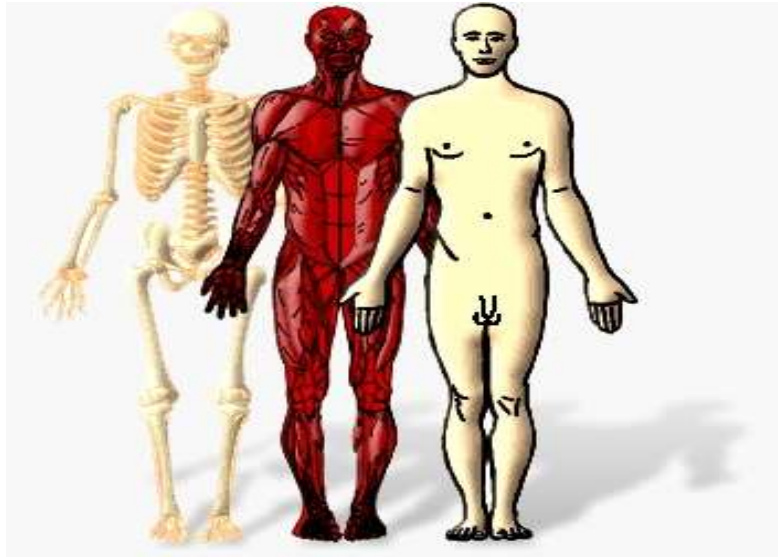
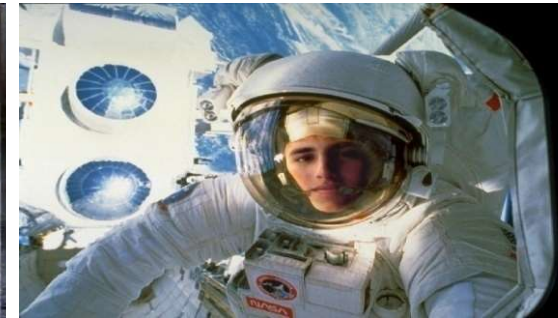


Introdução

Hardware X Software



Corpo Humano
Parte Física
Componentes



Capacidade de utilizar o corpo em atividades específicas explorando seus componentes

Introdução

Hardware

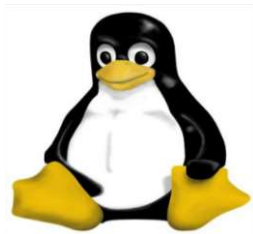
- Parte física: placas, periféricos, cabos e etc.
- Sozinho, não serve para nada.



Introdução

Software (programa)

Parte Lógica, em outras palavras, constitui um conjunto de instruções que explora o hardware visando a geração de resultados ou a execução de uma tarefa.



Linux



Telegram

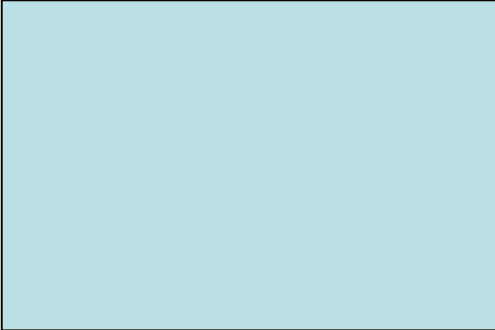


chrome



Instagram





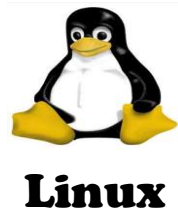
Introdução – Parte II

Software Aplicativo e Sistema Operacional

Introdução

Podemos classificar os softwares em:

➤ Sistemas operacionais



➤ Aplicativos



Introdução

Analogia:



Introdução

Analogia:

- ✦ A cozinha é o hardware (H)
- ✦ Ana Maria Braga é o Software Aplicativo (SA)
- ✦ A Maria é o Sistema Operacional (SO)



Introdução

Sistemas operacionais

- Softwares de sistema, que gerenciam a própria operação do computador;
- Controla os recursos do computador e fornece a base para os softwares de aplicação serem escritos;
- Escrever programas para controlar discos, leitores de CD/DVD e outras partes do hardware é demasiadamente complexo. Fazer todas as partes do hardware trabalharem de forma integrada e otimizada é mais complexo ainda;
- Para afastar o desenvolvedor (programador) da camada do hardware foi criada uma camada intermediária de software denominada S.O..

Softwares Aplicativos

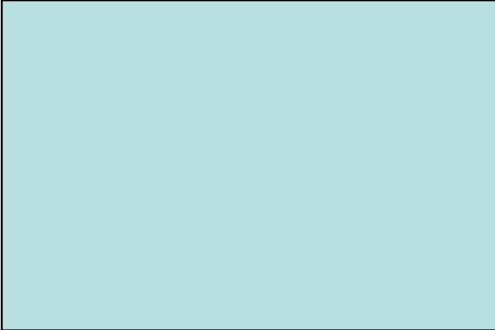
- Softwares de aplicação, que resolvem problemas específicos e/ou efetuam tarefas para os usuários.

Introdução

Organização do software de computadores

- Sendo assim o S.O. é um programa que se situa entre os softwares aplicativos e o hardware:
 - Gerenciando os recursos do computador (UCP, dispositivos periféricos, etc.).
 - Estabelece uma interface com o usuário.
 - Determina como o usuário interage com o hardware.
 - Provê e executa serviços para softwares aplicativos.





Introdução – Parte III
Classificação do software de computadores
quanto à propriedade intelectual

Software Proprietário

- Software garantido por copyright (direito autoral).
- Sua cópia, redistribuição ou modificação são de alguma forma restritos pelo seu criador ou distribuidor.

Introdução

Shareware

- Software garantido por copyright, distribuído gratuitamente para um período experimental.
- Se você gostar dele, deve registrá-lo, pagando uma taxa.
 - A maioria dos autores acrescenta documentação, ampliações, suporte e atualizações gratuitas para estimular as pessoas a fazer o registro.

Introdução

Freeware

- O autor fornece-o gratuitamente para ser usado por todos.
- O autor detém o copyright.
- Ele pode impor restrições ao uso.
- O software não pode ser alterado ou redistribuído sem permissão.

Introdução

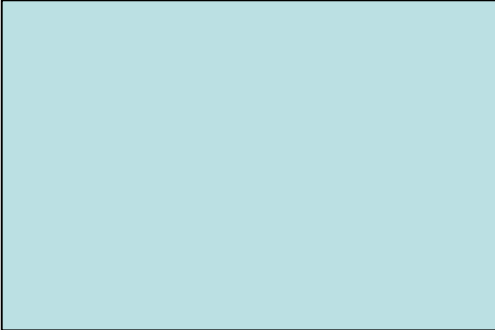
Software de fonte aberta

- Uma variação de freeware.
- O código-fonte é disponibilizado.
 - Código-fonte: legível por seres humanos.
- Outros programadores podem modificar e redistribuir os programas livremente.
 - Ajuda a identificar erros (*bugs*) e a criar melhorias mais facilmente.

Introdução

Domínio Público

- Software que não possui copyright.
- Pode ser usado e até mesmo alterado sem permissão.
- Tipicamente, é desenvolvido por universidades e/ou instituições de pesquisa, normalmente usando subsídios governamentais.



Linguagens de Programação – Parte I
Conceito e classificação com relação à
sua similaridade com a linguagem humana

Linguagens de Programação

Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador. Cada linguagem possui um conjunto único de palavras-chaves (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções dos programas.

Os programas de computador podem ser escritos em várias linguagens de programação, algumas diretamente compreensíveis pelo computador e outras que exigem passos de tradução intermediária.

Linguagens de Programação

As linguagens de programação podem ser divididas em três tipos, com relação à sua similaridade com a linguagem humana:

- Linguagem de máquina;
- Linguagem simbólica;
- Linguagem de alto nível.

Linguagens de Programação

Linguagem de máquina: é a linguagem de mais baixo nível de entendimento pelo ser humano e a única, na verdade, entendida pelo processador (UCP).

É constituída inteiramente de números (0's e 1's), o que torna praticamente impossível entendê-la diretamente. Cada UCP tem seu conjunto único de instruções que definem sua linguagem de máquina, estabelecido pelo fabricante do chip.

Linguagens de Programação

Uma instrução típica em linguagem de máquina seria algo como:

0100 1111 1010

Essa linguagem é também classificada como uma linguagem de primeira geração.

Linguagens de Programação

Linguagem simbólica: é a linguagem de nível imediatamente acima da linguagem de máquina.

Ela possui a mesma estrutura e conjunto de instruções que a linguagem de máquina.

Porém, permite que o programador utilize nomes (chamados mnemônicos) e símbolos em lugar de números.

Linguagens de Programação

A linguagem simbólica é também única para cada tipo de UCP, de forma que um programa escrito em linguagem simbólica para uma UCP poderá não ser executado em outra UCP de uma família diferente.

Nos primórdios da programação os programas eram escritos em linguagem simbólica.

Linguagens de Programação

Hoje a linguagem simbólica, é utilizada quando a velocidade de execução ou o tamanho do programa executável gerado são essenciais.

A conversão da linguagem simbólica para a linguagem de máquina se chama montagem, e é feita por um programa chamado montador (ou assembler).

Linguagens de Programação

Uma típica instrução em linguagem simbólica seria:

ADD A, B

Essa linguagem é também classificada como linguagem de segunda geração, e, assim como a linguagem de máquina, é considerada uma linguagem de baixo nível.

Linguagens de Programação

Linguagem de alto nível: São as linguagens de programação que possuem uma estrutura e palavras-chave que são mais próximas da linguagem humana.

Tornando os programas mais fáceis de serem lidos e escritos.

Esta é a sua principal vantagem sobre as linguagens de nível mais baixo.

Linguagens de Programação

Os programas escritos nessas linguagens são convertidos para a linguagem de baixo nível através de um programa denominado compilador ou de um interpretador.

Uma instrução típica de uma linguagem de alto nível é:

```
if (A>10) then A:=A-7;
```

Linguagens de Programação

Linguagem de alto nível

Linguagem simbólica

Linguagem de máquina

hardware

+

**Similaridade
com a linguagem
humana**

-



Linguagens de Programação – Parte II

Características Básicas

Linguagens de Programação

Características básicas

➤ Legibilidade

- facilidade com que os programas podem ser lidos e entendidos.

➤ Portabilidade

- está relacionada a possibilidade de se utilizar o código fonte em diversas arquiteturas.

Linguagens de Programação

Características básicas

➤ Confiabilidade

- Diz-se que um linguagem de programação é confiável se um programa construído com esta se comporta de acordo com suas especificações sob todas as condições.

➤ Segurança

- Está relacionada com a confiabilidade. Porém, associada com a proteção/integridade dos dados manipulados.

Linguagens de Programação

Características básicas

- Definição e uso de abstrações
 - abstração é a capacidade de definir e, depois, de usar estruturas ou operações complicadas de uma maneira que permita ignorar muitos detalhes.
- Expressividade
 - significa que uma linguagem tem formas relativamente convenientes, em vez de desajeitadas, de especificar computações.

Linguagens de Programação

Características básicas

➤ Flexibilidade

- capacidade de utilizar suas características em diversos contextos.

➤ Custo

- treinamento – capacitação dos programadores
- desenvolvimento – escrita e compilação do programa
- execução – verificação de tipos e interpretação

Linguagens de Programação

Características básicas

- Custo (continuação)
 - manutenção – capacidade de adaptar o programa existente
- Desempenho
 - relacionado com a otimização (compilação x execução)