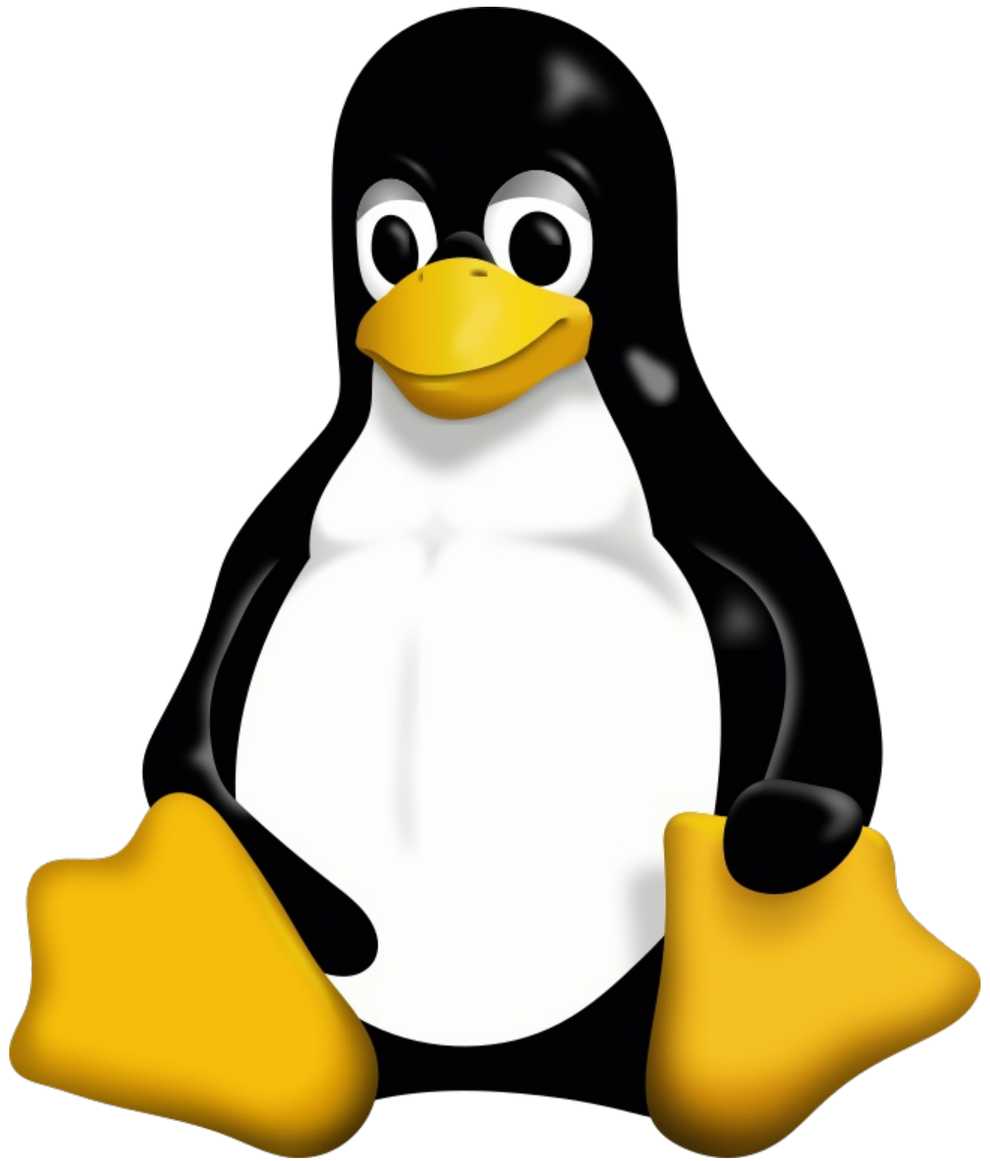
Sistemas Operacionais: visão geral



lewing@isc.tamu.edu Larry Ewing and The GIMP, CC0,
Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Tux.svg)

Prof. Abrantes Araújo Silva Filho

O que é um sistema operacional?

Tumisu, no Pixabay (https://pixabay.com/photos/man-thinking-doubt-question-mark-5723449/)

Difícil dizer com certeza absoluta pois realiza duas funções fundamentais e de certa forma não relacionadas:

- **Fornecer aos programas de usuário uma máquina virtual abstrata mais simples, melhor e mais limpa do que a complexidade do hardware;**

- **Gerenciar todos os recursos de hardware/software:**
  - **CPU**
  - **Memória**
  - **Dispositivos de I/O**
  - **Outros programas**

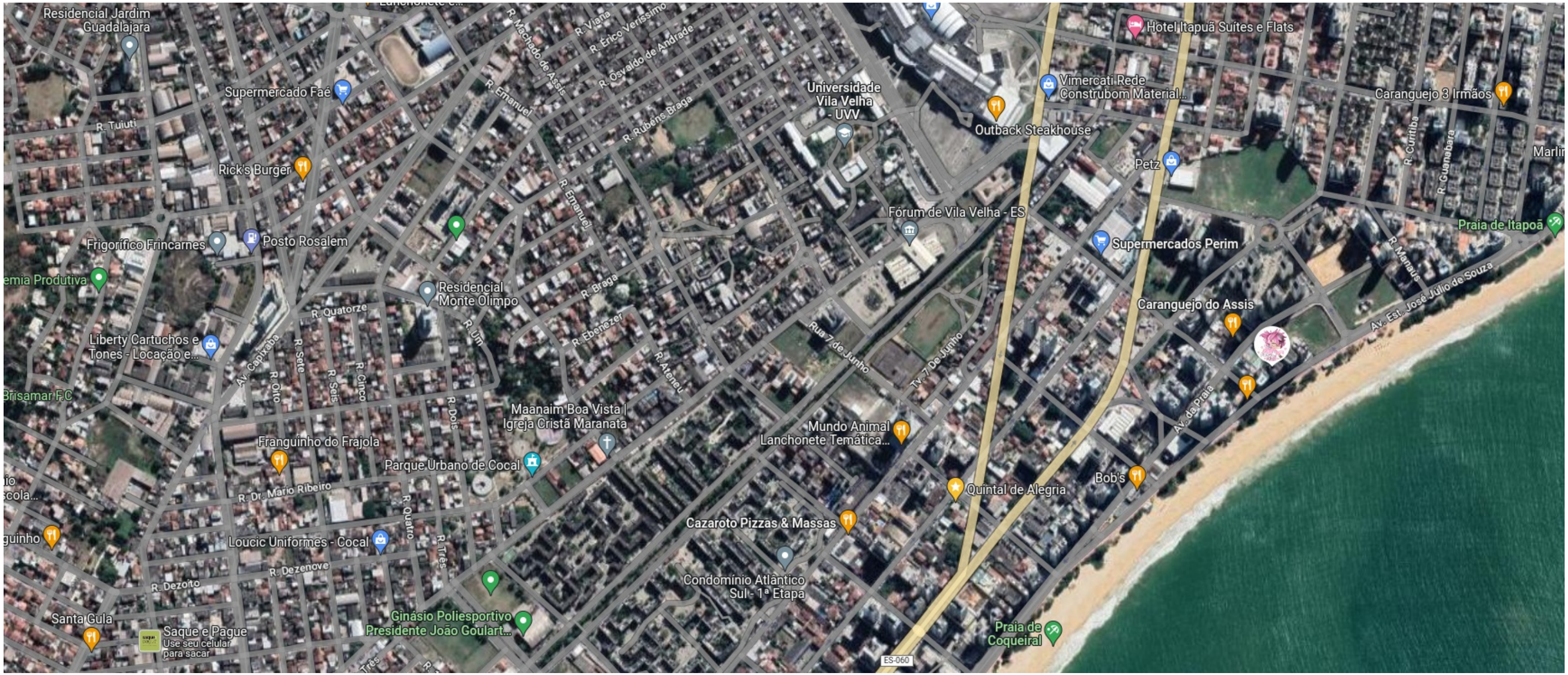
O que é abstração?

Veloce Cockpit, (https://www.velocecockpit.com.br/)

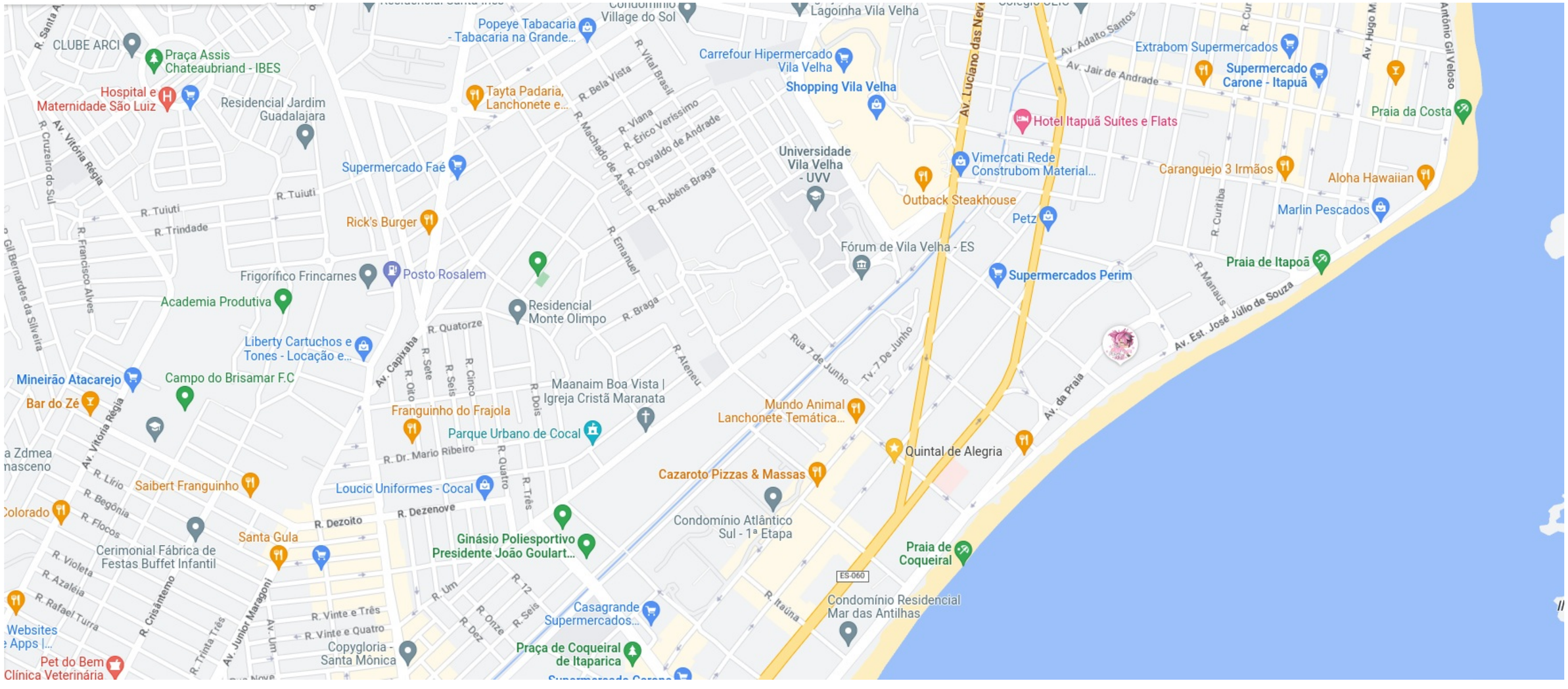
Capacidade de ignorar os detalhes internos não importantes e se concentrar no propósito básico e mecanismos de funcionamento e operação.

Faz parte do pensamento computacional.

O que é abstração?



O que é abstração?



O que é abstração?

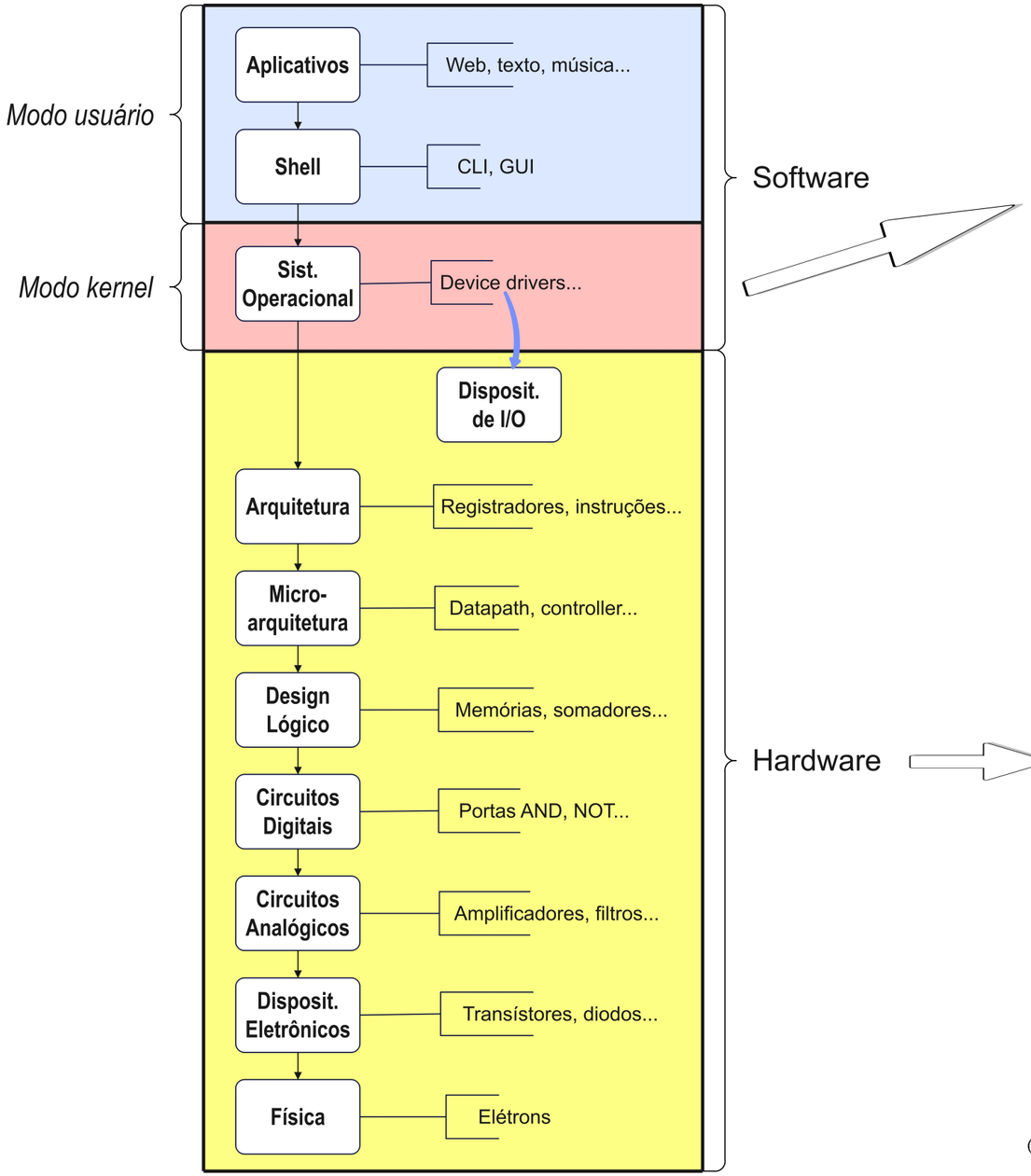
O que é abstração?

```c
#include <CRpaic.h>
#include <stdbool.h>
#include <stdio.h>

bool is_prime (int n);

int main (void)
{
    int n = 0;

    do
    {
        n = crpaic_get_int("Informe um inteiro (0 para sair): ");

        if (n == 0)
            printf("Programa finalizado.\n");
        else if (is_prime(n))
            printf("%d é primo.\n", n);
        else
            printf("%d não é primo.\n", n);
    }
    while (n != 0);

    return 0;
}
```

O que é abstração?

```
67 bool is_prime (int n)
68 {
69     if (n < 2)
70         return false;
71
72     if (n == 2 || n == 3)
73         return true;
74
75     if (n % 2 == 0 || n % 3 == 0)
76         return false;
77
78     for (int i = 5; i * i <= n; i += 6)
79     {
80         if (n % i == 0 || n % (i + 2) == 0)
81             return false;
82     }
83
84     return true;
85 }
```

Sistema operacional como máquina virtual abstrata

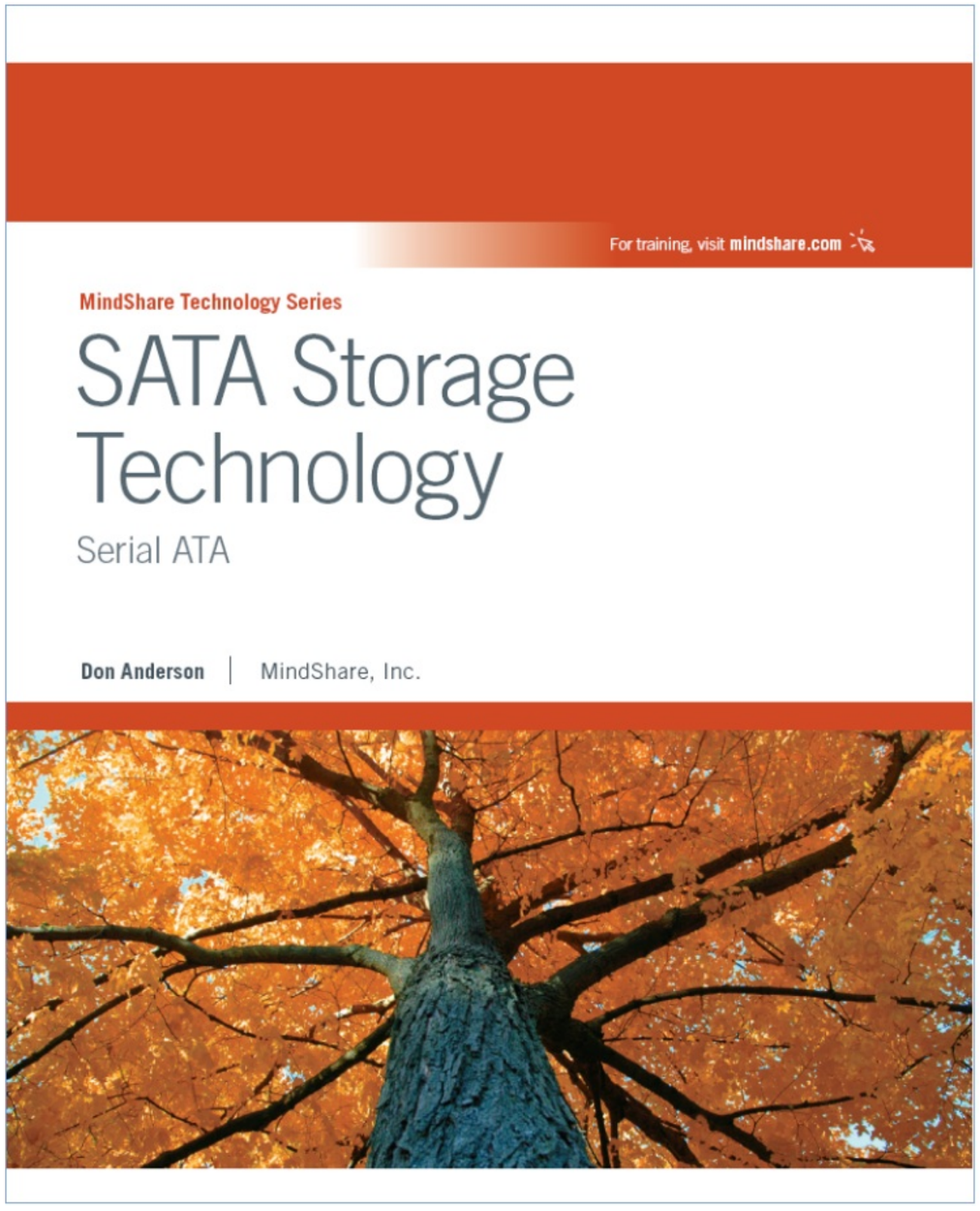


Daria Chudarina, no Pixabay
(https://pixabay.com/vectors/leaves-cat-nature-autumn-cute-7552915/)

Daria Chudarina, no Pixabay
(https://pixabay.com/vectors/leaves-cat-nature-autumn-cute-7552915/)

Cathe, no Pixabay
(https://pixabay.com/illustrations/ai-generated-mummy-revived-egypt-8684607/)

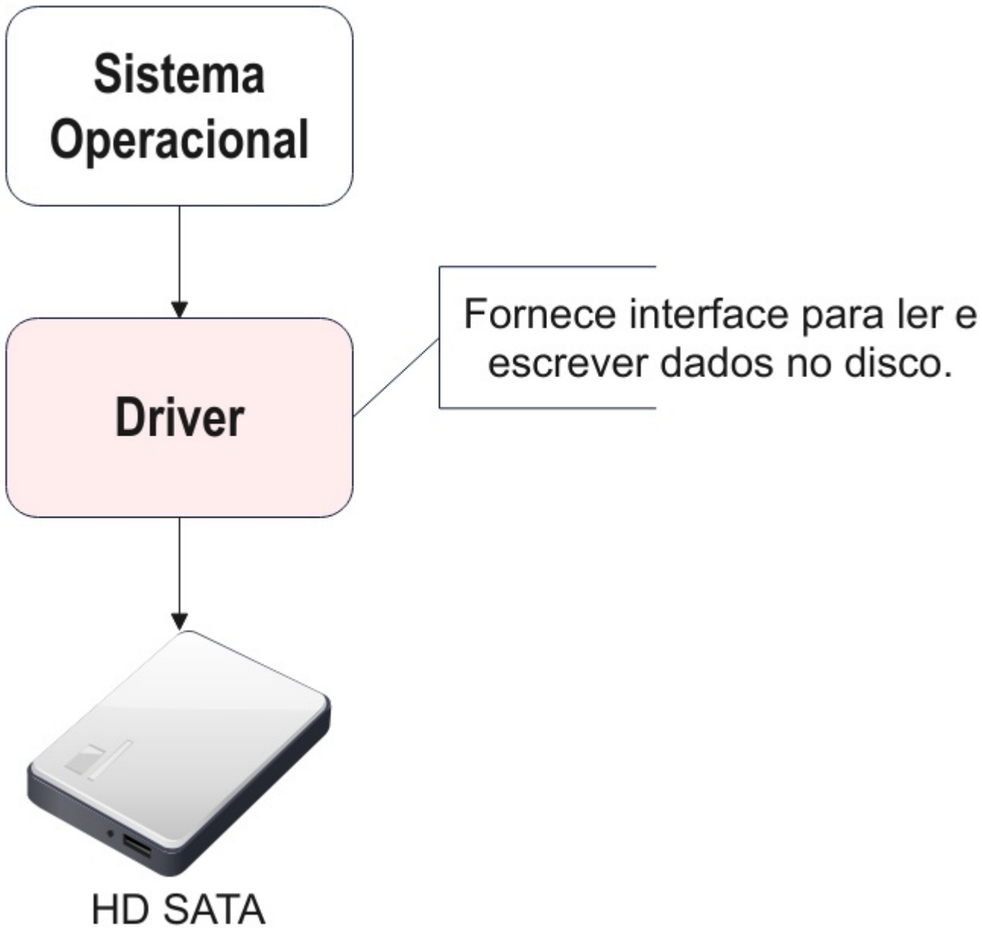
O que são essas abstrações?



Para aprender a usar um disco SATA para gravar e ler arquivos, um programador deveria ler um livro de quase 500 páginas...

Obviamente, ninguém quer lidar com discos em nível de hardware. Como resolver esse impasse? Como o SO "aprende" a usar o disco?

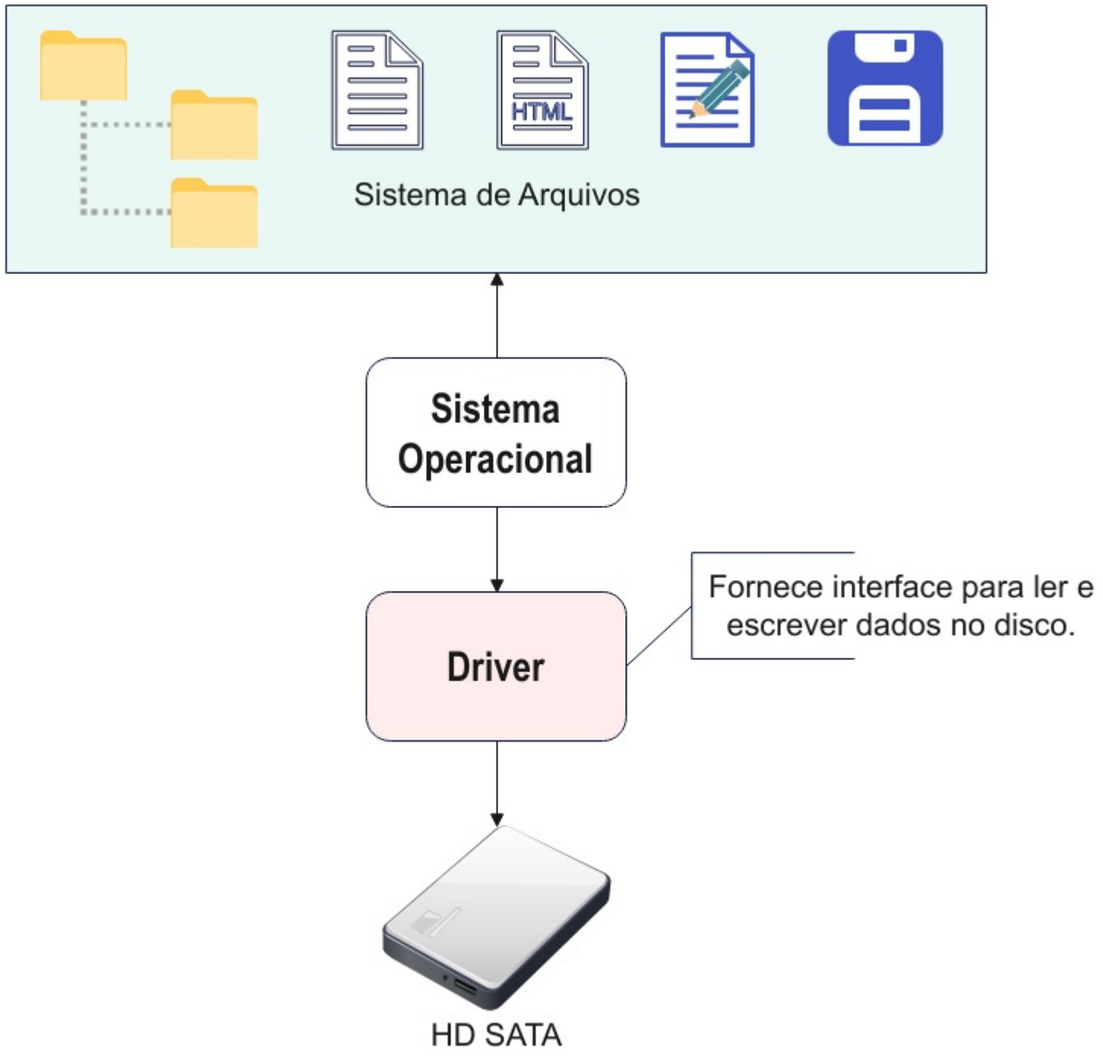
O que são essas abstrações?



Os fabricantes criam softwares especiais chamados de drivers, que fornecem uma interface através da qual é possível utilizar o hardware sem ter que se preocupar com os detalhes de baixo nível.

O driver "ensina" ao SO como usar o hardware. Mesmo assim esse nível é baixo demais para os usuários e aplicativos. Qual a solução então?

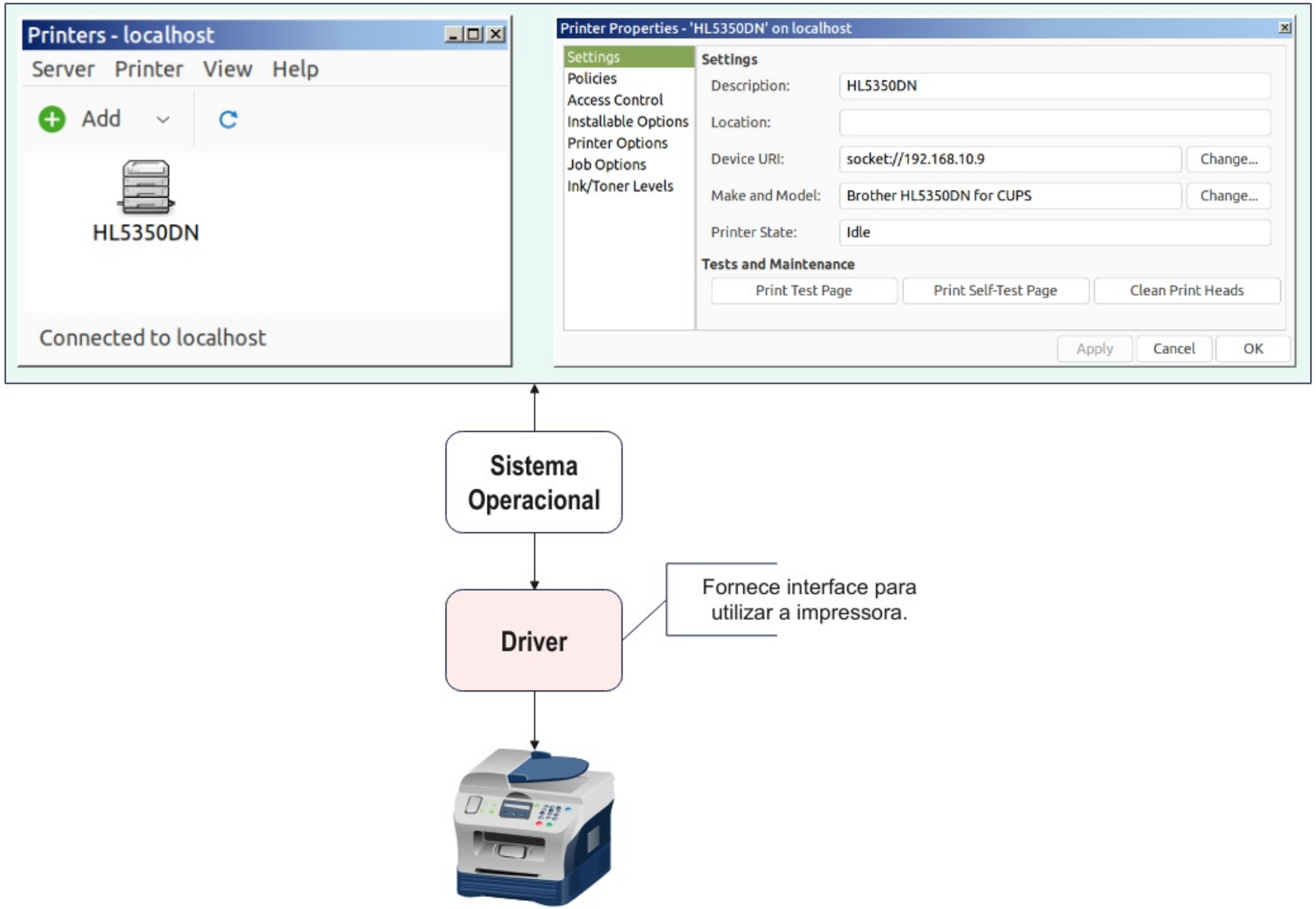
O que são essas abstrações?



O sistema operacional cria uma abstração, o sistema de arquivos. Usando essa abstração podemos ignorar os detalhes de como o hardware funciona.

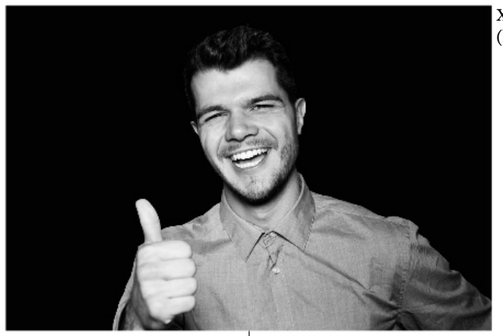
Os sistemas operacionais criam e gerenciam abstrações para permitir que os usuários e programas não precisem lidar com os detalhes complexos do hardware.

O que são essas abstrações?



Sistema operacional como máquina virtual abstrata

O conjunto de abstrações fornecidas pelo sistema operacional cria uma máquina virtual abstrata simples, amigável, limpa, que esconde o hardware: os usuários ou programas interagem com essa máquina virtual amigável e o sistema operacional é responsável pelos detalhes de baixo nível. É a interface entre o usuário e a máquina.



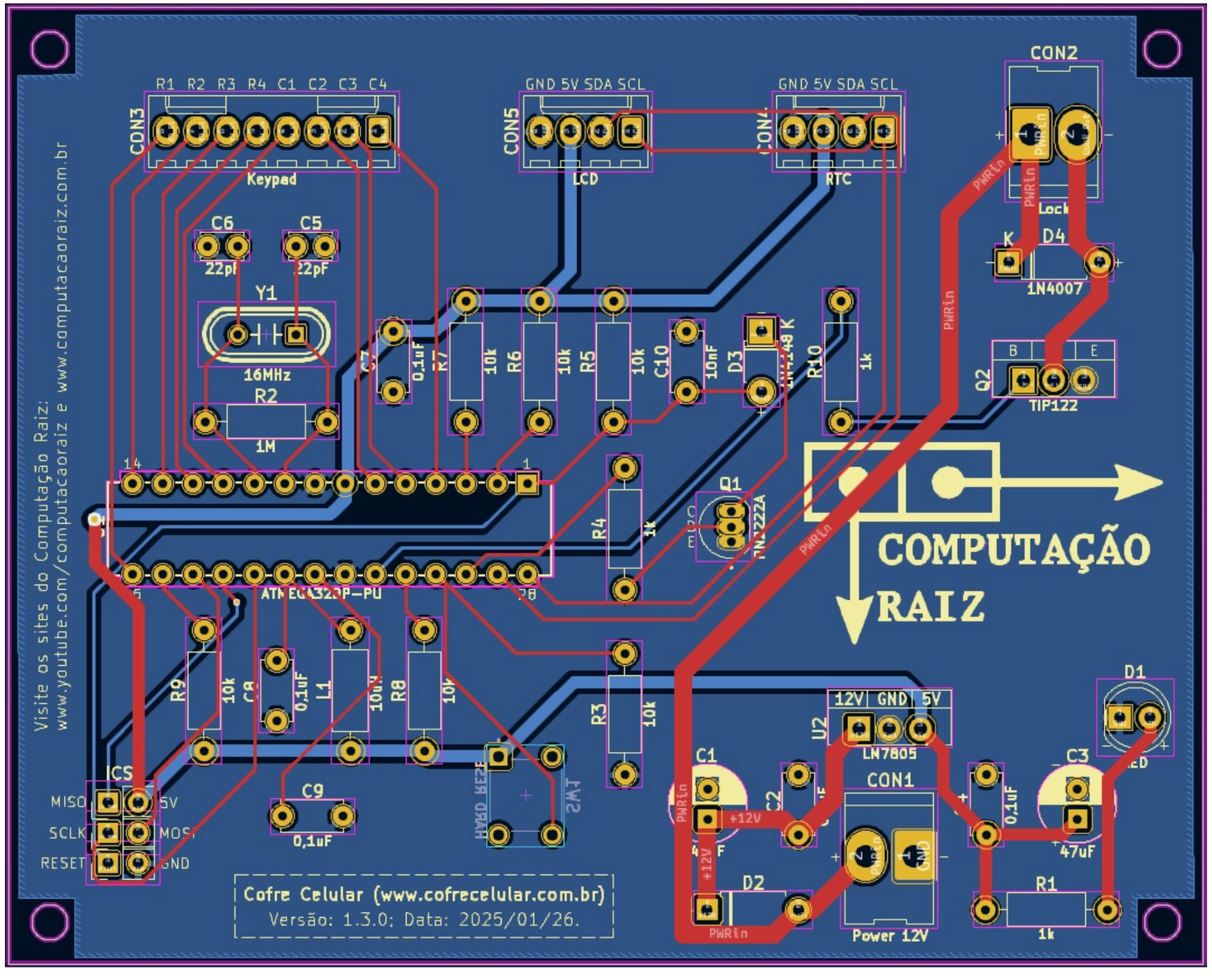
Xylito, no Pixabay
(https://pixabay.com/photos/thumbs-up-man-checked-laugh-6973391/)

Os clientes reais dos sistemas operacionais são os programas aplicativos: são eles que lidam com as abstrações fornecidas pela interface (shell: CLI ou GUI).

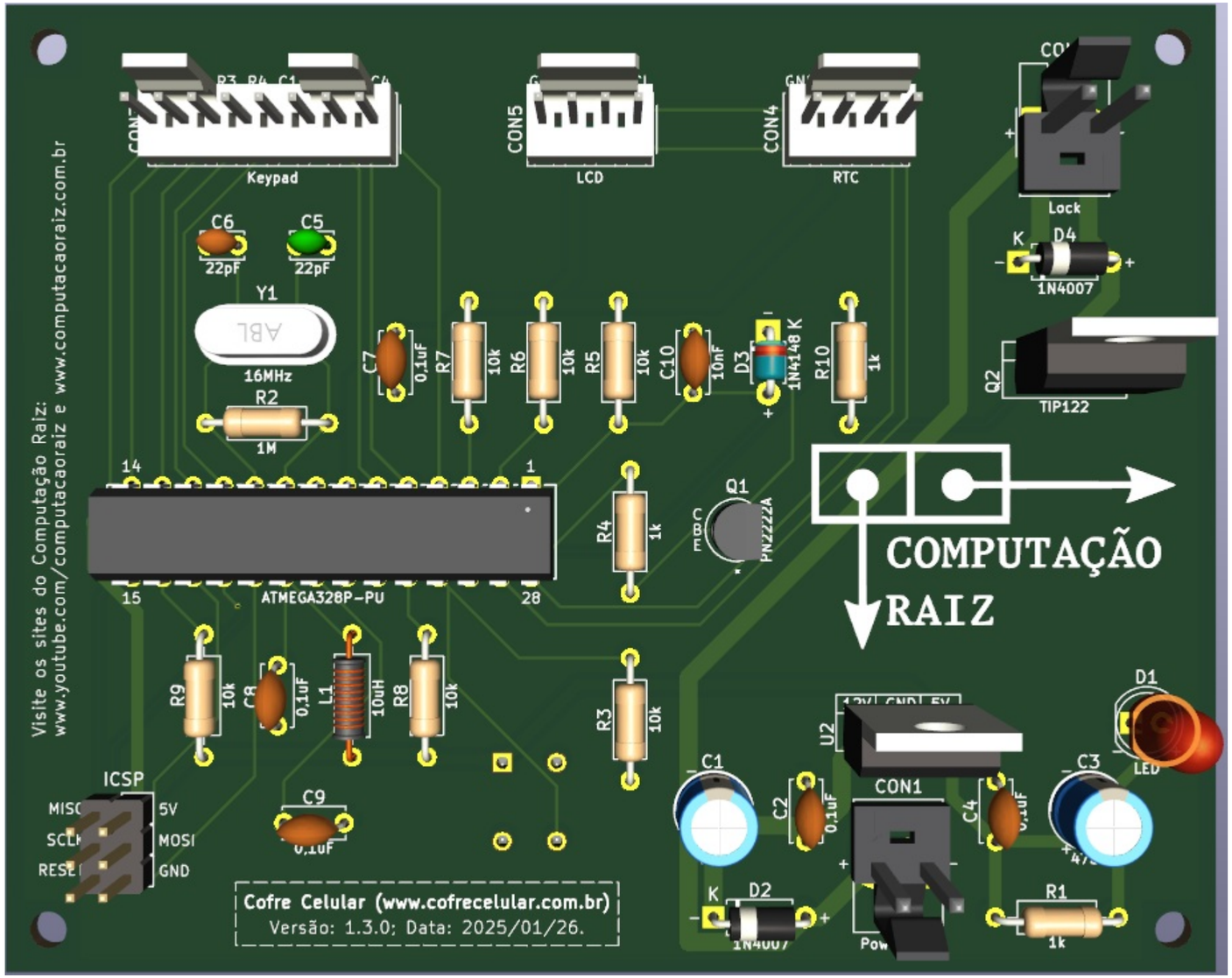
Daria Chudarina, no Pixabay
(https://pixabay.com/vectors/leaves-cat-nature-autumn-cute-7552915/)

Cathe, no Pixabay
(https://pixabay.com/illustrations/ai-generated-mummy-revived-egypt-8684607/)

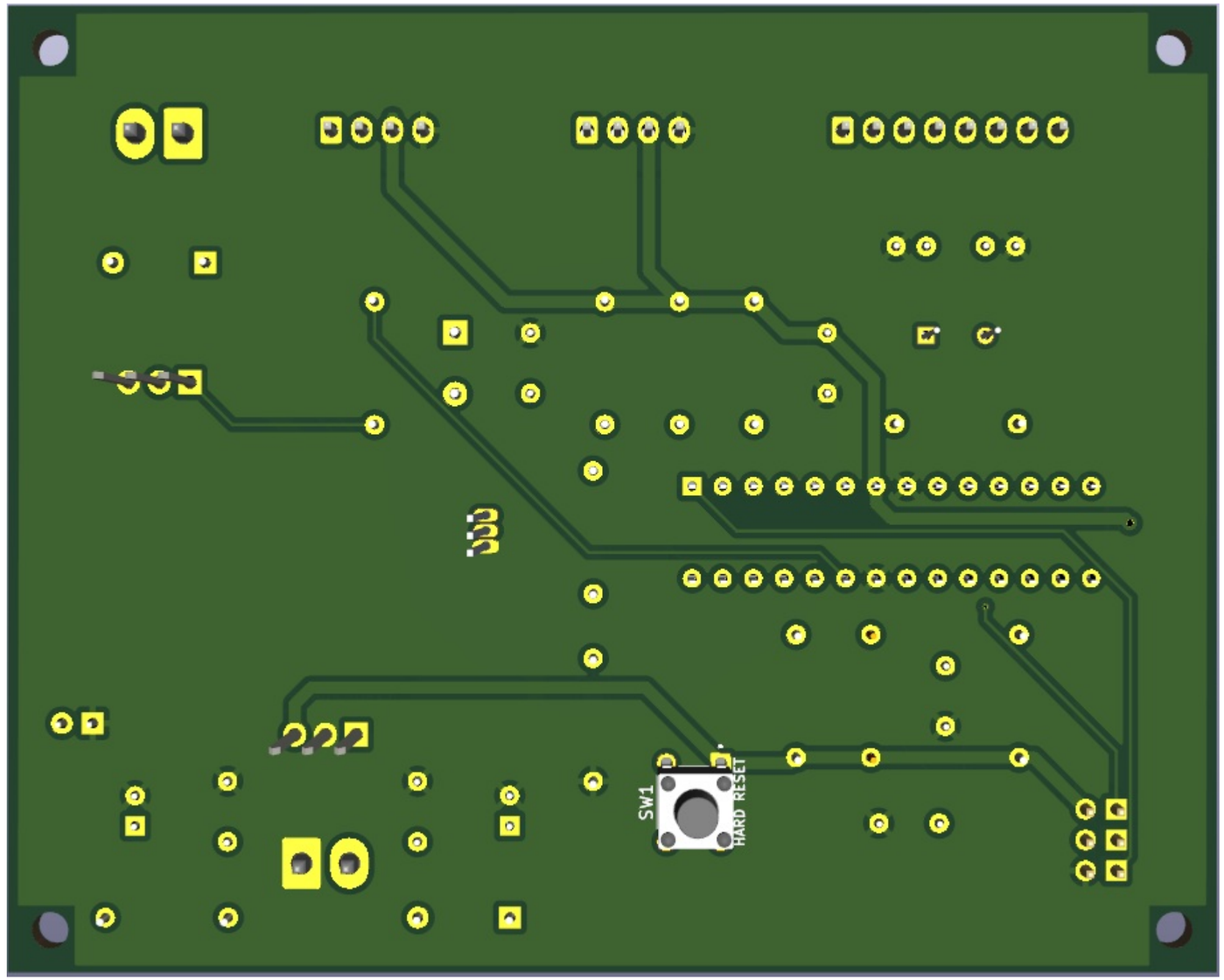
Sistema embarcado como máquina virtual abstrata



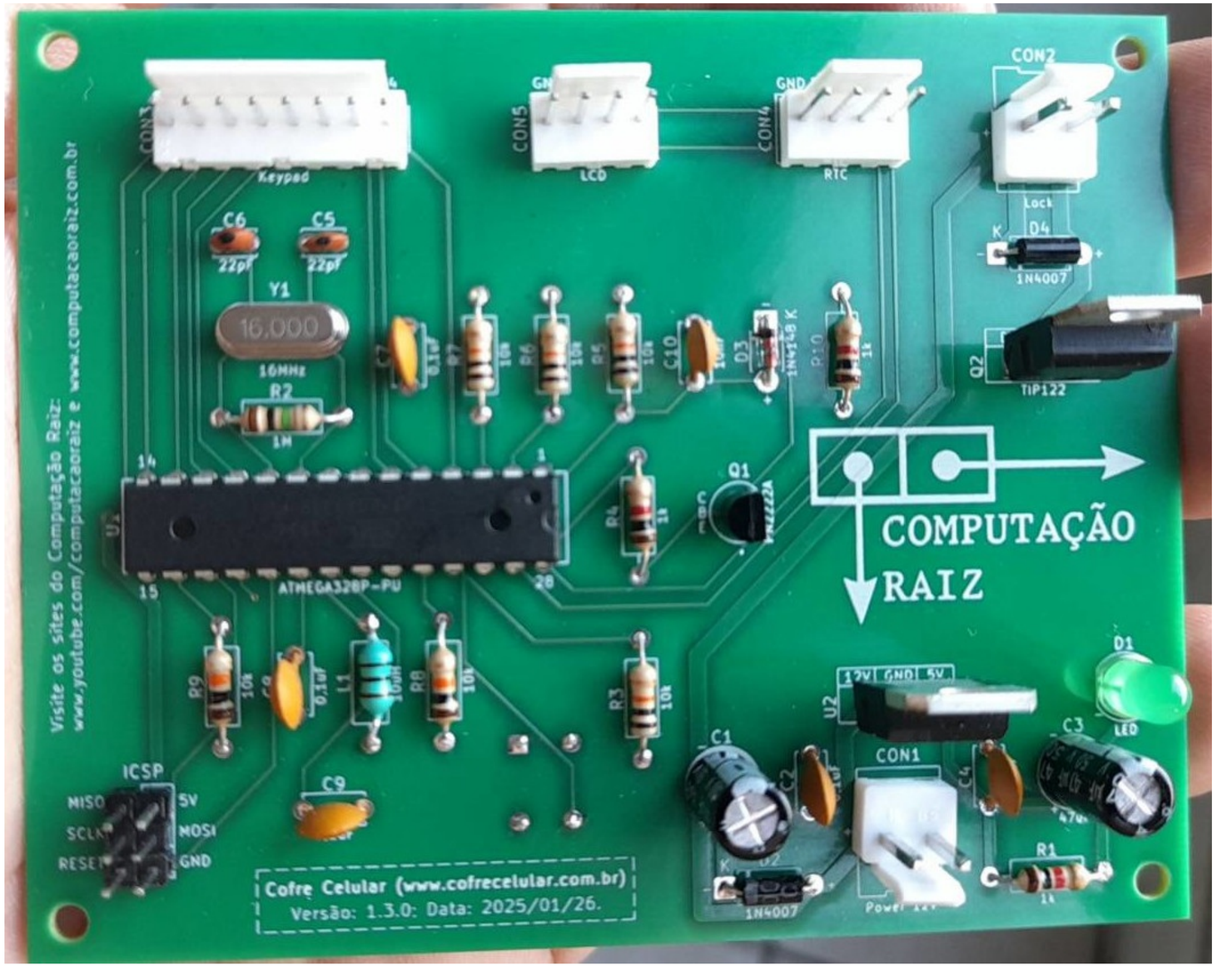
Sistema embarcado como máquina virtual abstrata



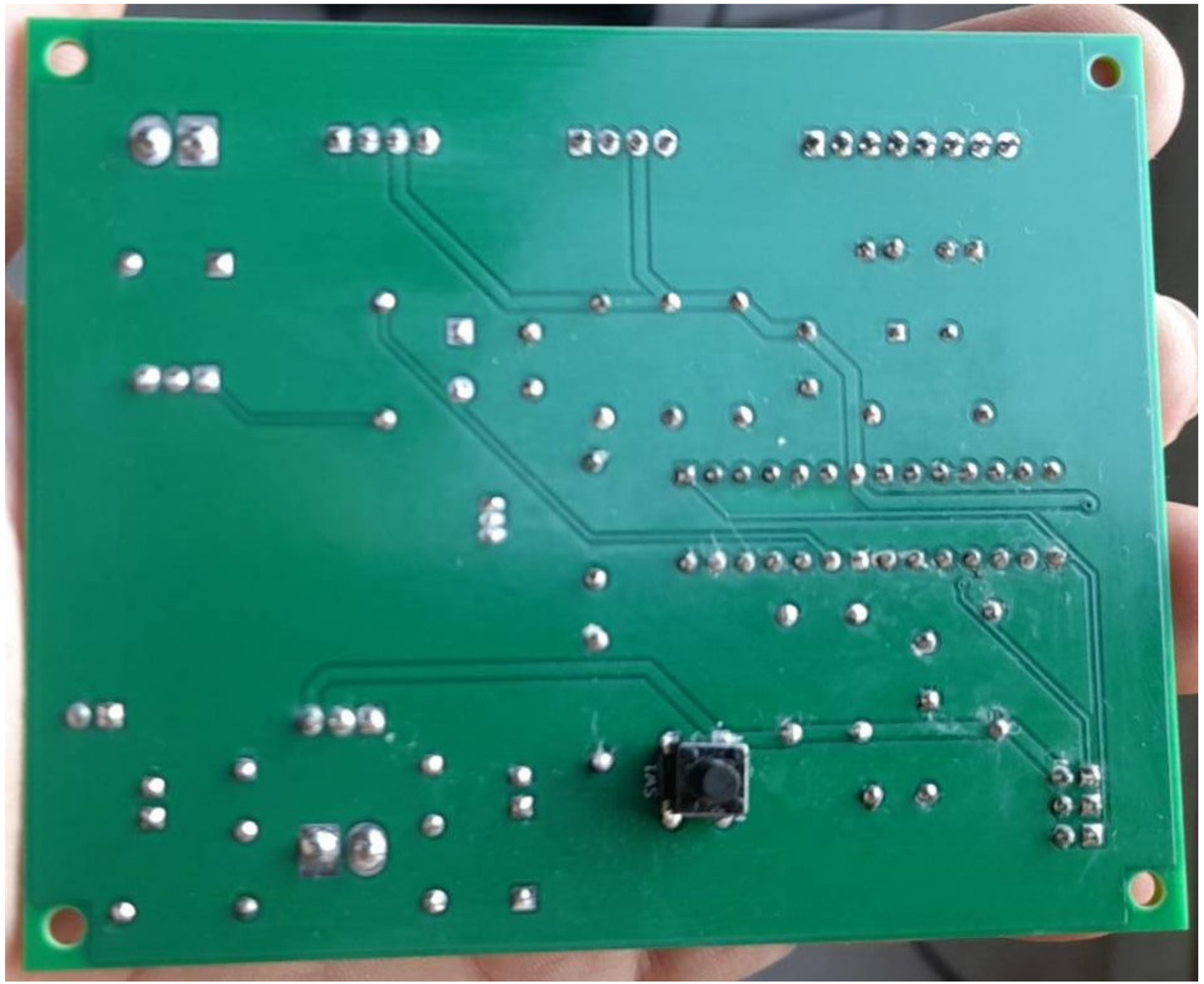
Sistema embarcado como máquina virtual abstrata



Sistema embarcado como máquina virtual abstrata

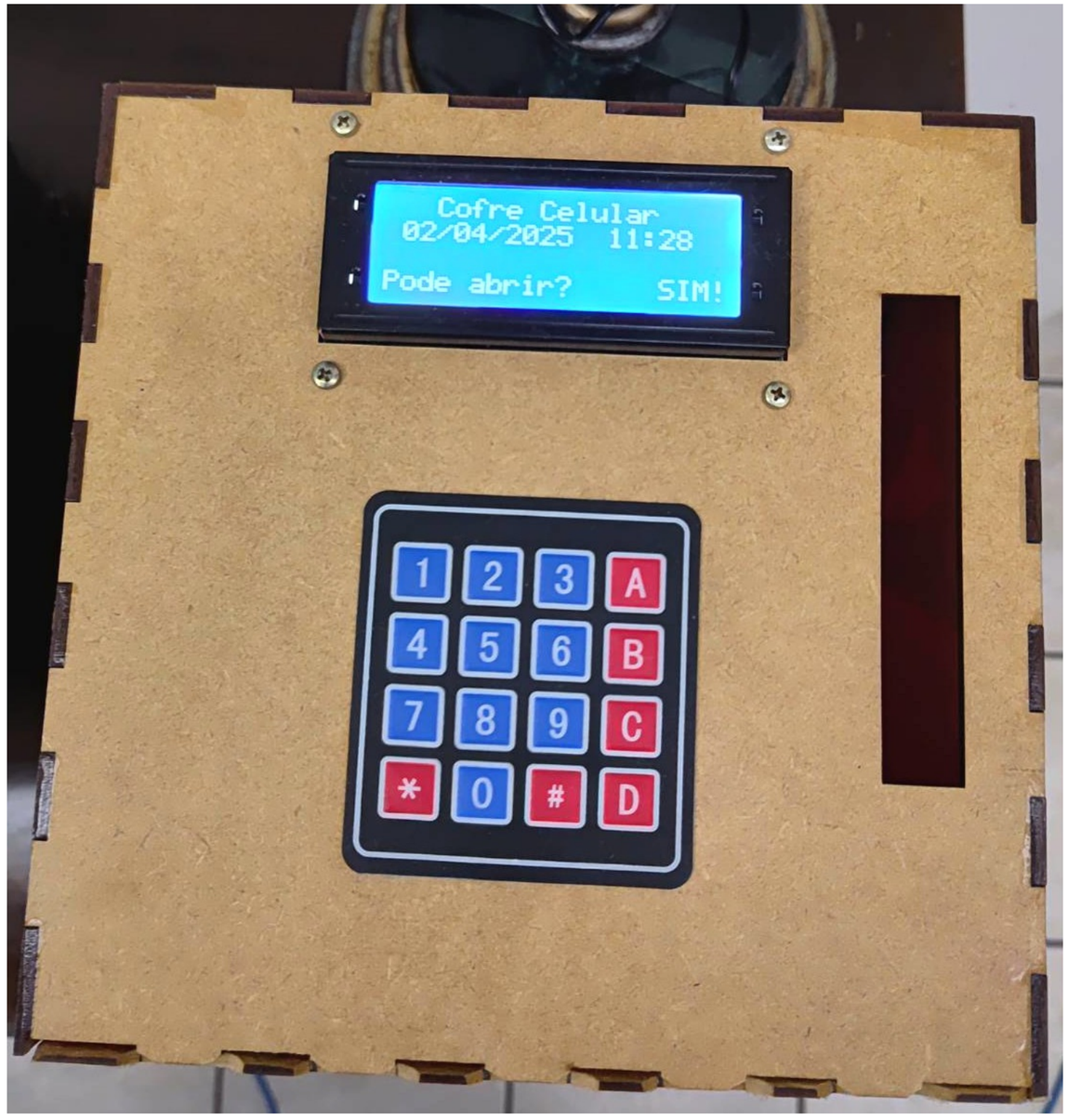


Sistema embarcado como máquina virtual abstrata



Sistema embarcado como máquina virtual abstrata

Sistema embarcado como máquina virtual abstrata



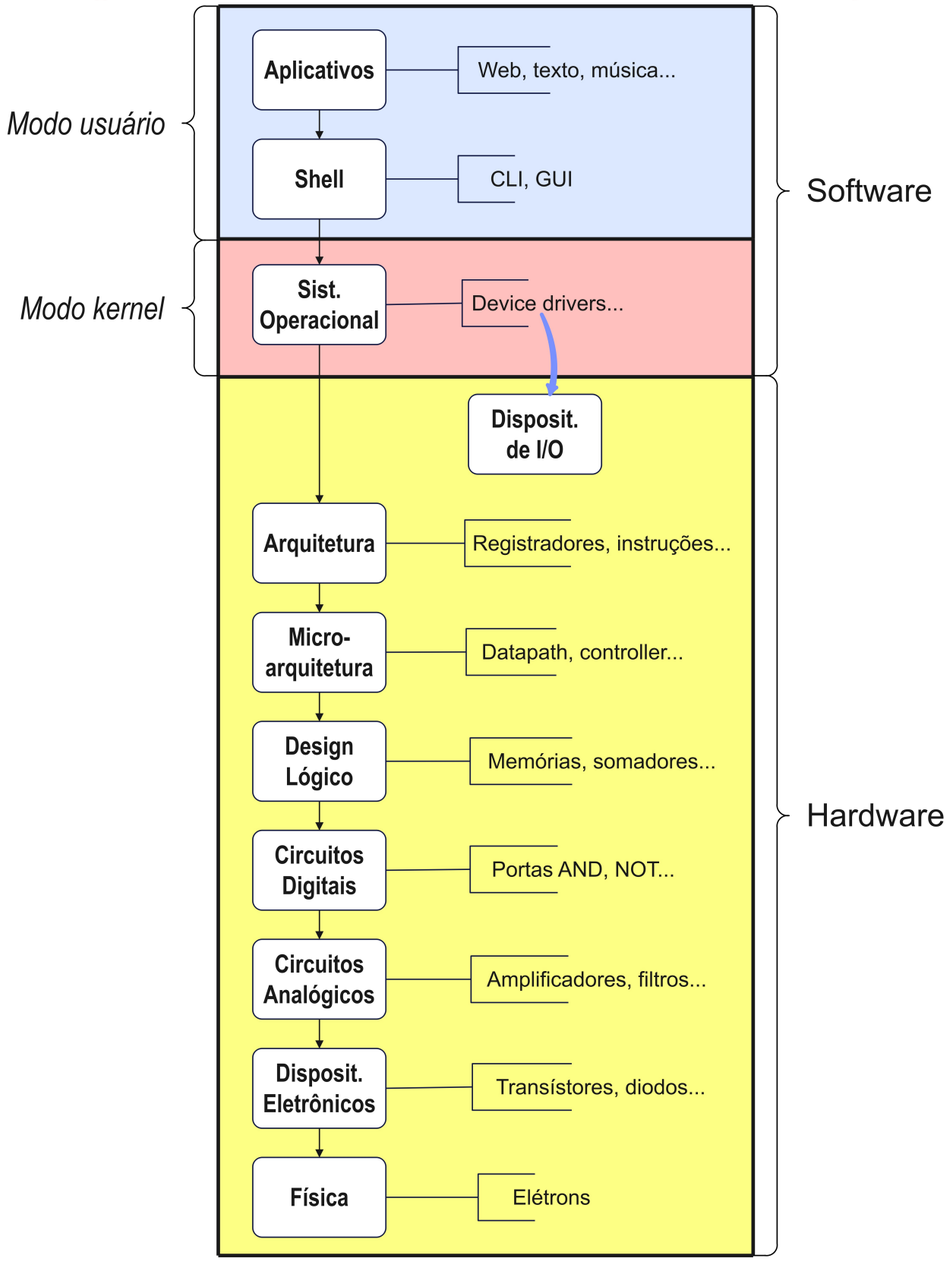
Sistema operacional como gerenciador de recursos

O sistema operacional gerencia o uso de hardware e software de maneira eficiente. De modo didático costumamos dividir um sistema computacional (Linux, por exemplo) em três níveis:

1. **Hardware**
   - **Processador, Memória, Discos, Rede, ...**
2. **Kernel**
   - **Gerenciamento de processos**
   - **Gerenciamento de memória**
   - **Gerenciamento de dispositivos**
   - **Chamadas de sistema**
   - **Autorização**
3. **Processos de usuário**
   - **GUI, CLI, TUI, Shell**
   - **Servidores**
   - **Autenticação**
   - **...**

É o núcleo do sistema operacional que está sempre residente em memória, e que gerencia o hardware e os demais processos de usuário.

Alguns conceitos básicos importantes

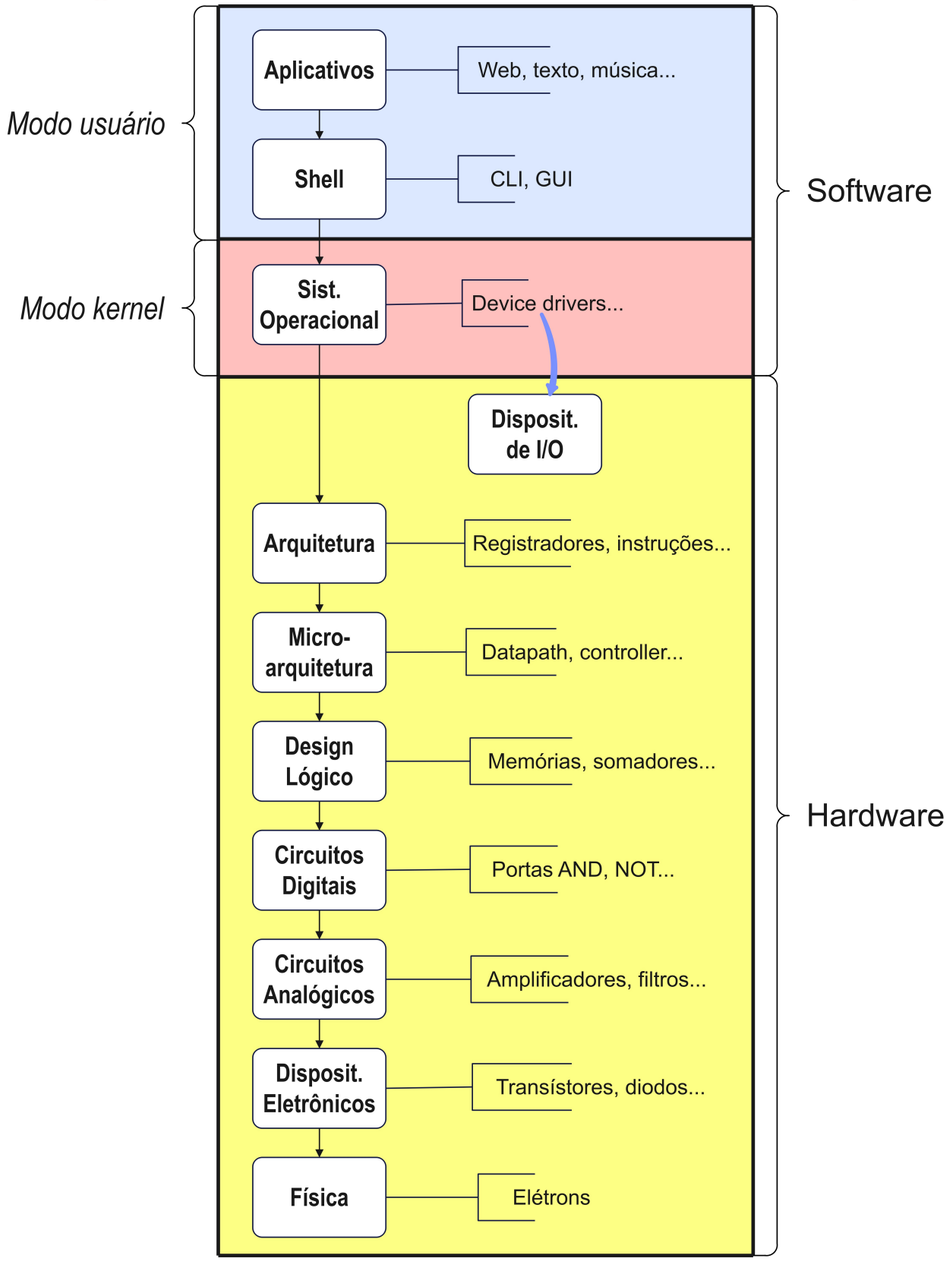


As CPUs dos computadores operam em dois "modos" principais:

- Modo Kernel: **acesso completo a todo o hardware; pode executar qualquer instrução que afete o controle da máquina ou faça I/O.**

- Modo Usuário: **apenas uma parte das instruções está disponível; não pode rodar instruções de controle ou I/O.**

Alguns conceitos básicos importantes

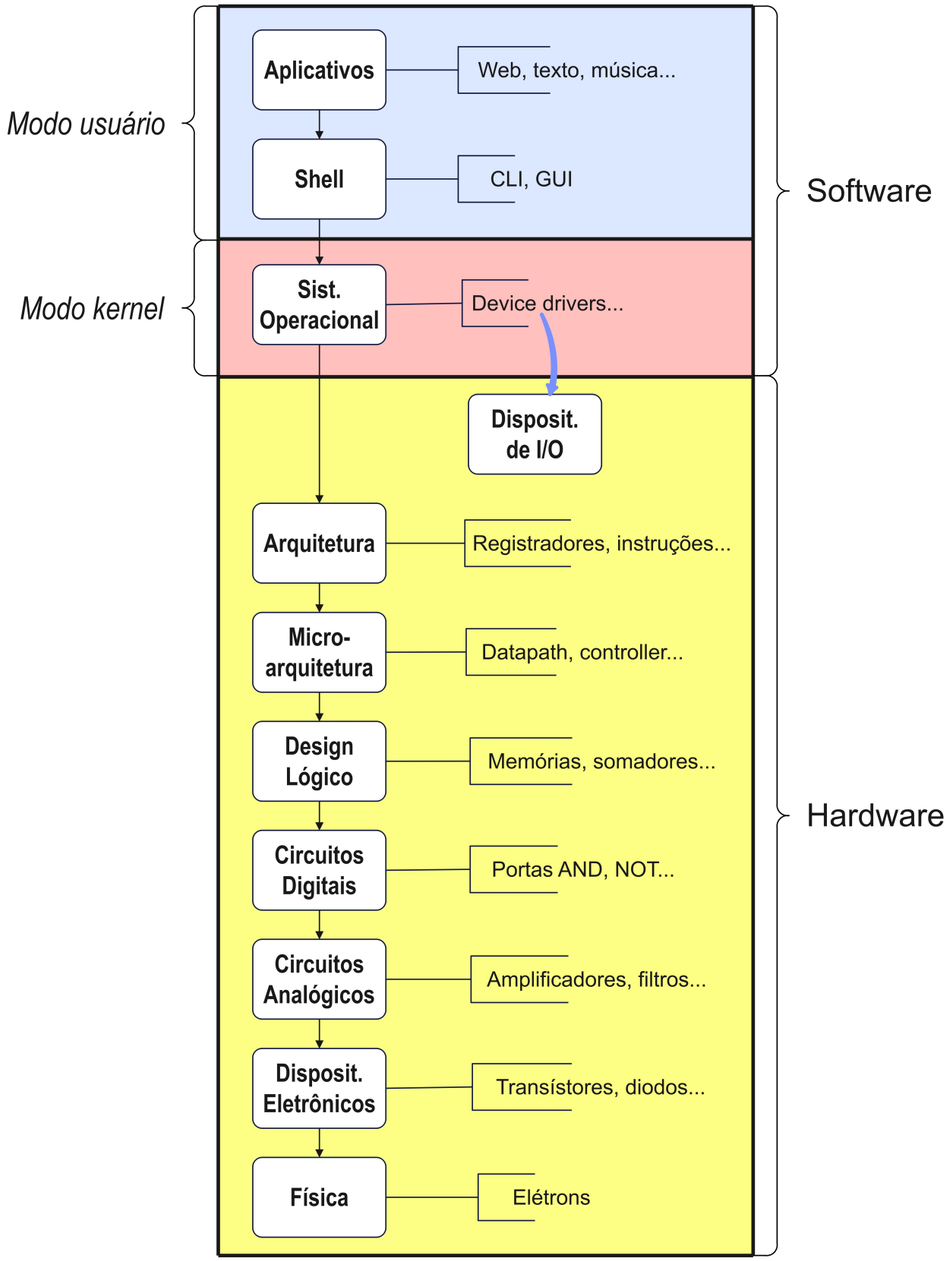


Também temos dois "espaços" de memória que são separados:

- Espaço Kernel: área de memória onde o kernel está; só o kernel tem acesso à sua própria área de memória

- Espaço Usuário: área de memória para armazenar os processos de usuário; em geral cada processo tem uma área separada dos demais

Alguns conceitos básicos importantes



Shell: **é um programa que disponibiliza os serviços do sistema operacional para os usuários ou outros programas. Pode ser:**
- Interface de Linha de Comando (CLI)
- Interface Gráfica do Usuário (GUI)
- Interface de Texto do Usuário (TUI)

Obs.: pode ser local ou remoto.

O Kernel: o que é e funções principais

O Kernel é o núcleo do sistema operacional que está sempre residente em memória (no espaço kernel) e que gerencia o hardware e demais processos de usuário. As 4 funções principais do Kernel são:

- **Gerenciamento de processos**

- **Gerenciamento de memória**

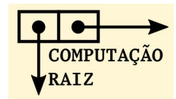
- **Gerenciamento de dispositivos**

- **Chamadas de sistema**

Exerce diversas outras funções, por exemplo:

- **Autorização**
- **Pseudodispositivos**

O Kernel: gerenciamento de processos



O gerenciamento de processos refere-se ao agendamento, início, pausa, reinício e finalização de processos.

PERGUNTA FUNDAMENTAL:

Um computador com uma CPU com um único core executa mais de um programa (processo) ao mesmo tempo?

O Kernel: gerenciamento de processos

O gerenciamento de processos refere-se ao agendamento, início, pausa, reinício e finalização de processos.

PERGUNTA FUNDAMENTAL:

Um computador com uma CPU com um único core executa mais de um programa (processo) ao mesmo tempo?

NÃO! A CPU só consegue executar um único processo de cada vez! Mesmo que você tenha vários programas abertos no computador, apenas 1 único processo está em execução em um determinado instante no tempo.

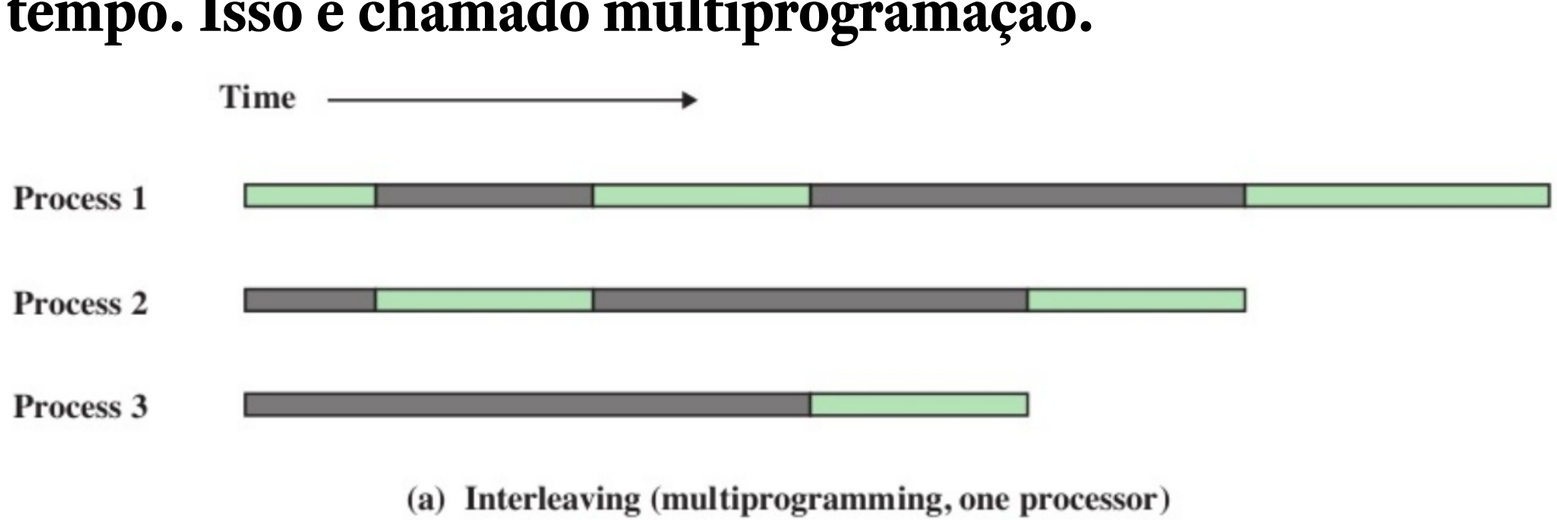
A CPU te "engana": ela executa cada processo por uma minúscula fração de tempo (time slice) e faz a troca de contexto (context switch), ou seja, passa a executar outro processo por uma pequena fração de tempo.

O Kernel: gerenciamento de processos

O gerenciamento de processos refere-se ao agendamento, início, pausa, reinício e finalização de processos.

MULTIPROGRAMAÇÃO:

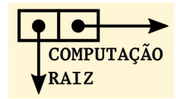
Como a troca de contexto é feita muito rapidamente, em escala de tempo imperceptível aos humanos, PARECE que a CPU está executando vários processos ao mesmo tempo. Isso é chamado multiprogramação.



(a) Interleaving (multiprogramming, one processor)

E onde o Kernel entra nessa história?

O Kernel: gerenciamento de processos



O gerenciamento de processos refere-se ao agendamento, início, pausa, reinício e finalização de processos.

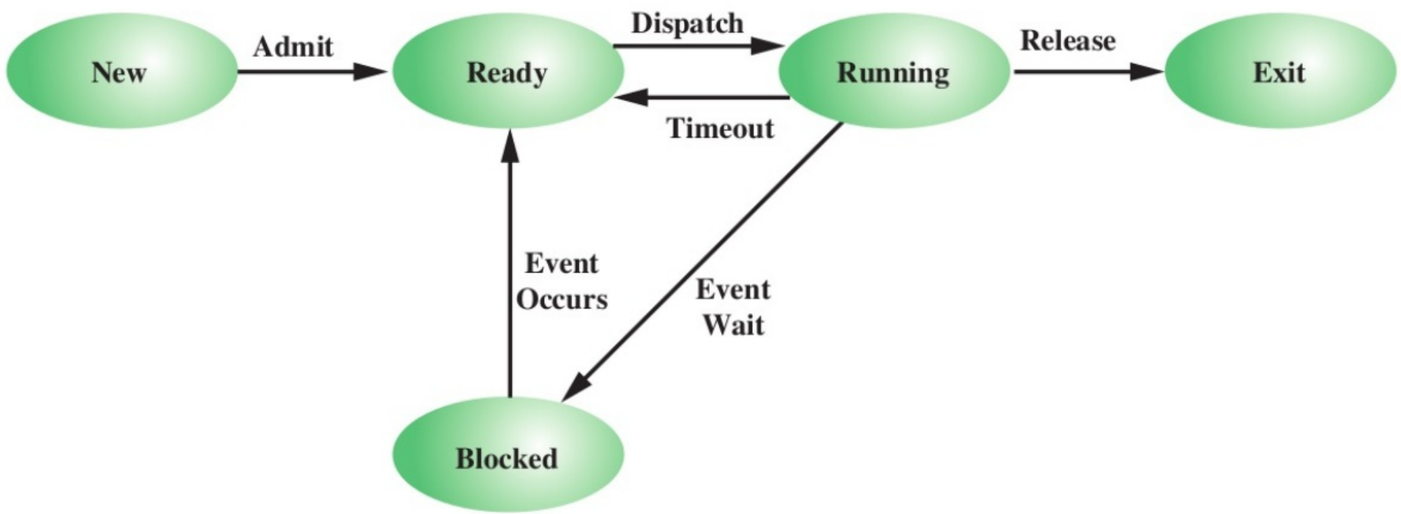
O kernel é responsável pela troca de contexto!

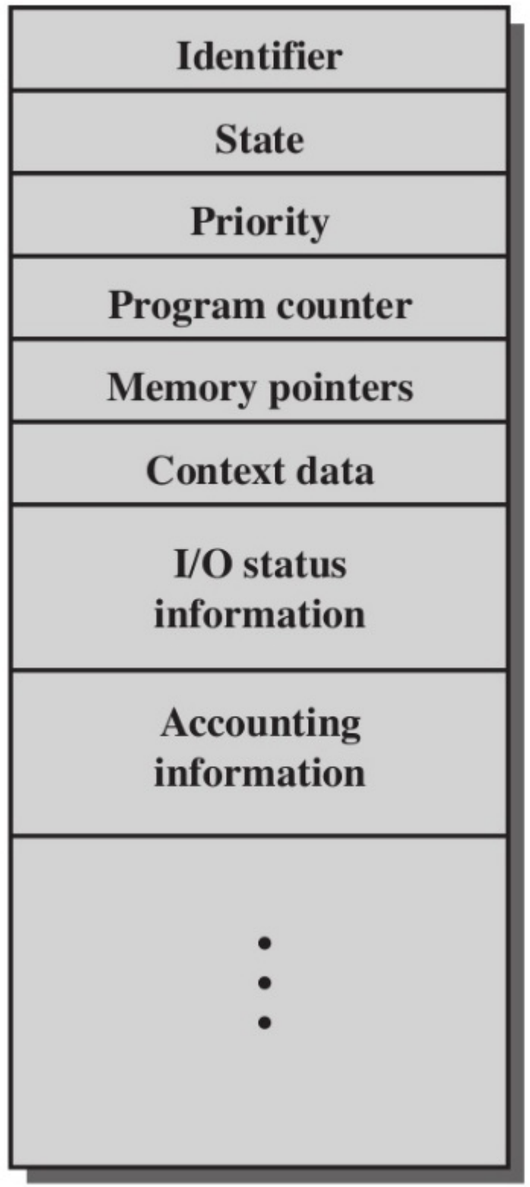
1. **A CPU interrompe o processo de usuário atual (p. ex. se o time slice terminar), vai para o modo kernel e passa o controle para o kernel.**
2. **O kernel salva o estado atual da CPU e da memória em uma estrutura chamada de bloco de controle do processo (PBC: process control block), que é necessário para o reinício futuro do processo.**
3. **O kernel realiza qualquer tarefa que possa ter ficado pendente durante o úlitmo time slice (coletar dados de input/output por exemplo).**
4. **O kernel verifica todos os processos em espera e escolhe o que será executado através do seu agendador de tarefas.**
5. **O kernel busca as informações de CPU e memória desse processo e prepara a CPU.**
6. **O kernel informa o time slice para o processo à CPU.**
7. **O kernel coloca a CPU em modo usuário e passa o controle da CPU para o processo.**

O Kernel: gerenciamento de processos

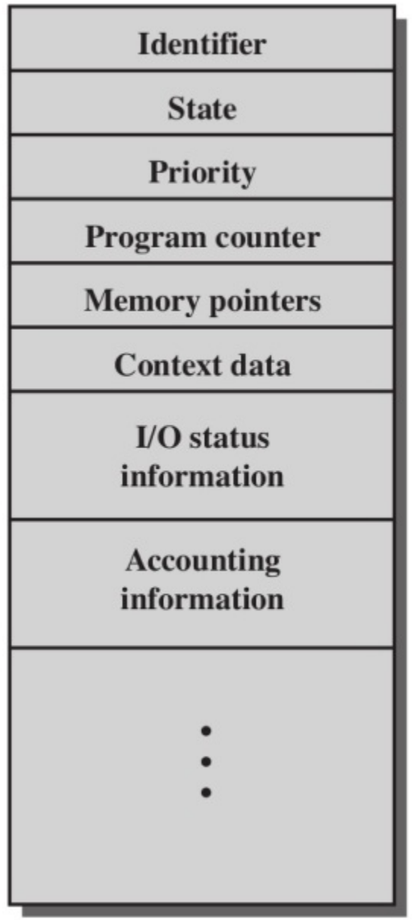
Mas afinal: o que é um processo?

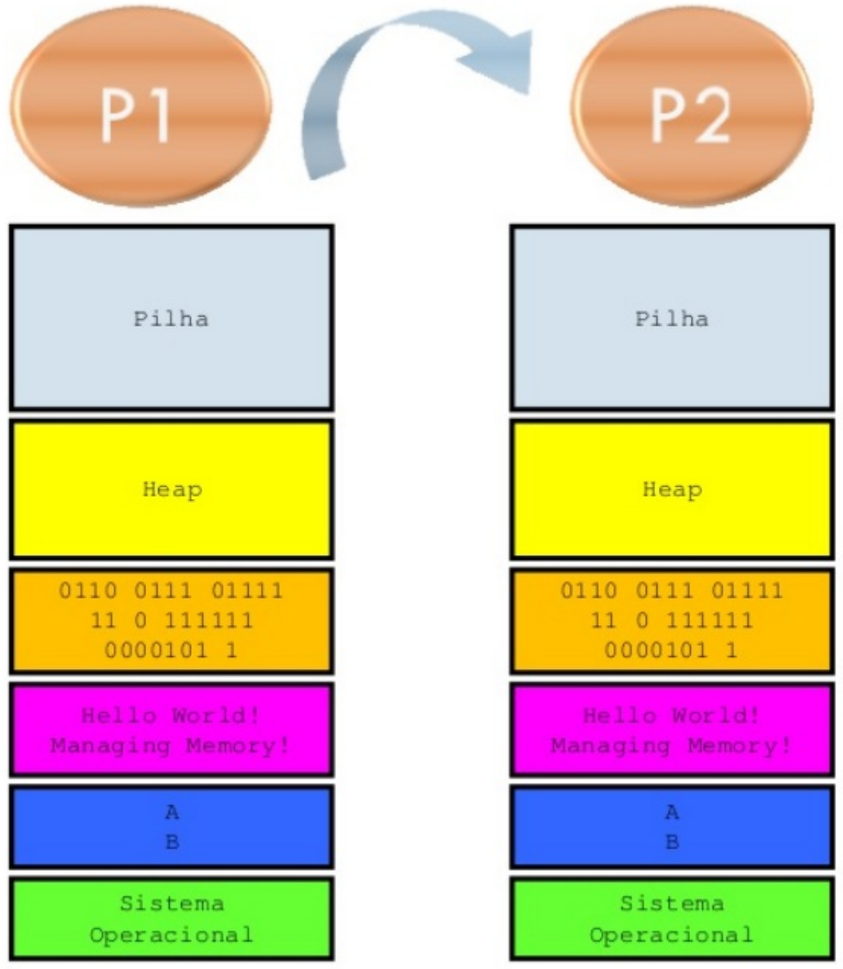
- **é um programa em execução, uma instância de um programa executando no computador**
- **pode ter a posse/controle de recursos do computador (registradores, memória, descritores, I/O, etc...)**
- **pode estar em diversos estados**
- **o SO mantém um "bloco de controle de processo" (PCB) para CADA processo**

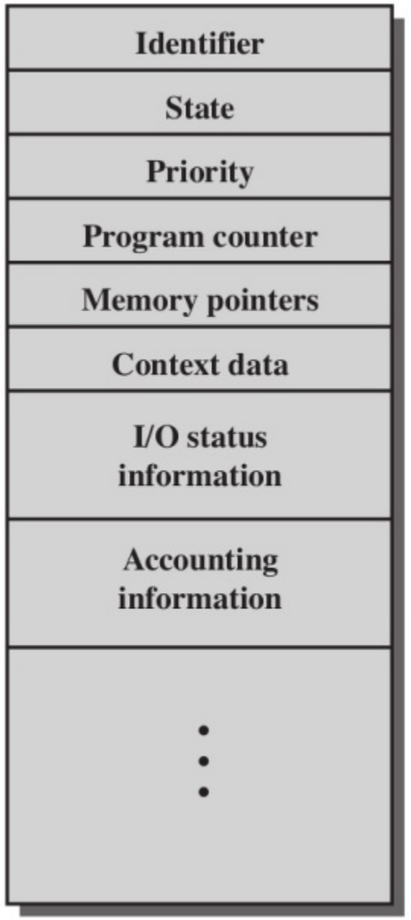




O Kernel: gerenciamento de processos







O Kernel: gerenciamento de memória

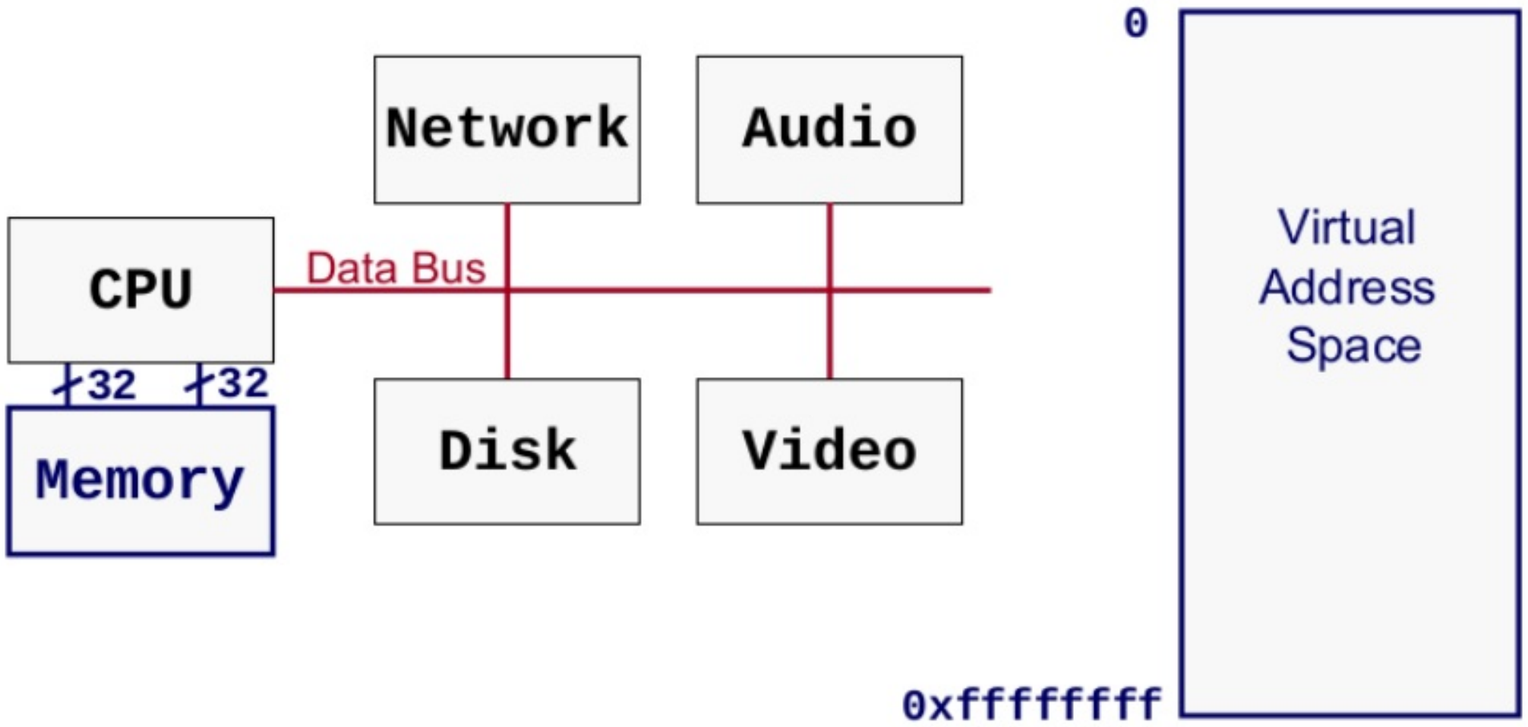
Durante a troca de contexto e em diversos outros momentos o kernel precisa gerenciar toda a memória do computador, incluindo:

- **sua própria área de memória (espaço kernel)**
- **a área de memória de cada processo (espaço usuário)**
- **áreas de memória compartilhada entre processos de usuário**
- **área de troca de memória (swap)**

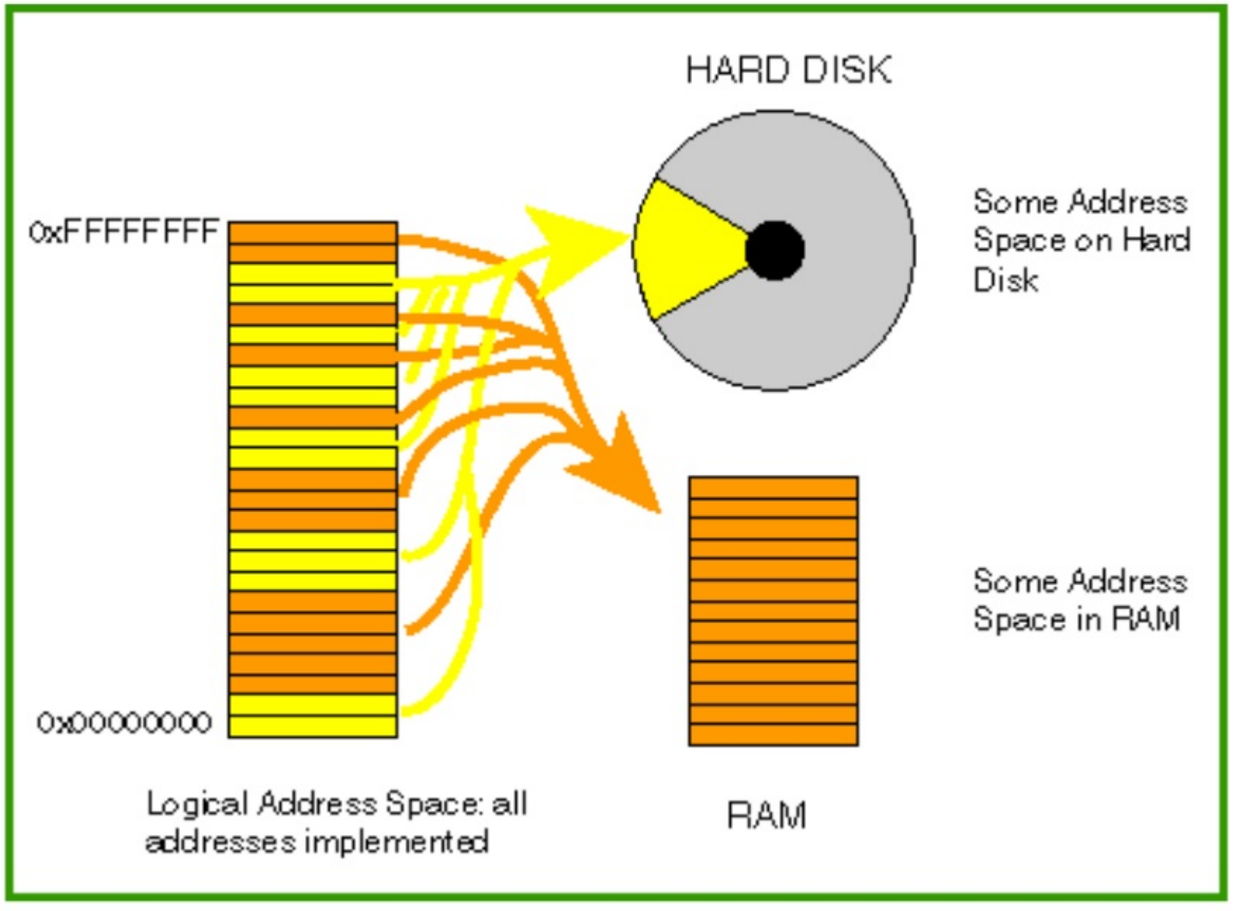
CPUs modernas incluem um hardware específico para auxiliar o kernel, o MMU (Memory Management Unit), Unidade de Gerenciamento de Memória, que fornece ao kernel mecanismos de implementação do conceito de memória virtual, que permite que os processos de usuário usem todo o espaço de endereçamento de memória disponível (mais memória do que realmente existente).

A MMU faz o mapeamento entre os endereços virtuais e reais através de uma mapa de memória (tabela de paginação - Page Table) gerenciado pelo kernel.

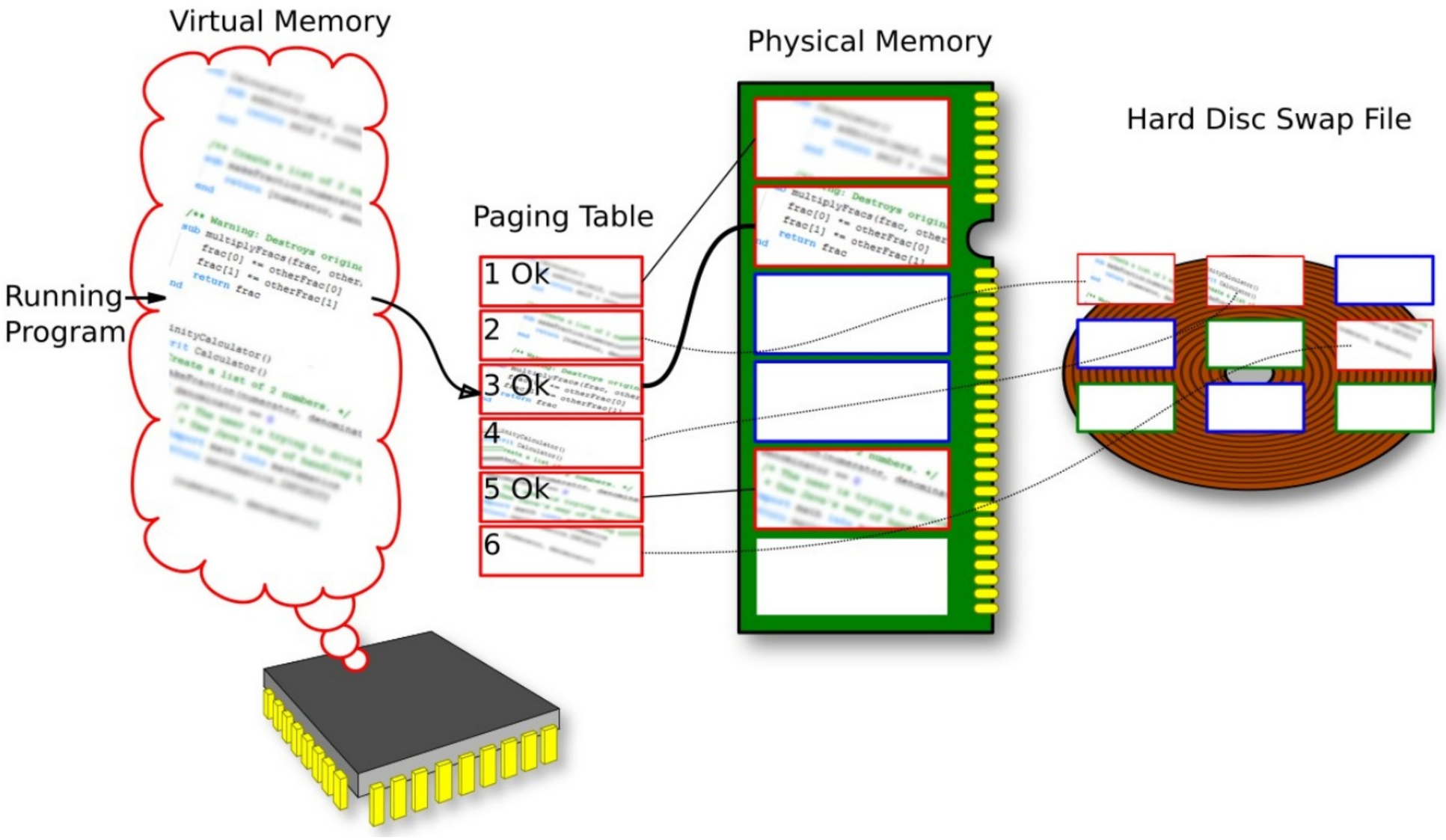
O Kernel: gerenciamento de memória



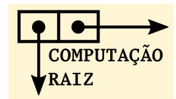
O Kernel: gerenciamento de memória



O Kernel: gerenciamento de memória



O Kernel: gerenciamento de dispositivos



É relativamente simples para o Kernel trabalhar com diversos dispositivos tais como teclados, mouses, discos rígicos, sistemas de som, monitores, placas de vídeo, placas de rede, mesas digitalizadores, impressoras, scanners, etc. POR QUÊ?

Todo o acesso é feito em modo kernel e os fabricantes fornecem os drivers com os programas específicos de acesso ao dispositivo.

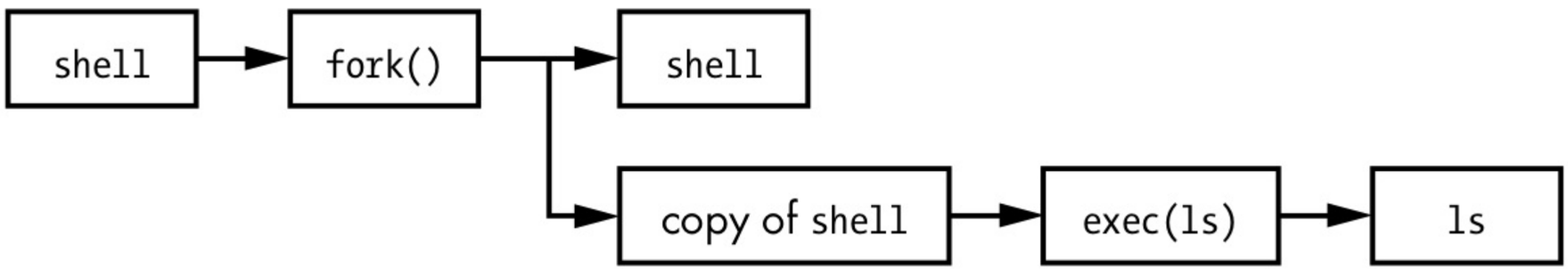
O kernel não precisa saber como acessar o dispositivo, ele precisa saber qual driver usar e passar a solicitação para o driver!

O Kernel: chamadas de sistema

São "portas de entrada" para que os processos de usuário possam acessar e solicitar serviços aos kernel. As System Calls (syscalls) realizam tarefas que um processo de usuário não pode realizar por conta própria (como acessar um arquivo no HD, por exemplo). Uma syscall é uma interação entre um processo e o kernel.

Chamadas de sistema também são importantes para iniciar processos, por exemplo:

- **`fork()`** **o kernel cria uma cópia do processo**
- **`exec()`** **o kernel carrega e inicia algum programa**



Com exceção de um processo chamado INIT, todos os outros processos no Linux são iniciados com um fork() e, na maioria das vezes, com exec().

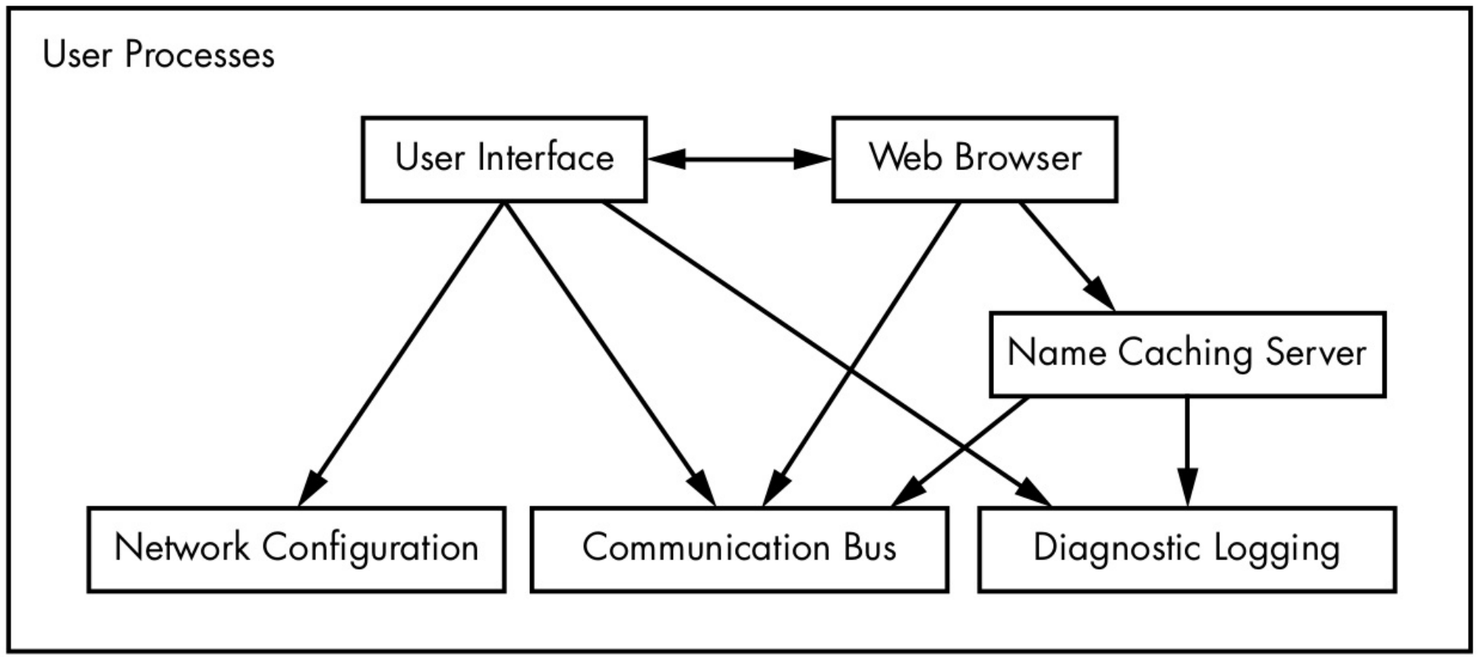
Processos de usuário: o que são?

Todo programa que está em execução no computador, exceto o Kernel, é um processo de usuário. Formam a maioria absoluta dos programas em execução:

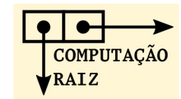
- **navegadores web**
- **players multimídia**
- **processadores de texto**
- **editores de código**
- **leitores de PDF**
- **...**

Mas também:

- **serviços de rede**
- **serviços de comunicação**
- **serviços servidores diversos**
- **serviços diagnósticos**
- **...**



O que é um usuário?



Um usuário (user) é uma entidade que pode executar processos e ser dono de arquivos. Um usuário está associado a um nome (username) e a um identificador (UID).

Todo processo tem um dono (owner), e falamos que o processo roda como o usuário que é seu dono. Um usuário pode gerenciar seus processos, mas não pode interferir no processo de outros usuários.

Sistemas operacionais têm muito mais usuários do que pessoas reais que usam o sistema, e esses usuários existem para executar processos específicos.

O usuário mais importante é o root, que é o superusuário administrador que pode fazer tudo no sistema (inclusive estragar o sistema inteiro).

Também existem grupos (group) de usuários identificados por um nome e um GID.

O que é um usuário?

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:115::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:116::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
```

O que é um usuário?

```
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
lightdm:x:114:121:Light Display Manager:/var/lib/lightdm:/bin/false
speech-dispatcher:x:115:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:116:123:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:117:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:118:125::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:119:126:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:120:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:121:127::/nonexistent:/bin/false
colord:x:122:128:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:123:129::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:124:130:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
abrantesasf:x:1000:1000:,,,:/home/abrantesasf:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
nvidia-persistenced:x:125:134:NVIDIA Persistence Daemon,,,:/nonexistent:/usr/sbin/nologin
mysql:x:126:136:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:127:137:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
gdm:x:129:139:Gnome Display Manager:/var/lib/gdm3:/bin/false
opendkim:x:130:140::/run/opendkim:/usr/sbin/nologin
fwupd-refresh:x:131:141:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
postgres:x:132:142:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
monitoria:x:1001:1001:monitoria,,,,:/home/monitoria:/bin/bash
ntp:x:133:144::/nonexistent:/usr/sbin/nologin
```