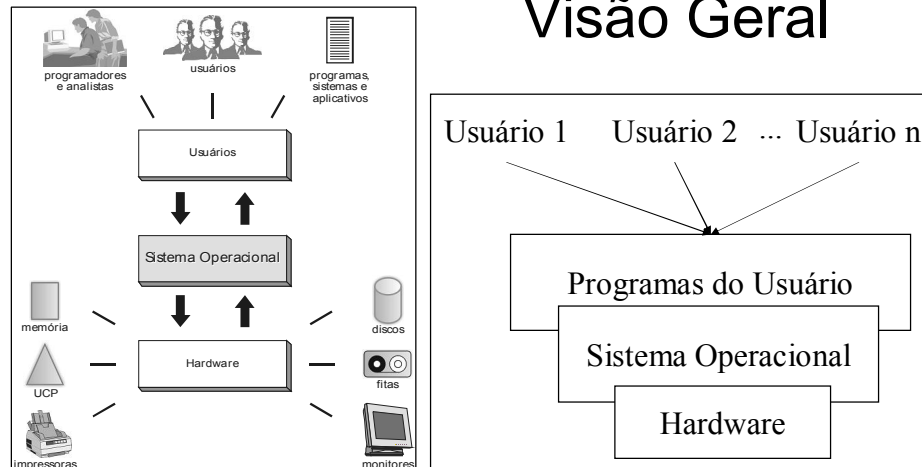
1100101010010010001111



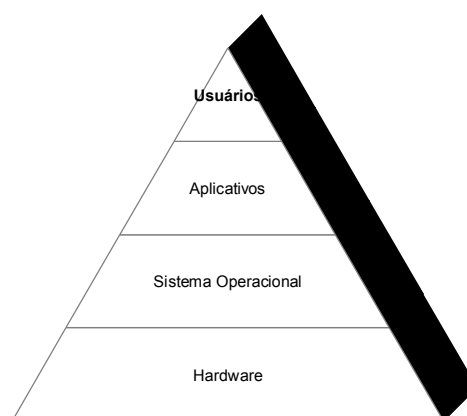
Sistema Operacional

- Também chamado de sistema básico.
- Contém um conjunto de programas necessários ao funcionamento do computador
 - É através do SO que o usuário pode se comunicar com o computador

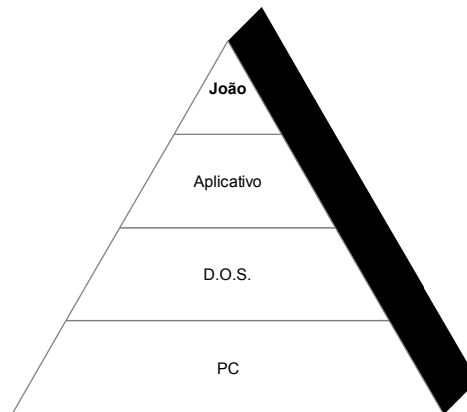
Sistema Operacional Visão Geral



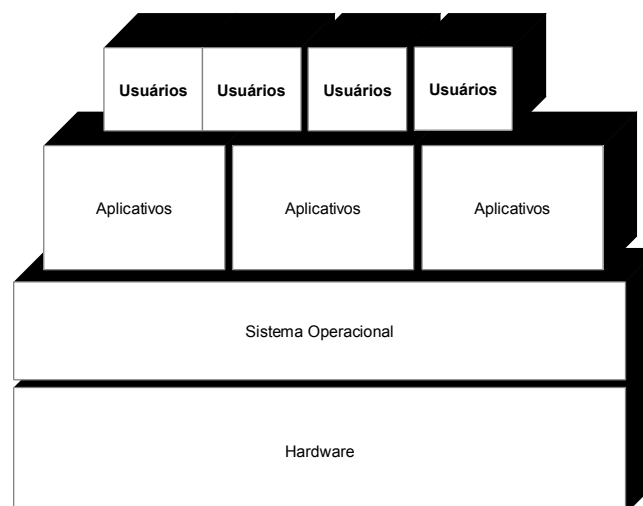
Sistema Operacional



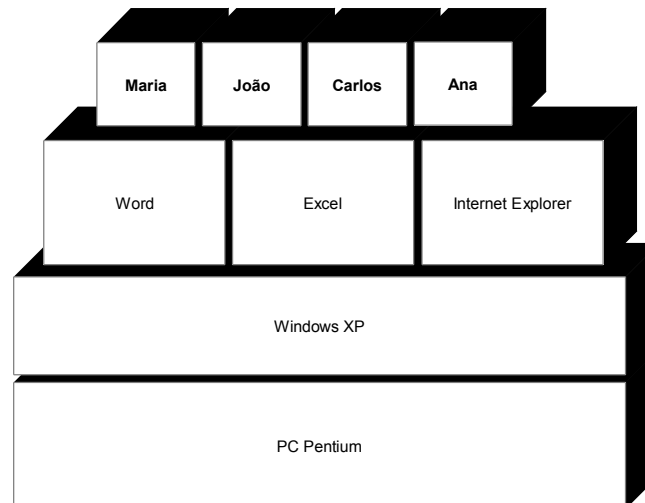
Sistema Operacional



Sistemas Operacionais – Multiusuário e Multitarefa



Sistemas Operacionais – Multiusuário e Multitarefa



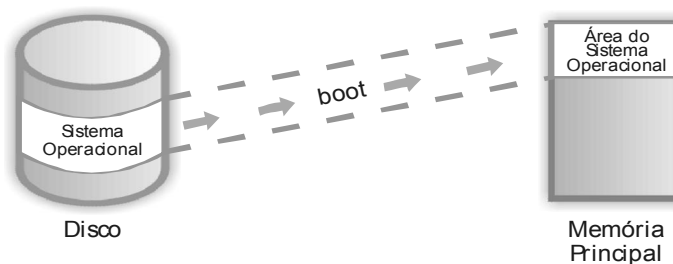
Ao ligar o computador...

- Autoteste
 - Identifica memória, discos, teclado, sistema de vídeo e qualquer dispositivo ligado a ele.
- Boot (BIOS)
 - Iniciação do sistema.(POST – Power On Self Test)
 - Procura o SO a ser iniciado
 - Objetivo
 - Colocar o computador em condições de funcionamento
 - Carregar o computador com rotinas essenciais ao atendimento dos diversos programas de aplicação que lhe serão posteriormente submetidos.

BIOS (Basic I/O Service)

- O código de bips varia de acordo com a marca do BIOS (Award ou AMI, por exemplo) podendo também haver pequenas mudanças de uma placa mãe para outra. Geralmente, o manual da placa mãe traz uma tabela com as seqüências de bips usadas. As instruções a seguir servem como referência:
 - 1 bip curto – POST Normal - sistema esta ok.
 - 2 bips curto – erro no POST – um código de erro é mostrado na tela.
 - Não bip – fonte de alimentação, sistema com problema, CPU desconectada, ou som desconectado.
 - Bip Contínuo – fonte de alimentação, sistema, ou problema no teclado.
 - Repetidos bips curtos – fonte de alimentação ou problema no sistema ou no teclado.
 - 1 longo bip, 1curto bip – sistema com problema.
 - 1 bip longo, 2 bips curtos – adaptador de vídeo com problema;
 - 3 Bips longos: Falha nos primeiros 64 KB da memória RAM (Base 64k memory failure) Foi detectado um problema grave nos primeiros 64 KB da memória RAM.
 - 4 Bips Longos: Timer não operacional: O Timer 1 não está operacional ou não está conseguindo encontrar a memória RAM.
 - 5 Bips: Erro no processador.
 - 7 Bips: Processor exception (interrupt error): o processador gera uma interrupção
 - 8 Bips: Erro na memória da placa de vídeo (display memory error).
 - 9 Bips: Erro na memória ROM (ROM checksum error).

Ao ligar o computador...



- ... as instruções internas orientam-no para encontrar e executar o SO
- O SO opera todo o tempo em que a máquina estiver ligada

Sistema Operacional

- Exemplos de atividades realizadas:
 - Permite que os programas armazenem e recuperem informações.
 - Esconde dos programas os detalhes específicos de hardware.
 - Permite que programas sejam executados sem a interferência de outros programas.
 - Controla o fluxo de dados entre os componentes de um computador.
 - Responde às solicitações dos usuários.
 - ...

Justificativa do S.O.

- Porque o uso de uma camada intermediária de software chamada de SO?
 - Escrever programas para controlar discos, leitores de CD/DVD e outras partes do hardware é bastante complexo.
 - Fazer todos as partes do hardware trabalharem de forma integrada e otimizada é mais complexo ainda.
 - Para afastar o desenvolvedor da camada do hardware (Aumento de produtividade).

Conceitos Básicos sobre S.O.

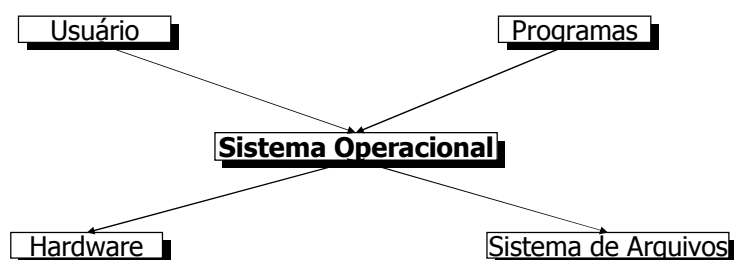
- Processo
 - Também chamado de Tarefa
 - Programa em execução e o seu contexto (código executável, arquivos de dados, bibliotecas de funções sendo utilizados, etc.)
 - Entidade ativa que compete pelos recursos do sistema e interage com outros processos
- Kernel - Núcleo do SO
 - Controla suas funções centrais, contém código de máquina de baixo nível que controla o gerenciamento de hardware, permanece na memória o tempo todo.

Conceitos Básicos sobre S.O.

- Shell - Interpretador de comandos
 - Meio de interação entre o usuário e o computador .
 - Assume o controle da tela, recebe as inserções do usuário, as interpreta e as executa.
 - Exemplos de serviços: login/logout, manipulação de arquivos, principal interface com o KERNEL, entre outros.

Tarefas Básicas do S.O.

- I. Proporcionar interface usuário/computador
- II. Gerenciar dispositivos de hardware
- III. Gerenciar e manter sistemas de arquivos em disco
- IV. Dar suporte a todas os processos em execução.



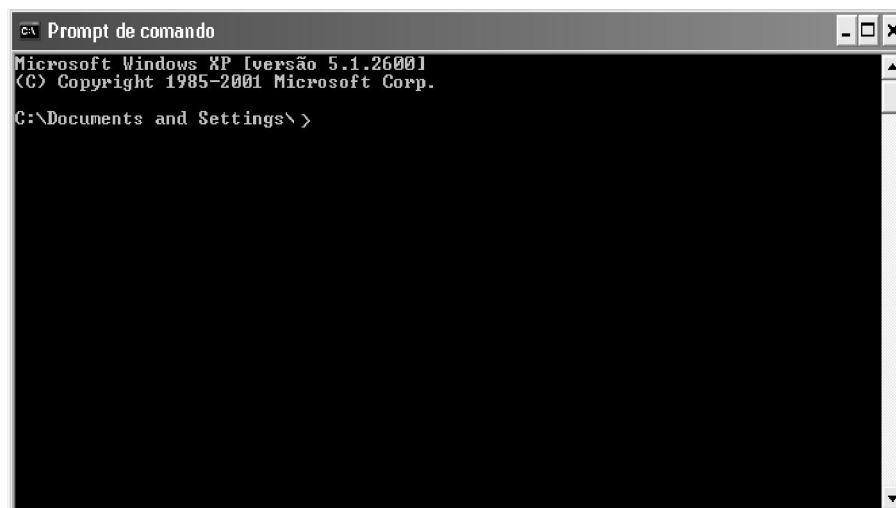
Sistema Operacional

- Interface de linha de comando
 - Palavras e símbolos são digitados no teclado do computador
- Interface Gráfica
 - Uso do mouse para selecionar ações, cliques em ícones.

Interface Linha de Comando

- Ex: DOS
- Usuário controla a execução de programas digitando comandos no aviso de comando (prompt)
 - Letra da unidade de disco atual seguida do sinal maior que (C>)
 - Pode exibir também o diretório atual
- O aviso de comando indica que o SO está pronto para receber comandos
- Digitado incorretamente o comando o SO retorna uma mensagem de erro

Interface de linha de comando



Interface Gráfica

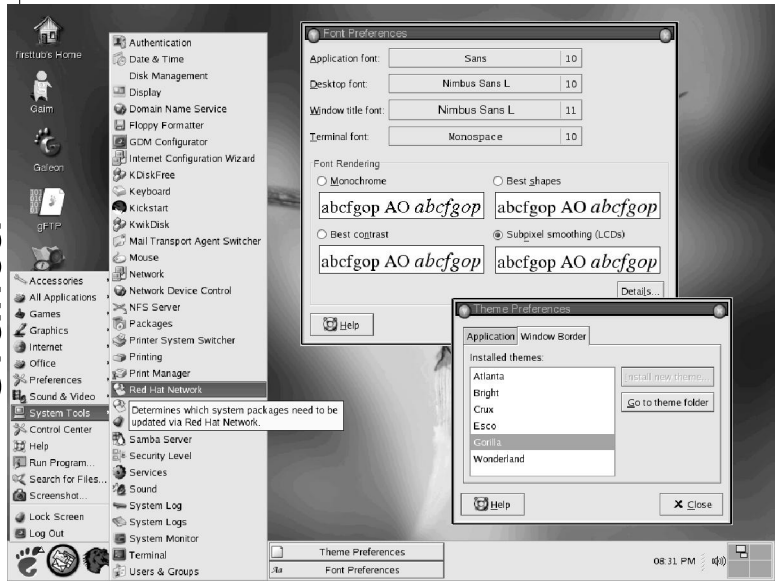
- Usa ícones, menus e janelas para acessar programas, discos e executar comandos do sistema
 - Windows 98/NT, MacOS e OS/2 Warp.
- Alguns GUIs não são sistemas operacionais propriamente mas são extensões de um sistema operacional com uma interface de linha de comando
 - Windows 3.1 para DOS e X-Windows para Unix
 - Amigável (simples para inexperientes e sofisticada para experientes)

Exemplo de Interface Gráfica



Microsoft Windows XP

Exemplo de Interface
Gráfica



Linux Red Hat Limbo

Exemplo de Interface
Gráfica



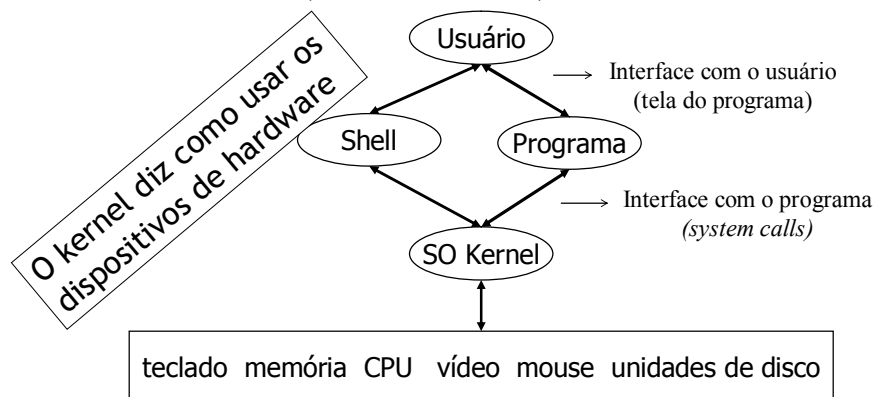
Mac OS X

Exemplo de Interface Gráfica



II. Gerenciar Dispositivos de Hardware

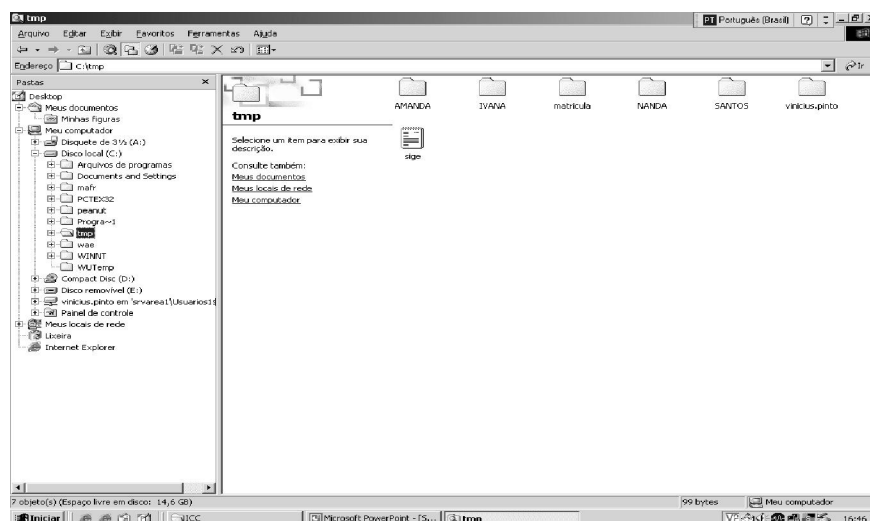
- Modo como o hardware é usado
- O SO é intermediário entre programas e hardware
 - Uso de memória, unidades de disco, ...

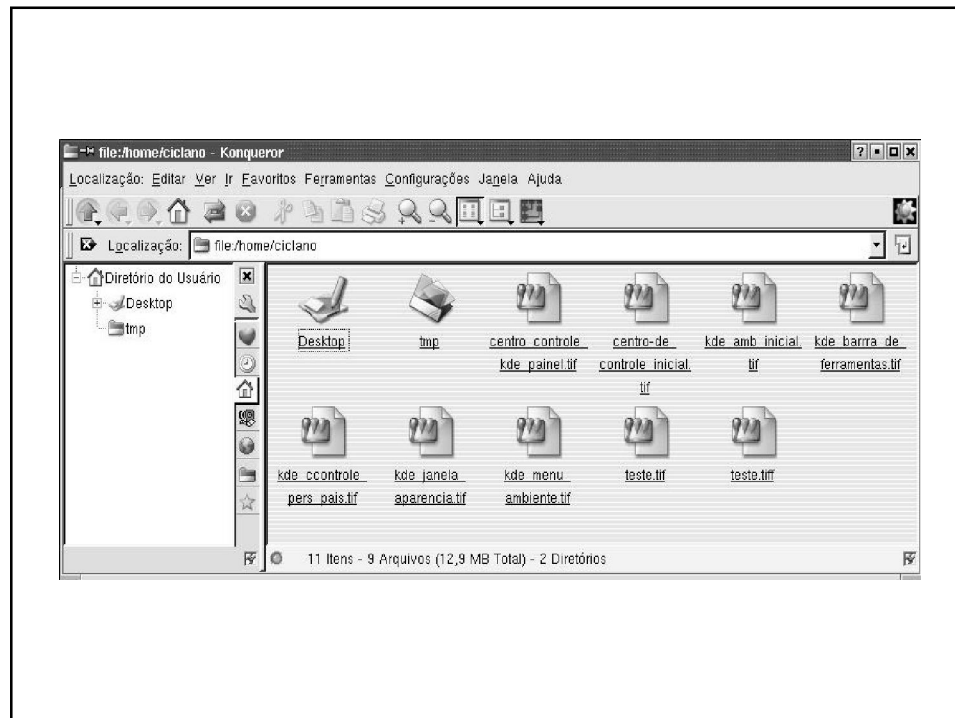


II. Gerenciando Dispositivos de Hardware

- Ex: Digitar comando para listar arquivos ou dar um clique em uma pasta
 - O SO interpreta a ação
 - A UCP é interrompida pelo kernel e instruída a ir até a unidade de disco especificada e recuperar os nomes dos arquivos que forem encontrados.
 - O SO intercepta o fluxo de dados (nomes dos arquivos) que está retornando e exibe-o na tela.

Peter Norton pg. 249





III. Gerenciar e manter os sistemas de arquivos em disco

- Agrupa os dados em compartimentos lógicos para armazená-los em disco
 - Chamados de Arquivos
 - Os arquivos podem conter
 - instruções de programas; ou
 - dados criados ou usados por um programa
 - Alguns programas permitem que o usuário crie seu próprio conjunto de dados e armazene-os como arquivo nomeado
- O SO mantém a lista de arquivos contidos em disco

III. Gerenciar e manter os sistemas de arquivos em disco

- Nomenclatura básica:
 - Diretório/Pasta: Forma de agrupar um conjunto de arquivos sob um nome para organizá-los.
 - Subdiretórios/Subpastas: grupos menores.
 - Diretório raiz/Pasta raiz/Pasta da unidade
 - Árvore que se ramifica a partir do primeiro diretório
 - Hierarquia de diretórios

Referências Bibliográficas

- Aplicativos e Linguagens de programação
 - Fernando Velloso
 - Capítulos 5 e 7
 - Alcalde
 - Capítulos 5 e 7
 - Norton
 - Capítulos 9, 10, 11, 12 e 13
- Sistemas Operacionais
 - Tanenbaum
 - Capítulo 1
 - Norton
 - Capítulo 8