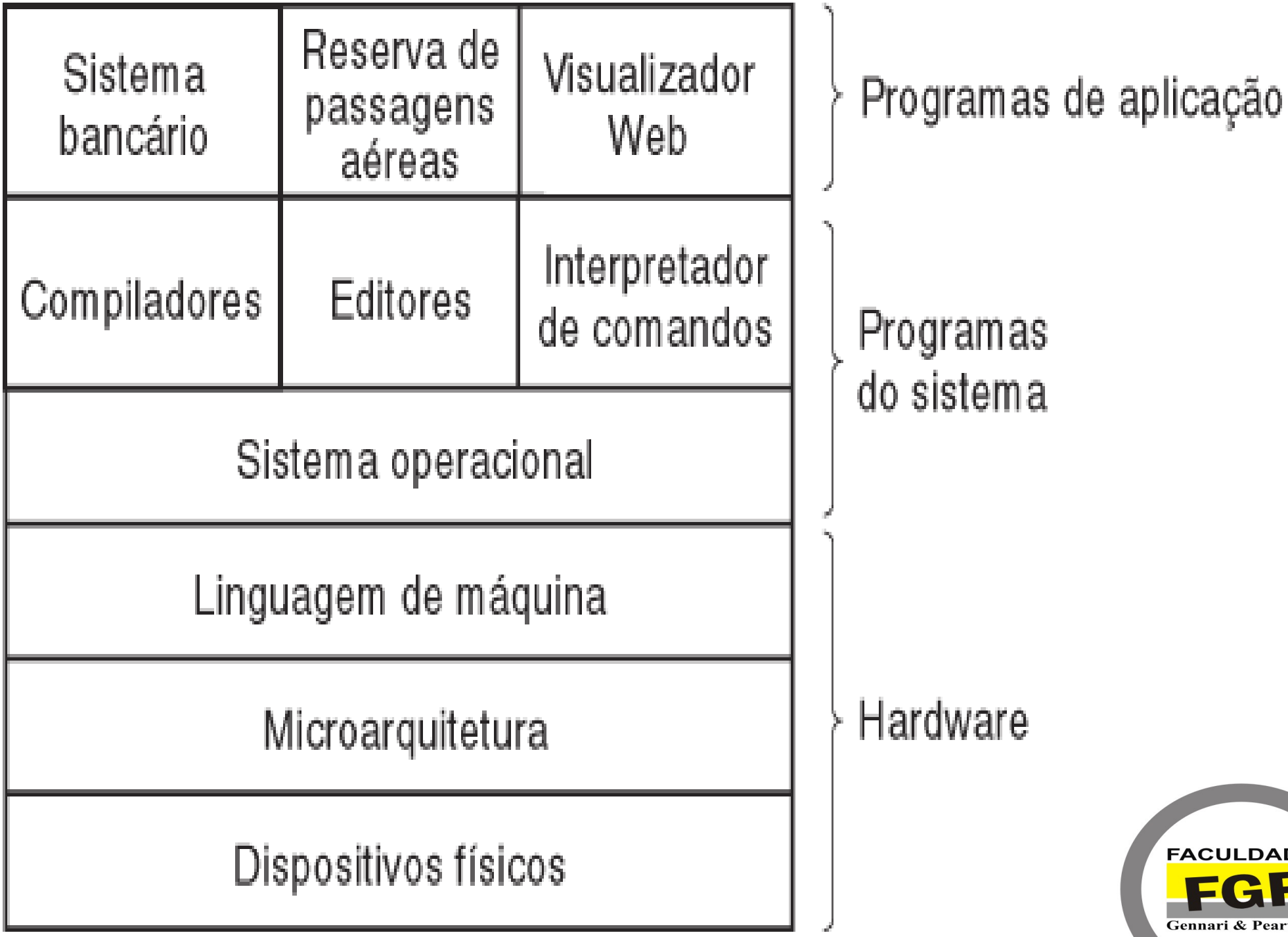
Sistemas Operacionais

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


Sistema bancário
Reserva de passagens aéreas
Visualizador Web
Programas de aplicação
Compiladores
Editores
Interpretador de comandos
Programas do sistema
Sistema operacional
Linguagem de máquina
Microarquitetura
Hardware
Dispositivos físicos

Dispositivos Fisicos: Chips, fios, fontes, etc

Micro Arquitetura – Agrupamentos de dispositivos fisicos em unidades funcionais. Acesso a registradores, caminhos de dados, etc (controlado por hardware ou por um microprograma).

ISA (instruction set architecture) – Hardware + instrucoes de controle do mesmo (unicas por arquitetura). Controla-se dispositivos por registradores do proprio dispositivo.

SO – Oculta o nivel ISA dos usuarios

Programas do Sistema - Não fazem parte do SO, mas o acompanham (SO normalmente estara em modo supervisor – exceto em maquinas mais antigas/simples) – Micro Kernel confunde este conceito

Programas de Aplicativos – Programas dos usuarios

Micro Kernel x Kernel Monolitico

Monolitico: Todo sistema operacional esta no mesmo espaco de memoria, portanto cada procedimento (funcao) do mesmo pode requisitar/modificar qualquer outro procedimento

Microkernel: Existe divisao clara do sistema operacional em "programas", muitos dos quais podem executar em modo usuario. O nucleo supervisor consiste de recursos de passagem de mensagens entre tais subsistemas, requisitando ou respodendo as requisicoes dos outros subsistemas.

O que e um Sistema Operacional?

* Maquina Estendida ou Maquina Virtual
Abstrai do usuario/programador as dificuldades do hardware

Ex: Leitura/Gravacao de dados, temporizador, memoria

* Sistema Gerenciador de Recursos
O SO procura organizar os recursos de hardware e monitorar seu uso, controlando as requisicoes conflitantes entre multiplos usuarios/sistemas

Ex: Spool de impressao

Tipos de Sistemas Operacionais

De Mainframe (Linux, OS/390)

- Suportam processamento simultaneo de milhares/milhores de jobs
- Gerenciam quantidade enorme de E/S (algoritmos de elevador)
- Oferecem 3 tipos de servicos:
  * Modo lote: Relizacao de grandes operacoes (computacao dos dados do censo por exemplo)
  * Processamento de transacoes: Verificar grande quantidade de pequenas transacoes (tais como operacoes bancarias)
  * Tempo compartilhado: Permite que multiplos usuarios realizem suas tarefas simultaneamente

Tipos de Sistemas Operacionais

De Servidores (Linux/Unix/Windows)

- Oferecem servicos a uma rede de computadores
- Caracteristicas de protecao/seguranca de dados

De Multiprocessadores

- Suporte a mais de 1 CPU (atualmente praticamente todos os sistemas operacionais modernos possuem esta caracteristica)

Multiprocessamento x Gerenciamento de Processos

Filas de execucao

Chama-se de fila de execucao a estrutura de dados basica utilizada pelo scheduler.

Existe uma fila de execucao definida por processador (em sistemas multiprocessados (SMP) como o Linux).

Cada processo necessariamente estara em uma (e apenas uma) fila de execucao.

No Linux a estrutura runqueue esta definida em kernel/sched.c e nao em um arquivo de cabecalho (.h) para que apenas partes da estrutura sejam acessiveis ao resto do kernel atraves de metodos especificos.

Multiprocessamento x Gerenciamento de Processos

Prioridades

Cada fila de execucao contem 2 arrays de prioridades (um ativo e outro expirado).

Estas filas contem os processos executaveis em cada nivel de prioridade, sendo que sao 140 niveis pro sistema (default) e utiliza-se de um mapa de bits (bitmap) para a descoberta eficiente do processo a executar (tal mapa esta definido como um array long de 5 elementos = 160 bits).

Como mencionado, cada nivel de prioridade contem os processos daquele nivel em sua lista, sendo a escolha do proximo processo a executar tao simples quanto descobrir o proximo processo da lista.

Multiprocessamento x Gerenciamento de Processos

Complexidade Algoritimica

Diversas formas existem para se medir o comportamento de um determinado codigo (algoritmo). Dentre estas, uma comumente utilizada e durante toda a apresentacao referenciada chama-se notacao Big-O.

Esta notacao visa medir o comportamento assintotico de um algoritmo, ou seja, como este se comporta quando suas entradas de dados tornam-se cada vez maiores, proximas do infinito.

Ex:

O(1) -> Constante
O(n) -> Linear
O(log n) -> Logaritimico
O(2 ^ n) -> Exponencial (pessimo)
O(n!) -> Fatorial (muito pessimo)

Multiprocessamento x Gerenciamento de Processos

Recalculo de tempo

Algoritmo O(N) -> Problemas com bloqueios da lista, levando tambem a disputas de bloqueio:

for (lista_tarefas_do_sistema){ recalcular(prioridade); recalcular(time_slice);}

Algoritmo O(1) -> Por oferecer 2 arrays de prioridades (ativo e expirado), quando a fatia de tempo de um processo atinge 0, a mesma sera recalculada antes de o processo ser movido para a lista de expirados. Quando todos atingirem 0, os ponteiros dos arrays serao invertidos (em *schedule()*):

```
struct prio_array array = rq->active;
if (!array->nr_active) {
    rq->active=rq_expired;
    rq->expired=array; }
```

Tipos de Sistemas Operacionais

De Computadores Pessoais (Linux/Windows/Panther)

- Normalmente interface com usuario facilitada
- Recursos graficos poderosos para melhor interacao
- Caracteristicas de seguranca mais pobres

De Tempo Real (QNX/VxWorks/Linux?)

- Prazo rigido para realizacao de tarefas (atrasos nao sao aceitos)
- Tempo real critico (acoes devem ocorrer necessariamente naquele instante)
- Tempo real nao critico (descumprimento ocasional de um prazo pode ser aceito) -> Linux Scheduler 2.6 tenta atender aqui priorizando processos orientados a E/S

Processos orientados a E/S x orientados a Processador

Dizemos que um processo esta voltado (orientado) para E/S quando este passa boa parte de seu tempo de processamento dormindo (ou seja, aguardando uma determinada E/S, tal como dados digitados pelo teclado ou uma leitura de disco).

Em contrapartida, dizemos que um processo esta voltado a processador quando o mesmo passa a maior parte do tempo em execucao (exemplo, realizando calculos matematicos ou reenderizacao de imagens).

O Linux beneficia (e muito) processos orientados a E/S, para oferecer otimos recursos de interatividade (todos os aplicativos interativos, esperam E/S do usuario).

Tipos de Sistemas Operacionais

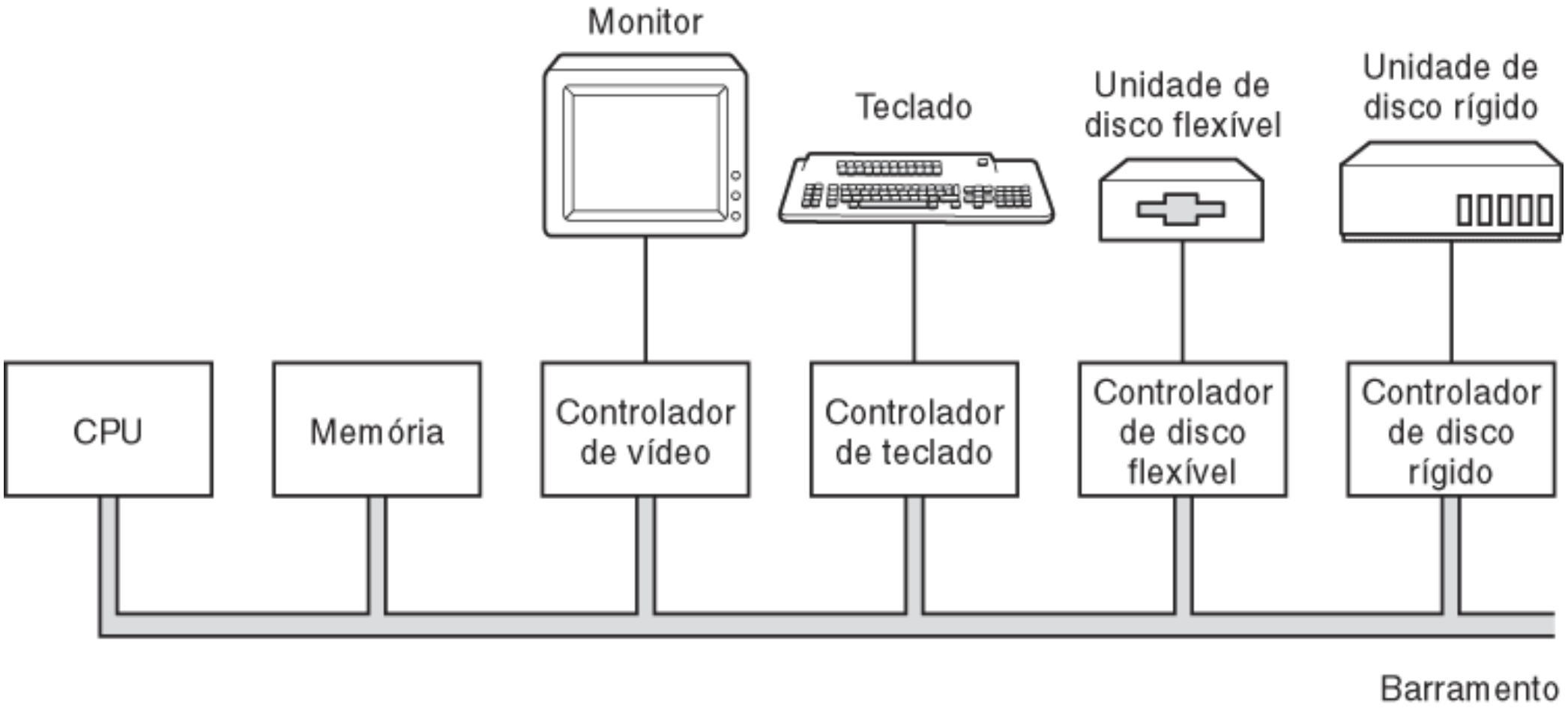
De Sistemas Embarcados (PalmOS,WinCE,WinXPEmbbeded,Linux)

- 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

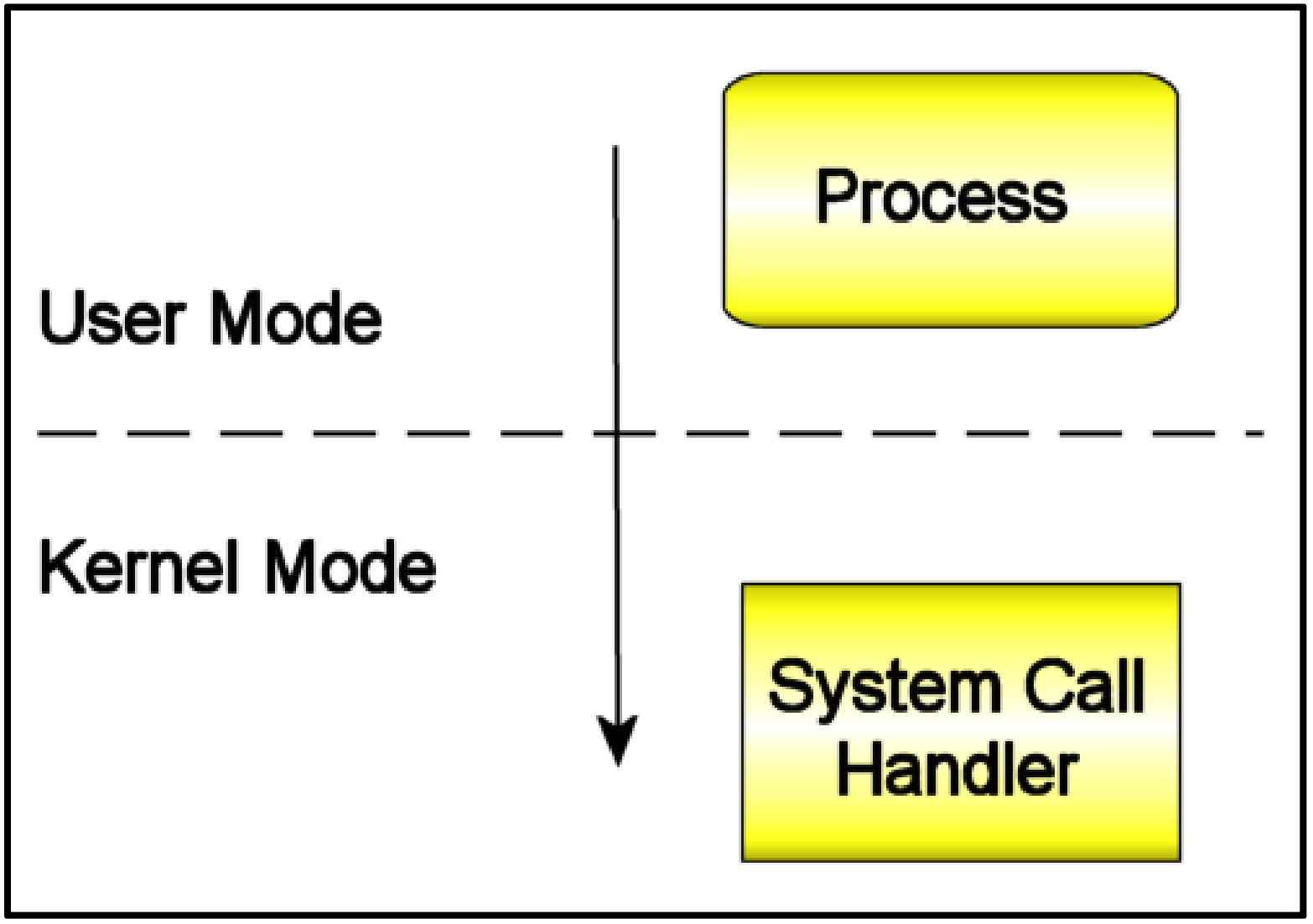
- Um bit do registrador PSW (program status word) controla em qual modo o sistema se encontra

- 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

FIM! Será mesmo?

DÚVIDAS?!?

Rodrigo Rubira Branco
rodrigo@kernelhacking.com