



SISTEMAS OPERACIONAIS NÚCLEO GERENCIA DE PROCESSOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

COMPETÊNCIA 3 (FUNDAMENTAL)

- Objetivo:
 - Distinguir os componentes de um sistema operacional para diferenciar as gerências e suas funcionalidades distintas

COMPETÊNCIA 3



Componentes do S.O.



Núcleo



Shell



Gerência de processo



Gerência de memória



Gerência de arquivos

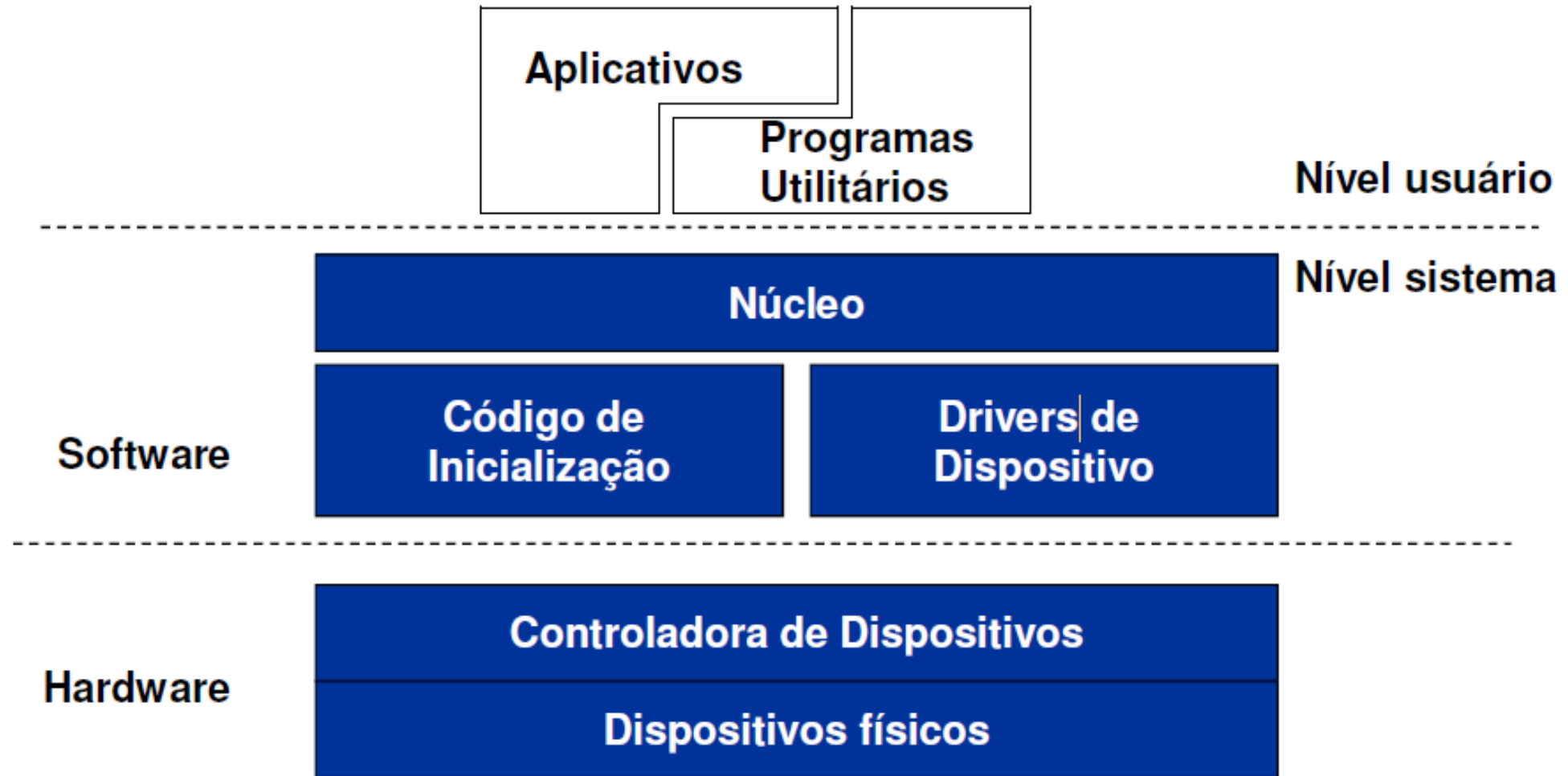


Gerência de entrada e saída

S.O. – ESTRUTURA DO SISTEMA

Um sistema operacional é composto de diversos componentes que têm objetivos e funcionalidades complementares. Os mais relevantes de um sistema operacional típico são:

- Código de inicialização
- Drivers de Dispositivos
- Núcleo



S.O. – ESTRUTURA DO SISTEMA

Código de inicialização : a inicialização do hardware requer uma série de tarefas complexas, como reconhecer os dispositivos instalados, testá-los e configurá-los adequadamente para seu uso posterior. Outra tarefa importante é carregar o núcleo do sistema operacional em memória e iniciar sua execução.

S.O. – ESTRUTURA DO SISTEMA

Drivers : módulos de código específicos para acessar os dispositivos físicos. Existe um driver para cada tipo de dispositivo, como discos rígidos IDE, SCSI, portas USB, placas de vídeo, etc. Muitas vezes o driver é construído pelo próprio fabricante do hardware e fornecido em forma binária para ser acoplado ao restante do sistema operacional.

S.O. – ESTRUTURA DO SISTEMA

Núcleo : é o coração do sistema operacional, responsável pela gerência dos recursos do hardware usados pelas aplicações. Ele também implementa as principais abstrações utilizadas pelos programas aplicativos.

S.O. – PROTEÇÃO DO NÚCLEO

- Um sistema operacional deve gerenciar os recursos do hardware, fornecendo-os às aplicações conforme suas necessidades.
- Para assegurar a integridade dessa gerência, é essencial garantir que as aplicações não consigam acessar o hardware diretamente, mas sempre através de pedidos ao sistema operacional, que avalia e intermedia todos os acessos ao hardware.

S.O. – NÍVEIS DE PRIVILÉGIO

Para permitir a diferenciação de privilégio de acesso entre os diferentes tipos de software, os processadores modernos contam com dois ou mais níveis de privilégio de execução. Esses níveis são controlados por flags especiais nos processadores, e a mudança de um nível de execução para outro é controlada por condições específicas.

S.O. – NÍVEL NÚCLEO

Nível núcleo : também denominado nível supervisor, sistema, monitor ou ainda kernel space. Para um código executando nesse nível, todo o processador está acessível: todos os registradores, portas de entrada/saída e áreas de memória podem ser acessados em leitura e escrita. Além disso, todas as instruções do processador podem ser executadas.

S.O. – NÍVEL USUÁRIO

Nível usuário (ou userspace): neste nível, somente um subconjunto das instruções do processador, registradores e portas de entrada/saída estão disponíveis. Instruções “perigosas” como HALT (parar o processador) e RESET (reiniciar o processador) são proibidas para todo código executando neste nível. Além disso, o hardware restringe o uso da memória, permitindo o acesso somente a áreas previamente definidas.

nível
usuário

aplicação

aplicação

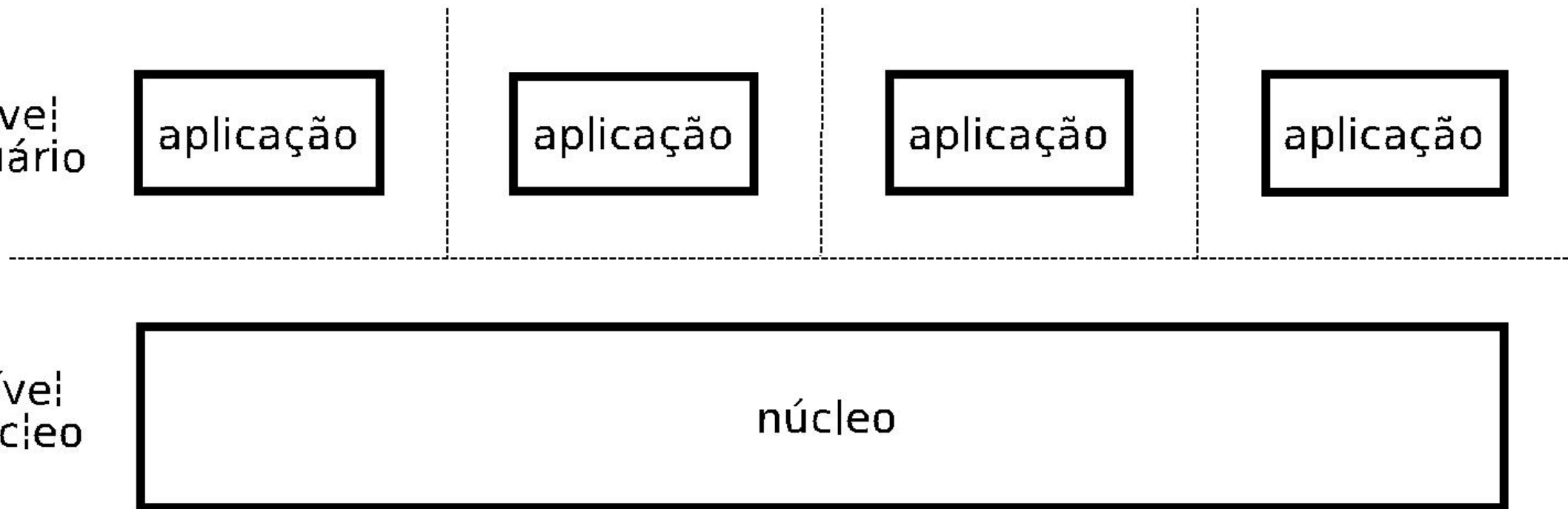
aplicação

aplicação

nível
núcleo

núcleo

hardware



S.O. – EXCEÇÃO

Caso o código em execução tente executar uma instrução proibida ou acessar uma área de memória inacessível, o hardware irá gerar uma exceção, desviando a execução para uma rotina de tratamento dentro do núcleo, que provavelmente irá abortar o programa em execução.

“Este programa executou uma instrução ilegal e será finalizado”

S.O. – CHAMADAS DO SISTEMA

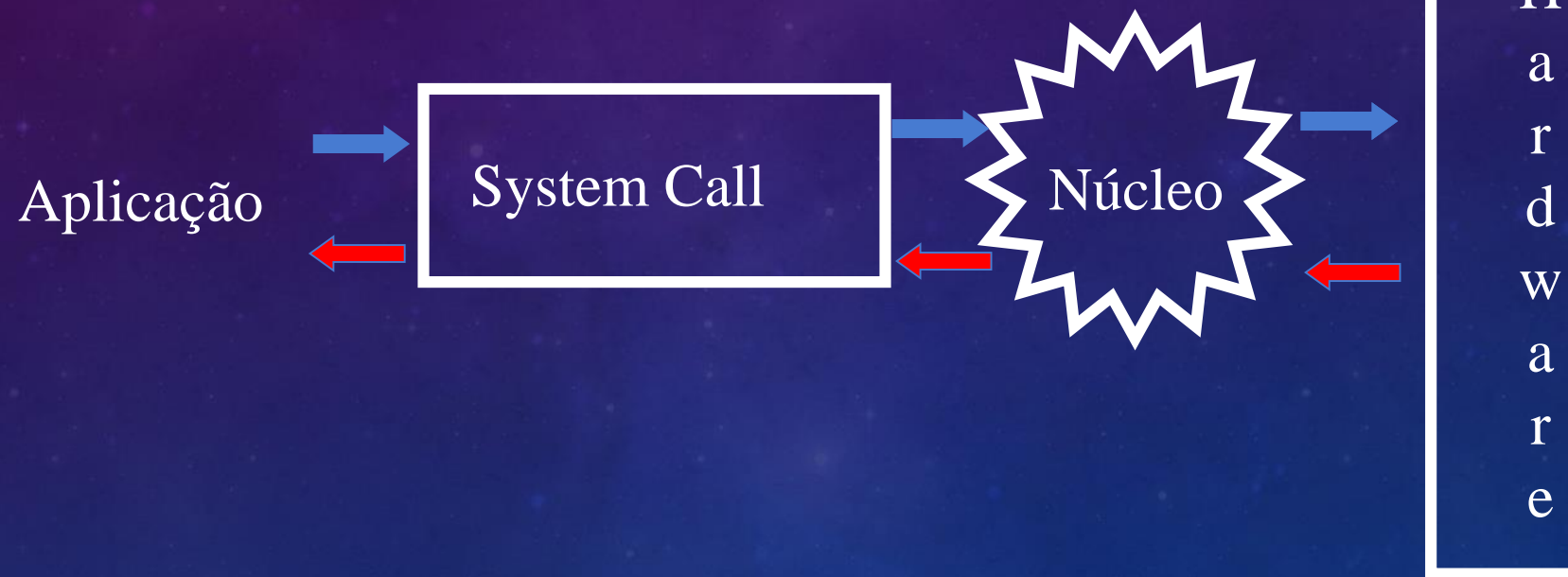
Pergunta: Como chamar, a partir de uma aplicação, as rotinas oferecidas pelo núcleo para o acesso ao hardware e suas abstrações?

Resposta: Os processadores implementam uma instrução especial que permite acionar o mecanismo de interrupção de forma intencional, sem depender de eventos externos ou internos. Ao ser executada, essa instrução (int no Pentium, syscall no MIPS) comuta o processador para o nível privilegiado e procede de forma similar ao tratamento de uma interrupção. Por essa razão, esse mecanismo é denominado **interrupção de software**.

S.O. – CHAMADAS DO SISTEMA

- A ativação de procedimentos do núcleo usando interrupções de software é denominada **chamada de sistema (system call ou syscall)**.
- Os sistemas operacionais definem chamadas de sistema para todas as operações envolvendo o acesso a recursos de baixo nível (periféricos, arquivos, etc) ou abstrações lógicas (criação e finalização de tarefas, operadores de sincronização e comunicação, etc).
- Geralmente as chamadas de sistema são oferecidas para as aplicações em modo usuário através de uma biblioteca do sistema (**system library**), que prepara os parâmetros, invoca a interrupção de software e retorna à aplicação os resultados obtidos.

S.O. – CHAMADAS DO SISTEMA

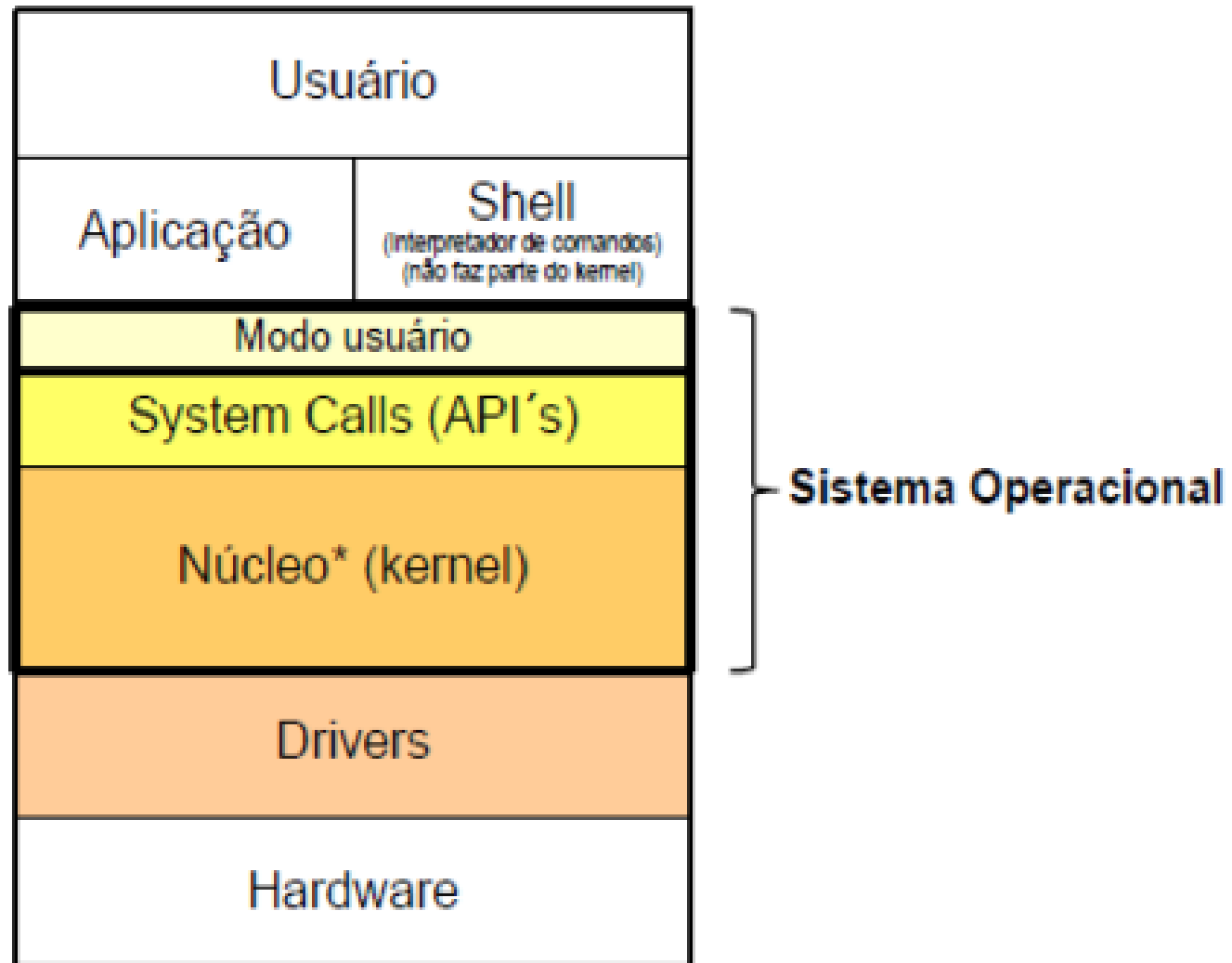


Funções	System calls
Gerência de processos e threads	Criação e eliminação de processos e threads Alteração das características de processos e threads Sincronização e comunicação entre processos e threads Obtenção de informações sobre processos e threads
Gerência de memória	Alocação e desalocação de memória
Gerência do sistema de arquivos	Criação e eliminação de arquivos e diretórios Alteração das características de arquivos e diretórios Abrir e fechar arquivos Leitura e gravação em arquivos Obtenção de informações sobre arquivos e diretórios
Gerência de dispositivos	Alocação e desalocação de dispositivos Operações de entrada/saída em dispositivos Obtenção de informações sobre dispositivos

• SHELL:

- Define uma interface entre o sistema operacional e seus usuários
 - fornece uma comunicação natural com os usuários do computador
 - shells antigos se comunicavam com os usuários, via teclado e monitor, por mensagens textuais
 - shells modernos fornecem uma interface gráfica com o usuário
 - GUI – Graphical User Interface
 - na qual objetos a serem manipulados (arquivos e programas) são representados graficamente através de ícones na tela
 - permite ao usuário executar um comando do sistema apontando e deslocando ícones





GERENCIAMENTO DE PROCESSOS.

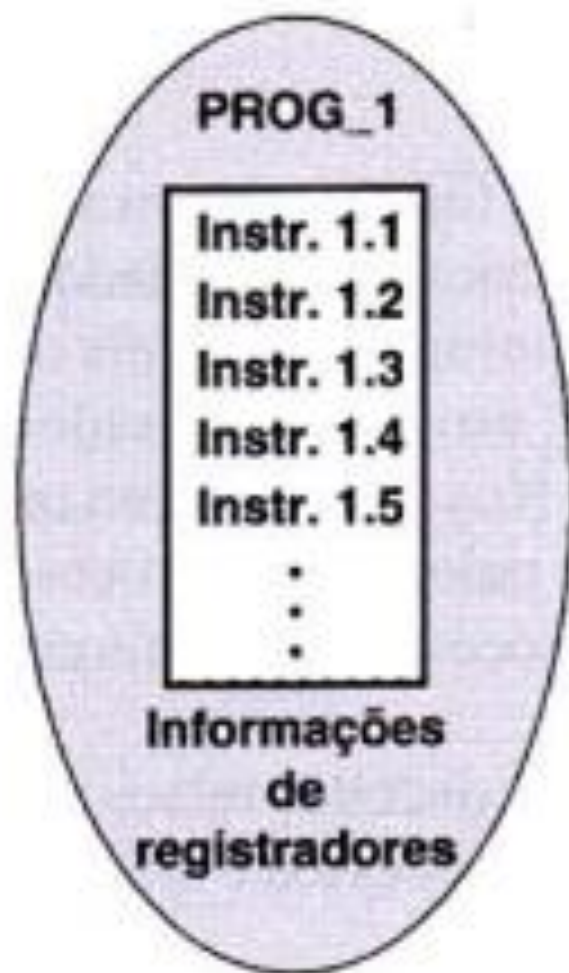
Um programa é:

- Uma seqüência finita de instruções;
- Uma entidade passiva (que não se altera com o passar do tempo).
- Armazenado em disco.

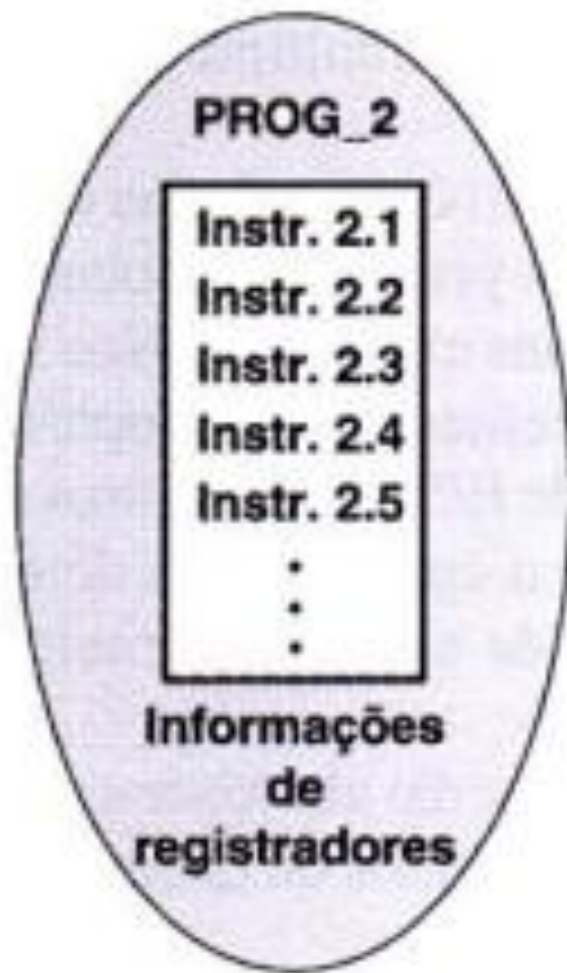
GERENCIAMENTO DE PROCESSOS.

Um processo é:

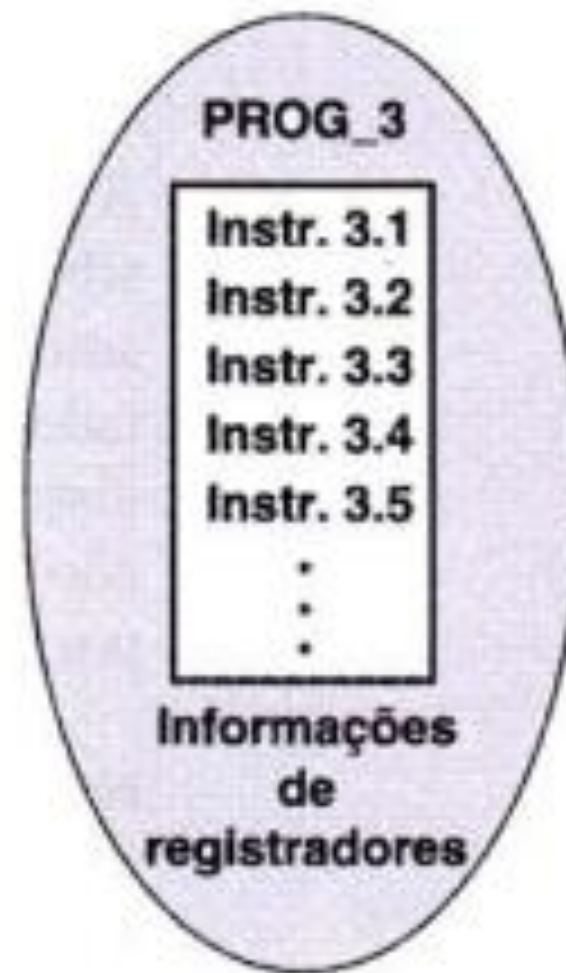
- Uma abstração que representa um programa em execução;
- Uma entidade dinâmica: seu estado se altera conforme for executando.
- Armazenado na memória.
- Pode-se encontrar mais de um processo instanciando um programa único



Processo X



Processo Y



Processo Z

PROCESSOS

Imagem de um programa

- Segmento de código
- Conjunto de recursos de HW alocados pelo Sis. Operacional.
- Registradores (PC, Stack Pointer...);
- Espaço de endereçamento (memória);
- Espaço no disco (arquivos de E/S).
- Unidade de escalonamento
- Estado;

ESTRUTURA DE UM PROCESSO



PROCESSOS.

Bloco de controle de processo:

Representa o processo no sistema operacional, mantendo a informação associada com cada um.

Constitui-se de uma área de memória reservada ao processo e que mantêm suas informações, tais como:

- PID (número identificador do processo)
- Estado do processo (rodando, bloqueado, etc...)
- PC (endereço da próxima instrução)
- Registradores da CPU
- Informações de gerenciamento de memória (limite, utilizado, etc...)
- Etc...

PROCESSOS.

*Bloco de controle de
processo:*

PID
Estado do processo
<i>PC</i>
Registradores
Limites de memória
Lista de arquivos abertos
⋮



CARACTERÍSTICAS DA ESTRUTURA DO PROCESSO

COMUNICAÇÃO ENTRE PROCESSOS

- A **comunicação entre processos**, em inglês **Inter-Process Communication** (IPC), é o grupo de mecanismos que permite aos processos transferirem informação entre si.
- A execução de um processo pressupõe por parte do sistema operacional, entre outras coisas, a criação de um contexto de execução próprio que, de certa forma, abstrai o processo dos componentes reais do sistema.
- Devido a esta virtualização dos recursos, o processo não tem conhecimento acerca dos outros processos e, como tal, não consegue trocar informação.
- O sistema operacional provê recursos para a troca de informações.

COMUNICAÇÃO ENTRE PROCESSOS

Classificação:

- Mecanismos locais
 - Pipes nomeados (FIFO)
 - Pipes unidireccionais
 - Filas de mensagens
 - Memória Compartilhada
- Mecanismos cliente/servidor
 - Sockets
 - Remote Procedure Calls

RELACIONAMENTO ENTRE PROCESSOS.

Sem relacionamento (Caso mais simples): os processos são independentes.

Grupo de processos

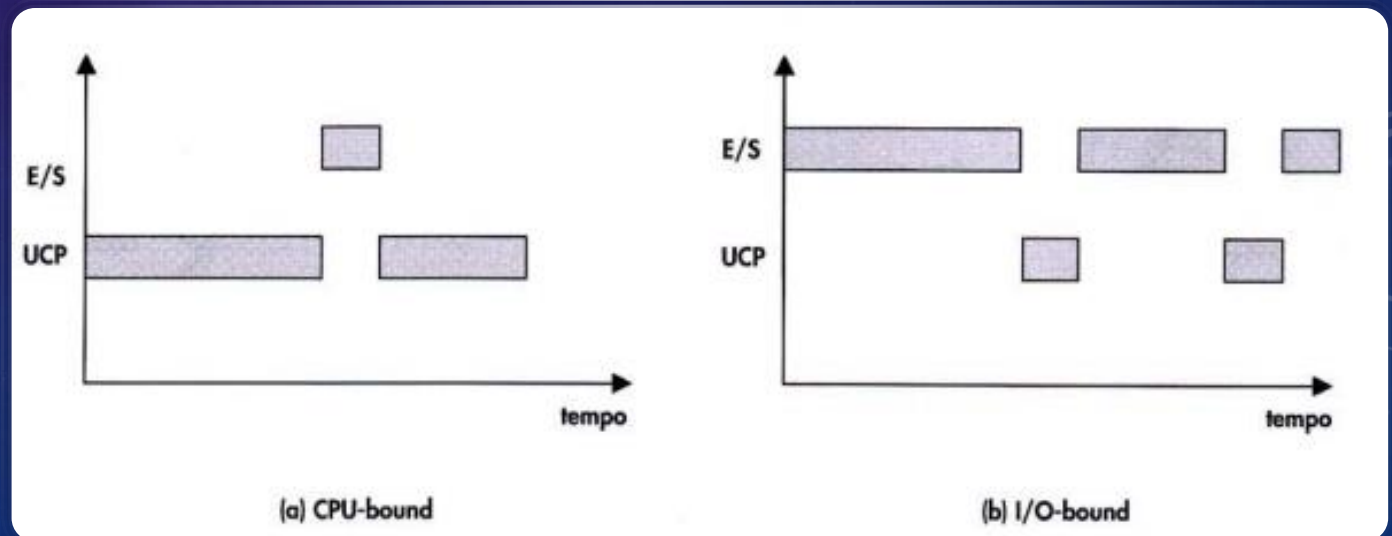
Compartilhamento de recursos

- Baseados em *hierarquia* de processos:
- Um processo pai cria processos filhos;
- Os filhos podem executar o mesmo código, ou trocá-lo;
 - Obtem-se uma árvore de processos.
 - Implica na definição da semântica de termino de um processo:
 - Só o processo morre;
 - Toda sua descendência morre.

CATEGORIAS DE PROCESSOS

Os processos se separam em duas categorias, relacionadas ao comportamento:

- CPU-Bound: Usam a CPU - Cálculo, operações sobre a memória e/ou registradores
- I/O Bound: fazem Entrada/saída (liberam a CPU) Escrita na tela, entrada teclado, som, vídeo, rede, disco...



CICLO DE VIDA DO PROCESSO

Os processos evoluem: Criado -> “em vida” -> morto

Durante a vida, passam de “uso de CPU” e a “uso de E/S” (e reciprocamente), via chamada de sistema, interrupção, ou por causa de um evento.

Ao ser criado, um processo estará pronto para usar a CPU.

ESTADOS DE PROCESSO



Criado (Novo): processo novo, sem utilizar a cpu ainda.



Pronto: Contexto carregado e esperando o uso de cpu



Em execução: Instrução em uso da cpu pertence ao processo em execução.s

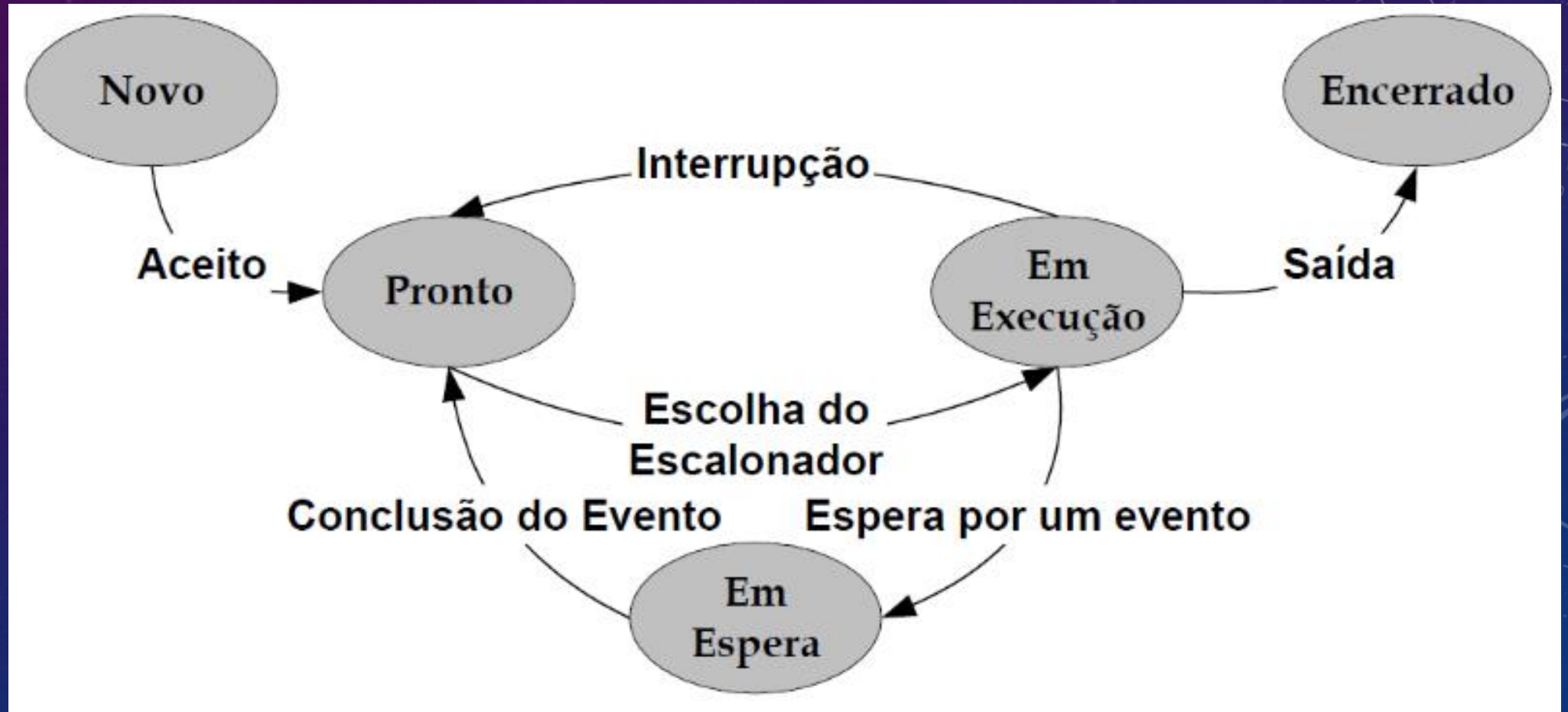


Bloqueado (Em espera): aguardando evento (E/S, sinal, etc...)



Encerrado: processo terminado.

TRANSIÇÕES DE ESTADOS



TRANSIÇÕES DE ESTADOS

Pronto -> executando - Algoritmo de escalonamento

Executando -> pronto

- Interrupção de tempo

- Interrupção devida ao escalonador

- Decisão espontânea (yield)

Executando -> bloqueado

- E/S – sincronização

TRANSIÇÕES DE ESTADOS

Bloqueado -> pronto
Interrupção

Executando -> encerrado
Interrupção (CTRL-C)
Término normal

Bloqueado, pronto -> encerrado
interrupção

TRANSIÇÕES DE ESTADOS

Mais um estado : Suspenso

Dois problemas principais para gerar os recursos:

- A CPU é muito mais rápida do que a memória;
- A memória é de tamanho finito.

Precisa-se, então, poupar a memória.

- Processos bloqueados que estão na memória podem ser transferidos para o disco (swap) até sua E/S ser acabada.
- Processos prontos podem também ser descarregados para o disco.

Threads

A evolução dos softwares e dos componentes de hardware fizeram com que as linhas de instruções dos processos adquiriram características únicas, que possibilitaram separá-las para execuções em diferentes núcleos. Essas linhas de instruções ficaram conhecidas como threads, mas muita gente preferiu traduzir a palavra “thread” para tarefa.

Threads

Threads são pequenas tarefas que o computador precisa executar de forma concorrente, ou seja, operações que um ou mais programas abertos precisam realizar e disputam pela "atenção" do processador.

Logo, Thread é:

Uma divisão do processo principal de um programa.



Multi-threading

Capacidade do processador que possui 2 ou mais núcleos de lidar com várias threads ao mesmo tempo, e é devido a essa tecnologia presente na maioria dos processadores modernos que podemos executar várias aplicações ao mesmo tempo, necessitando apenas de um sistema operacional multitarefa.

