

Introdução a Computação

Aula 3/SOFTWARE/SI

Prof^a. Ma. Camilla Côrtes Carvalho Heitor



SOFTWARE

- É a parte lógica do computador.
- Um conjunto de instruções que deve ser executado pelo computador, a fim de realizar uma tarefa qualquer.



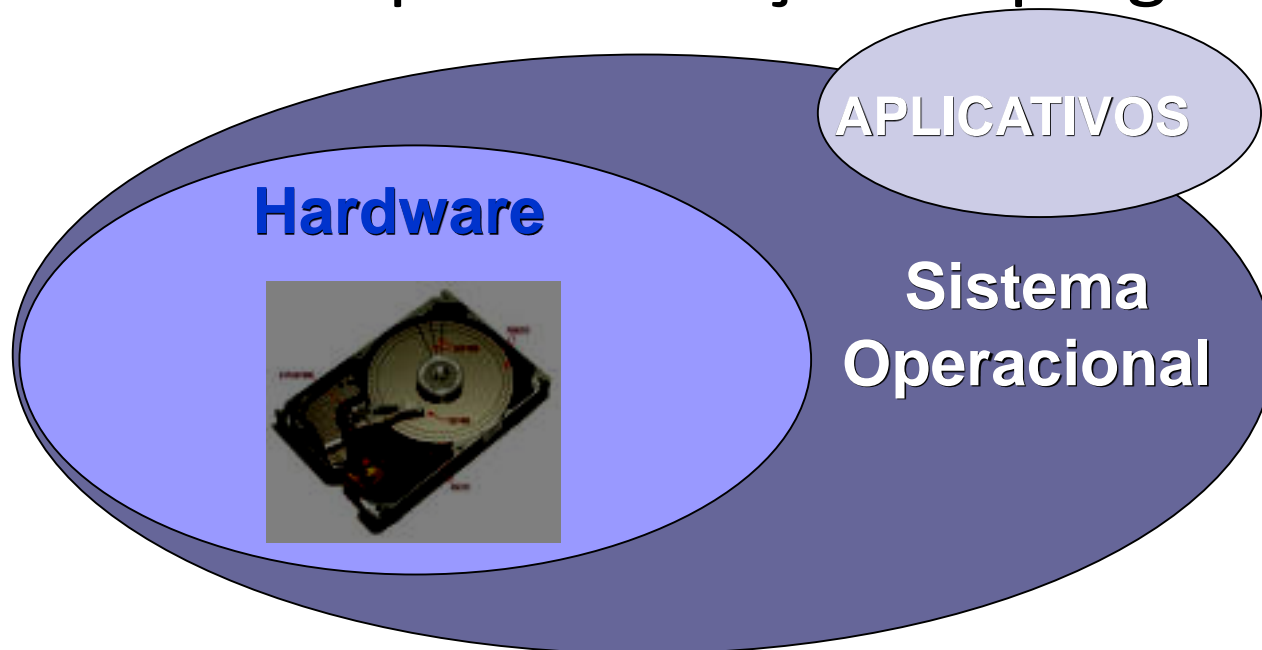
Tipos de softwares

- Software Básicos
- Software Aplicativos
- Software Utilitários



O que é um sistema operacional ?

- Software básico, um programa que atua como intermediário entre o usuário e o hardware de um computador com propósito de fornecer um ambiente para execução de programas.



Sistemas Operacionais (SO)

- Programa especial que gerencia as funções básicas do computador.
- Uma coleção de programas (softwares) que inicializam o hardware do computador.
- Fornece gerência de rotinas básicas para controles e dispositivos.
- Proporciona interação entre o hardware e software.
- Mantem integridade do sistema.

7 Serviços do SO

- Execução de programas;
- Manipulação de Sistemas de arquivos;
- Comunicação;
- Detecção de Erros;
- Alocação de Recursos;
- Contabilização;
- Proteção.

Serviços do SO

- **Execução de programas:**

capacidade do sistema de carregar um programa na memória e executá-lo.

- **Manipulação de Sistemas de arquivos:**

os programas necessitam ler, gravar, criar e remover arquivos.

- **Comunicação:**

troca de informações entre processos que estão sendo executados em um mesmo computador ou em computadores diferentes interligados por uma rede. Implementado via memória compartilhada ou passagem de mensagem.

- **Detecção de Erros:**

garantir a correta computação detectando-se erros na CPU, memória, rede ou em programas do usuário.

Serviços do SO

- **Alocação de Recursos (dar recursos):**

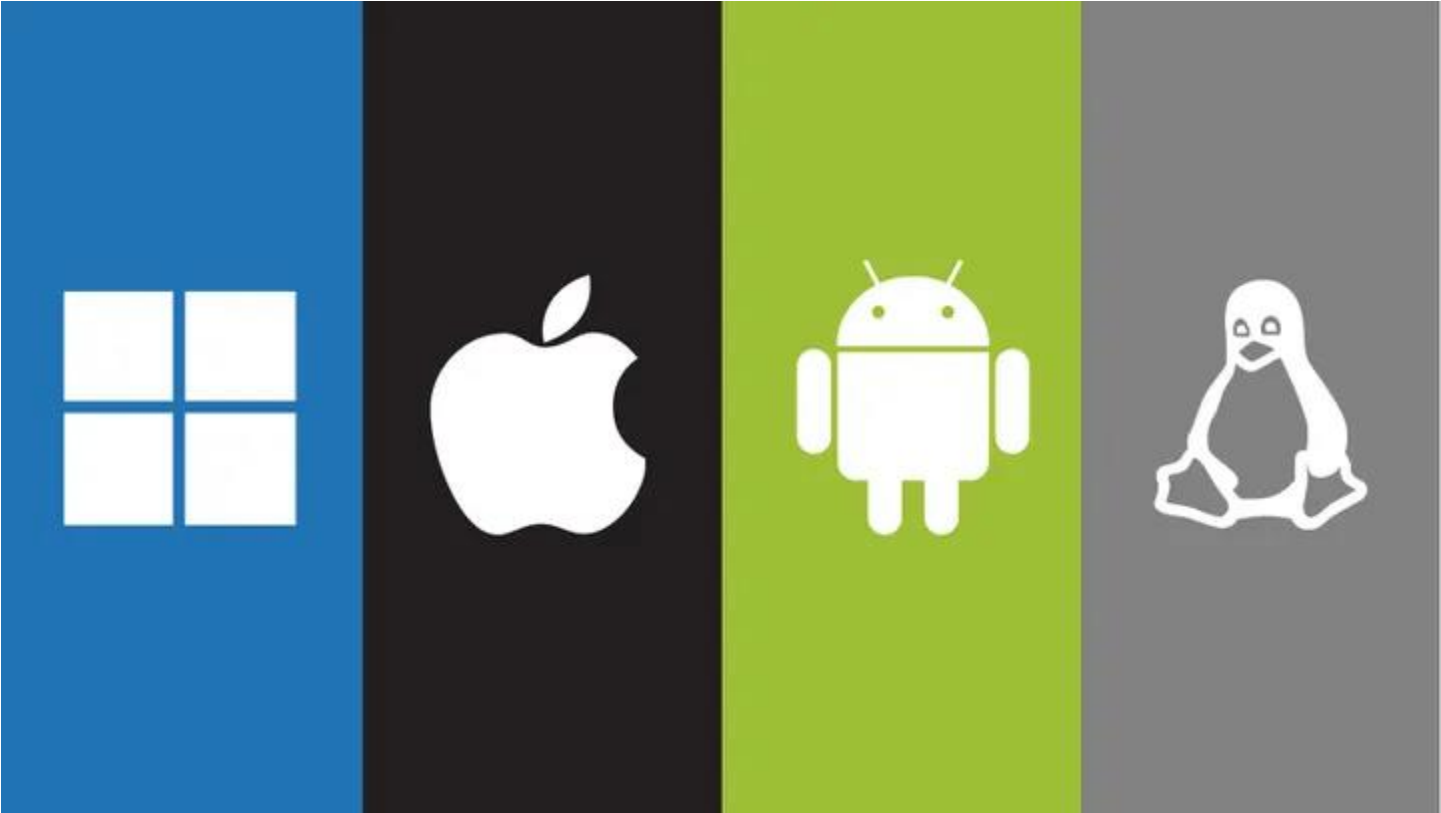
alocar recursos para múltiplos usuários ou múltiplos processos executando concorrentemente ou ao mesmo tempo, ou seja, distribuir recurso do sistema (CPU, memória, dispositivos) entre os diferentes processos que estão em execução, garantia que tenha recurso necessário para executar de forma eficiente.

- **Contabilização (registrar o uso):**

registrar os usuários que utilizam os recursos do computador, quais tipos de recursos, em que quantidade e como os utilizam. Exemplo: quanto tempo de CPU um processo usou, medir o quanto de memória consumida por usuários

- **Proteção:**

garantir que todos os acessos aos recursos do sistema sejam controlados.



Software Utilitários

- São programas que suprem a deficiência do sistema operacional.
- Executam tarefas de manutenção, análise ou otimização do sistema, para manter o funcionamento adequado e melhorar desempenho.
- Podem ser usados para compactação de dados, aumento de desempenho da máquina, etc.
- Ex: Ferramentas de backup, antivírus, limpador de arquivos.

SOFTWARES UTILITÁRIOS



➤ **Produtividade Pessoal**

- Editores de Texto, Planilhas de Cálculo, SGBD, Apresentação,
- Publicação, Editores Gráficos,
- Projetos Técnicos, etc

➤ **Internet**

- Browsers: MS Internet Explorer, Netscape, Mozilla FireFox,
- E-mail: MS Outlook Express, Mozilla Firebird, Opera.
- Conversação: Yahoo, Messenger
- Plug-in's – Windows Media Player, ShockWave, Acrobat, Winamp, MP3 etc

➤ **Específicos:**

- Compactadores (WinZip),
- Anti-virus (Norton, Trend Micro, McAfee Viruscan),
- Backup (ArcServe),
- Comunicação privativa (PC Anywhere, Carbon Copy) etc

Software Aplicativos

- Objetivo de auxiliar o usuário a desempenhar uma tarefa específica para o usuário final.



SOFTWARES APLICATIVOS

❖ **Empresariais Genéricos** (por Ramo de Atividade e/ou Negócio):

- ❖ Sistemas Integrados: SAP, Oracle, (J.D.Edwards, PeopleSoft), Datasul, Totvs (Microsiga, Logocenter RM), Senior, Radar etc
- ❖ Sistemas Funcionais: Contabilidade, Folha de Pagamento, Financeiro, Ativo Imobilizado, etc

❖ **Específicos, Técnicos, Funcionais** (por Problema/Processo/Projeto) :

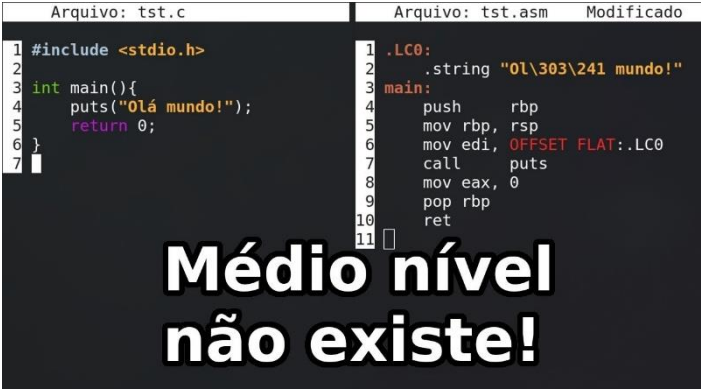
- ❖ Simuladores, Científicos
- ❖ Radares, Controle de Processos
- ❖ Roteirizadores, GPS
- ❖ Processos únicos, diferenciadores etc.

❖ **Especialistas, Tecnológicos** (normalmente junto com HW):

- ❖ Piloto Automático, Relógios, Medidores, Controles Automáticos,
- ❖ Controle Numéricos (CNC's),
- ❖ Centrais Telefônicas,
- ❖ Telefones Celulares, Eletrodomésticos

Linguagens de programação

- Existem dois tipos:
 - Linguagem de baixo nível
 - Linguagem de alto nível



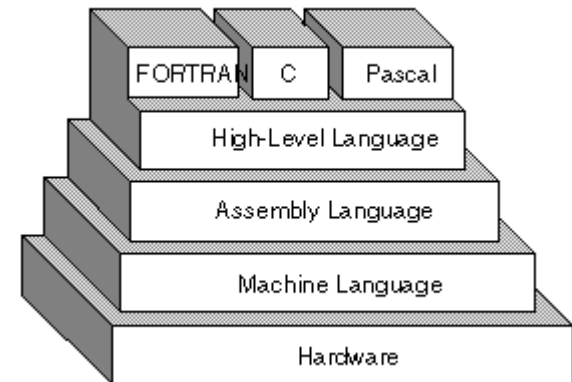
```
Arquivo: tst.c
1 #include <stdio.h>
2
3 int main(){
4     puts("Olá mundo!");
5     return 0;
6 }
7

Arquivo: tst.asm Modificado
1 .LC0:
2     .string "01\303\241 mundo!"
3 main:
4     push    rbp
5     mov     rbp, rsp
6     mov     edi, OFFSET FLAT:.LC0
7     call    puts
8     mov     eax, 0
9     pop     rbp
10    ret
11
```

**Médio nível
não existe!**

Linguagem de baixo nível

- Ela é compreensível pela arquitetura do computador.
- Utilizada somente instruções do processador, com isso necessita conhecimento registradores de máquina.
- Ex: Linguagem Assembly



Linguagens de baixo nível

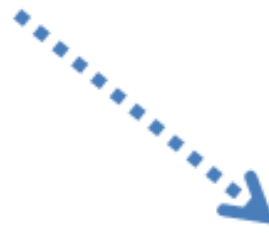


Voltadas para a máquina, são escritas utilizando as instruções do microprocessador do computador.
São genericamente chamadas de linguagens Assembly.



Vantagens

Os programas são executados com maior velocidade de processamento e ocupam menos espaço na memória.



Desvantagens

Um código Assembly gerado para um tipo de processador não serve para outro. Estes também não são estruturados, tornando a programação mais difícil.

Assembly Language	Machine Code
add \$t1, t2, \$t3	04CB: 0000 0100 1100 1011
addi \$t2, \$t3, 60	16BC: 0001 0110 1011 1100
and \$t3, \$t1, \$t2	0299: 0000 0010 1001 1001
andi \$t3, \$t1, 5	22C5: 0010 0010 1100 0101
beq \$t1, \$t2, 4	3444: 0011 0100 0100 0100
bne \$t1, \$t2, 4	4444: 0100 0100 0100 0100
j 0x50	F032: 1111 0000 0011 0010
lw \$t1, 16(\$s1)	5A50: 0101 1010 0101 0000
nop	0005: 0000 0000 0000 0101
nor \$t3, \$t1, \$t2	029E: 0000 0010 1001 1110
or \$t3, \$t1, \$t2	029A: 0000 0010 1001 1010
ori \$t3, \$t1, 10	62CA: 0110 0010 1100 1010
ssl \$t2, \$t1, 2	0455: 0000 0100 0101 0101
srl \$t2, \$t1, 1	0457: 0000 0100 0101 0111
sw \$t1, 16(\$t0)	7050: 0111 0000 0101 0000
sub \$t2, \$t1, \$t0	0214: 0000 0010 0001 0100

Linguagem de alto nível

- Longe da linguagem de máquina e próximo a linguagem humana.
- Linguagem distante a linguagem da arquitetura de computadores, ou seja, não necessita de conhecimento da linguagem do processador.
- Ex: Python



```

19 // perm for permission checks
20 // user for user variables
21 page_open(array("sess" => "Example_Session", "auth" => "Example_Auth", "perm" => "Examp
22
23 // page access requires that the user is authenticated and has "admin" permission
24 $perm->check("admin");
25
26 // s is a per session variable, u is a per user variable.
27
28 // If they've already been loaded in from the database, leave them alone
29 // Otherwise, set them to a value so we don't get PHP warnings later.
30 if(!isset($s)) { $s=0; };
31 if(!isset($u)) { $u=0; };
32
33 $sess->register("s");
34 $user->register("u");
35
36 ?>
37 <html>
38 <head>
39 <!-- Style sheet used by Table class below -->
40 <style type="text/css">
41 table.metadata { background-color: #eeeeee; border-width: 0; padding: 4 }
42 th.metadata { font-family: arial, helvetica, sans-serif }
43 td.metadata { font-family: arial, helvetica, sans-serif }
44 table.data { background-color: #cccccc; border-width: 0; padding: 4 }
45 th.data { font-family: arial, helvetica, sans-serif; horizontal-align: left; verti
46 td.data { font-family: arial, helvetica, sans-serif; horizontal-align: left; verti
47 </style>
48 </head>
49 <body bgcolor="#ffffff">
50
51 <a href="<?php $sess->pself_url() ?>">Reload</a> this page to see the counters increment
52 <a href="<?php $sess->purl("index.php3") ?>">Load</a> the simple page again.<br>
53 <a href="<?php $sess->purl("default.php3") ?>">Load</a> the default auth example.<br>

```



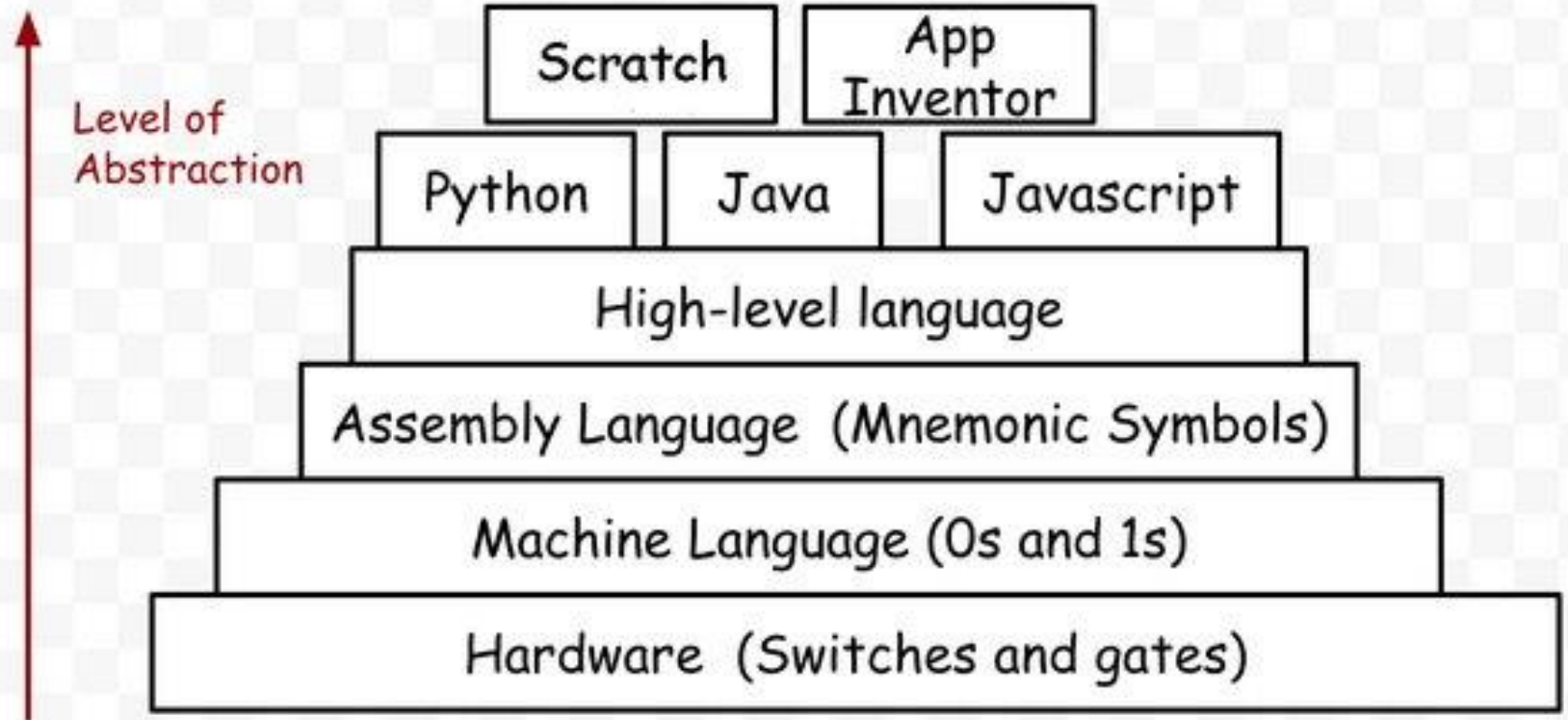
Alto Nível

- Aprendizado facilitado
- Menor custo de elaboração e manutenção de software

Baixo Nível

- Aprendizado dificultado
- Melhor aproveitamento da arquitetura da máquina e mais velocidade de processamento





Tipos de Linguagens

- Linguagens compiladas
 - Existe um processo de compilação e geração de um programa executável que pode ser executado pela UCP
 - Ex: Delphi, Visual Basic (VB), C/C++, COBOL, Pascal
- Linguagens interpretadas
 - Não existe compilação nem programa executável
 - Cada instrução do programa é interpretada e executada pela UCP
 - Pode gerar problemas de desempenho
 - Ex: JavaScript, Perl, PHP, Java, C#

SOFTWARE LIVRE
SOFTWARE PROPRIETÁRIO
SOFTWARE COMERCIAL



SOFTWARE COMERCIAL

Programas desenvolvidos por empresas com o objectivo de obter lucro com a sua utilização.

Note que “Comercial” e “Proprietário” não são a mesma coisa. Grande parte do software comercial é proprietário, mas existe software livre que é comercial, e existe software que não é livre mas que também não é comercial.



SOFTWARE PROPRIETÁRIO

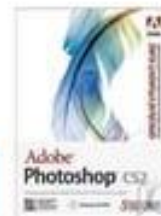
São programas cuja cópia, redistribuição ou modificação são proibidas pelo seu proprietário. Usualmente, é necessário uma licença para que possa ser utilizados.

Exemplos de software proprietário

- anti-Arbitration - Intel
- Winzip - Winzip
- Windows - Microsoft



SOFTWARE PROPRIETÁRIO





SOFTWARE LIVRE

**Uma questão de liberdade, não de preço e sim
liberdade de expressão!**

Software Livre é um conceito muito importante no mundo da computação. Consiste em um movimento social, que diz que o software precisa estar atrelado a aspectos de liberdade, a ideia do SW livre começou em 1983, por Richard Stallman, que fundou a Free Software Foundation (FSF), uma entidade sem fins lucrativos.

Software Livre

- Software que respeita a liberdade e senso dos usuários, cujo código fonte deve ser disponibilizado.
- Implica e não propriedade **do software**
- O usuário possui 6 liberdades:
 - liberdade de executar;
 - copiar;
 - distribuir;
 - estudar;
 - mudar;
 - melhorar o software.



“Quando o usuário não controla o programa, o programa controla os usuários.”

SOFTWARE LIVRE X SOFTWARE PROPRIETÁRIO



Software Livre



Software Proprietário

Software Livre x Software Proprietário

Software Livre	Software Proprietário
Sem custo de licença, manutenção ou anuidade	Custo de licença, manutenção ou anuidade
Necessidade de equipe com conhecimento	Custo do treinamento incluído na implantação
Atualizações mais lentas , por voluntários	Atualizações anuais, rápida resolução de bugs, e suporte disponível
Customização pode ser cara.	Customização incluída na implantação



SOFTWARE PROPRIETÁRIO



SOFTWARE LIVRE

Suíte de Escritório



Editor de imagem



Editor de gráfico vetorial



Editor de fotos digitais



Editoração eletrônica



Editor de áudio digital



Aplicativo CAD



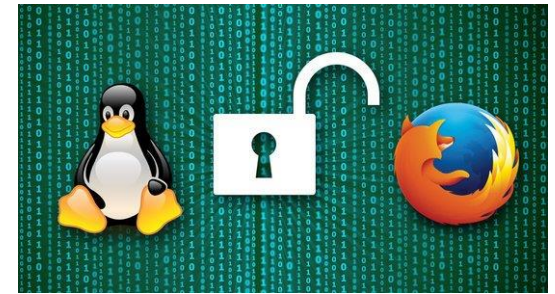
Modelagem 3D



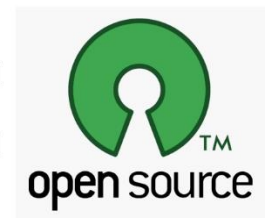
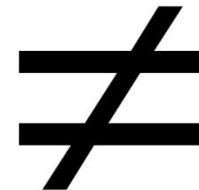
Editor de vídeo



Open Source



- Se define código fonte aberto, porem pode ter um dono. Mas somente quem vai promover alterações nele é o proprietário
- Open Source é um movimento que surgiu em 1998 por iniciativa principal de Bruce Perens, resultando na criação da Open Source Initiative (OSI). A OSI não tenta ignorar as liberdades da FSF, mas torná-las mais flexíveis!
- Para essa definição cumprir possuem dez critérios:
 - **Redistribuição livre**
 - **Código-Fonte**
 - **Obras Derivadas**
 - **Integridade do código fonte do autor**
 - **Não ter discriminação contra pessoas ou grupos**
 - **Não ter discriminação contra campos de trabalho**
 - **Distribuição da Licença**
 - **Licença não deve ser específica para um produto**
 - **Licença não deve restringir outro software**
 - **Licença deve ser tecnologicamente neutra**



PRÓXIMA AULA: PROVA

23/05/2025