Sistemas Operacionais

Rodrigo Rubira Branco
rodrigo@kernelhacking.com
rodrigo@fgp.com.br

Tipos de Sistemas Operacionais

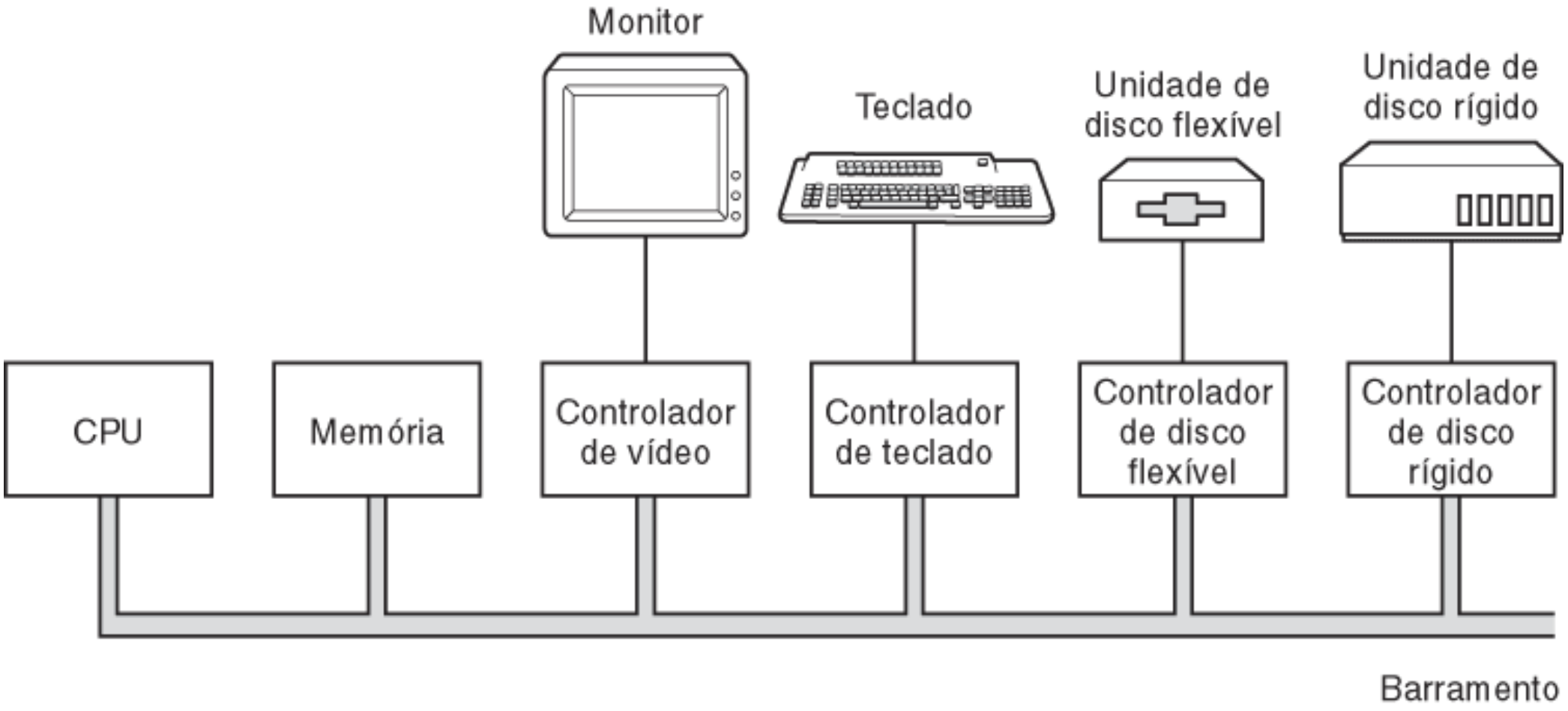
De Sistemas Embarcados (PalmOS,WinCE,WinXPEmbbeded,Linux)

- Hardware simples, especifico e com funcionalidade unica/limitada
- Pequenos, consumos de recursos minimos
- Sem componentes mecanicos (normalmente)
- NetBSD em torradeira?

De Cartoes Inteligentes ou outros dispositivos pequenos

- Exemplo: Certificacao Digital
- Exemplo: Tokens OTP (one-time-pass)
- Rodam em ROM ou similar (flash) e possuem limitacoes serias de memoria e de consumo de energia
- Usos especificos (cartoes bancarios por exemplo)

Hardware de Computadores e o Sistema Operacional



Arquitetura Simplificada

Processador

- Cerebro do computador, busca instrucoes a serem executadas na memoria, executa e busca proxima instrucao.

- Acesso a registradores muito mais rapido do que a memoria, entao as instrucoes (que diferem para cada arquitetura) normalmente manipulam os mesmos (copia-se da memoria para o registrador, faz-se as operacoes e depois copia-se novamente para a memoria)

Registradores Especiais

- Contador de Programa (PC ou EIP)

- Armazena o endereco da memoria da proxima instrucao a executar

- Ponteiro de Pilha (Stack Pointer ou ESP)

- Aponta o topo da pilha atual na memoria.
- Existe uma estrutura de pilha para cada procedimento chamado que ainda nao encerrou.
- Ela contem parametros de entrada e variaveis locais.

Ponteiro de Pilha (entendendo seu funcionamento)

- Ao se executar uma funcao, seus parametros serao passados via stack para a mesma
- Temos portanto:

ESP + 4 -> Parametro 1
ESP -> Return Address

- Se esta funcao definir 1 variavel local, a stack ira mudar para

ESP +8 -> Parametro 1
ESP +4 -> Return Address
ESP -> Variavel Local

- Porque? Porque o ESP sempre aponta para o TOPO da Pilha

Ponteiro de Pilha (entendendo seu funcionamento)

- Obviamente fica inviavel determinar facilmente onde os parametros da funcao estao sendo passados

- Para isso, existe outro registrador (EBP), que deve ter seu valor salvo no inicio da funcao e restaurado no final dela

- Apos salvar o valor original do EBP, a funcao iguala ele ao ESP e passa a utilizar apenas o EBP para referenciar suas variaveis (o ESP continuara sendo modificado), mas o EBP nao, portanto os parametros passados sao facilmente encontrados

Ponteiro de Pilha (entendendo seu funcionamento)

- Tem-se portanto:

ESP + 8 = EBP + 8 Parametro 1
ESP + 4 = EBP + 4 Return Address
ESP = EBP EBP Salvo

- Cada variavel local declarada, ira modificar o valor de ESP, mas nao o do EBP

- A funcao que salva o EBP e o iguala ao ESP chama-se PROLOGO:

push ebp
mov ebp, esp

- A funcao que restaura o EBP original, chama-se EPILOGO:

pop ebp
ret

Pipeline

- Buscar -> Decodificar -> Executar uma instrucao gera desperdicio de tempo

- O pipeline oferece um recurso de hardware que permite buscar uma instrucao, enquanto se decodifica outra e se executa uma terceira

- Processadores superescalares possuem multiplos niveis de pipeline (ex: um para aritmetica de inteiros, outro para pontos flutuantes e um outro para operacoes logicas)

- Organizar e gerenciar este tipo de recurso obviamente e complexo (salienta-se que cada CPU tem suas caracteristicas)

Modo Nucleo, Supervisor, Kernel, Privilegiado...

- Em geral, com excecao de hardware embarcado, os sistemas operam em modo nucleo ou modo usuario

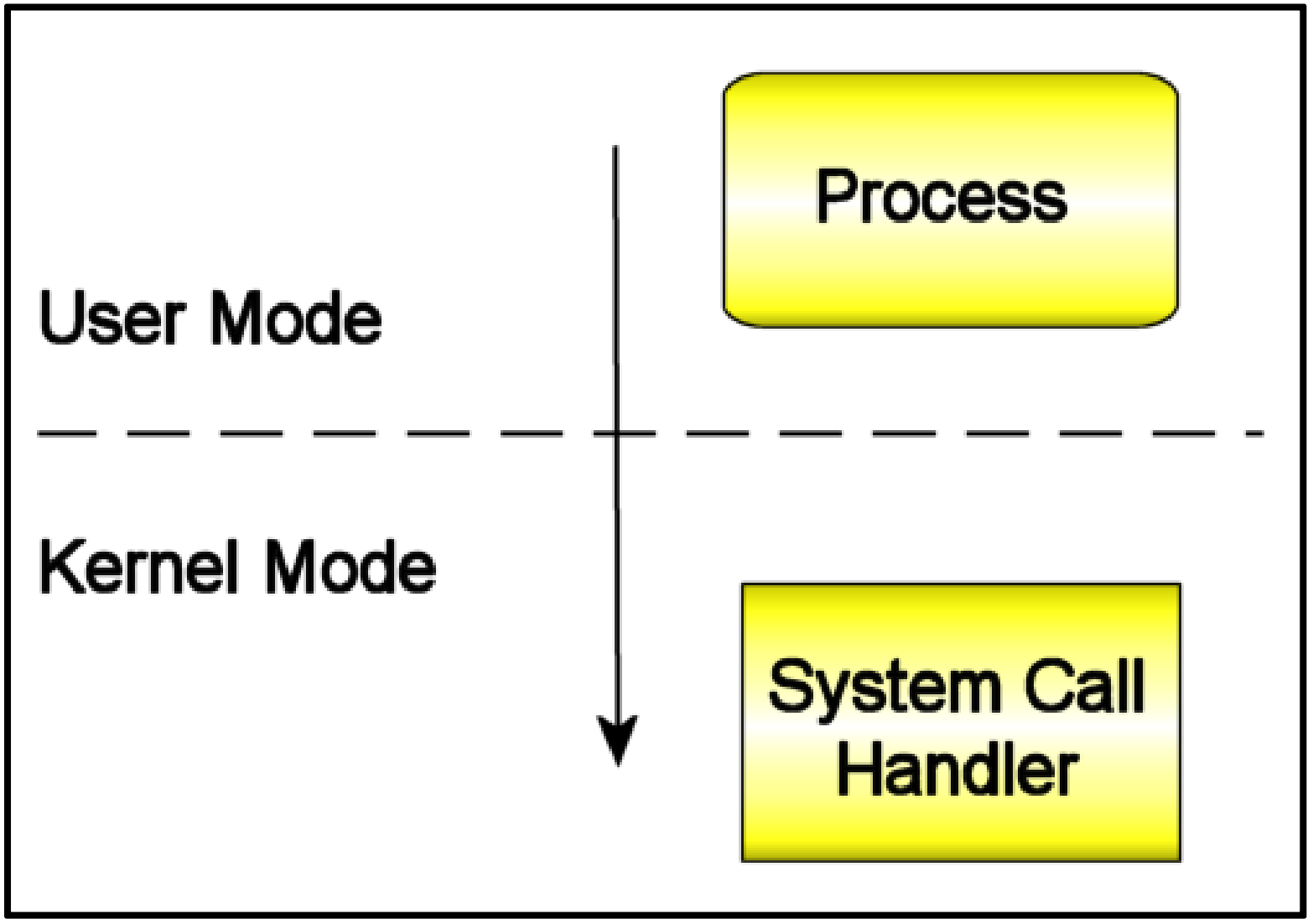
- Um bit do registrador crt0 (flag PE) controla em qual modo o sistema se encontra

- Modo privilegiado possui acesso a todo tipo de informacao e ao hardware enquanto o modo usuario nao

- Modo usuario possui acesso a algumas informacoes da PSW, inclusive podendo manipular certas partes da mesma, mas obviamente nao ao bit de controle de modo privilegiado

Chamadas ao Sistema

- Quando necessita de recursos do modo supervisor, o modo usuario realiza um trap (int 80 em plataforma intel) atraves de uma chamada ao sistema



Traps de Hardware

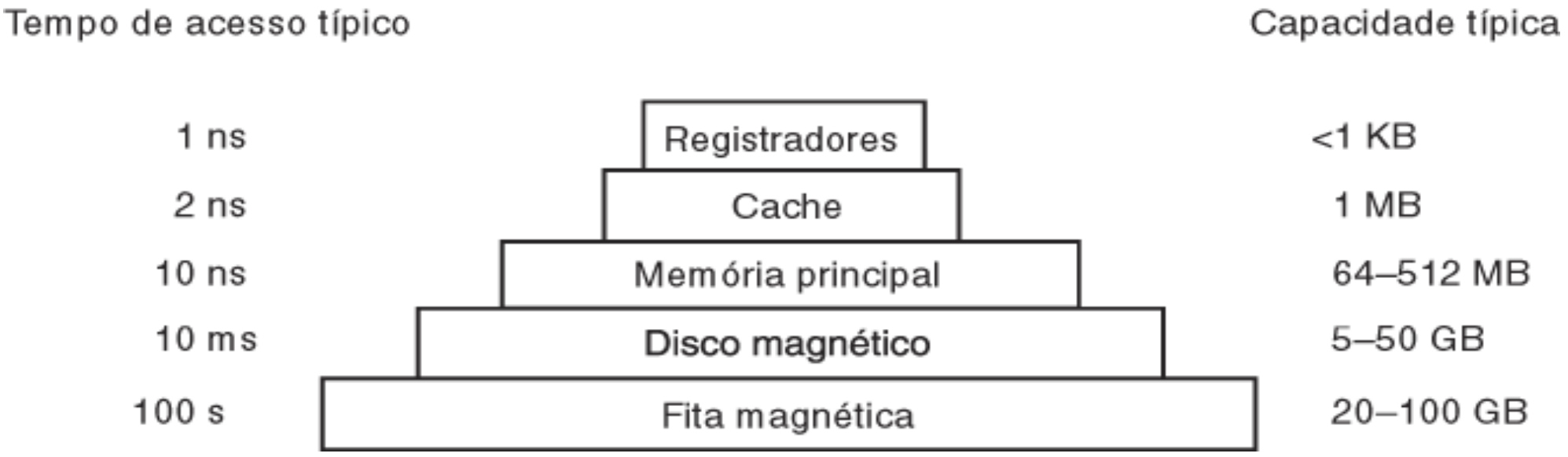
- Alem das interrupcoes (que serao vistas mais para frente na materia) ainda existe trap de hardware para o tratamento das chamadas excessoes

- Exemplos: Tentativa de uma divisao por 0 ou referencia a um ponto flutuante muito pequeno que nao possa ser representado (underflow).

Nota: Nao confundir este underflow com o underflow relacionado a falhas de seguranca, onde consegue-se gerenciar o ponteiro da pilha ou de outra estrutura (heap) para forcar a sobrescrita de dados

Memoria

- Por ser mais lenta do que a execucao de uma instrucao, ela acaba sendo construida em hierarquia de camadas.



Registradores

- Mesmo material e portanto tao rapido quanto a CPU
- Equipamentos possuem registradores conforme o tamanho da palavra do mesmo (32 bits ou 64 bits por exemplo)
- Gerenciados via software

Cache

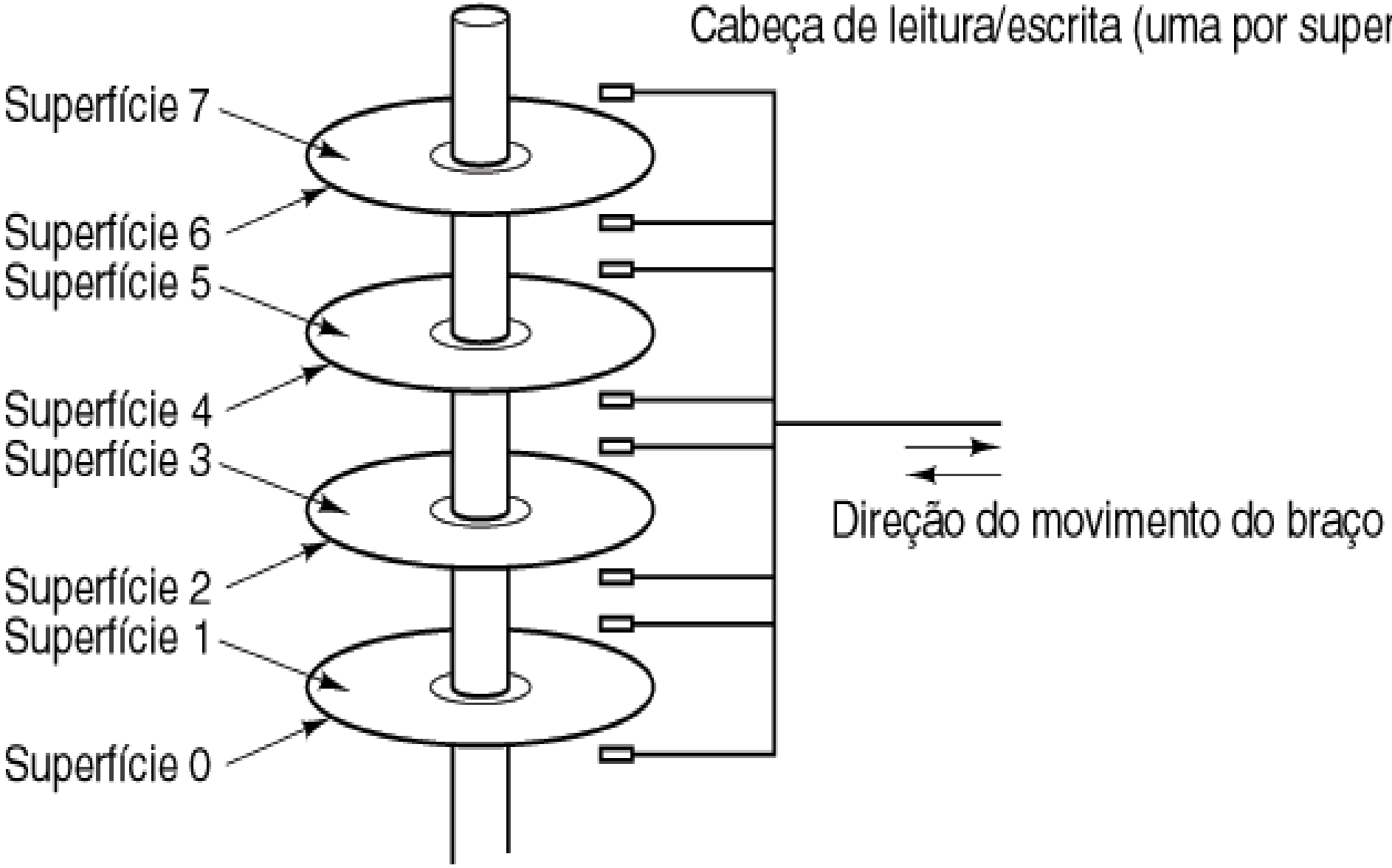
- Gerencia via hardware (normalmente, controlado pelo SO)
- Dividido em linhas de cache (64 bytes cada em geral)
- Dados mais frequentemente acessados ficam mais proximos a CPU
- Se o dado estiver no cache (cache hit), sera fornecido (2 ciclos CPU acesso)
- Se o dado nao estiver no cache (cache miss), devera buscar na memoria principal (isso significa atraso)
- Hardwares podem ter varios niveis de cache, cada um maior e mais lento

RAM (Random Access Memory)

- Memoria principal, onde os aplicativos deverao normalmente executar

Hard disk (disco magnetico)

- 1 ou mais pratos que rodam a 5400, 7200, 10800 ou + RPM's
- 1 braco mecanico que roda sobre os pratos
- Informacao escrita em circulos concentricos
- Cada posicao do braco existe uma cabeca que pode ler uma trilha
- Todas as cabecas (trilhas) de um braco formam um cilindro
- Cada trilha sera dividida em um certo numero de setores (512 bytes normalmente)


Cabeça de leitura/escrita (uma por superfície)
Superfície 7
Superfície 6
Superfície 5
Superfície 4
Superfície 3
Superfície 2
Superfície 1
Superfície 0
Direção do movimento do braço

ROM (Read only memory)

- Memoria de apenas leitura

EEPROM (electrically erasable ROM) e FLASH

- Podem ser escritas mas possuem acesso muito mais lento

Memorias volateis como CMOS

- Usadas para manter data e hora
- Linux pega data e hora da BIOS e depois gerencia sozinho (hwclock -w)
- BIOS as vezes fica armazenada na CMOS
- Consomem pouca energia (mantidas com a bateria de fabrica)

FIM! Será mesmo?

DÚVIDAS?!?

Rodrigo Rubira Branco
rodrigo@kernelhacking.com