

# Computador

Prof<sup>a</sup>. Rose Yuri Shimizu

# Computador - máquina hierárquica (camadas)

- Software
  - ▶ Parte lógica: sequência de instruções para controlar o hardware
- Hardware
  - ▶ Parte física: conjunto de circuitos eletrônicos que manipulam sinais elétricos, transformando sinais de entrada em sinais de saída.
- Computador
  - ▶ É um sistema de computação em camadas, com um conjunto de hardware capaz de efetuar computações conforme lógica do software
  - ▶ Server para o processamento automático dos dados - **INFOR**mação auto**MÁTICA**
- <https://fga.rysh.com.br/apc/aulas/0-hardware.pdf>
- <https://fga.rysh.com.br/apc/aulas/0-software.pdf>

# Camadas do sistema de computação

## SOFTWARE

Aplicação/software

Software básicos e sistema

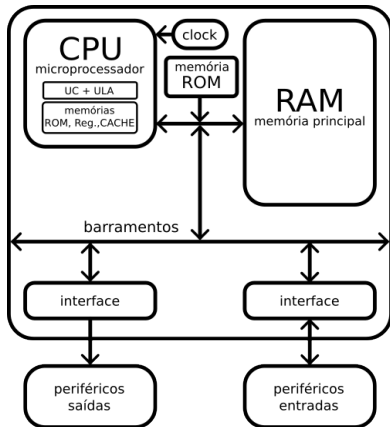
## HARDWARE

Arquitetura

### Organização

- Parte física
- Fabricantes: diferentes implementações da mesma arquitetura

# Hardware - Organização



- Entradas e saídas
- Memórias
- Microprocessador

# Hardware - Organização

- Entradas e saídas

- ▶ Comunicação entre o computador e seu ambiente:

- ★ Entrada: equipamentos que fornecem o que computador irá processar
    - ★ Saída: equipamentos que recebem o que computador processou

- ▶ Dispositivos de entrada:

- ★ **Teclado** - lê os caracteres digitados pelo usuário (entrada padrão)
    - ★ Mouse - lê os movimentos e toque de botões
    - ★ Microfone, scanner, sensores, etc.

- ▶ Dispositivos de saída:

- ★ **Monitor** - imprime na tela caracteres e gráficos (saída padrão)
    - ★ Alto-falante - comunicação com o usuário através de som
    - ★ Impressora, etc.

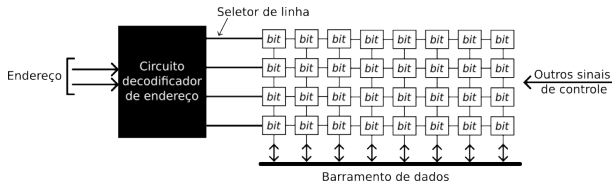
- ▶ Dispositivos de entrada e saída:

- ★ Disco rígido - grava e lê dados.
    - ★ Modem, etc.

# Hardware - Organização

- Memórias

- ▶ Conjunto de componentes eletrônicos capazes de conservar estados
  - ★ Armazenam programas e dados de forma binária (estados)
  - ★ Convencionou-se: 1 (alta tensão) e 0 (baixa tensão)



- ▶ Medidas

- ★ Bit (binary digit): menor unidade informação na computação (0 ou 1)
- ★ Byte correspondem a 8 bits
- ★ Kbytes (kilobytes) correspondem a 1024 bytes ou ( $2^{10}$  bytes)
- ★ MBytes (megabytes) correspondem a 1024 KBytes ou ( $2^{20}$  bytes)
- ★ Gbytes (gigabytes) correspondem a 1024 Mbytes ou ( $2^{30}$  bytes)

# Hardware - Organização

- Memórias

- ▶ Memórias não-voláteis - ROM (Read-only memory)
  - ★ Somente leitura
  - ★ Usada para armazenar programas estáticos (que não alteram)
  - ★ Não necessitam de um fornecimento constante de energia

# Hardware - Organização

- Memórias

- ▶ Memórias não-voláteis - ROM (Read-only memory)
- ▶ Memórias não-voláteis - secundária (HD, SSD)
  - ★ Leitura e escrita
  - ★ Alta capacidade de armazenamento e baixo custo
  - ★ Não necessitam de um fornecimento constante de energia
  - ★ Armazena dados e programas (não-execução)
  - ★ Ex.: HD, SSD



# Hardware - Organização

- Memórias

- ▶ Memórias não-voláteis - ROM (Read-only memory)
- ▶ Memórias não-voláteis - secundária (HD, SSD)
- ▶ Memórias voláteis - RAM (Random access memory)
  - ★ Armazenar/executar programas e dados que estão em execução
  - ★ Necessitam de um fornecimento constante de energia

# Hardware - Organização

- Memórias

- ▶ Memórias não-voláteis - ROM (Read-only memory)
- ▶ Memórias não-voláteis - secundária (HD, SSD)
- ▶ Memórias voláteis - RAM (Random access memory)
  - ★ Armazenar/executar programas e dados que estão em execução
  - ★ Necessitam de um fornecimento constante de energia
  - ★ Memória cache - SRAM: RAM estática
    - Mais rápidas (e caras): ↑ desempenho
    - Mantém um cópia de dados e instruções da memória principal
    - Diminuir acesso à memória principal

# Hardware - Organização

- Memórias

- ▶ Memórias não-voláteis - ROM (Read-only memory)
- ▶ Memórias não-voláteis - secundária (HD, SSD)
- ▶ Memórias voláteis - RAM (Random access memory)
  - ★ Armazenar/executar programas e dados que estão em execução
  - ★ Necessitam de um fornecimento constante de energia
  - ★ Memória cache
  - ★ Memória principal - DRAM: RAM dinâmica
    - Mais simples e baratas: ↑ capacidade armazenamento
    - Processos: programas em execução

# Hardware - Organização

## ● Processador - CPU

- ▶ CPU - Central Processor Unit
- ▶ Circuito integrado: pequeno e de altíssima velocidade de processamento
- ▶ Parte ativa do computador: **cálculo e manipulação dos dados, tomadas de decisão, execução de instruções, comando de outros chips**

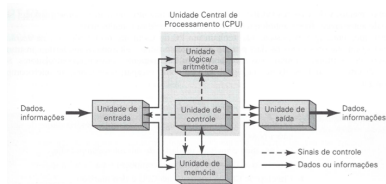


FIGURA 1.11 Diagrama funcional de um computador digital.

- ▶ Executa um conjunto limitado de operações/instruções (arquitetura)
- ▶  $\neq$  fabricantes = arquitetura  $\neq$  projetos (organização)

# Camadas do sistema de computação

## SOFTWARE

Aplicação/software

Software básicos e sistema

## HARDWARE

### Arquitetura

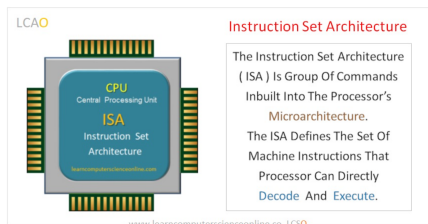
- O que faz: operações/instruções específicas que cada conjunto de componentes é projetado para executar
- Regras para padronização: entradas e saídas, chamadas (formato)

Organização

# Hardware - Arquitetura

## ● Instruction Set Architecture (ISA)

- ▶ Especificam os conjunto de instruções e seu funcionamento: tudo o que o computador sabe fazer
- ▶ O que faz: operações/instruções específicas que cada conjunto de componentes é projetado para executar
- ▶ Regras para padronização: entradas e saídas, chamadas (formato)
- ▶ Exemplos: x86, x86-64, ARM
- ▶ Microarquitetura: implementação do conjunto de instruções em determinado processador (core, zen, fusion).



# Hardware - Arquitetura

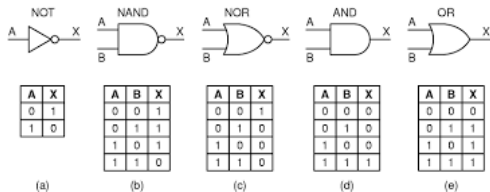
- Instruction Set Architecture (ISA)
- Desenvolvido a partir da álgebra/lógica booleana
  - ▶ Matemático inglês George Boole - meados do século XIX
  - ▶ Representar matematicamente (precisão) problemas de lógica (pensamento humano).

A	$\sim A$	A	B	$A \rightarrow B$
V	F	V	V	V
V	F	V	F	F
F	V	F	V	V
F	V	F	F	V

- ▶ Opera sobre valores verdade: Verdadeiro ou Falso.

# Hardware - Arquitetura

- Instruction Set Architecture (ISA)
- Desenvolvido a partir da álgebra/lógica booleana
  - ▶ Implementada com circuitos lógicos-digitais
  - ▶ Componentes básicos: portas lógicas AND, OR, NOT e flip-flop
  - ▶ Combinação das portas: implementam toda a álgebra de Boole
  - ▶ Convencionou-se: estado 0(F), ausência(menos) de tensão; estado 1(V), presença(mais) de tensão



## ▶ Equivalência lógica

A	B	A or B
V	V	V
V	F	V
F	V	V
F	F	F

A	B	A and B
V	V	V
V	F	F
F	V	F
F	F	F

A	B	$A \rightarrow B$	$\sim A$ or B
V	V	V	F or V = V
V	F	F	F or F = F
F	V	V	V or V = V
F	F	V	V or F = V



# Hardware - Arquitetura

- Instruction Set Architecture (ISA)
- Desenvolvido a partir da álgebra/lógica booleana
- Interface entre Hardware x Software
  - ▶ Instruções que comandam um hardware

# Hardware - Arquitetura

- Instruction Set Architecture (ISA)
- Desenvolvido a partir da álgebra/lógica booleana
- Interface entre Hardware x Software
  - ▶ Instruções que comandam um hardware - Habilitar um circuito distinto
    - Instrução Máquina: sequência de bits que identificam uma operação (tarefa a ser executada)

# Hardware - Arquitetura

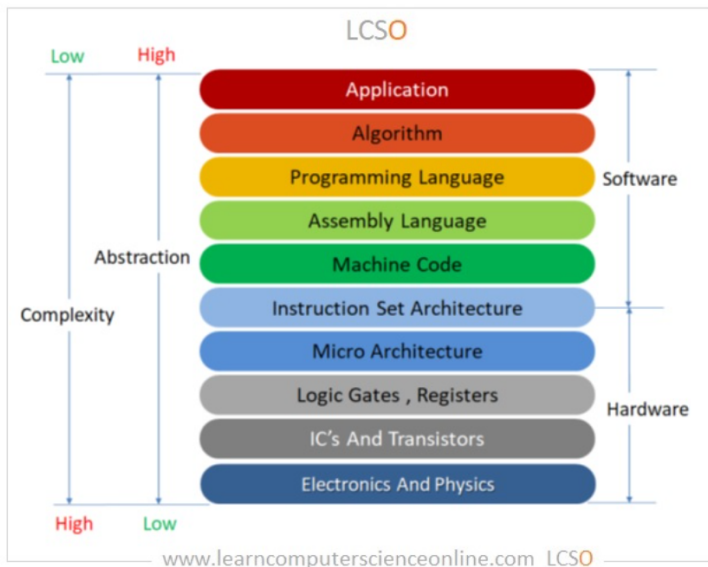
- Instruction Set Architecture (ISA)
- Desenvolvido a partir da álgebra/lógica booleana
- Interface entre Hardware x Software
  - ▶ Instruções que comandam um hardware
  - ▶ Instruções utilizados pelo software
    - ★ Instrução Assembly: mnemônico (“palavra-chave”) que correlaciona a funcionalidade e a instrução de máquina
    - ★ Lógicas/aritméticas (add, sub, and, or)
    - ★ Transferência de dados - memória (lw, sw)
    - ★ Controle de fluxo - comparações e desvios (beq, bne, j, jr, jal)
    - ★ Interface com a E/S (in, out, insw, outsw)
    - ★ Cada instrução possui um código binário que o identifica

C	assembly
<code>int V[NNN];</code>	<code>la r1, V # r1 &lt;- &amp;(V[0])</code>
<code>...</code>	<code>lw r4, 4(r1) # r4 &lt;- M[r1+1*4]</code>
<code>V[0] = V[1] + V[2]*16;</code>	<code>lw r6, 8(r1) # r6 &lt;- M[r1+2*4]</code>
	<code>sll r6, r6, 4 # r6*16 = r6&lt;&lt;4</code>
	<code>add r7, r4, r6</code>
	<code>sw r7, 0(r1) # M[r1+0*4] &lt;- r4+r6</code>

# Computador

- Conjunto de componentes eletrônicos capazes de conservar estados
- Convencionou-se: 1 (alta tensão) e 0 (baixa tensão)
- Computador =  
[ sistema binário (dados) + álgebra booleana (lógica) ] +  
circuitos de comutação e conversação de estados

# Camadas do sistema de computação



# Camadas do sistema de computação

## SOFTWARE

Aplicação/software

**Software básicos e sistema**

## HARDWARE

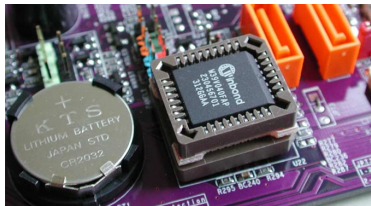
Arquitetura

Organização

# Boot - Inicialização da máquina

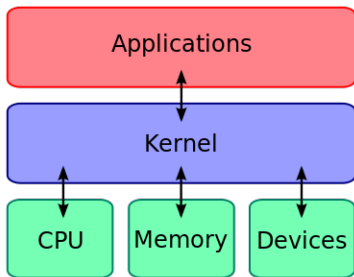
## 1 BIOS

- ▶ “Basic Input/Output System”
- ▶ Sistema Básico de Entrada/Saída
- ▶ Inicializa os dispositivos básicos e o **sistema operacional**
- ▶ **Gravado em uma memória**



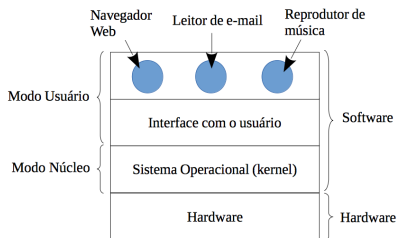
## 2 Sistema Operacional

- ▶ Gerenciador de recursos e serviços
- ▶ Interface do software x hardware



# Sistema Operacional

- Dividido em modo núcleo x usuário
  - ▶ Modo núcleo: acesso ao hardware, instruções de máquina
  - ▶ Modo usuário: acesso ao subconjunto de instruções que o modo núcleo disponibiliza
- Base sobre a qual são executados diversos programas
- Interface do hardware: chamadas de sistema para as aplicações utilizarem o hardware de forma simplificada
- Programas em execução: processo gerenciado pelo SO





# Sistema Operacional - Exemplos

- Windows
  - ▶ Criado pela Microsoft
  - ▶ Copyright (direitos autorais)
- Linux (kernel Linux)
  - ▶ Baseado no sistema operacional UNIX
  - ▶ Está sob a licença GNU/GPL
    - ★ Liberdade de executar, alterar e redistribuir
    - ★ Não pode “fechar” → qualquer alteração não pode ser restrita
  - ▶ Distribuições Linux
    - ★ Debian, Mint, Fedora, RedHat, Slackware, Ubuntu
- BSD
  - ▶ Baseado no sistema operacional UNIX
  - ▶ O MacOS é derivado do BSD
  - ▶ Licença BSD: código livre mas pode ser “fechado”
- FreeBSD é derivado do BSD (usado no ps4)
- Android é baseado no núcleo Linux e desenvolvido pela Google



Mac OS



FreeBSD.



# Sistema Operacional para a disciplina

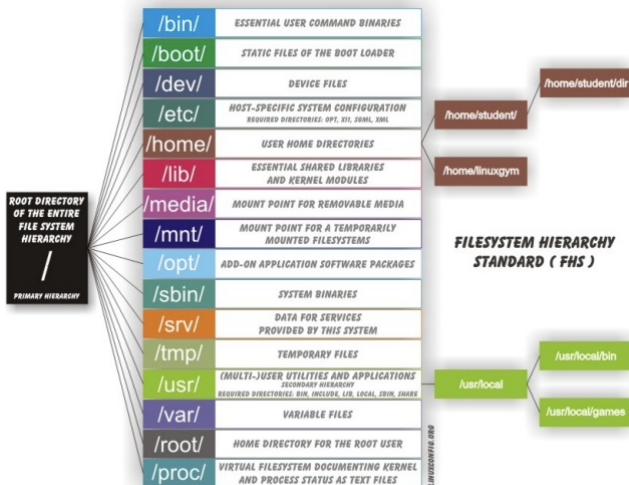
- Distribuição Linux:
  - ▶ kernel Linux + gerenciador de pacotes + repositório
- Ubuntu (baseado no Debian):
  - ▶ <https://ubuntu.com/download>
- Opções: <https://fga.rysh.com.br/apc/index.html#ambientes>

# Sistema Operacional - Linux

- Estrutura: arquivos e diretórios estão situados sob o diretório raiz “/”

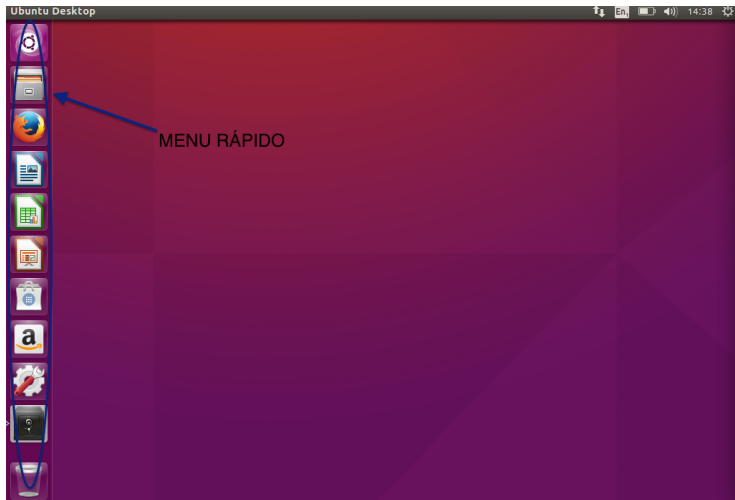
Diretório	Descrição
/bin/	Comandos binários essenciais para todos os usuários (ex: cat, ls, cp)
/boot/	Arquivos do Boot loader (inicialização)
/dev/	Dispositivos
/etc/	Arquivos de configuração específicos do computador
<b>/home/</b>	Diretórios de usuários
/lib/	Diretório com as bibliotecas essenciais
/mnt/	Sistemas de arquivos “montados” temporariamente (ex. partições)
/media/	Pontos de “montagem” para mídia removível, como CD-ROMs
/proc/	Processos do sistema
/root/	Diretório home para o super usuário (root).
/sbin/	Arquivos binários para propósito de administração do sistema
/tmp/	Arquivos temporários
/usr/	Hierarquia secundária (mesma hierarquia mas sem dados essenciais para sistema) para dados compartilhados de usuários (pacotes instalados)
/var/	Arquivos “variáveis”, como logs, base de dados, páginas Web e arquivos de e-mail

# Sistema Operacional - Linux - Estrutura de diretórios

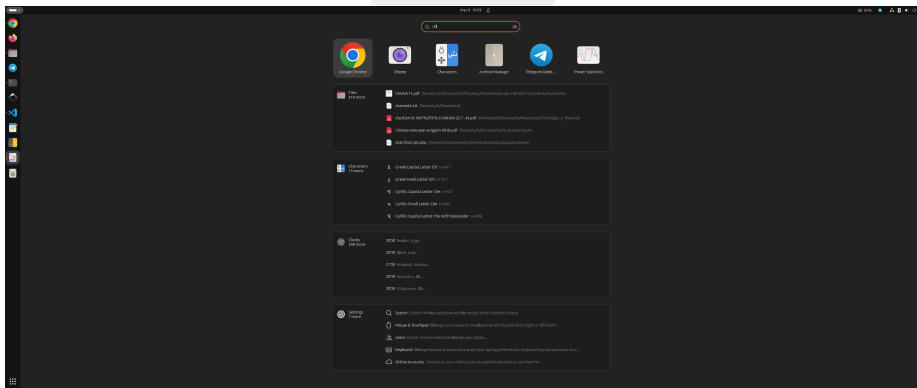


- Usuário possui permissão de alteração (gravação) somente da sua **“home”**
- Para alterar outros diretórios, precisa ter a senha de **“root”** (super-usuário)

# Acessar arquivos e programas - Menu rápido



# Acessar arquivos e programas - Tecla windows



# Acessar arquivos e programas - Tecla windows

- Abrir Navegador Web (acessar listas de exercícios): *firefox* ou *chrome*
- Abrir Editor de Texto (escrever o código-fonte): *text editor*
- Abrir Terminal (acessar arquivos, compilar e executar): *terminal*



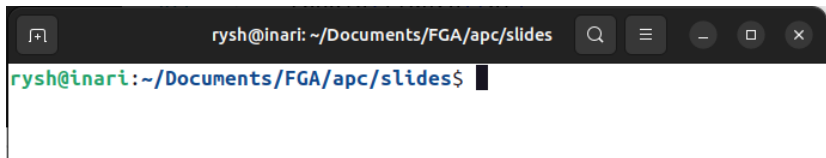
# Programas

- Navegador Web
  - ▶ <https://fga.rysh.com.br/apc/>
  - ▶ <https://moj.naquadah.com.br/>
- Editor de texto:
  - ▶ Novo arquivo/documento: ctrl + n
  - ▶ Nova aba/documento: ícone “+”
  - ▶ Abrir arquivo: ctrl + o
  - ▶ Salvar arquivo: ctrl + s



# Terminal

- Shell (interface de usuário)
- Interface gráfica de usuário (GUI): janelas + ícones
- Interface de linha de comando (Command-Line Interface - CLI)
  - ▶ Shell modo texto: processo que recebe comandos via teclado, interpreta-os e passa para sistema operacional executar
  - ▶ Exemplos de serviços: login/logout, manipulação de arquivos, execução de programas, etc.
  - ▶ Programas utilizados para acessar o shell
    - ★ Windows: Command Prompt (cmd.exe ou cmd) e PowerShell (novo prompt de comando do Windows)
    - ★ Linux: Terminal



A screenshot of a terminal window. The title bar shows the window name 'rysh@inari: ~/Documents/FGA/apc/slides' and standard window control buttons (search, menu, minimize, maximize, close). The terminal content shows the prompt 'rysh@inari: ~/Documents/FGA/apc/slides\$' followed by a cursor. The prompt is colored green and blue.

- Prompt: indicativo de entrada de comando
  - ▶ <user>@<hostname>:<path>
  - ▶ ~: /home/<user>

# Terminal - Comando básicos

- **cd / + <enter>** : “abre” o diretório raiz
- **cd + <enter>** : “abre” o diretório home do usuário logado
- **cd Documents** : “abre” o diretório “Documents” (se existir) do diretório atual
  - ▶ Atenção para o diretório atual e qual deseja Abrir
  - ▶ Exemplo:
    - 1 Árvore de diretórios: ~/Documents/FGA/apc
    - 2 Diretório atual: ~
    - 3 Abrir diretório apc: **cd Documents/FGA/apc**
    - 4 Diretório atual: ~/Documents/FGA/apc
    - 5 Voltar 1 diretório: **cd ..**
    - 6 Diretório atual: ~/Documents/FGA
    - 7 Abrir diretório apc: **cd apc**
    - 8 Diretório atual: ~/Documents/FGA/apc
- **ls + <enter>**: listar o conteúdo do diretório
- **ls exemplo**: listar o conteúdo do diretório “exemplo”
- **mkdir exemplo**: criar o diretório “exemplo” dentro do diretório atual
- **gcc**: compilador da linguagem de programação C