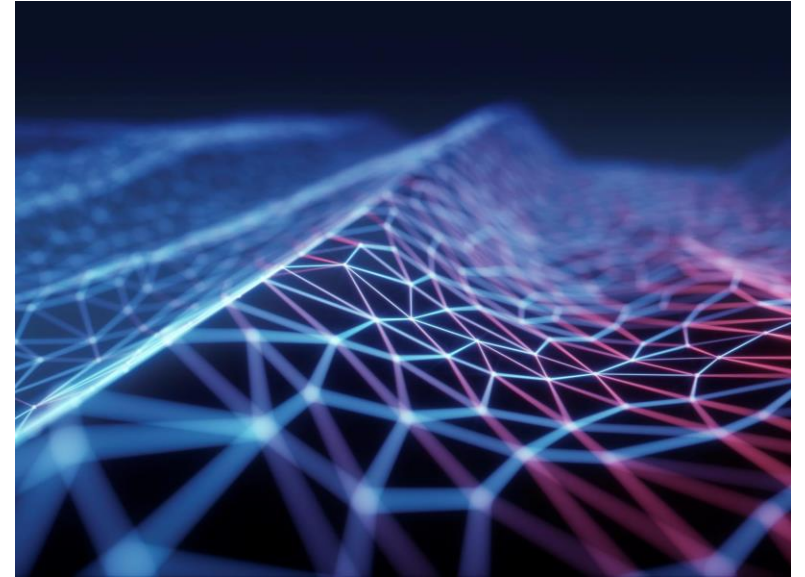


01 - SISTEMAS OPERACIONAIS

PROFESSORES:

OSMAR DE OLIVEIRA BRAZ JUNIOR

JORGE WERNER



OBJETIVOS

- Visão Geral dos Sistemas Operacionais.
- Tipos de Sistemas Operacionais e suas funções.
- Concorrência
- Estrutura e arquitetura
- Comunicação entre processos(Sincronização)
- Gerência do Processador

ATIVIDADE MENTIMETER

Acesse o Link

<https://www.menti.com>

Entre com o código

14574479



CONCEITO



○ que é um Sistema Operacional?

- É um software cuja função é controlar e gerenciar o funcionamento de um computador e seus diversos recursos.

SISTEMA OPERACIONAL

Quais as funções de um Sistema Operacional?

QUAIS AS FUNÇÕES DO SO?



- Facilidade de acesso aos recursos do sistema
- Compartilhamento dos recursos de forma organizada

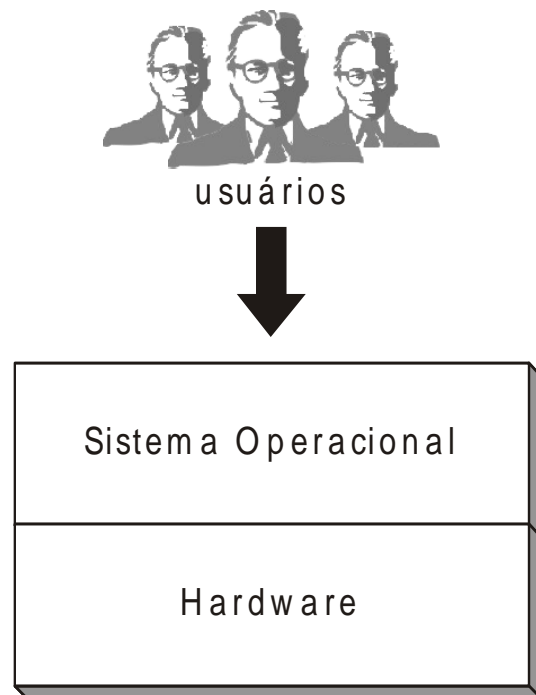
SISTEMA OPERACIONAL

Como é visto o Sistema Operacional...

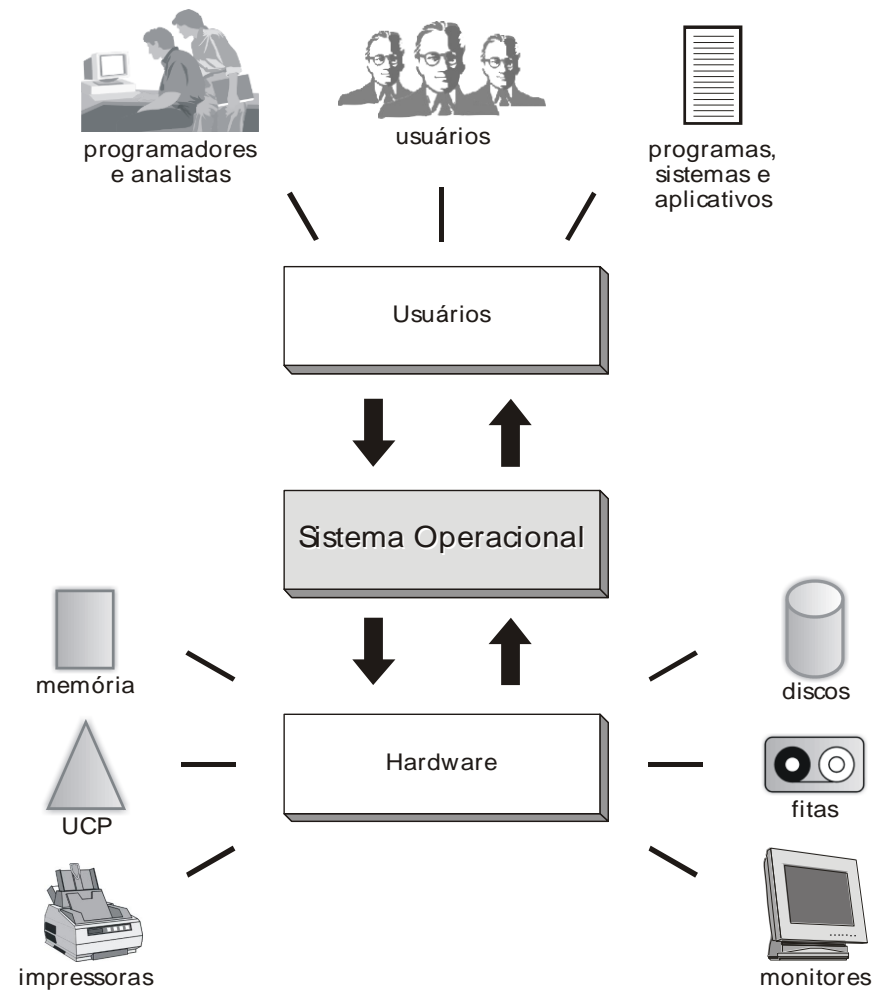
Sob a ótica do usuário?

Sob a ótica do Sistema Computacional?

VISÃO DO USUÁRIO



VISÃO DA MÁQUINA



SISTEMA OPERACIONAL

Fica a dúvida...

Seria possível usar o hardware sem o Sistema Operacional?

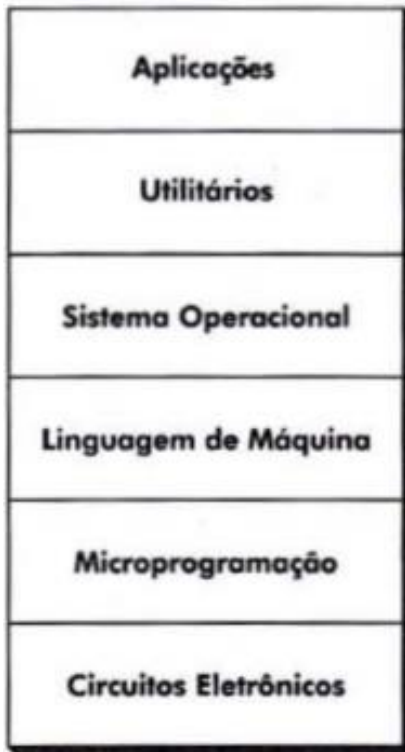
MÁQUINA DE CAMADAS



Qual a sua opinião?

Uma operação efetuada pelo software pode ser implementada em hardware, enquanto uma instrução executada pelo hardware pode ser simulada via software.

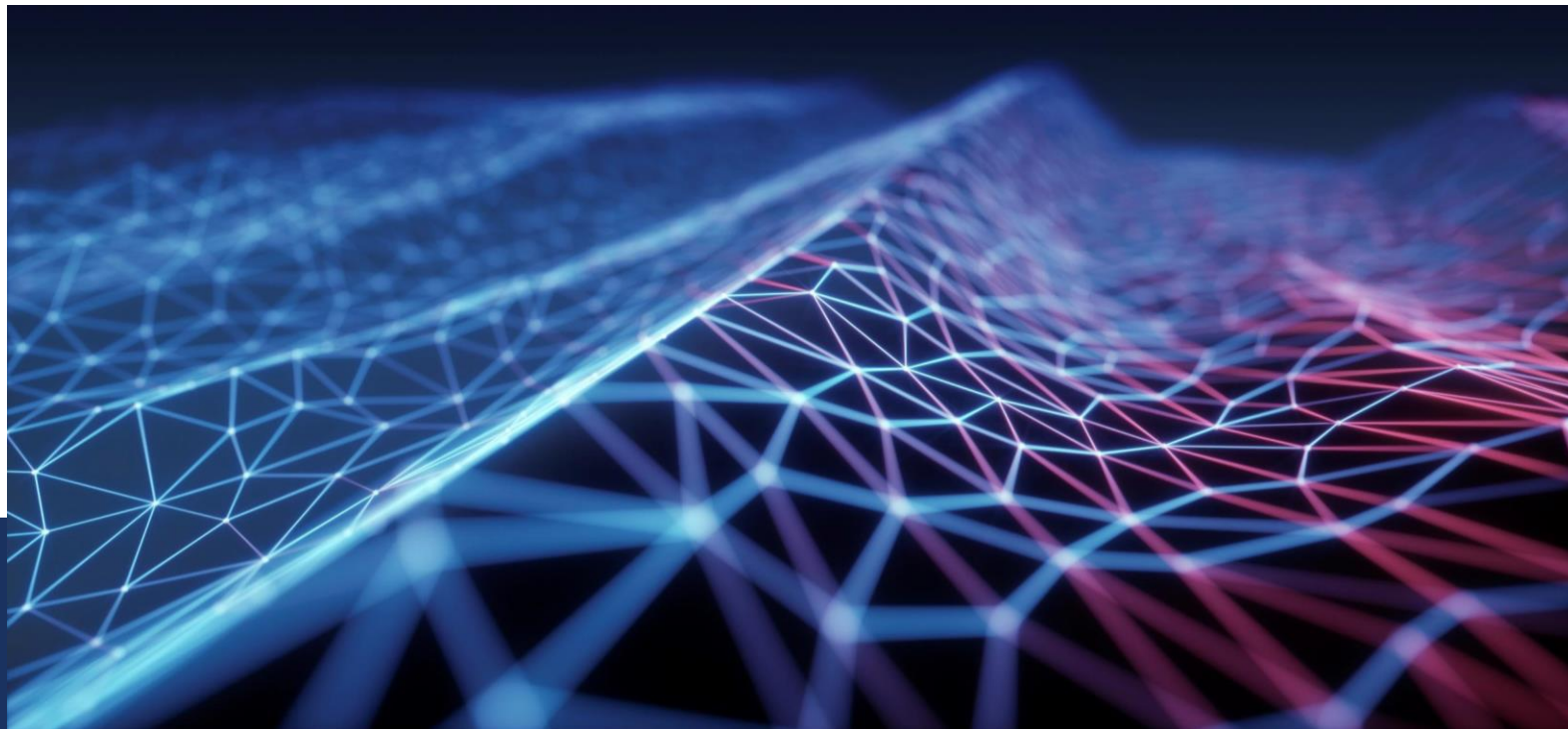
MÁQUINA DE CAMADAS



- Temos ao lado uma visão mais alinhada aos modelos dos computacionais atuais;
- Usuário interage com o SO, abstraindo as questões de acesso ao hardware;
- A decisão do que é implementado a nível de hardware ou software fica a cargo do projetista;
- Essa visão modular e abstrata é chamada de máquina virtual.

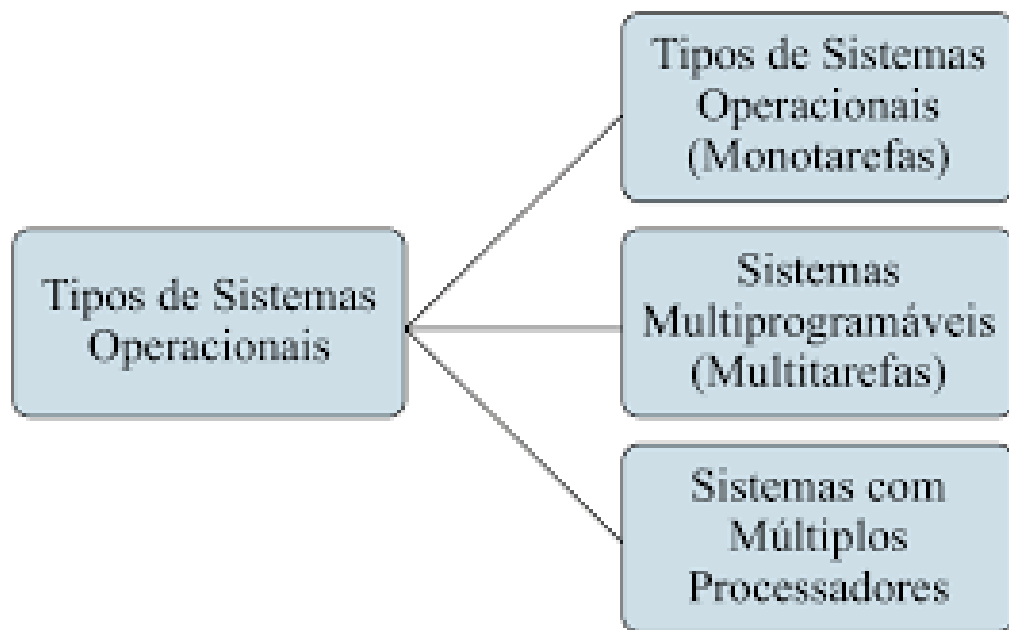
O QUE FICOU NA MENTE?

- Como seria utilizar um computador sem um sistema operacional? Quais são suas duas principais funções?
- Quais as principais dificuldades que um programador teria no desenvolvimento de uma aplicação em um ambiente sem um sistema operacional?
- Explique o conceito de máquina virtual. Qual a grande vantagem em utilizar este conceito?
- Defina o conceito de uma máquina de camadas.



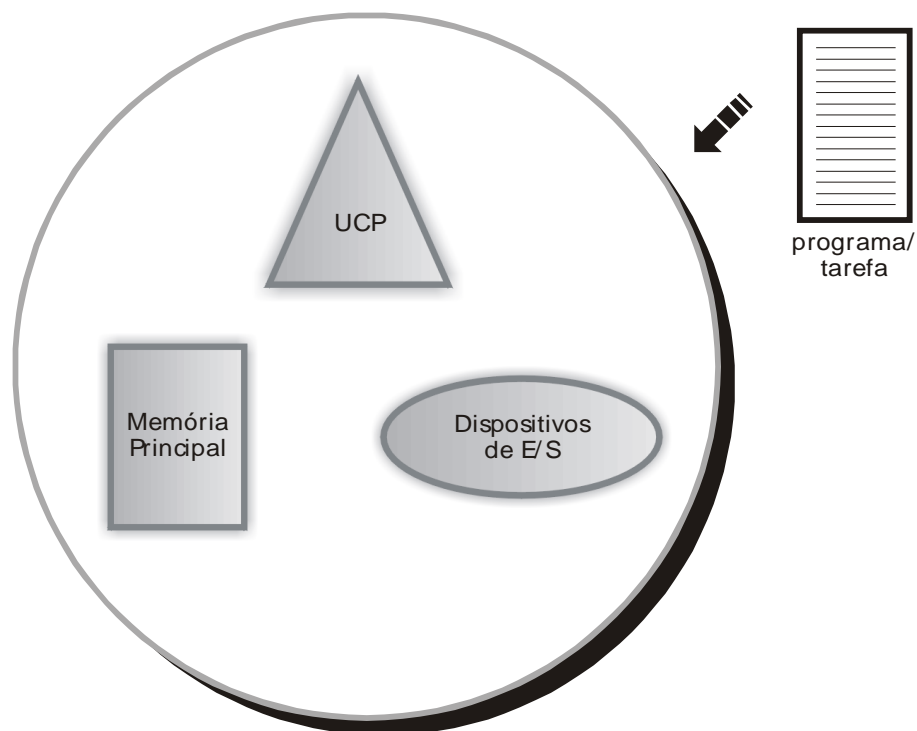
TIPOS DE SISTEMAS OPERACIONAIS

Considerando o gerenciamento de processos:



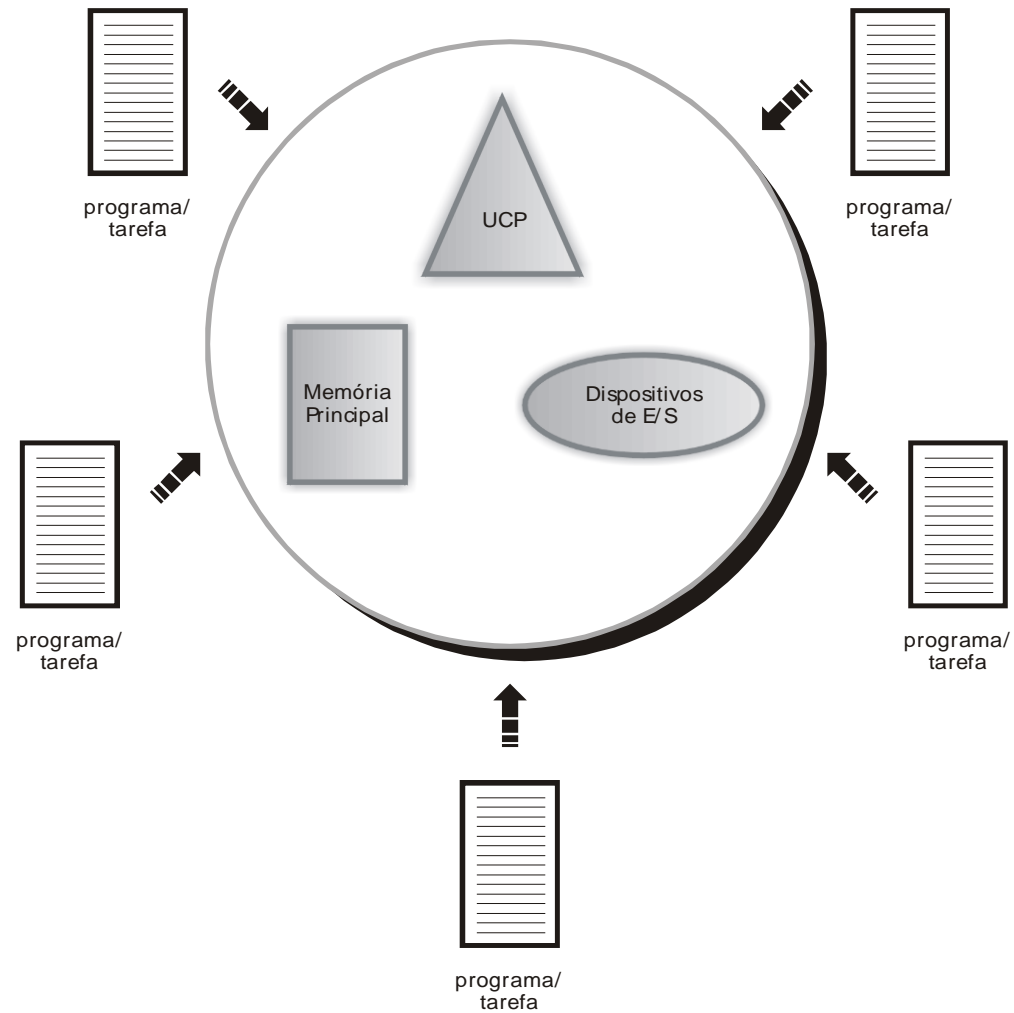
TIPOS DE SISTEMAS OPERACIONAIS

TIPOS DE SISTEMAS OPERACIONAIS - MONOPROGRAMÁVEIS/MONOTAREFA



- *Sistema operacional gerenciando um programa por vez;*
- *Sistema dedicado a uma única tarefa.*
- *Implementação simplificada, por abstrair concorrência e compartilhamento de recursos.*

TIPOS DE SISTEMAS OPERACIONAIS MULTIPROGRAMÁVEIS/MULTITAREFA

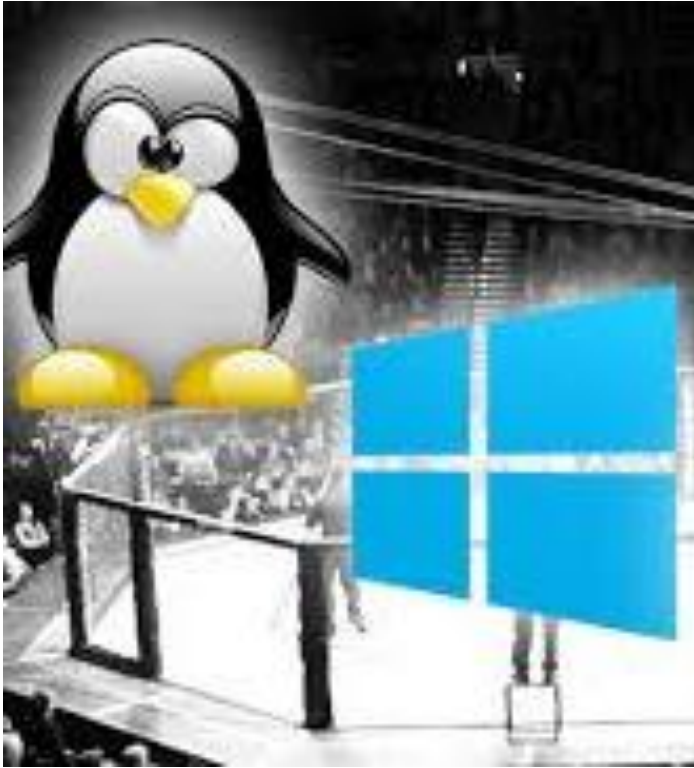


SOS MULTIPROGRAMÁVEIS/MULTITAREFA



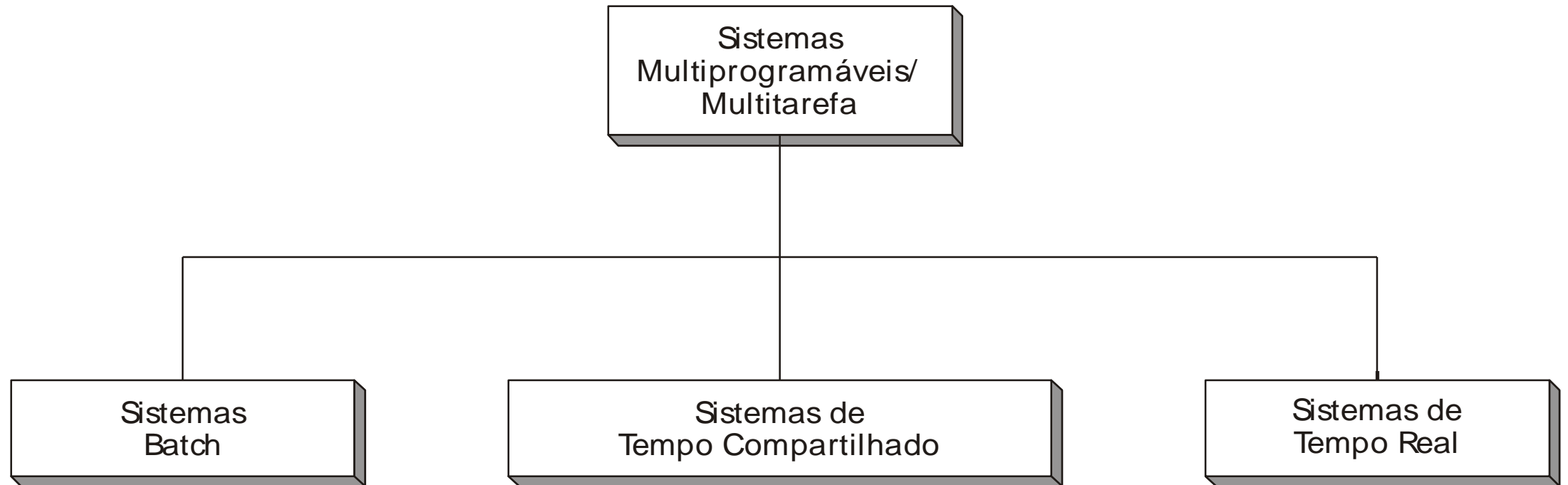
- Evolução dos sistemas monoprogramáveis;
- Enquanto um programa espera uma operação de E/S outros podem estar processando neste intervalo;
- Vantagem: redução de custos pelo compartilhamento.

MULTIPROGRAMÁVEIS - MULTITAREFAS



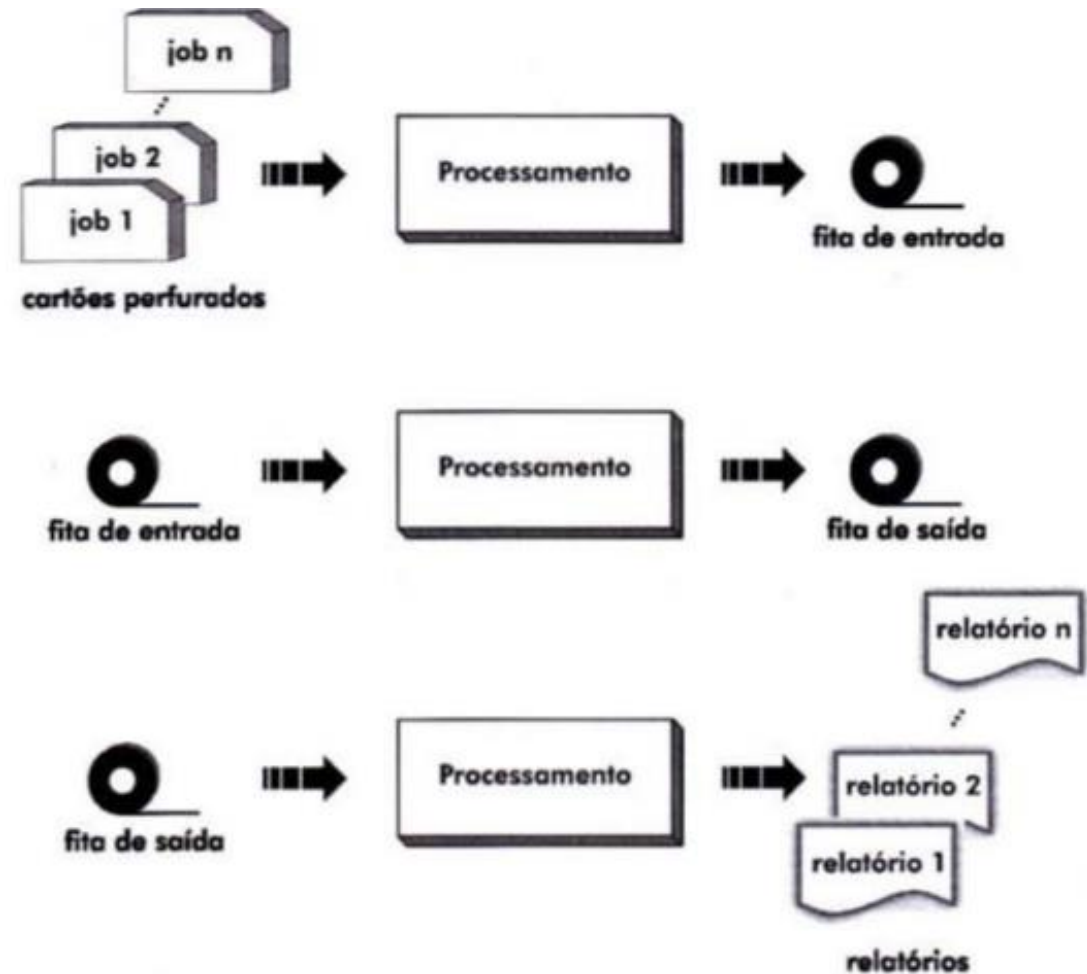
- Sistemas Operacionais Multitarefa podem ter formas diferentes de gerenciar essas tarefas...
- Sim, e eles podem ser separados em três grupos :)

TIPOS DE SISTEMAS OPERACIONAIS MULTIPROGRAMÁVEIS /MULTITAREFA

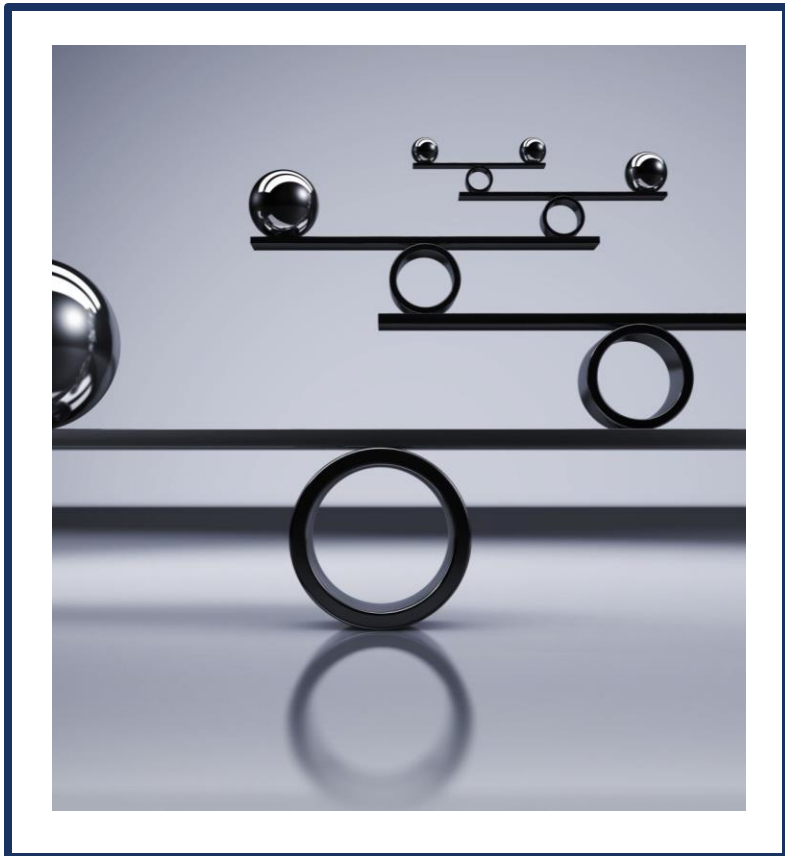


SISTEMAS BATCH

- Os processamentos conhecidos como Jobs, iniciavam por cartão ou fita, executavam um processamento, gerando uma saída em disco ou fita;
- Sem comunicação intermediária com o usuário até o final do processamento.



UMA DÚVIDA...



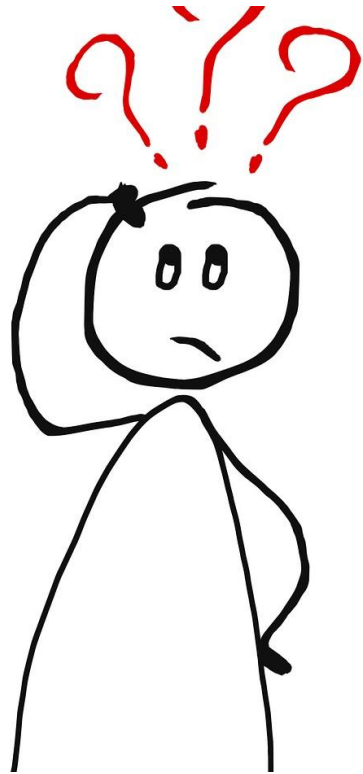
- Ainda usamos o modelo dos sistemas batch hoje?

TIPOS DE SISTEMAS OPERACIONAIS

- Sistemas de Tempo Compartilhado
 - *Time-sharing* – divisão do tempo do processador em pequenos intervalos (*time-slice*)
 - Conceito de fatia de tempo -> causa a impressão que o recurso computacional está inteiramente a disposição do usuário.
 - O job tem sua execução interrompida ao final do tempo alocado (terminada ou não a tarefa).

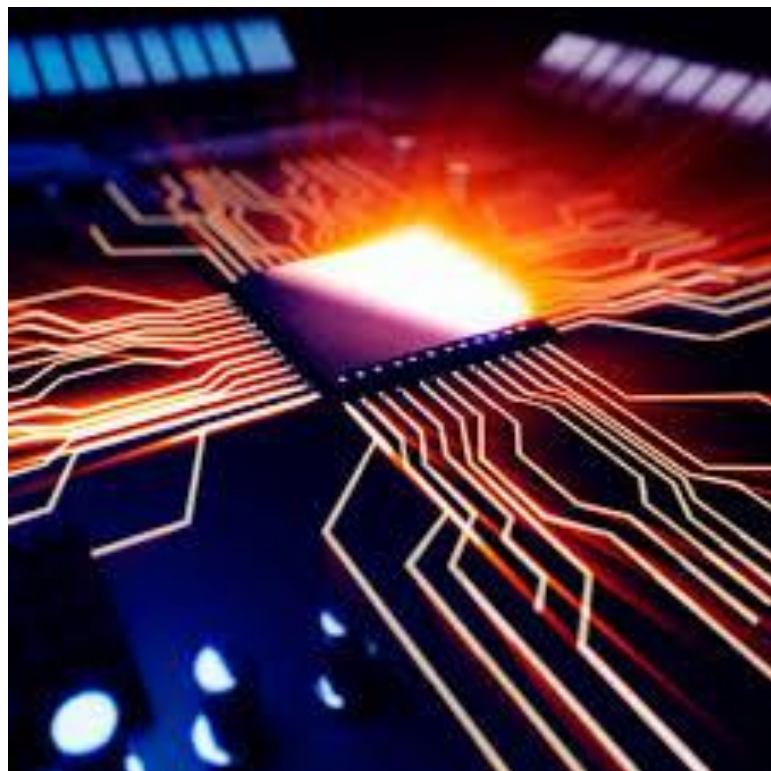


UMA DÚVIDA?



Tirar um Job não terminado do processamento tem ligação com o conceito de preemptividade?

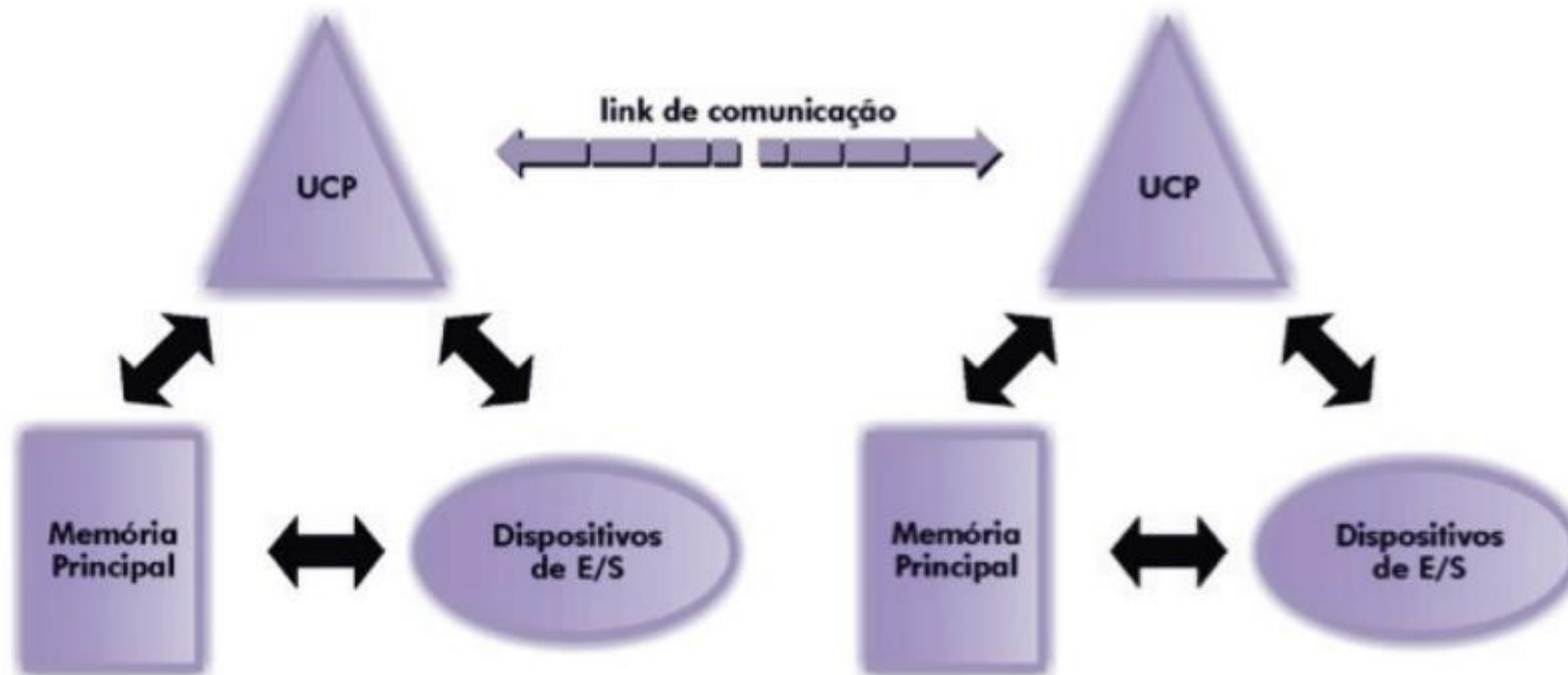
TIPOS DE SISTEMAS OPERACIONAIS



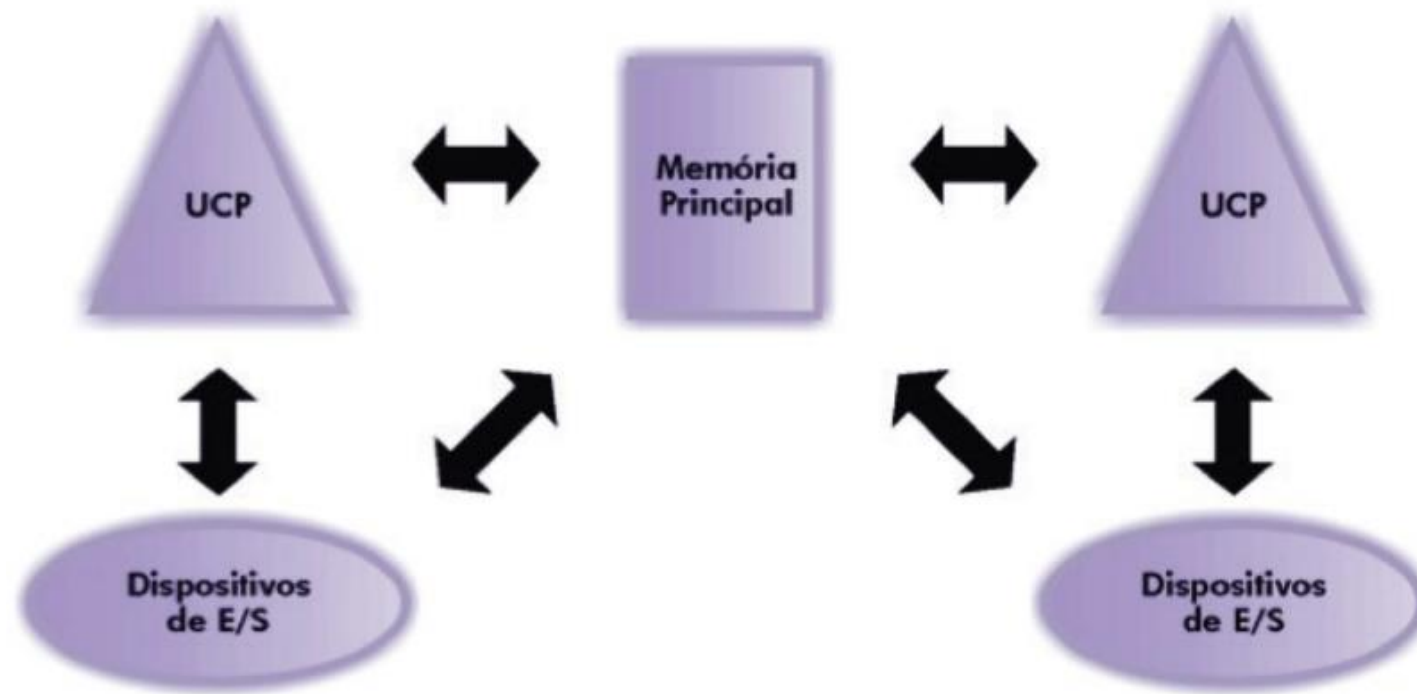
- **Sistemas de Tempo Real**
 - Caracteriza-se pela restrição do tempo exigido no processamento das aplicações.(limites rígidos)
 - Não existe ideia de fatia de tempo.
 - Prioridade de execução.
 - Aplicações de controle de processos, monitoramento de refinarias de petróleo, controle de tráfego aéreo, usinas termoeletricas, etc.

SISTEMAS COM MÚLTIPLOS PROCESSADORES

- Sistemas que operam com mais de um processador;
- Viabilizou grande desenvolvimento nas áreas de processamento de imagens, pesquisas aeroespaciais e simuladores;
- Vantagens adicionais: Escalabilidade, Disponibilidade e balanceamento de carga;
- A forma de construção dos sistemas operacionais para múltiplos processadores, tem na comunicação entre os processadores e compartilhamento da memória um fator-chave de funcionamento.

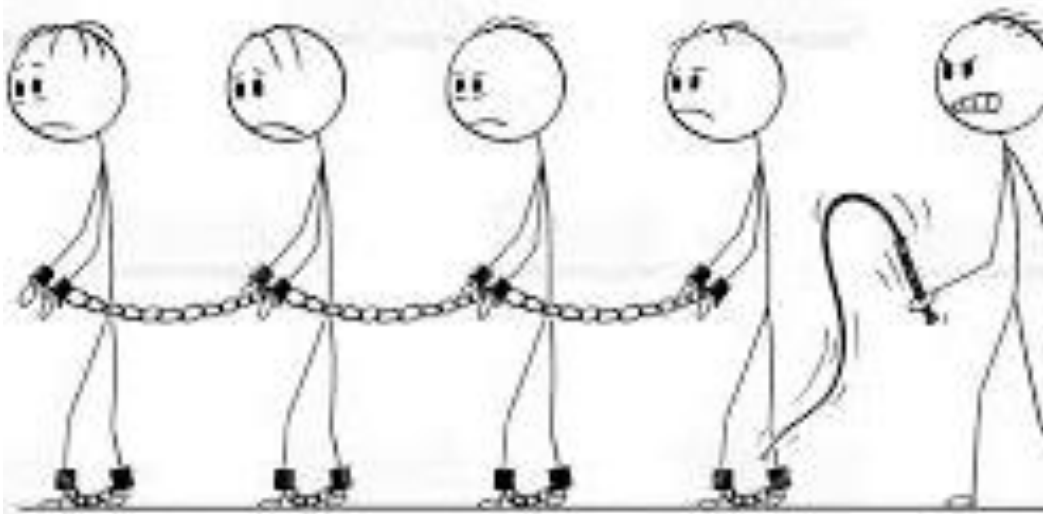


SISTEMAS FRACAMENTE ACOPLADOS



SISTEMAS FORTEMENTE ACOPLADOS

SISTEMAS FORTEMENTE ACOPLADOS ASSIMÉTRICOS



- Apenas um processador pode executar funções do sistema operacional.
- Os demais processadores (escravos) devem solicitar autorização do processador mestre
- O processador mestre é aquele que concentra a execução de todas as instruções privilegiadas, aquelas que executam apenas no modo kernel
- Os processadores escravos executam instruções não privilegiadas e são gerenciados pelo processador mestre

SISTEMAS FORTEMENTE ACOPLADOS SIMÉTRICOS



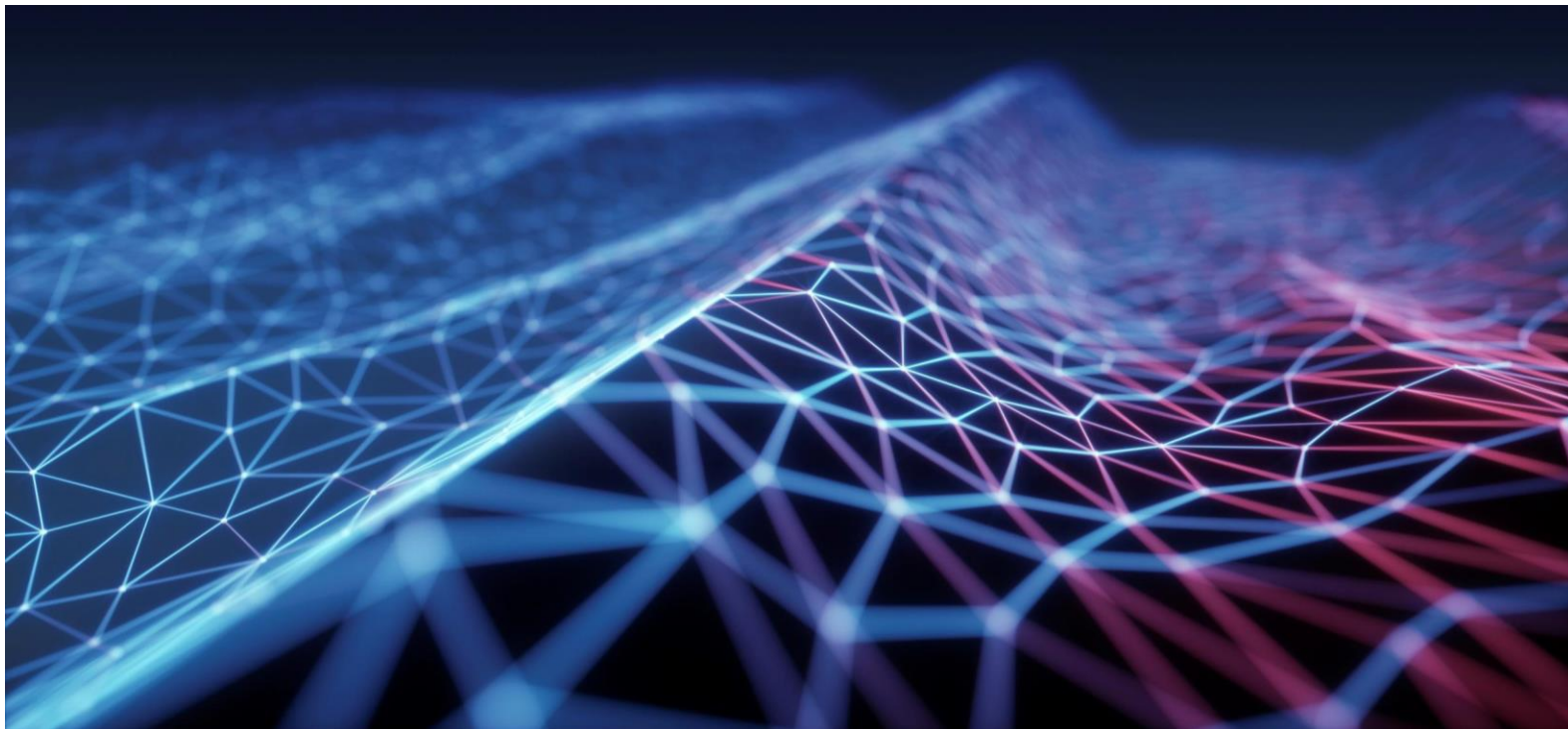
- Os processadores podem executar os serviços do sistema operacional.
- Apenas algumas funções ficam a cargo de um processador.
 - Boot de sistema (inicialização do sistema)
- Não há problema se algum processador falhar,
 - Ocorre apenas a diminuição da capacidade de computação.

QUANTIDADE DE USUÁRIOS

- Quanto à quantidade de usuários que podem utilizar o SO concorrentemente:
 - Monousuário: apenas um usuário por vez (apesar de poder suportar recursos como troca de usuário).
 - Ex.: Windows. Esse também pode ser acessado por terminais e conexão remota.
 - Multiusuário: vários usuários usam o computador ao mesmo tempo, seja por diversos terminais, seja por conexão remota como o SSH.
 - Ex.: Linux, Unix.

O QUE FICOU NA MENTE?

1. Com base no que estudamos, na sua opinião qual a grande diferença entre sistemas monoprogramáveis e multiprogramáveis?
2. Enquanto uma tarefa aguarda uma ação do usuário, outra tarefa está em execução. Qual tipo de Sistema?
2. O que caracteriza o processamento batch? Quais aplicações podem ser processadas neste tipo de ambiente?
3. Como funcionam os sistemas de tempo compartilhado? Quais as vantagens em utilizá-los?
4. Para quais finalidades os sistemas de tempo real são mais indicados?
5. Explique o que você entendeu sobre sistemas fortemente acoplados e fracamente acoplados.
6. Os sistemas operacionais de uso doméstico já utilizam os recursos de multiprocessadores? Explique de onde tirou a sua conclusão.
7. Qual seria o tipo de sistema operacional recomendável para uso como servidor de aplicações em um ambiente corporativo?

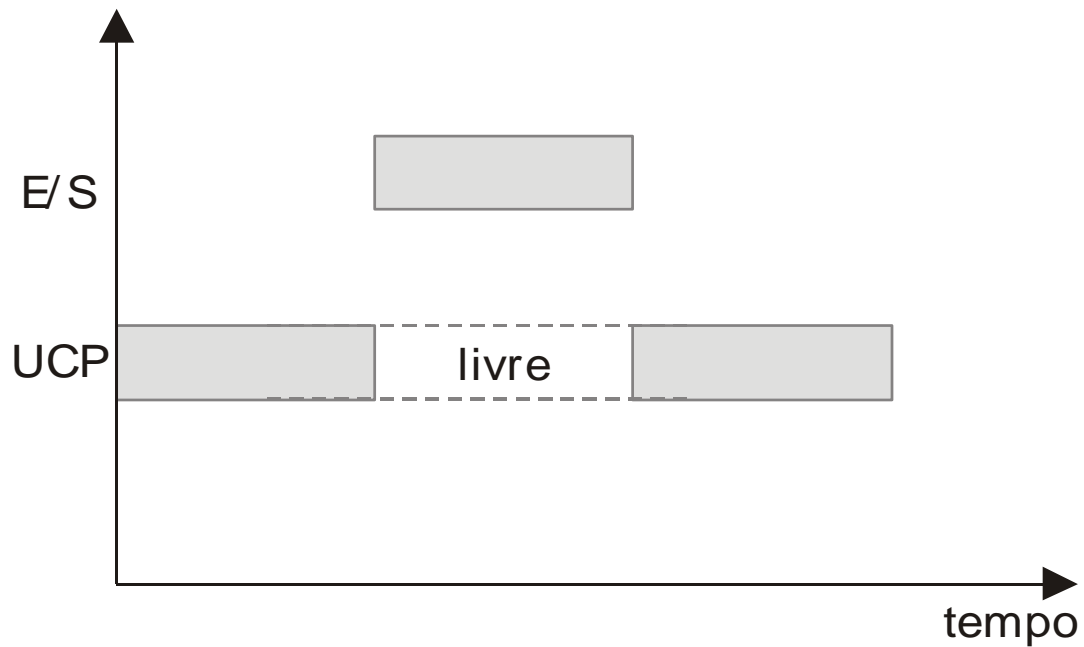


CONCORRÊNCIA

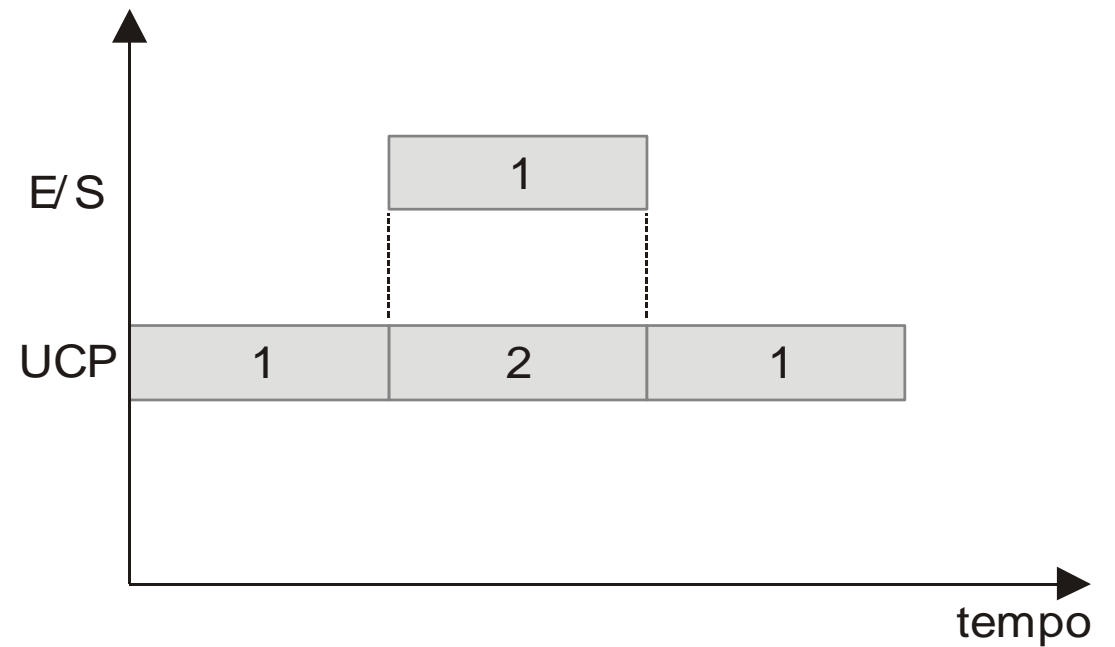
OBJETIVOS

- Interrupção e exceção
- Operações de E/S
- Buffering
- Spooling
- Reentrância

SISTEMA MONOPROGRAMÁVEL X MULTIPROGRAMÁVEL

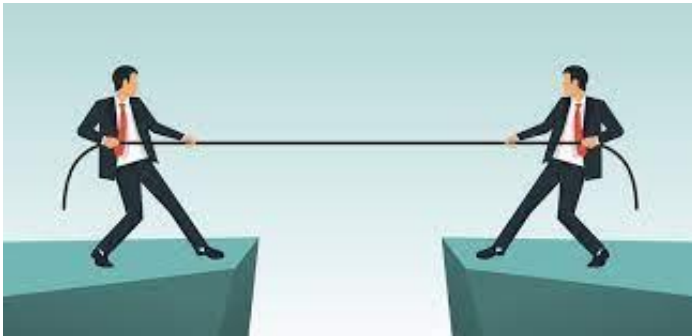


(a) Sistema Monoprogramável



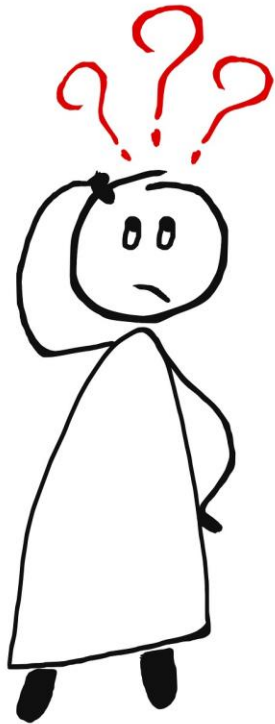
(b) Sistema Multiprogramável

CONCORRÊNCIA



- Sistemas operacionais podem ser vistos como um conjunto de rotinas executadas de forma **concorrente** e ordenada (Pinkert, 1990);
- A capacidade do processador executar instruções e ao mesmo tempo solicitar operações de E/S, permite que diversas tarefas sejam executadas ao mesmo tempo;
- O conceito de concorrência é o princípio básico para o projeto e a implementação dos sistemas multiprogramáveis.

UMA DÚVIDA...

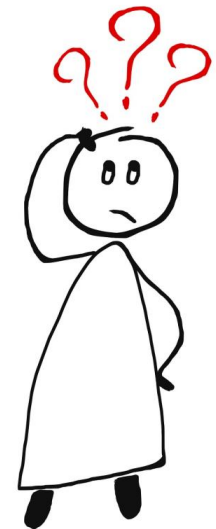


- Seria correto afirmar que apenas o processamento seria melhor aproveitado nos sistemas multiprogramáveis?

CONCORRÊNCIA NA PRÁTICA

	Monoprogramação	Multiprogramação
Utilização da UCP	17%	33%
Utilização da memória	30%	67%
Utilização de disco	33%	67%
Utilização de impressora	33%	67%
Tempo total de processamento	30 min	15 min
Taxa de throughput	6 prog./hora	12 prog./hora

Características	Prog1	Prog2	Prog3
Utilização da UCP	Alta	Baixa	Baixa
Operação de E/S	Poucas	Muitas	Muitas
Tempo de processamento	5 min	15 min	10 min
Memória utilizada	50 Kb	100 Kb	80 Kb
Utilização de disco	Não	Não	Sim
Utilização de terminal	Não	Sim	Não
Utilização de impressora	Não	Não	Sim



SISTEMAS MONOPROGRAMÁVEIS

- 93% em operações de E/S
- Subutilização da Memória Principal
- Programas executados sequencialmente
- Ex:
 - Prog 1 – 5 min
 - Prog 2 – 15 min
 - Prog 3 – 10 min

RESUMINDO A CONCORRÊNCIA...



- Sob o âmbito de sistemas operacionais podemos definir concorrência como:
 - A execução de tarefas concorrentes, significando na prática, vários processos disputando entre si pela utilização dos diversos periféricos e dispositivos do sistema.

CONCORRÊNCIA - VANTAGENS

Aumento da utilização da CPU;

Utilização mais eficiente da memória principal;

Compartilhamento dos dispositivos de e/s pelos vários usuários;

Melhor tempo de resposta na execução das tarefas.

INTERRUPÇÕES E EXCEÇÕES

- Eventos inesperados que provocam um desvio no fluxo de execução de um programa;
- São provenientes da sinalização de algum dispositivo de hardware ou pela execução de instruções do próprio programa;
- A interrupção é o mecanismo que tomou possível a implementação da concorrência nos computadores, sendo o fundamento básico dos sistemas multiprogramáveis;
- A diferença entre interrupção e exceção é dada pelo tipo de evento ocorrido.

INTERRUPÇÃO



- Ao se executar programas podem haver situações em que o S.O. precisa intervir, ocorrendo a Interrupção
- A execução do programa é desviada para uma rotina de tratamento
- Este mecanismo é básico para a implementação da concorrência e conseqüentemente para a implementação da multiprogramação
- Sincroniza a execução de todas as rotinas e controle dos dispositivos.

INTERRUPÇÃO INTERNA (EXCEÇÃO)



- Resultado direto da execução do programa.
- Pode ser escrito pelo próprio programador.
- A instrução do programa gera a interrupção e é executada de forma síncrona

- Exemplo:
 - erro de divisão por zero (overflow)

INTERRUPÇÃO EXTERNA



- Gerado pelo Sistema Operacional
- O programa que está sendo executado é interrompido para atender a solicitação de interrupção do dispositivo externo (impressora, hd, mouse)
- Evento assíncrono

INTERRUPÇÃO E EXCEÇÃO

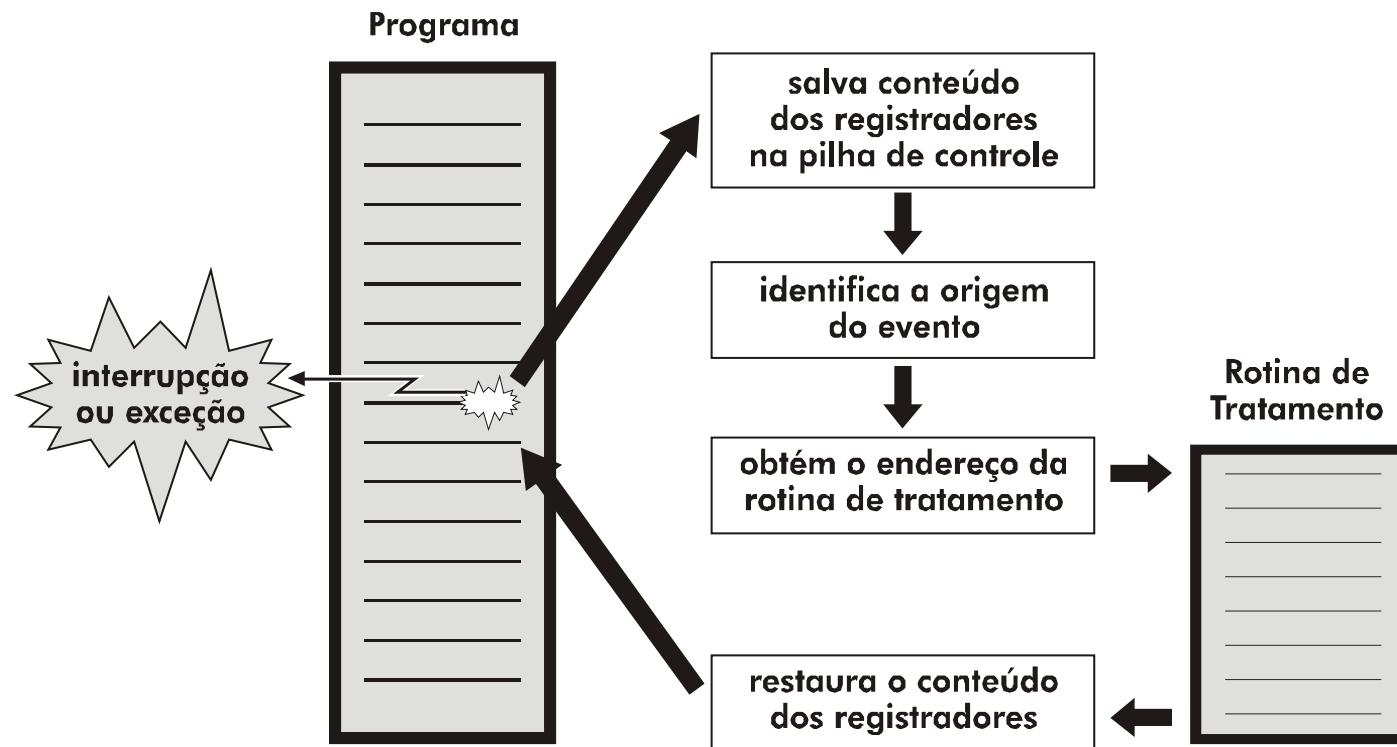


**Questão boa
para uma
prova!**

Uma interrupção é sempre gerada por algum evento externo ao programa e, nesse caso, independe da instrução que está sendo executada;

A exceção é resultado direto da execução de uma instrução do próprio programa, como a divisão de um número por zero ou a ocorrência de overflow.

INTERRUPÇÕES - FLUXO DE ATENDIMENTO



TRATAMENTO DE INTERRUPÇÕES E EXCEÇÕES

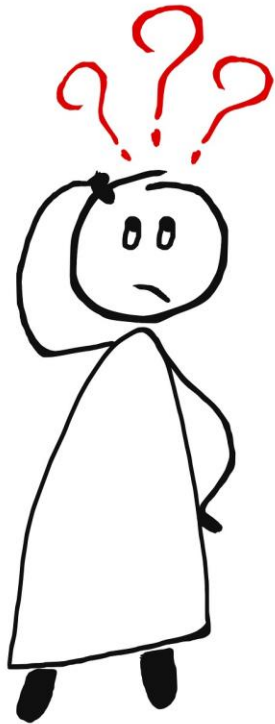


- Temos diferentes tipos de interrupções e cada qual necessita de uma rotina de tratamento
- As instruções da rotina de tratamento são mantidas no vetor de interrupções, o qual contém o endereço inicial da sequência de instruções da rotina de tratamento
- No caso das interrupções internas, o próprio desenvolvedor pode escrever o tratamento que quer dar para uma interrupção

EVENTOS SÍNCRONOS E ASSÍNCRONOS

-
- Um evento síncrono é resultado direto da execução de um programa corrente, sendo tais eventos previsíveis e, por definição, só podem ocorrer um de cada vez;
 - Os eventos assíncronos podem acontecer múltiplas vezes, como no caso de diversos dispositivos de E/S informarem ao processador que estão prontos para receber ou transmitir dados;
 - A diferença fundamental entre exceção e interrupção é que a primeira é gerada por um evento síncrono, enquanto a segunda é gerada por eventos assíncronos.

UMA DÚVIDA...



Um programa coleta dois números para dividir, se o divisor informado for zero, a rotina que trata a exceção ocorrerá sempre no mesmo bloco de instruções?

TRATANDO AS INTERRUPÇÕES

- Por serem eventos assíncronos (e imprevisíveis), com possibilidade múltiplas de incidências, é necessário mecanismos para tratar a simultaneidade;
- Uma das maneiras é desabilitando as interrupções, quando uma está sendo tratada (interrupções mascaráveis);
- Outro caminho é usar o controlador de pedidos de interrupção, responsável por avaliar as interrupções geradas e suas prioridades de atendimento.

OPERAÇÕES DE ENTRADA E SAÍDA



Inicialmente o controle das operações (E/S) era executada pelo processador que deveria monitorar o término;



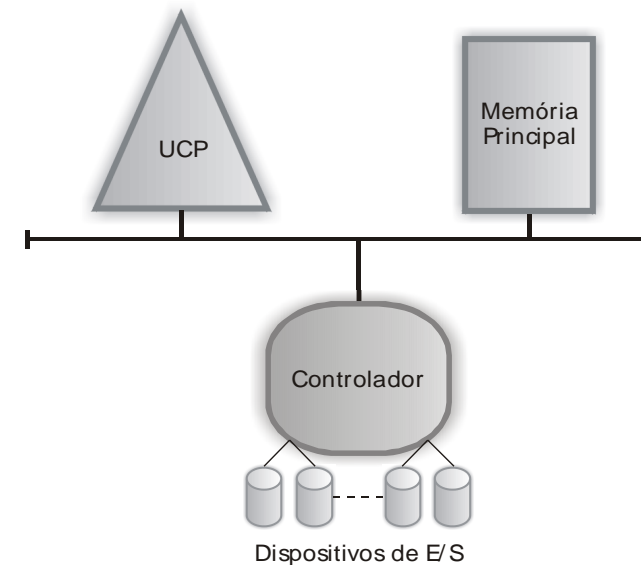
Com o uso do recurso das interrupções, os periféricos passaram a informar quando finalizaram a operação solicitada.

OPERAÇÕES DE ENTRADA/SAÍDA

- Inicialmente executadas pelo próprio processador
- Surgimento do controlador
- Gerenciamento das operações
 - E/S controlada por programa – mantinha o processador ocupado até o término
 - Polling – o sistema operacional testava o dispositivo p/ saber o término da operação
 - E/S controlada por interrupção – o controlador interrompia o processador para avisar o término da operação

CONTROLADOR DE INTERRUPÇÕES

- Mecanismo implementado no hardware que avalia as interrupções que são geradas e qual a sua prioridade de atendimento.
- Permite que o processador realize outras tarefas enquanto a interrupção é tratada pelo dispositivo

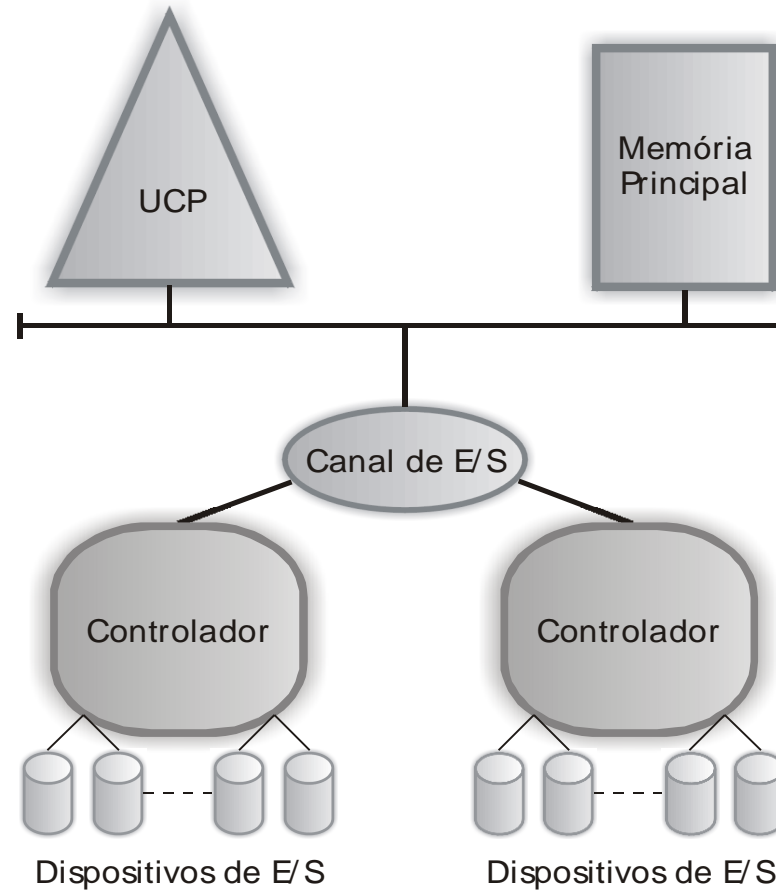


DMA

-
- Técnica de DMA – *Directy Memory Acess*
 - Permite que um bloco de dados seja transferido entre a memória principal e os dispositivos sem a intervenção do processador, exceto no início e no final da transferência.

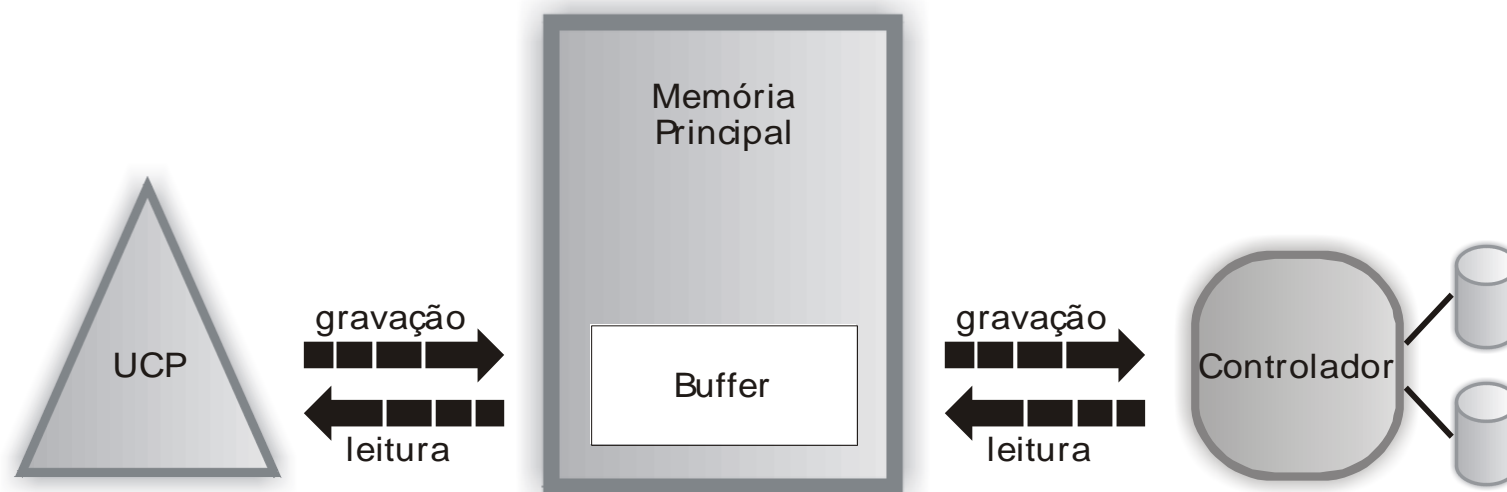
OTIMIZANDO AS OPERAÇÕES DE E/S

- Canal de E/S



BUFFERING

- Operações de E/S

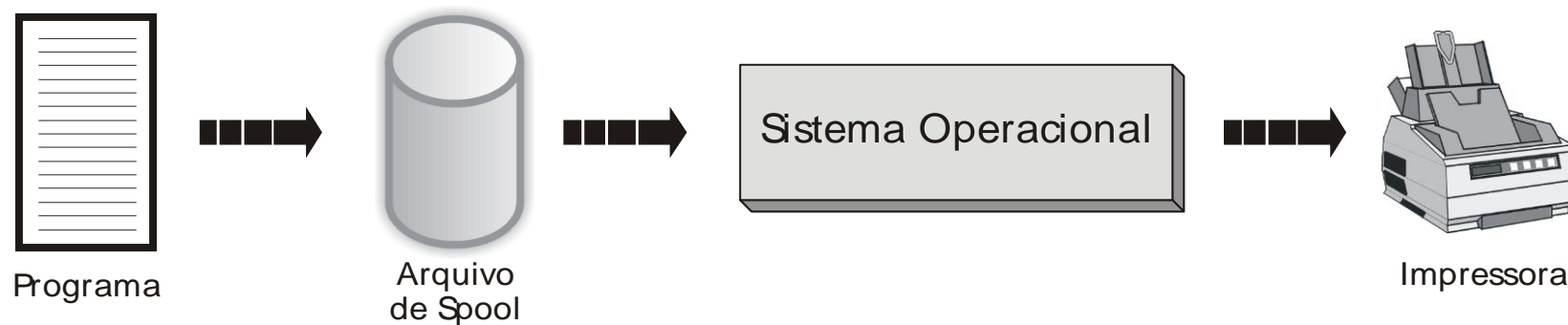


BUFFERING

- Qual a vantagem da técnica???
- Permite minimizar o problema de disparidade da velocidade de processamento entre o processador e os dispositivos de E/S.
- Objetivo: manter o processador e dispositivos ocupados.

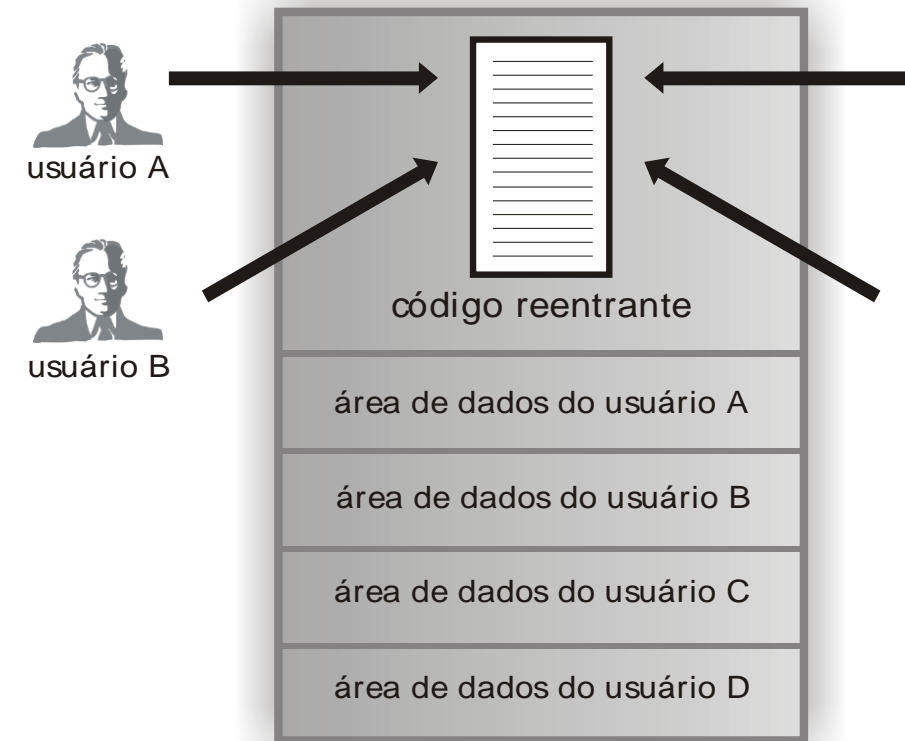
SPOOLING

- Utilização de uma área em disco rígido (HD) para armazenamento temporário de dados
- Essa implementação permite maior grau de compartilhamento e utilização de impressoras



REENTRÂNCIA

- O mesmo código de programa residente em memória é utilizado por vários usuários ao mesmo tempo.
- Proporciona grande economia de memória.
- O código reentrante não pode ser modificado pelo usuário.
- Exemplo:
 - Editores de texto;
 - Compiladores.



Memória Principal

O QUE FICOU NA MENTE?

1. O que é concorrência e como este conceito está presente nos sistemas operacionais?
2. Por que o mecanismo de interrupção é fundamental para a implementação da multiprogramação?
3. Descreva os passos que são executados quando um programa trata uma interrupção ou exceção.
4. O que são eventos síncronos e assíncronos? Como estes eventos estão relacionados ao mecanismo de interrupção e exceção?
5. O que é DMA e qual a vantagem desta técnica?
6. Como a técnica de buffering permite aumentar a concorrência em um sistema computacional?
7. Explique o mecanismo de spooling de impressão.



ESTRUTURA E ARQUITETURA DO SISTEMA OPERACIONAL

OBJETIVOS

- Proteção do Sistema
- System calls
- Modos de acesso
- Arquitetura monolítica
- Arquitetura em camadas
- Máquina virtual
- Arquitetura microkernel

PROTEÇÃO DO SISTEMA – ESTADO DE EXECUÇÃO

- Característica associada ao programa em execução
- Determina se ele pode ou não executar certas instruções ou rotinas

PROTEÇÃO DO SISTEMA

ESTADO USUÁRIO

SÓ EXECUTA INSTRUÇÕES QUE
NÃO AFETEM OUTROS
PROGRAMAS.

PROTEÇÃO DO SISTEMA - ESTADO SUPERVISOR

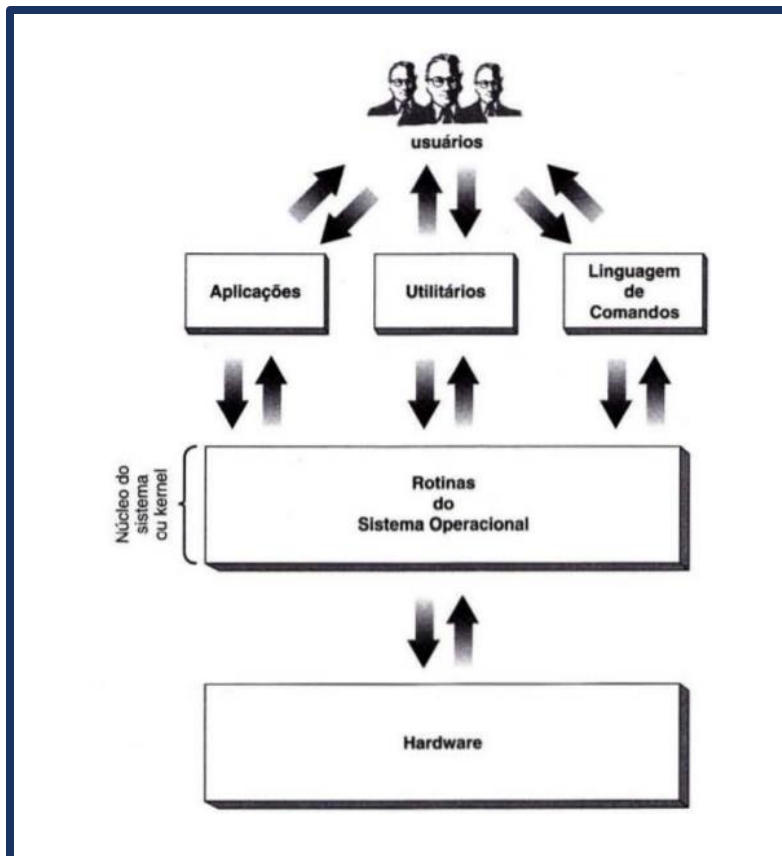


- Executa instruções privilegiadas
- Pode executar qualquer instrução do sistema operacional
- É o estado permanente do sistema operacional

ESTRUTURA DO SISTEMA OPERACIONAL

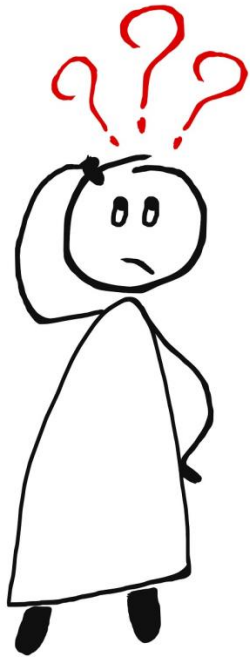
- O sistema operacional é formado por um conjunto de rotinas que oferece serviços aos usuários e às suas aplicações;
- Esse conjunto de rotinas é denominado núcleo do sistema ou kernel;
- Entretanto, nem todo o conjunto de rotinas e aplicativos fazem parte do núcleo do sistema operacional, mas servem de ponte ao kernel.

ESTRUTURA DO SISTEMA OPERACIONAL



- **A Comunicação com o Sistema Operacional pode ocorrer de 3 maneiras:**
 - por intermédio das chamadas de rotinas do sistema realizadas por aplicações;
 - acionando os utilitários do próprio sistema operacional;
 - Usando a linguagem de comandos nativa do sistema operacional (compiladores);
- **Serão apresentadas as funções do núcleo e os conceitos relativos à segurança, proteção do sistema, modos de acesso, rotinas do sistema, system calls, linguagem de comandos e ativação/desativação do sistema**

UMA DÚVIDA...

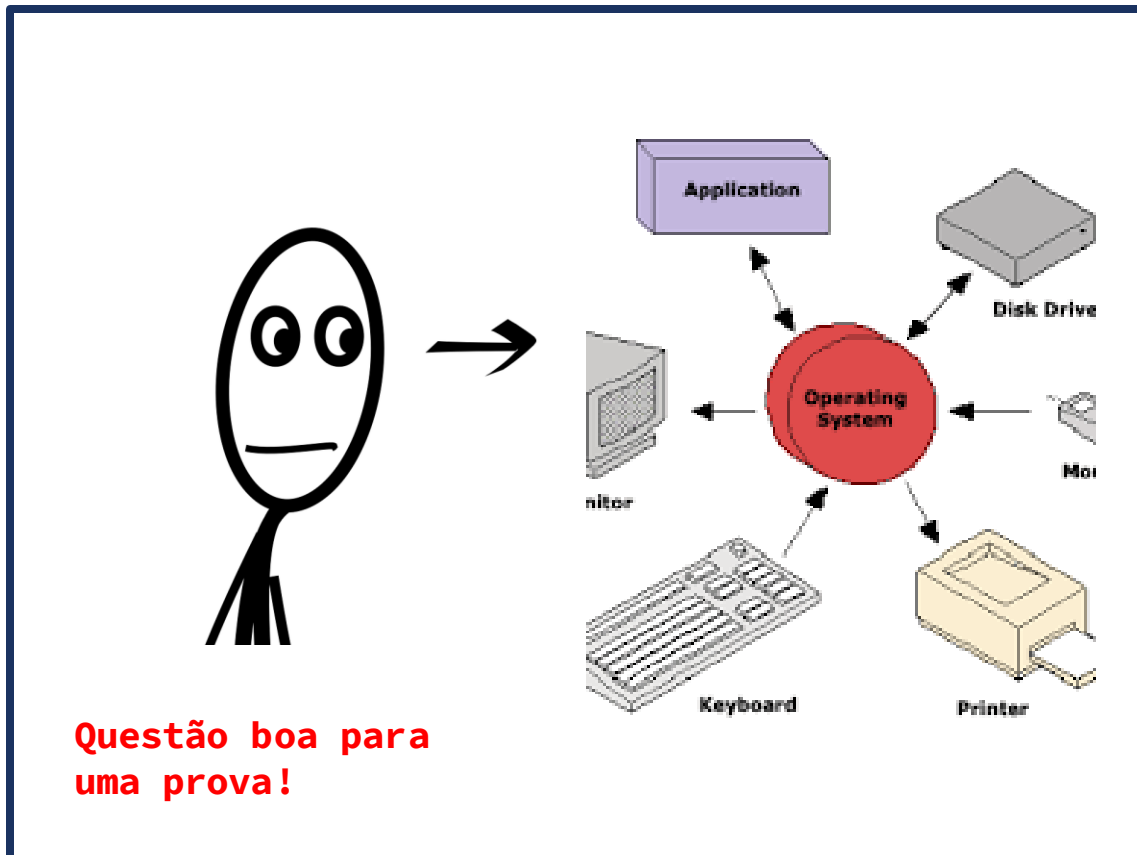


- Quando acionamos o gerenciador de tarefas do Sistema Operacional, de que maneira estamos interagindo com ele?
- Aplicação
- Utilitário
- Linguagem de Comando

FUNÇÕES DO KERNEL(NÚCLEO)

- A compreensão da estrutura e do funcionamento de um sistema operacional não é simples, pois as rotinas do sistema são executadas concorrentemente sem uma ordem pré definida, com base em eventos dissociados do tempo (eventos assíncronos);
- Muitos desses eventos estão relacionados ao hardware e a tarefas internas do próprio sistema operacional;
- Trabalharemos um conjunto dessas funcionalidades para compreender o modus operandi dos Sistemas Operacionais.

FUNÇÕES DO NÚCLEO DO S.O.



- Tratamento de interrupções e exceções.
- Criação e eliminação de processos e threads.
- Sincronização e comunicação entre processos e threads.
- Escalonamento e controle de processos e threads.
- Gerência de memória.
- Suporte a redes locais e distribuída.
- Contabilização do uso do sistema.
- Auditoria e segurança do sistema.
- Gerência do sistema de arquivos.
- Gerência de dispositivos de e/s.

Machado e Maia (2007),

SYSTEM CALLS

- Funções
 - Gerência de processos e threads
 - Gerência de memória
 - Gerência do sistema de arquivos
 - Gerência de dispositivos

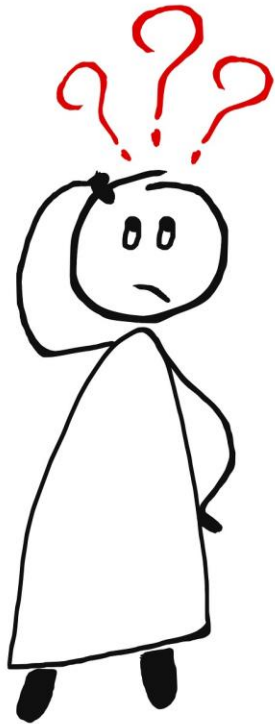
COMO PROTEGER O NÚCLEO DO SISTEMA OPERACIONAL?

- “Caso uma aplicação, com acesso ao núcleo, realize uma operação que altere a integridade, todo sistema fica comprometido.”
- *Modos de acesso: Modo Kernel e Modo Usuário*
- *Instruções privilegiadas e não-privilegiadas*

MODOS DE ACESSO

- Tipos de instruções
 - Privilegiadas
 - Não-privilegiadas
- Modos de acesso
 - Usuário
 - Kernel ou supervisor

UMA DÚVIDA...



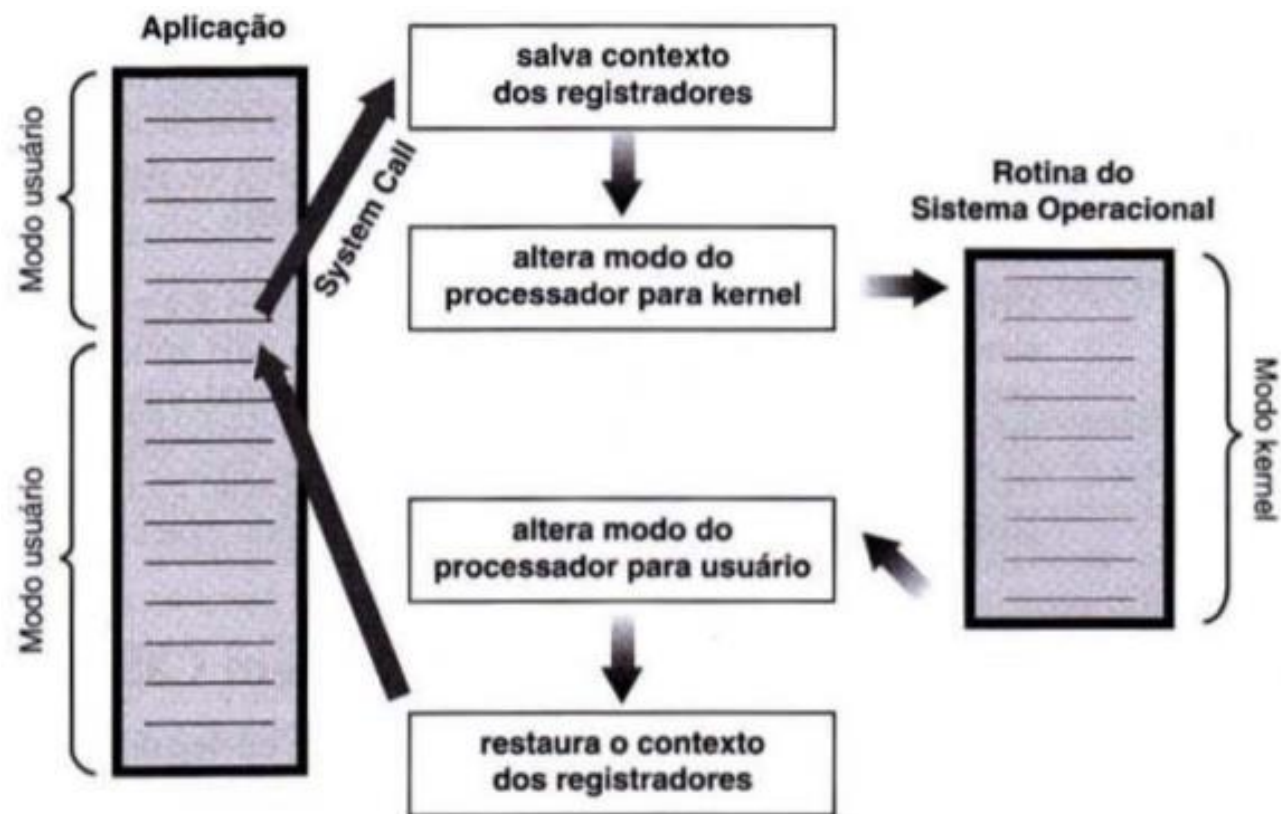
O aplicativo DEFRAG do Windows opera em qual modo?

- Usuário
- Kernel

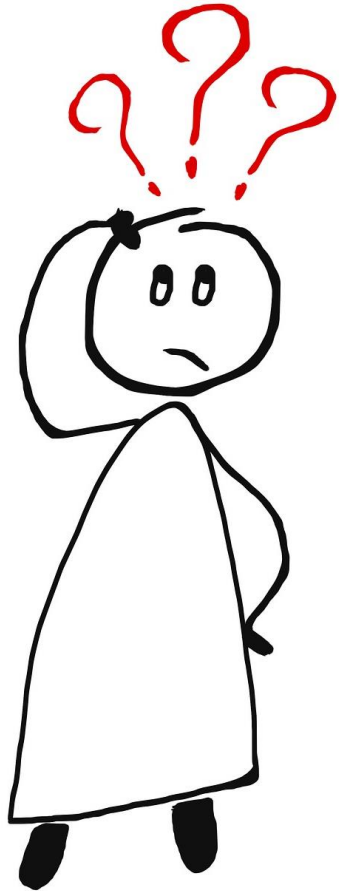
MODOS DE ACESSO – CHAMADA A UMA ROTINA DO S.O.

- Sempre que acontece uma chamada de sistema, o sistema é alterado de modo usuário para modo kernel,

O fluxo ao lado mostra com uma aplicação com privilégio de execução é tratada pelo Sistema Operacional

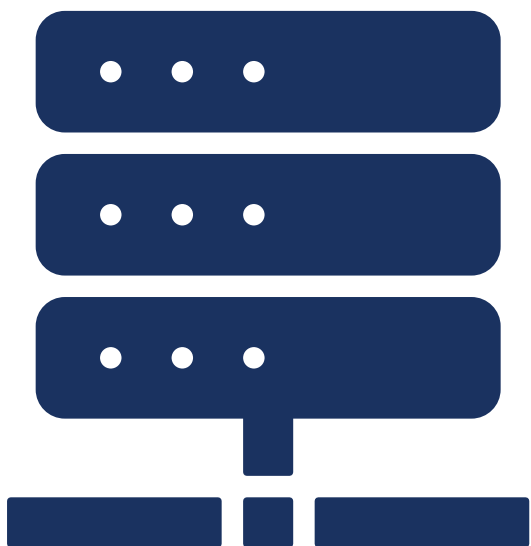


UMA DÚVIDA...



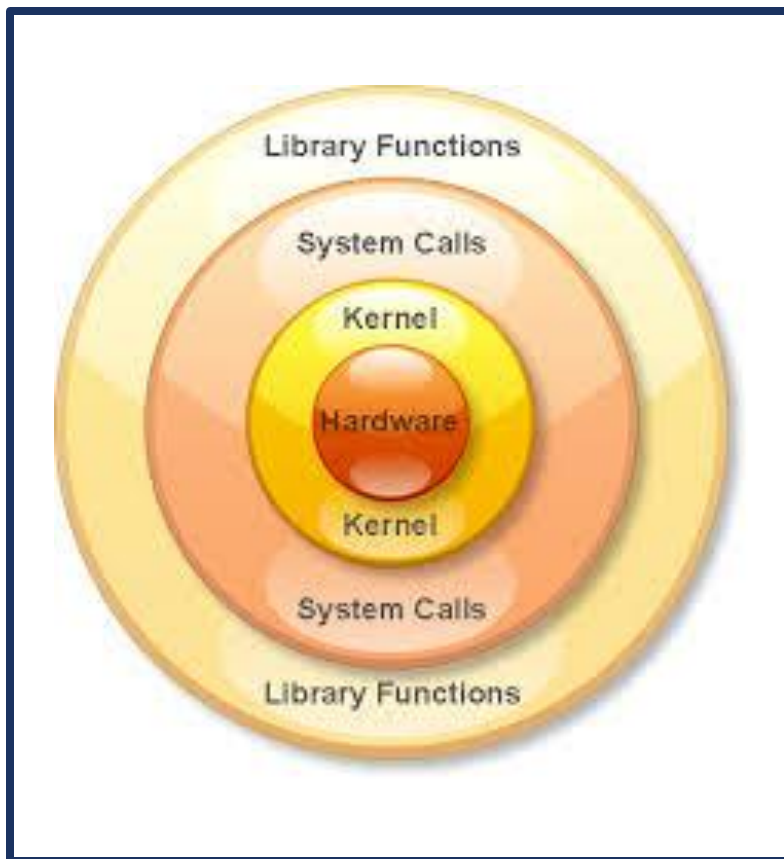
Quando você salva um arquivo do Word no disco, podemos dizer que existe a troca do modo de acesso de usuário para kernel?

TIPOS DE ARQUITETURA DOS SISTEMAS OPERACIONAIS



- Em relação ao seu projeto(arquitetura) segundo Tanenbaum(2011):
 - Arquitetura monolítica ou monobloco.
 - Arquitetura em camadas.
 - Monitor de máquinas virtuais.
 - Arquitetura microkernel.

SYSTEM CALLS

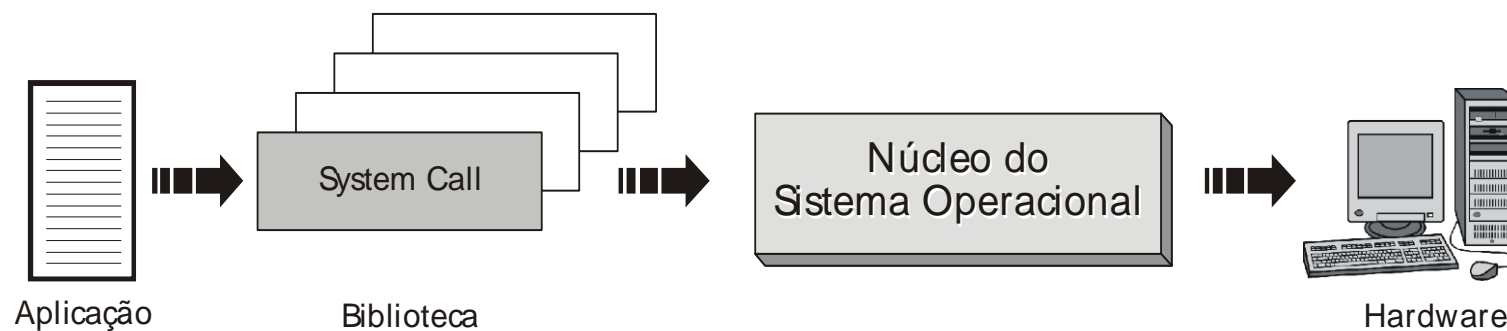


- Chamada ao sistema, system call, é um mecanismo criado para proteger o núcleo do sistema operacional.
- É a porta de entrada para o núcleo do sistema

SYSTEM CALLS

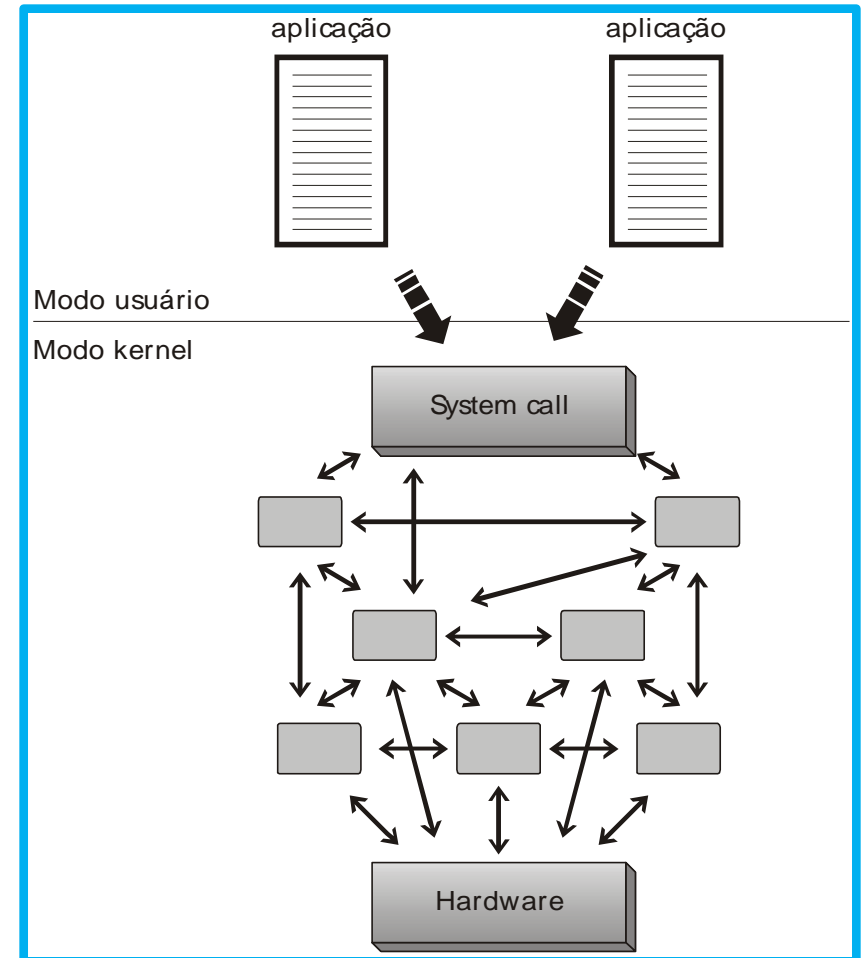


- Para cada serviço de um determinado sistema operacional, existe uma system call com parâmetros e formas de ativação específicos.
- Por isso, as aplicações que são desenvolvidas para serem executadas em um determinado sistema não são portáveis para outros.



ARQUITETURA MONOLÍTICA

- As rotinas do sistema são desenvolvidas e depois é gerado o código executável, formando um único arquivo de programa executável.
- Desvantagem:
 - Difícil manutenção e depuração
- Vantagem
 - Ótimo desempenho
 - Simplicidade



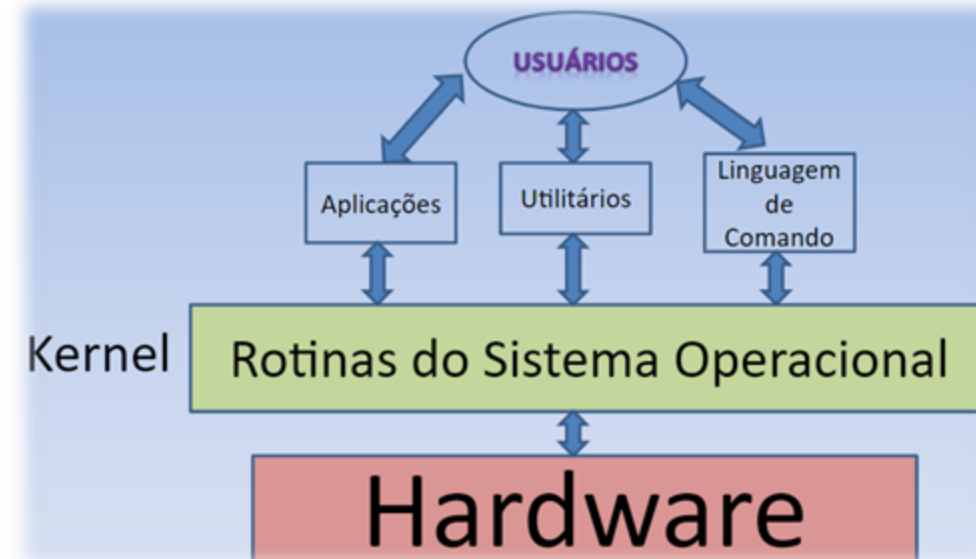
ARQUITETURA DE CAMADAS



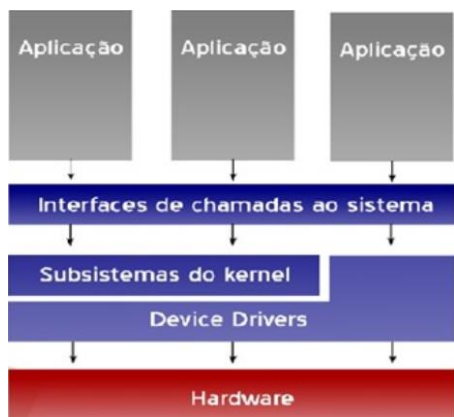
- O sistema é dividido em camadas sobrepostas
- Cada camada possui funções que atendem às camadas superiores.
- Objetiva isolar as funções do sistema operacional, facilitando
 - a depuração;
 - a manutenção;
 - Possibilita maior segurança, criando uma hierarquia de acesso ao kernel.

ACESSO AO NÚCLEO

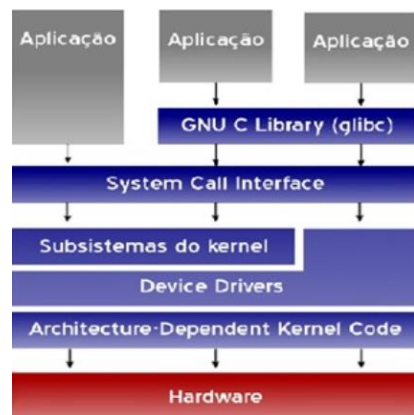
- Como o usuário pode se comunicar com o núcleo do sistema operacional (kernel)?
 - Rotinas do sistemas através das aplicações;
 - Utilitários;
 - Linguagem de comandos.



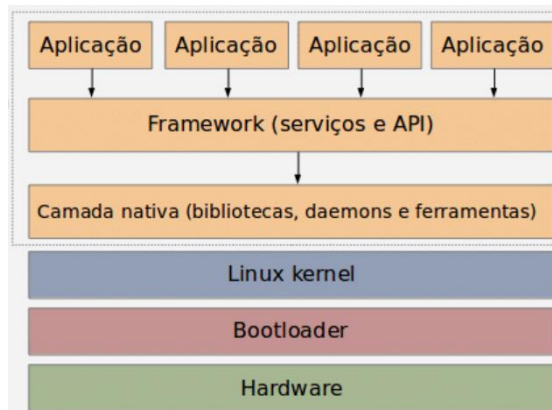
ARQUITETURAS DE S.O.



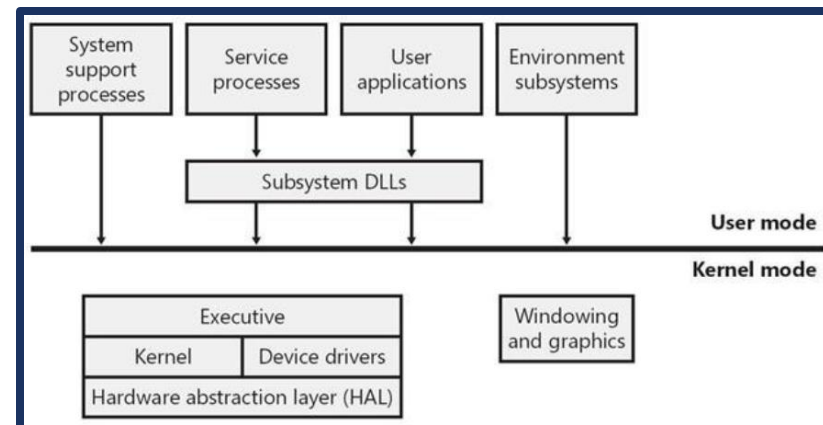
Linux



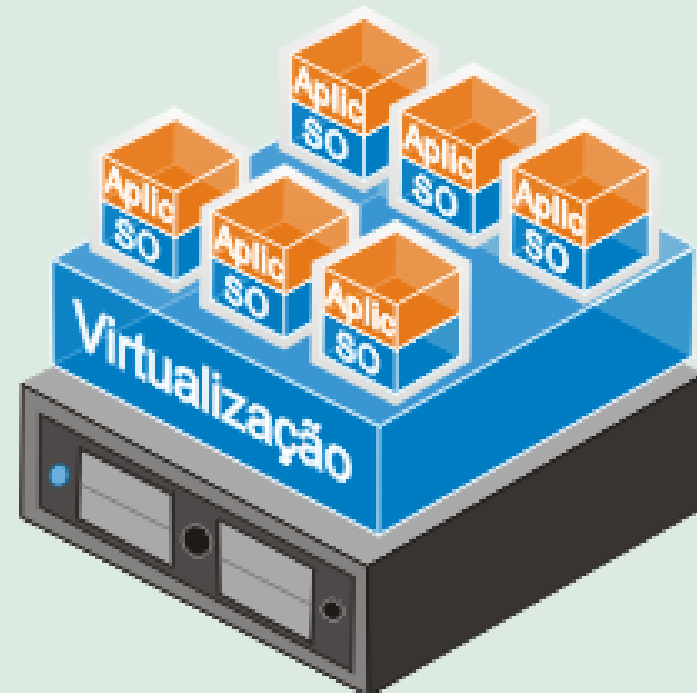
GNU/Linux



Android

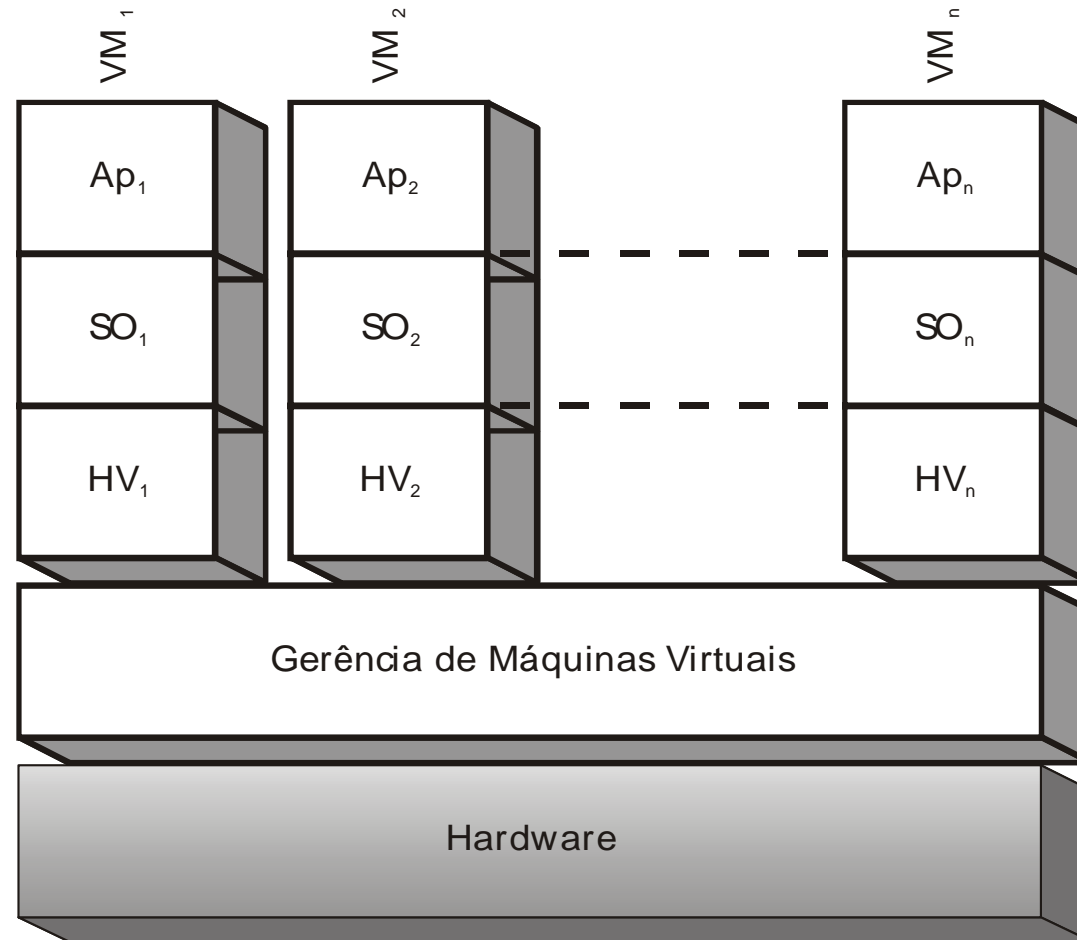


Windows

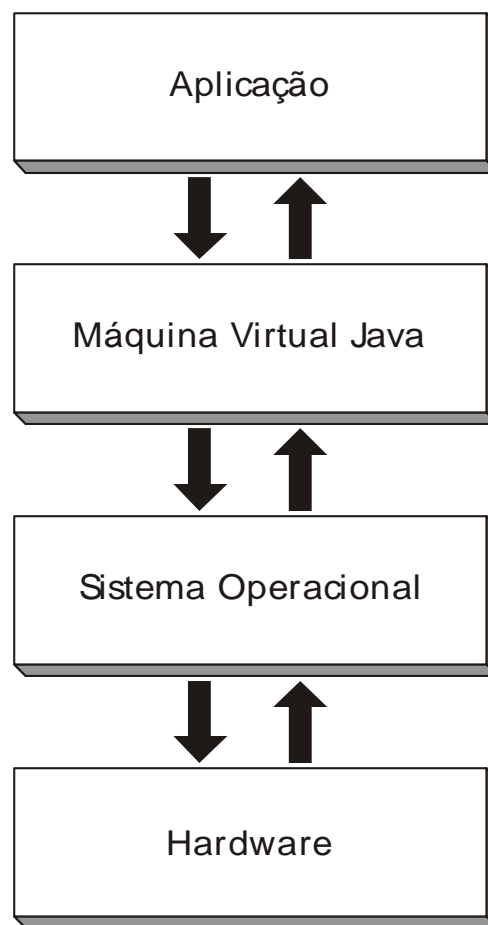


VIRTUALIZAÇÃO

MÁQUINA VIRTUAL



JAVA VIRTUAL MACHINE – ARQUITETURA DE FUNCIONAMENTO

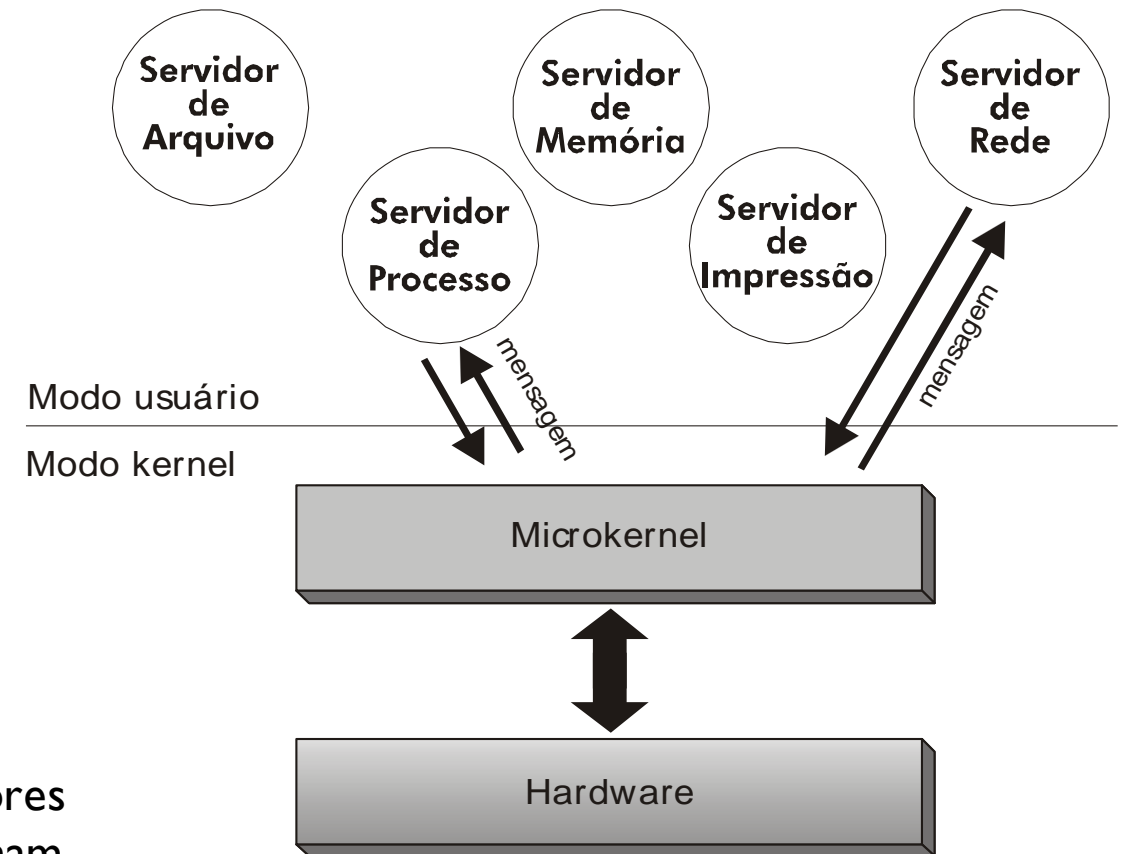


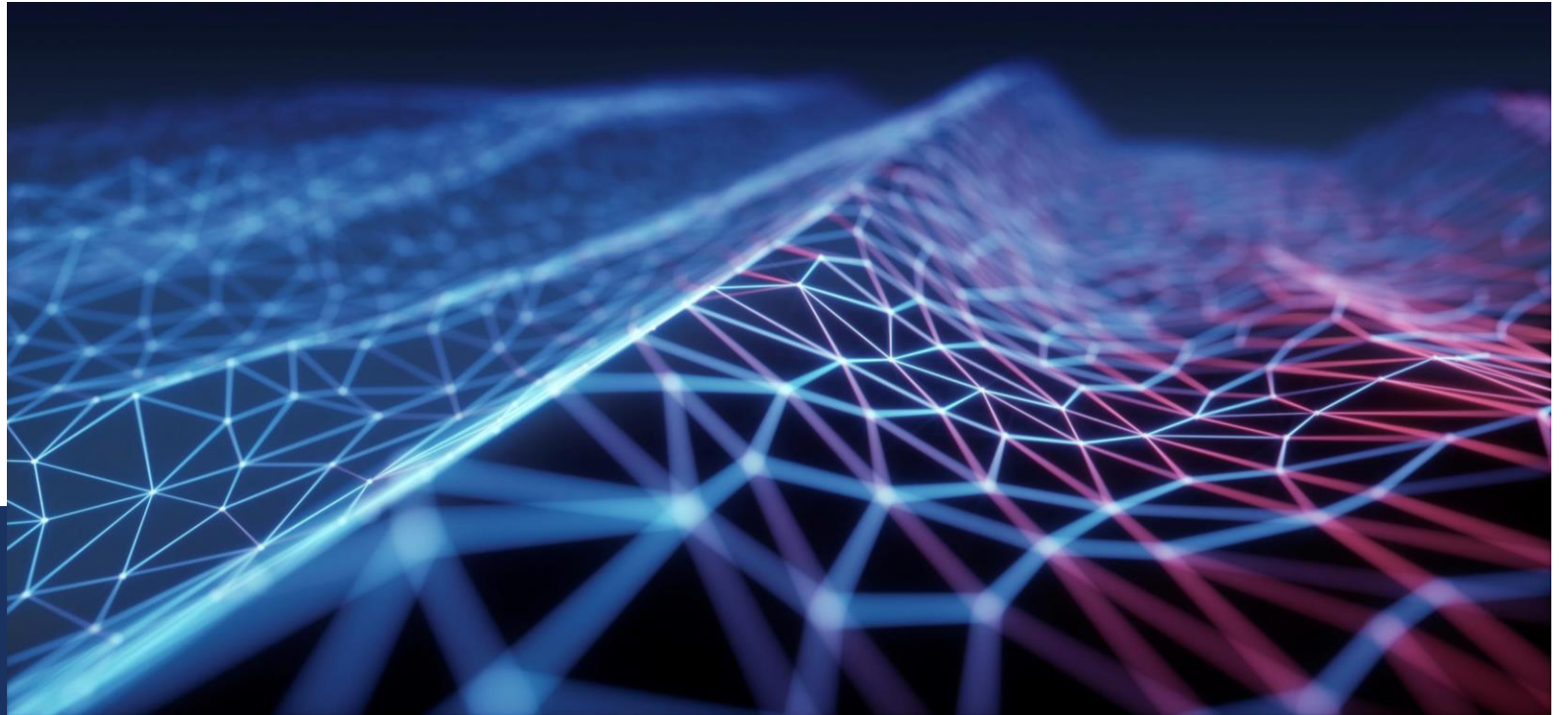
ARQUITETURA MICROKERNEL - CARACTERÍSTICAS

- Ou micronúcleo ou modelo cliente servidor
- Tornar o núcleo do sistema menor e mais simples;
- Os serviços do sistema operacional são oferecidos por meio de processos que recebem pedidos das aplicações, executam o pedido e devolvem a resposta à aplicação;
- O kernel fica encarregado apenas de controlar a comunicação entre o cliente (aplicação) e o servidor (rotinas do sistema operacional).

“A utilização deste modelo permite que os servidores executem em modo usuário, ou seja, não tenham acesso direto a certos componentes do sistema.”

(MACHADO; MAIA, 2007, p. 61).





PROCESSOS

OBJETIVOS

- Introdução
- Estrutura do processo
- Estados do processo
- Mudanças de estado do processo
- Criação e eliminação de processos
- Processos independentes, subprocessos e threads
- Processos foreground e background
- Processos do sistema operacional
- Processos CPU-bound e I/O-bound
- Sinais

SISTEMA OPERACIONAL

O que você entende por Processo no SO?

PROCESSOS

- Um programa para estar em execução deve estar associado a um **processo** no sistema operacional;
- A gerência de processos permite aos programas alocar recursos, compartilhar dados, trocar informações e sincronizar suas execuções;
- Nos sistemas multiprogramáveis os processos são executados concorrentemente, compartilhando o uso do processador e memória principal;
- Nos sistemas com múltiplos processadores não só existe a concorrência de processos pelo uso do processador como também a possibilidade de execução simultânea de processos nos diferentes processadores.

Um programa para estar em execução deve estar associado a um **processo** no sistema operacional;

A gerência de processos permite aos programas alocar recursos, compartilhar dados, trocar informações e sincronizar suas execuções;

Nos sistemas multiprogramáveis os processos são executados concorrentemente, compartilhando o uso do processador e memória principal;

Nos sistemas com múltiplos processadores não só existe a concorrência de processos pelo uso do processador como também a possibilidade de execução simultânea de processos nos diferentes processadores.

PROCESSOS

PROCESSOS



O conceito de processo pode ser definido como sendo o conjunto necessário de informações para que o sistema operacional implemente a concorrência de programas;



Para que a concorrência entre os programas ocorra sem problemas, é necessário que todas as informações do programa interrompido sejam guardadas para que, quando este voltar a ser executado, não lhe falte nenhuma informação necessária à continuação do processamento.

Gerenciador de Tarefas

Arquivo Opções Exibir

Processos Desempenho Histórico de aplicativos Inicializar Usuários Detalhes Serviços

Nome	PID	Status	Nome de usuário	CPU	Memória (co...	Virtualização d...
ApplicationFrameHos...	15992	Em execução	msi	00	996 K	Desabilitado
AppVShNotify.exe	4532	Em execução	SISTEMA	00	20 K	Não permitido
AppVShNotify.exe	9184	Em execução	msi	00	20 K	Desabilitado
audiodg.exe	18252	Em execução	SERVIÇO LOCAL	03	102.720 K	Não permitido
bdredline.exe	4576	Em execução	SISTEMA	00	20 K	Não permitido
Calculator.exe	13176	Suspenso	msi	00	0 K	Desabilitado
chrome.exe	13624	Em execução	msi	01	178.192 K	Desabilitado
chrome.exe	880	Em execução	msi	00	504 K	Desabilitado
chrome.exe	8804	Em execução	msi	00	29.096 K	Desabilitado
chrome.exe	8824	Em execução	msi	00	2.812 K	Desabilitado
chrome.exe	14768	Em execução	msi	00	257.808 K	Desabilitado
chrome.exe	16292	Em execução	msi	00	1.076 K	Desabilitado

PROCESSOS - VISUALIZAÇÃO WINDOWS

```
[root@localhost ~]# ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  12.4  1.3   3136  2516 ?        S    13:01   0:09 /bin/sh /sbin/i
root         2   0.0  0.0     0     0 ?        S    13:01   0:00 [kthreadd]
root         3   0.0  0.0     0     0 ?        I    13:01   0:00 [kworker/0:0]
root         4   0.0  0.0     0     0 ?        I<   13:01   0:00 [kworker/0:0H]
root         5   0.0  0.0     0     0 ?        I    13:01   0:00 [kworker/u2:0]
root         6   0.0  0.0     0     0 ?        I<   13:01   0:00 [mm_percpu_wq]
root         7   0.1  0.0     0     0 ?        S    13:01   0:00 [ksoftirqd/0]
root         8   2.7  0.0     0     0 ?        S    13:01   0:01 [kdevtmpfs]
root         9   0.0  0.0     0     0 ?        I<   13:01   0:00 [netns]
root        10   0.0  0.0     0     0 ?        S    13:01   0:00 [oom_reaper]
root        11   0.0  0.0     0     0 ?        I<   13:01   0:00 [writeback]
root        12   0.0  0.0     0     0 ?        I<   13:01   0:00 [crypto]
root        13   0.0  0.0     0     0 ?        I<   13:01   0:00 [kblockd]
root        14   0.0  0.0     0     0 ?        S    13:01   0:00 [kswapd0]
root        15   0.0  0.0     0     0 ?        I    13:01   0:00 [kworker/0:1]
root        32   0.0  0.0     0     0 ?        S    13:01   0:00 [khvcd]
root        42   0.0  0.7   1944  1472 ?        Ss   13:02   0:00 dhcpcd
root        47   1.0  1.5   6204  2940 hvc0    Ss   13:02   0:00 sh -l
root        78   0.0  0.0     0     0 ?        I    13:02   0:00 [kworker/u2:1]
root       103   0.0  1.6   8944  3008 hvc0    R+   13:03   0:00 ps -aux
```

PROCESSOS - VISUALIZAÇÃO LINUX

PROCESSOS - WINDOWS

Gerenciador de Tarefas

Arquivo Opções Exibir

Processos Desempenho Histórico de aplicativos Inicializar Usuários Detalhes Serviços

Nome	PID	Status	Nome de usuário	CPU	Memória (co...	Virtualização d...
ApplicationFrameHos...	15992	Em execução	msi	00	996 K	Desabilitado
AppVShNotify.exe	4532	Em execução	SISTEMA	00	20 K	Não permitido
AppVShNotify.exe	9184	Em execução	msi	00	20 K	Desabilitado
audiodg.exe	18252	Em execução	SERVIÇO LOCAL	03	102.720 K	Não permitido
bdredline.exe	4576	Em execução	SISTEMA	00	20 K	Não permitido
Calculator.exe	13176	Suspenso	msi	00	0 K	Desabilitado
chrome.exe	13624	Em execução	msi	01	178.192 K	Desabilitado
chrome.exe	880	Em execução	msi	00	504 K	Desabilitado
chrome.exe	8804	Em execução	msi	00	29.096 K	Desabilitado
chrome.exe	8824	Em execução	msi	00	2.812 K	Desabilitado
chrome.exe	14768	Em execução	msi	00	257.808 K	Desabilitado
chrome.exe	16292	Em execução	msi	00	1.076 K	Desabilitado

ESTRUTURA DO PROCESSO

- Estrutura do processo



GERÊNCIA DE PROCESSOS

- A troca de um processo por outro no processador, efetuada pelo Sistema Operacional, é chamada de mudança de contexto;
- O programa executa dentro de um contexto de hardware, ciente das limitações de memória, cpu e disco;
- Cada processo tem seu usuário vinculado pelo Gerenciador de Processos;
 - A figura apresentada no canto superior direito mostra os três componentes principais de um processo.

PARA QUE SERVE A ESTRUTURA
DO PROCESSO?

PARA QUE SERVE A ESTRUTURA DO PROCESSO?

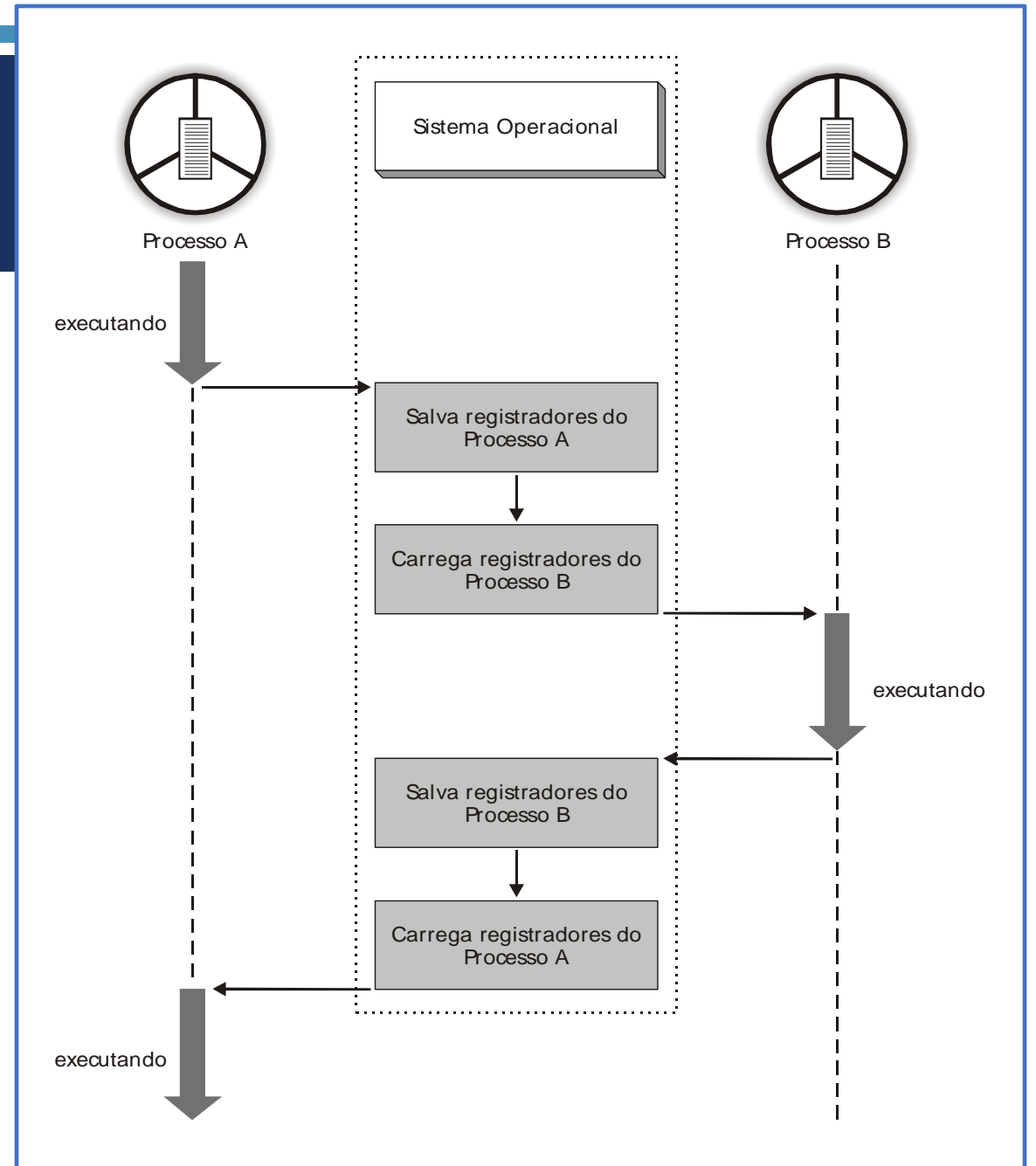
Quais as informações do contexto de software em um processo?



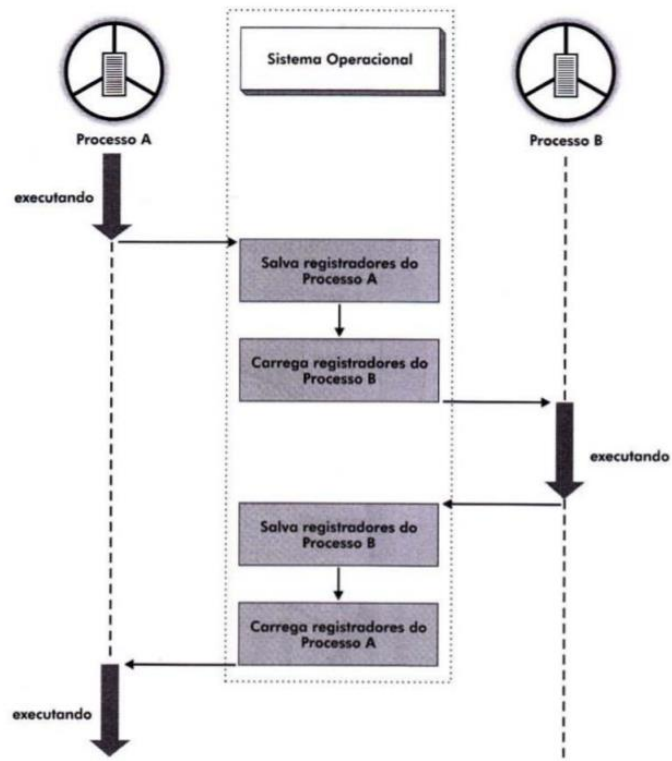
Como acontece a Mudança de Contexto?

CONTEXTO DE HARDWARE

- Mudança de contexto



CONTEXTO DE HARDWARE



- Armazena o conteúdo dos registradores gerais da UCP, além dos registradores de uso específico, como program counter (PC), stack pointer (SP) e registrador de status;
- A operação se resume em substituir o contexto de hardware de um processo pelo de outro, conforme mostra a figura ao lado.

CONTEXTO DE SOFTWARE - IDENTIFICAÇÃO

- Cada processo criado pelo sistema recebe uma identificação única (PID - process identification) representada por um número;
- Alguns sistemas, além do PID, identificam o processo através de um nome, além da identificação do usuário ou processo que o criou (owner);
- Cada usuário possui uma identificação única no sistema (UID - user identification);
- A UID permite implementar um modelo de segurança, onde apenas os objetos (processos, arquivos, áreas de memória etc.) que possuem a mesma UID do usuário (processo) podem ser acessados.

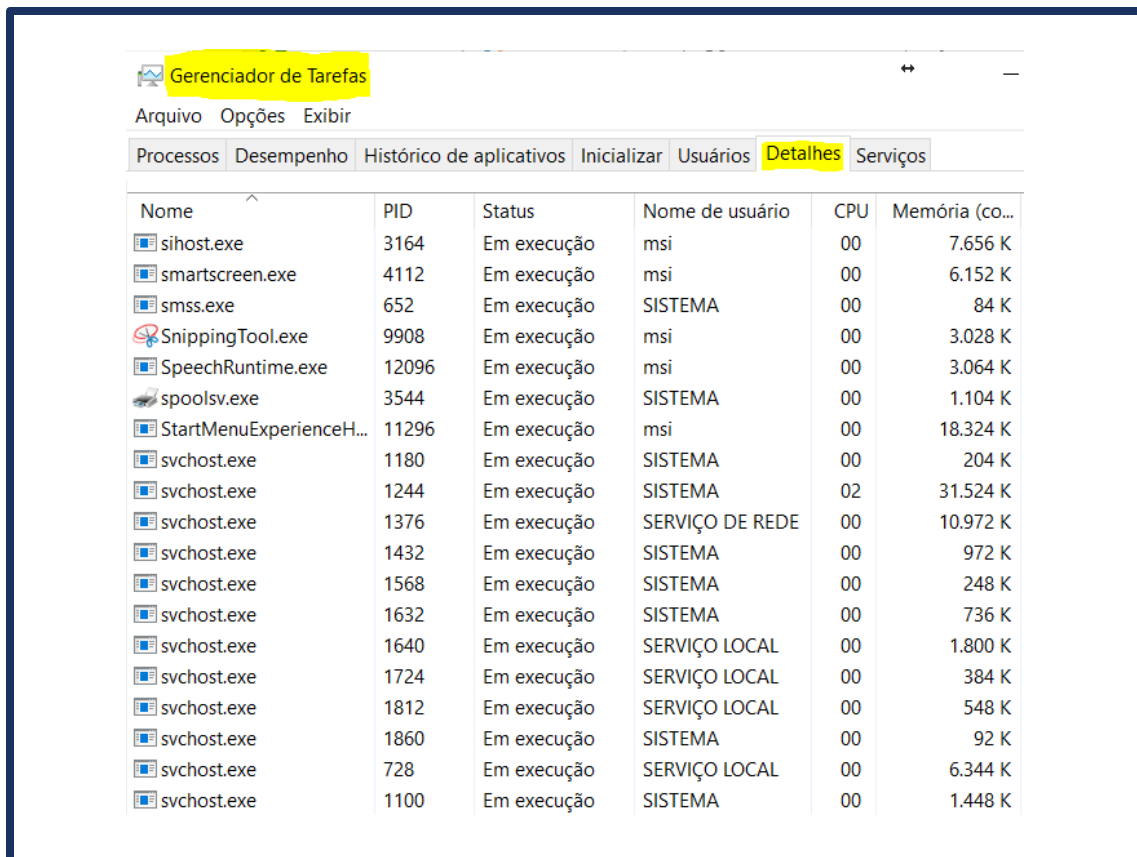
Contexto de Software

- × Nele é especificado os limites e características dos recursos que podem ser alocados pelo processo, como o número máximo de arquivos abertos simultaneamente, prioridade de execução e tamanho do buffer para operações de E/S;
- × Muitos dos recursos são definidos na criação e outros podem ser ajustados/requeridos durante a execução;
- × A maior parte das informações do contexto de software do processo provém de um arquivo do sistema operacional, conhecido como arquivo de usuários;
- × **O contexto de software é composto por três grupos de informações sobre o processo: identificação, quotas e privilégios.**



**Questão
boa para
uma
prova!**

CONTEXTO DE SOFTWARE - IDENTIFICAÇÃO



Nome	PID	Status	Nome de usuário	CPU	Memória (co...
sihost.exe	3164	Em execução	msi	00	7.656 K
smartscreen.exe	4112	Em execução	msi	00	6.152 K
smss.exe	652	Em execução	SISTEMA	00	84 K
SnippingTool.exe	9908	Em execução	msi	00	3.028 K
SpeechRuntime.exe	12096	Em execução	msi	00	3.064 K
spoolsv.exe	3544	Em execução	SISTEMA	00	1.104 K
StartMenuExperienceH...	11296	Em execução	msi	00	18.324 K
svchost.exe	1180	Em execução	SISTEMA	00	204 K
svchost.exe	1244	Em execução	SISTEMA	02	31.524 K
svchost.exe	1376	Em execução	SERVIÇO DE REDE	00	10.972 K
svchost.exe	1432	Em execução	SISTEMA	00	972 K
svchost.exe	1568	Em execução	SISTEMA	00	248 K
svchost.exe	1632	Em execução	SISTEMA	00	736 K
svchost.exe	1640	Em execução	SERVIÇO LOCAL	00	1.800 K
svchost.exe	1724	Em execução	SERVIÇO LOCAL	00	384 K
svchost.exe	1812	Em execução	SERVIÇO LOCAL	00	548 K
svchost.exe	1860	Em execução	SISTEMA	00	92 K
svchost.exe	728	Em execução	SERVIÇO LOCAL	00	6.344 K
svchost.exe	1100	Em execução	SISTEMA	00	1.448 K

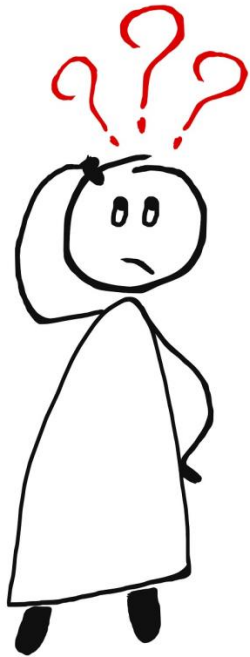
■ Atentar para as informações principais do Gerenciador de Tarefas do Windows (aba detalhes):

- Nome do processo;
- PID
- Status
- Nome do Usuário
- Consumo de CPU/Mem

CONTEXTO DE SOFTWARE - QUOTAS

- As quotas são os limites de cada recurso do sistema que um processo pode alocar;
- Caso uma quota seja insuficiente, o processo poderá ser executado lentamente, interrompido durante seu processamento ou mesmo não ser executado;
- Alguns exemplos de quotas presentes na maioria dos sistemas operacionais são:
 - número máximo de arquivos abertos simultaneamente;
 - tamanho máximo de memória principal e secundária que o processo pode alocar;
 - número máximo de operações de E/S pendentes;
 - tamanho máximo do buffer para operações de E/S;
 - número máximo de processos, subprocessos e threads que podem ser criados.

UMA DÚVIDA...



- As quotas estão ligadas exclusivamente aos recursos de hardware?
- Exemplos:
 - Um programa pode não executar pelo seu tamanho (muito grande para a memória disponível)
 - Um programa pode não executar por limitações específicas de um usuário

CONTEXTO DE SOFTWARE - PRIVILÉGIOS

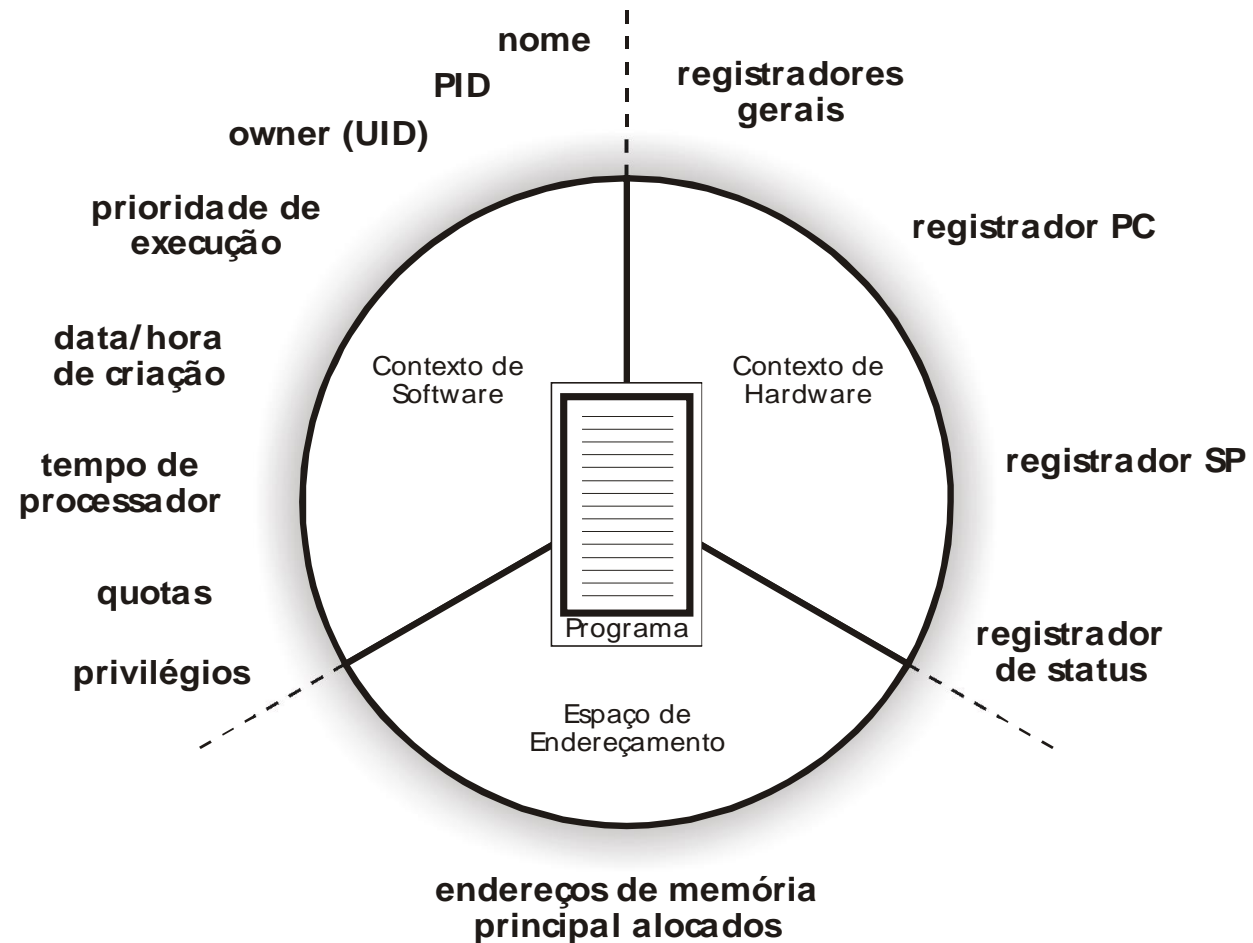
- Os privilégios ou direitos definem as ações que um processo pode fazer em relação a ele mesmo, aos demais processos e ao sistema operacional;
- Privilégios que afetam o sistema são os mais amplos e poderosos, pois estão relacionados à operação e à gerência do ambiente;
- A maioria dos SOs disponibiliza uma conta de acesso com todos estes privilégios disponíveis, com o propósito de o administrador gerenciar o sistema operacional. No sistema Unix existe a conta "root", no Windows a conta "administrator" e no OpenVMS existe a conta "system" com este mesmo perfil.

ESPAÇO DE ENDEREÇAMENTO

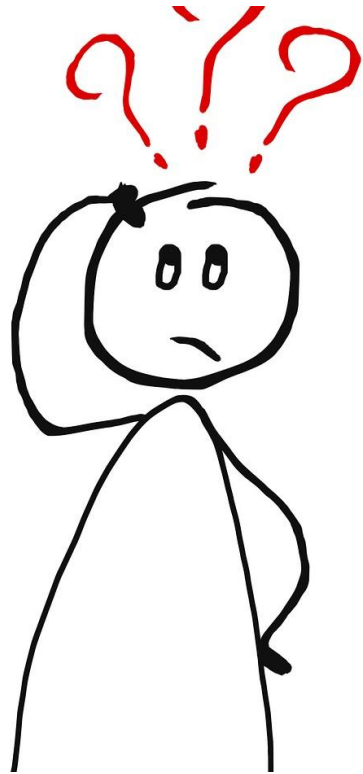
-
- O espaço de endereçamento é a área de memória pertencente ao processo onde instruções e dados do programa são armazenados para execução;
 - Cada processo possui seu próprio espaço de endereçamento, que deve ser devidamente protegido do acesso dos demais processos;
 - Diversos mecanismos de implementação e administração do espaço de endereçamento podem ser executados (item será discutido mais à frente).

ESPAÇO DE ENDEREÇAMENTO

- Estrutura de um processo



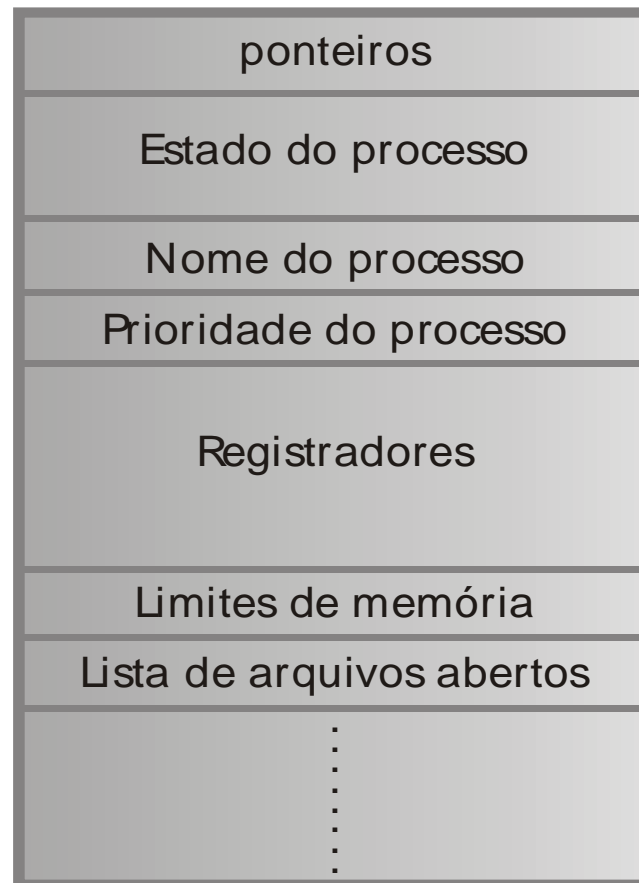
UMA DÚVIDA...



- Qual estrutura é utilizada pelo Sistema Operacional para a implementação do processo?
- Quais os estados que podem assumir um processo?

BLOCO DE CONTROLE DO PROCESSO

- PCB



BLOCO DE CONTROLE DE PROCESSOS

PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	C
1	0	0	75	0	-	378	schedu	?	00:00:04	i
2	1	0	75	0	-	0	contex	?	00:00:00	}
3	1	0	94	19	-	0	ksofti	?	00:00:00	}
6	1	0	85	0	-	0	bdfly	?	00:00:00	l
4	1	0	75	0	-	0	schedu	?	00:05:35	}
5	1	0	75	0	-	0	schedu	?	00:03:45	}
7	1	0	75	0	-	0	schedu	?	00:00:00	}
8	1	0	85	0	-	0	md_thr	?	00:00:00	n
21	1	0	75	0	-	0	end	?	00:05:40	}
253	1	0	75	0	-	0	end	?	00:00:00	}
254	1	0	75	0	-	0	end	?	00:00:00	}
255	1	0	75	0	-	0	end	?	00:55:28	}
579	1	0	75	0	-	399	schedu	?	00:02:00	s
583	1	0	75	0	-	383	do_sys	?	00:00:00	}
600	1	0	75	0	-	414	schedu	?	00:00:00	l
619	1	0	85	0	-	416	schedu	?	00:00:00	l
631	1	0	75	0	-	393	schedu	?	00:00:00	n
702	1	0	75	0	-	917	schedu	?	00:00:30	s
716	1	0	75	0	-	539	schedu	?	00:00:00	}
745	1	0	75	0	-	398	schedu	?	00:00:00	q
765	1	0	75	0	-	607	schedu	?	00:00:16	c

- Visão dos processos em execução agora a partir de um sistema Linux;
- As informações de identificação são semelhantes ao apresentados no Sistema Windows;
- Atentar aos identificadores PRI e TIME, que indicam respectivamente a prioridade do processo e o tempo de utilização do processador.

ESTADOS DO PROCESSO



EXECUÇÃO
(RUNNING)



PRONTO (READY)

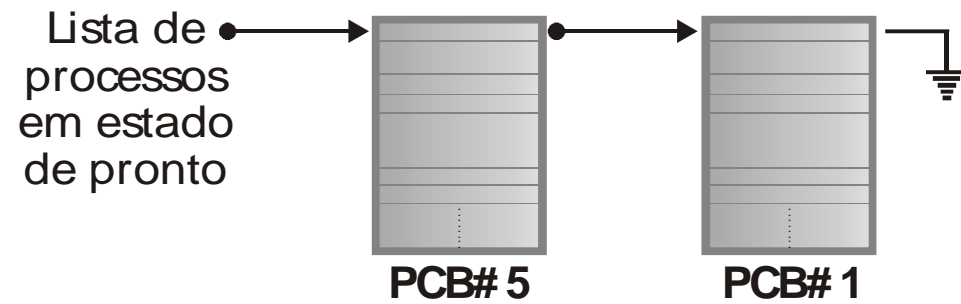


ESPERA (WAIT)

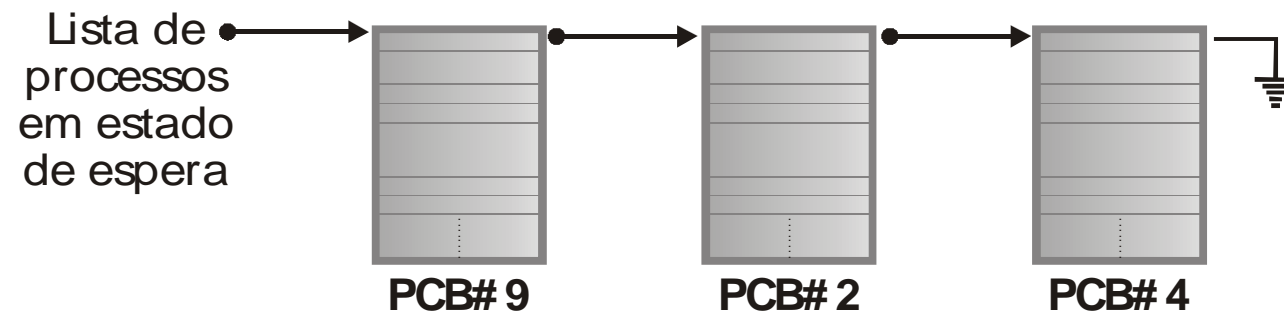
ESTADOS DOS PROCESSOS

- Em um Sistema Multiprogramável, para evitar o uso exclusivo do processador, os processos passam por diferentes estados ao longo do processamento:
 - EXECUÇÃO (running) - É dito como em execução quando está sendo processado pela UCP. Se usando uma máquina monoprocessada, apenas um processo por vez é tratado;
 - PRONTO (ready) - Os processos neste estado aguardam apenas pela alocação do sistema operacional para entrar em execução, em um mecanismo conhecido como “escalonamento”;
 - ESPERA (wait) - Quando um processo aguarda, por exemplo, por um recurso de E/S, é colocado no estado de espera. Um evento externo fará a comunicação de que o recurso solicitado foi liberado para seguir o processamento.

ESTADOS DO PROCESSO

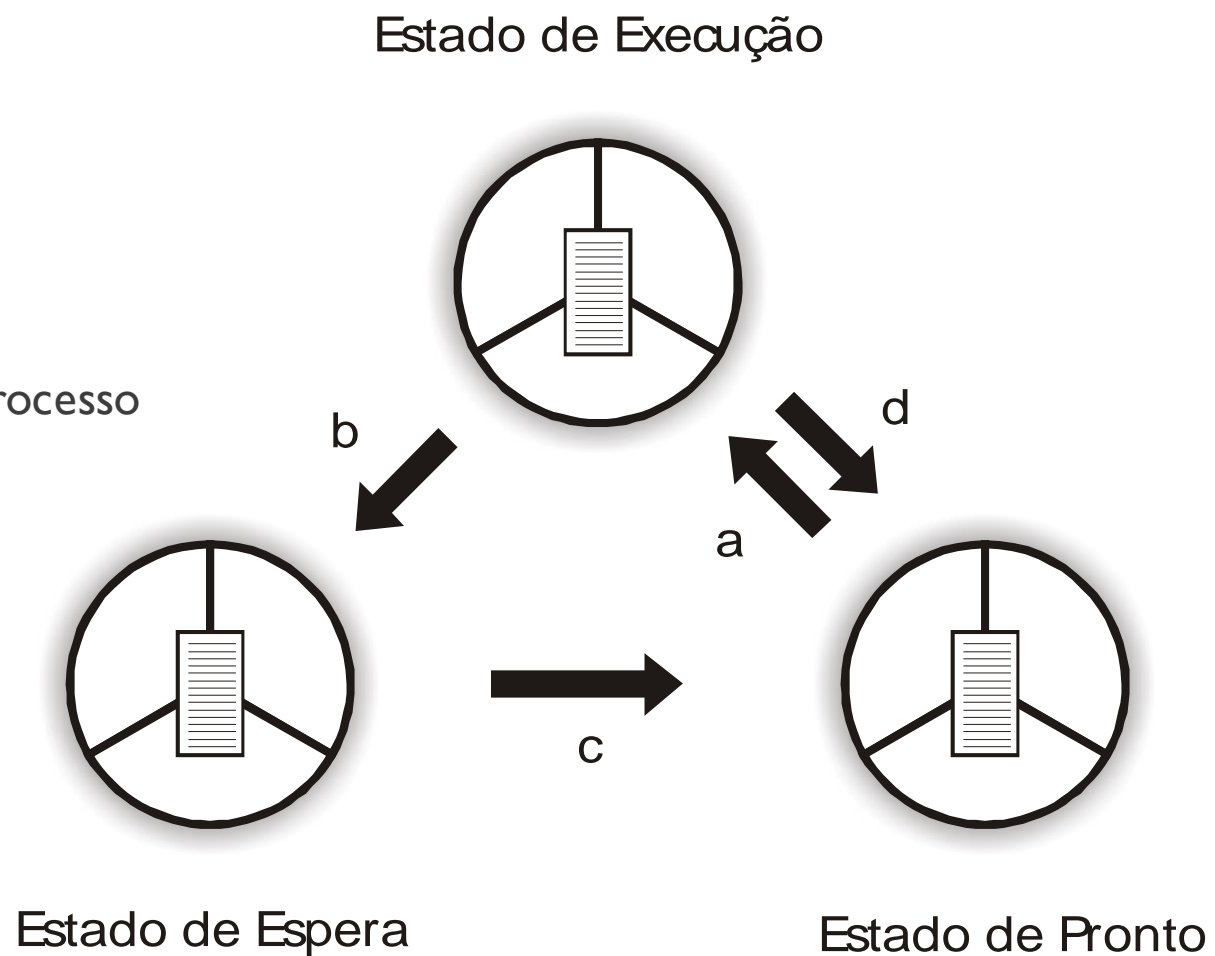


- Lista de PCBs

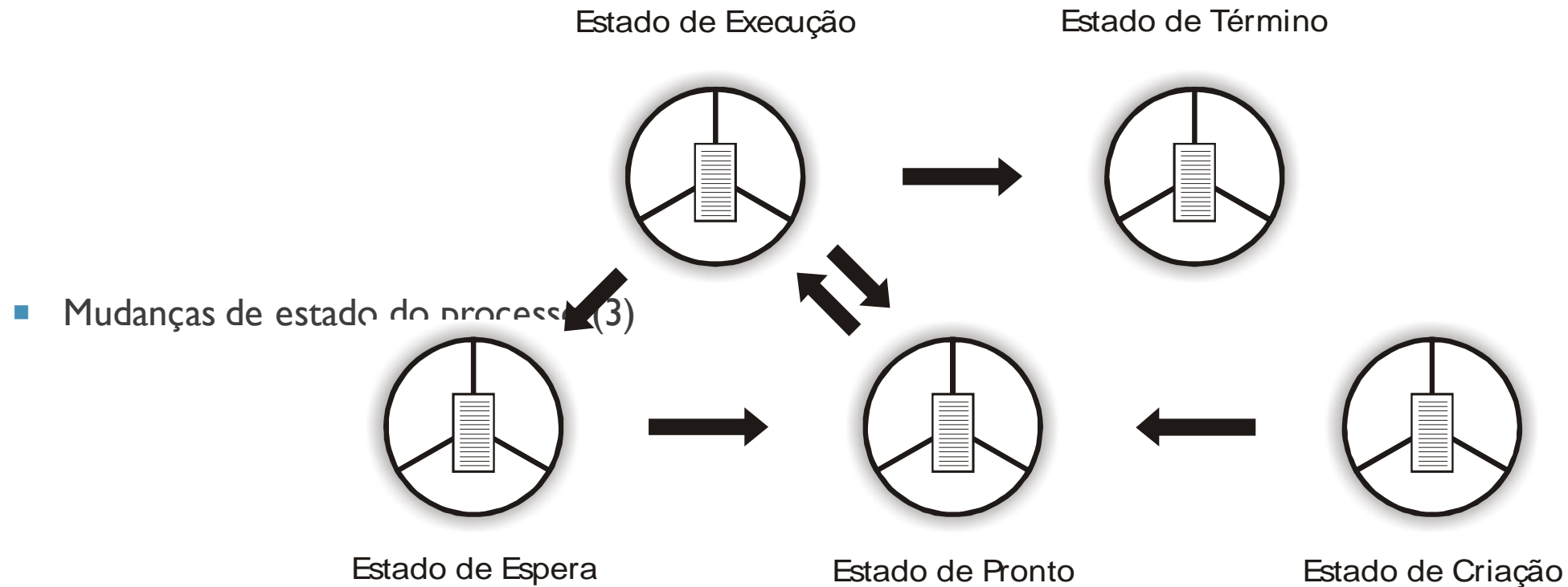


MUDANÇAS DE ESTADO

- Mudanças de estado do processo

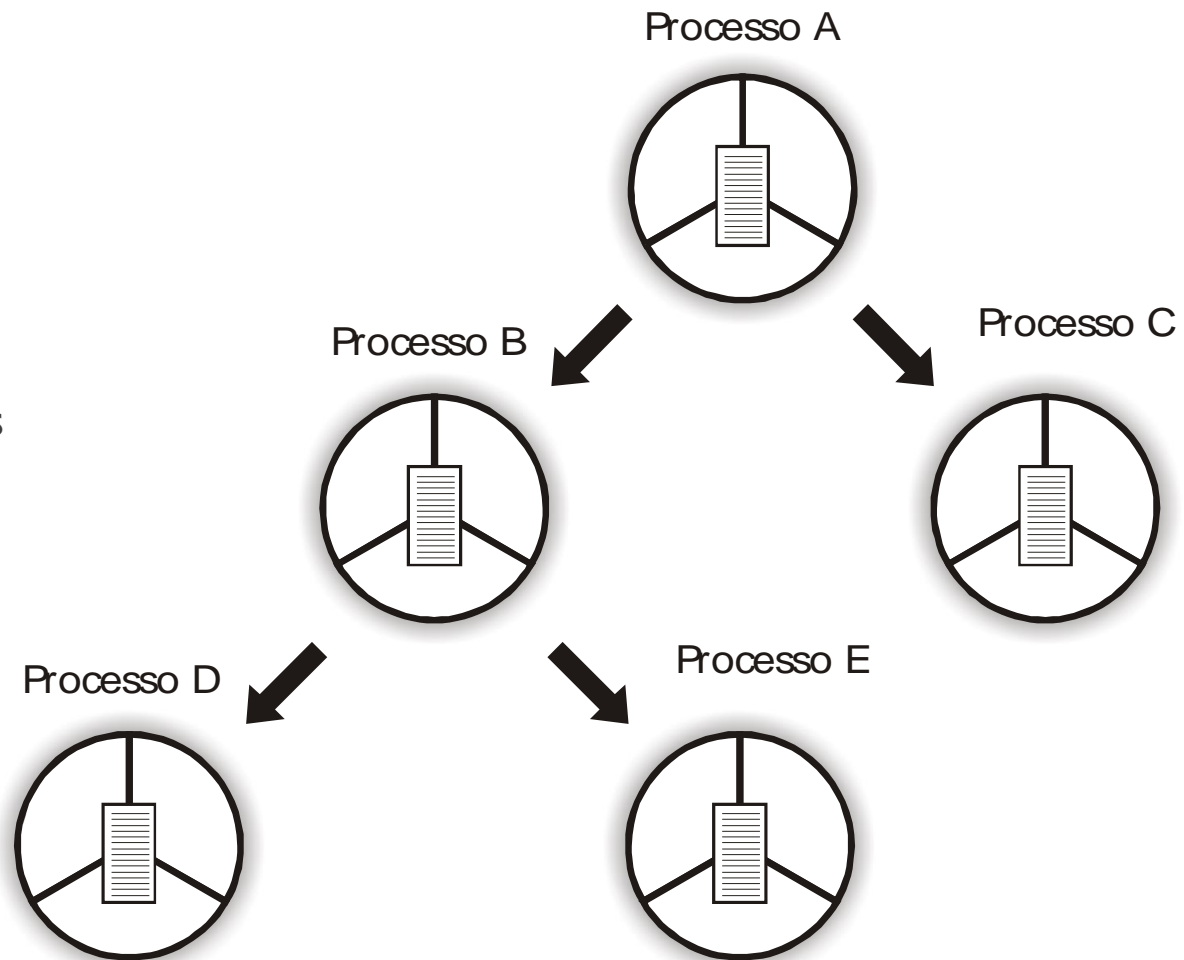


CRIAÇÃO E ELIMINAÇÃO

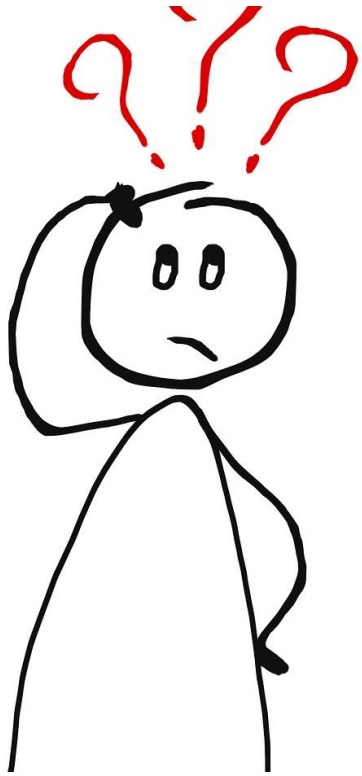


PROCESSOS INDEPENDENTES, SUBPROCESSOS E THREADS

- Estrutura de processos e subprocessos

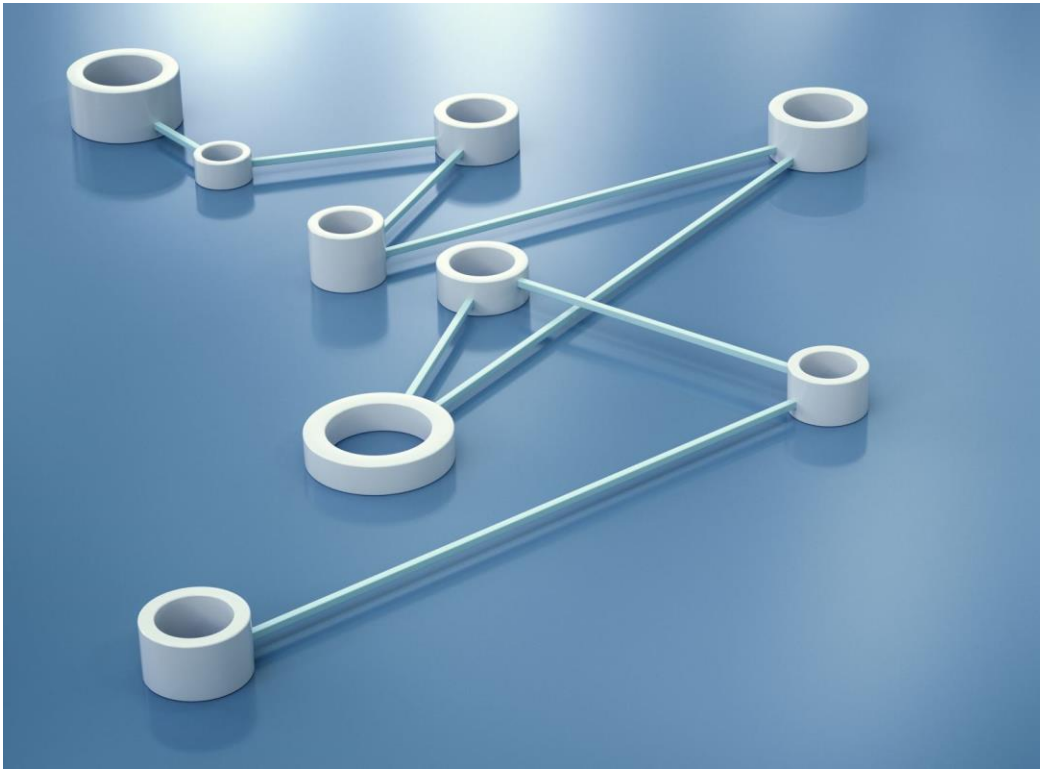


UMA DÚVIDA...



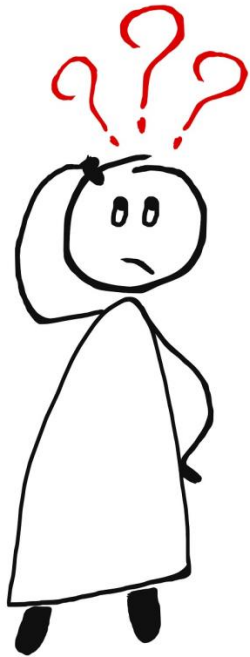
Respondendo pelo Chat, que tipo de situação pode fazer crescer o encadeamento de uma lista de processos em “espera”?

O QUE FICOU NA MENTE?



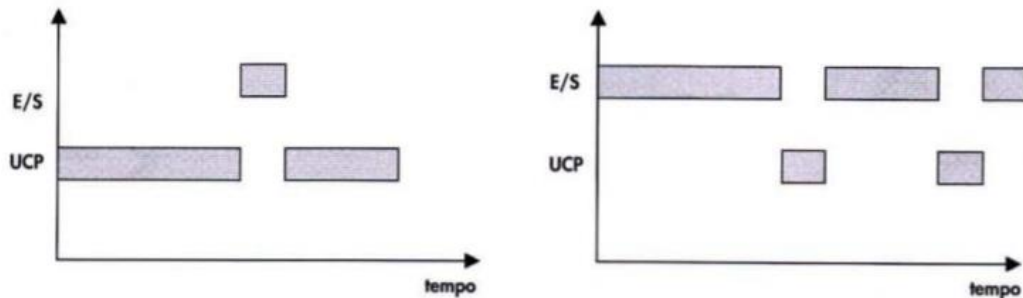
1. Defina o conceito de processo.
2. Por que o conceito de processo é tão importante no projeto de sistemas multiprogramáveis?
3. É possível que um programa execute no contexto de um processo e não execute no contexto de um outro? Por quê?
4. Quais partes compõem um processo?

O QUE FICOU NA MENTE?



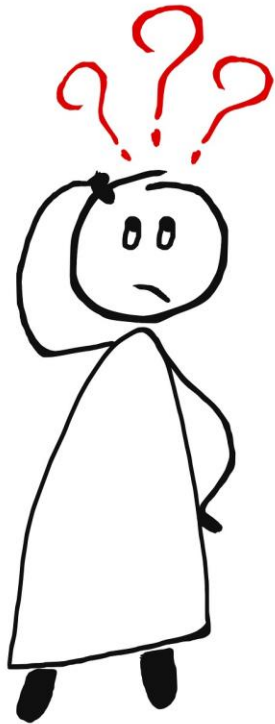
5. O que é o contexto de hardware de um processo e como é a implementação da troca de contexto?
6. Qual a função do contexto de software? Exemplifique cada grupo de informação.
7. O que é o espaço de endereçamento de um processo?
8. Como o sistema operacional implementa o conceito de processo? Qual a estrutura de dados indicada para organizar os diversos processos na memória principal?

PROCESSOS I/O BOUND E CPU BOUND



- Classificação usada para terminar os processos que consomem mais recursos de CPU ou I/O (E/S);
- A identificação desse comportamento se dá pelo tempo em que processo passa no estado de “execução” ou “espera”;

UMA DÚVIDA...



Os programas muito interativos, que frequentemente carecem de respostas dos usuários para seguir execução, poderiam ser considerados I/O Bound?

PROCESSOS FOREGROUND E BACKGROUND



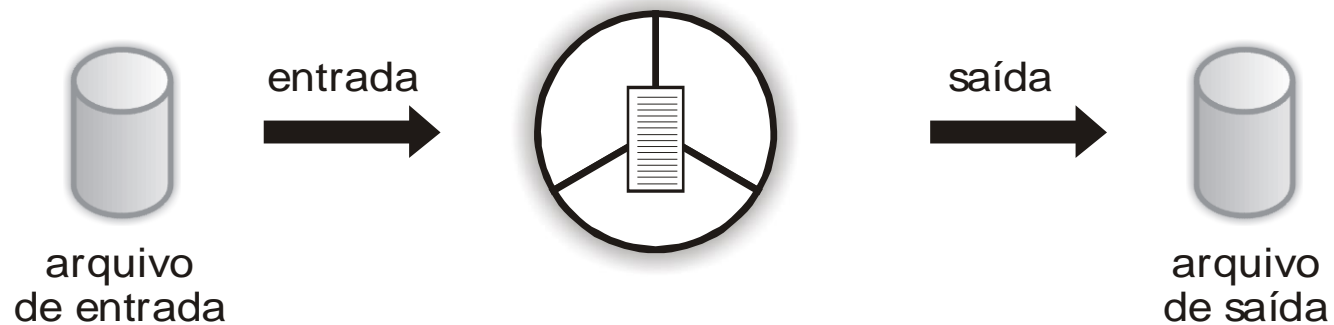
- Um processo possui sempre pelo menos dois canais de comunicação associados a sua estrutura, sendo entrada (Input) ou saída (Output);
- Um processo foreground é aquele que permite a comunicação direta do usuário com o processo durante o seu processamento;
- Um processo background é aquele onde não existe a comunicação com o usuário durante o seu processamento.

PROCESSOS FOREGROUND E BACKGROUND

(a) Processo Foreground

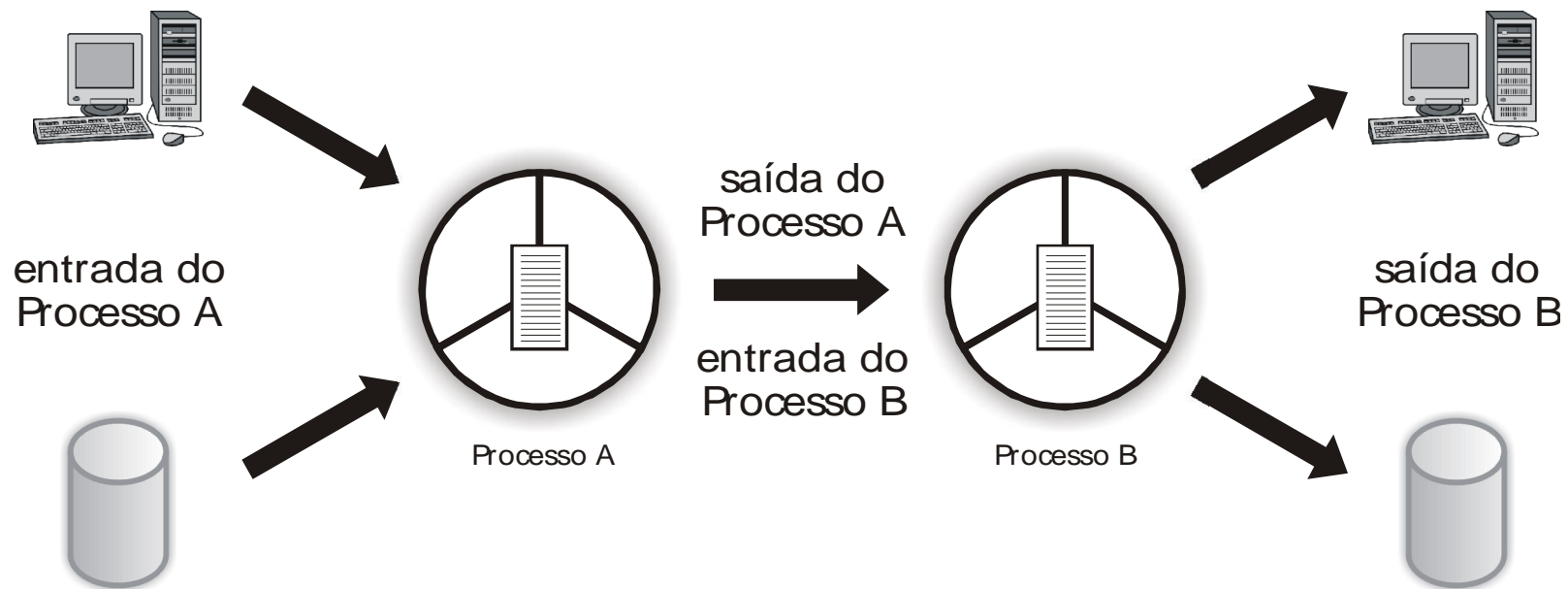


(b) Processo Background



PROCESSO FOREGROUND E BACKGROUND

- Pipe



PROCESSOS DO SISTEMA

- Auditoria e segurança
- Serviços de rede
- Contabilização do uso de recursos
- Contabilização de erros
- Gerência de impressão
- Gerência de jobs batch
- Temporização
- Comunicação de eventos
- Interface de comandos (shell)

PROCESSOS INDEPENDENTES

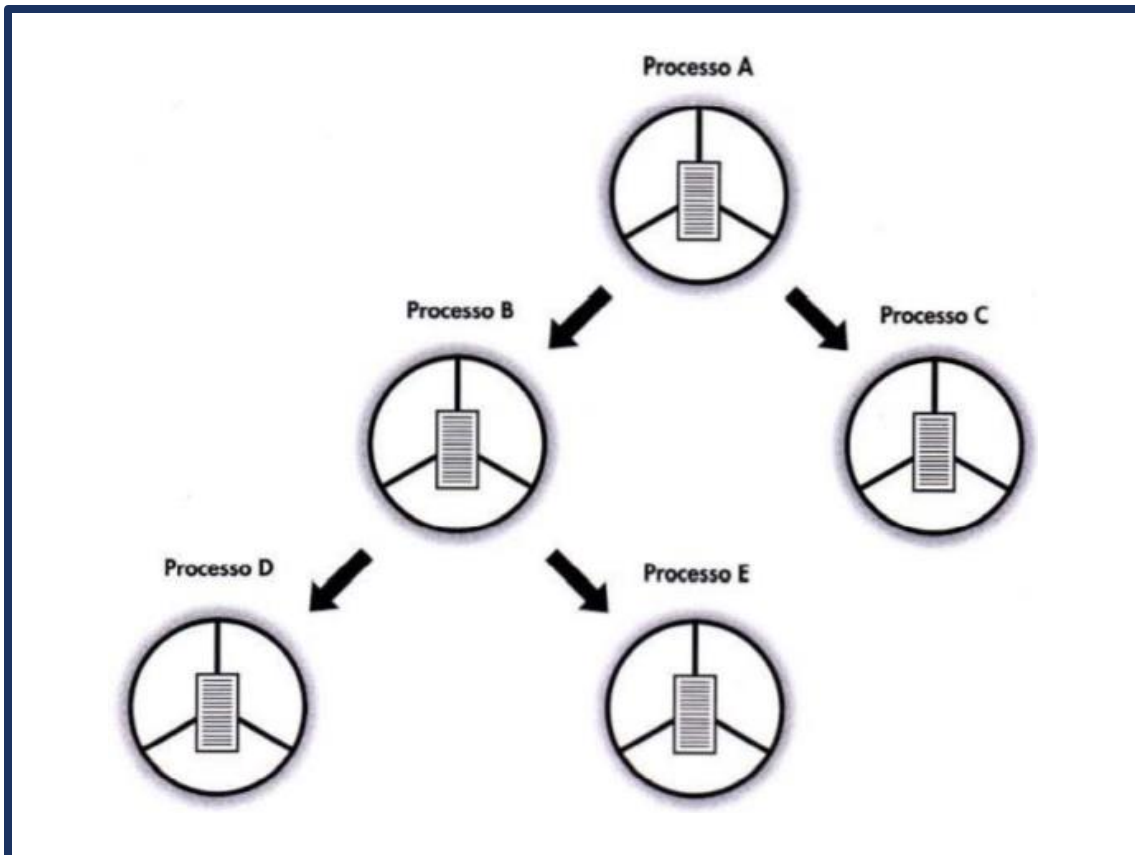
Forma mais simples de implementar a concorrência em ambientes multiprogramáveis;

Não existe a necessidade de vincular o processo criado ao seu criador;

Apresenta uma estrutura completa de processo, tendo contexto de software, hardware e espaço de endereçamento;

Nesse sentido, exige a alocação de um PCB.

SUBPROCESSOS



- São processos criados dentro de uma estrutura hierárquica;
- O processo criador é denominado processo-pai, enquanto o novo processo é chamado de subprocesso ou processo-filho;
- Dependência entre o processo criador e o subprocesso;
- Caso um processo-pai deixe de existir, os subprocessos subordinados são automaticamente eliminados;
- Os subprocessos possuem estrutura completa na PCB para identificá-los;
- Existe o compartilhamento de quotas com o processo-pai;

Threads



Surgiu pela necessidade de otimizar os recursos exigidos pelo SO para criação de processos;



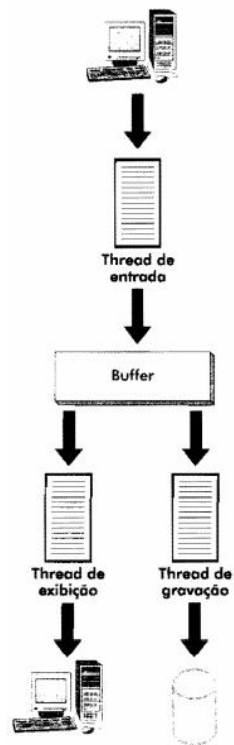
Compartilha o Contexto de Software e Espaço de Endereçamento;



Threads compartilham o processador da mesma maneira que um processo, ou seja, enquanto uma thread espera por uma operação de E/S, outra thread pode ser executada.

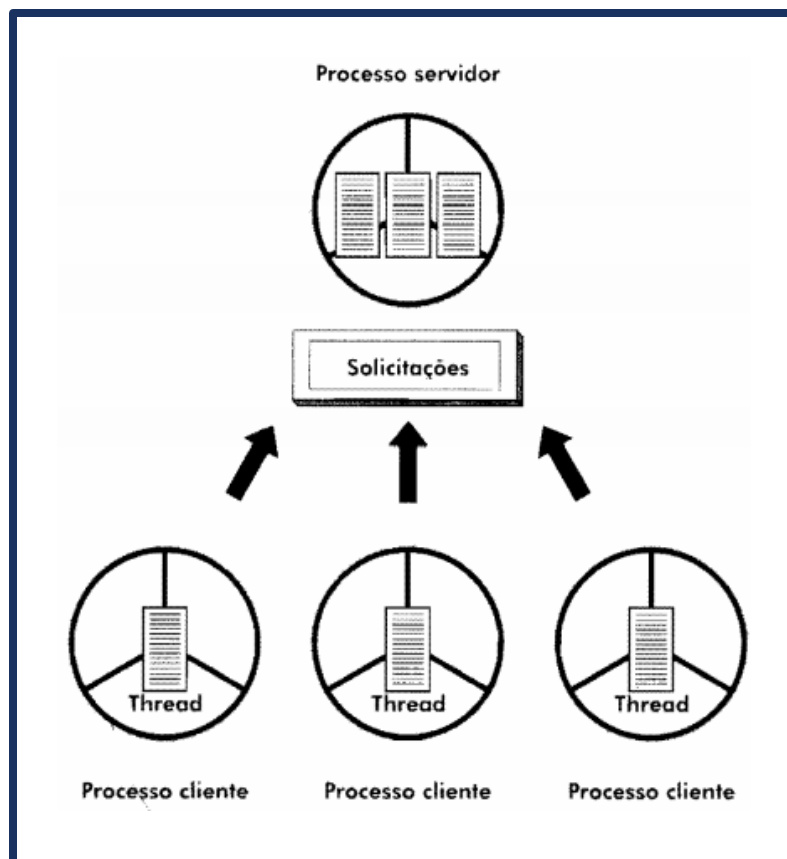


AMBIENTE MULTITHREAD



- Em algumas aplicações, a utilização de threads pode melhorar o desempenho apenas executando tarefas em background enquanto operações E/S estão sendo processadas;
- Editores de texto, planilhas, aplicativos gráficos e processadores de imagens são especialmente beneficiados quando desenvolvidos com base em threads.

AMBIENTE MULTITHREAD E AMBIENTE CLIENTE/SERVIDOR



- Threads são essenciais para solicitações de serviços remotos;
- Permite solicitar o serviço remoto, enquanto a aplicação pode continuar realizando outras atividades;
- Em um ambiente multithread, se uma aplicação solicita um serviço remoto, ela pode ficar esperando indefinidamente, enquanto aguarda pelo resultado.

DAEMONS/SERVIÇOS

- Em sistemas operacionais multitarefa, um **daemons** são programas que ficam rodando indefinidamente, sozinhos, e em background no computador. Eles não ficam interagindo diretamente com o usuário.
 - Por exemplo, no Linux, podemos citar o **daemon** SSH, que fica rodando no sistema aguardando um usuário solicitar uma sessão SSH (para logar remotamente), e o Apache, que fica rodando sem parar aguardando requisições web.
- Outro **daemon** super importante é o init, o processo de PID 1 no Linux. Ele é um daemon tão importante que é o primeiro daemon que começa a rodar em um ambiente Unix like e, se ele parar, o sistema também para.
- A maioria dos daemons são iniciados na hora do boot do sistema e finalizados no momento que o sistema desliga.

DAEMONS/SERVIÇOS FUNCIONAMENTO

- Um daemon é desenvolvido para nunca parar: ele até pode ter uma fase de inicialização e uma fase de finalização, mas entre essas fases ele permanece em um estado onde executa uma ação continuamente. Chamamos esse estado de loop principal (ou main loop).
- Por rodar em background e não poder interagir com o usuário de forma convencional (interface gráfica ou console), um daemon usa mecanismos alternativos para se comunicar com o usuário.
 - Por exemplo, ele podem ler parâmetros de arquivos de configuração, responder a pacotes que recebe via rede, informar ao administrador o que está ocorrendo escrevendo informações em logs, ou realizam ações ao receber sinais.
- Pode ser serviços: serviços de rede, servidor web, servidor de e-mail.

O QUE FICOU NA MENTE?

1. Como uma aplicação pode implementar concorrência em um ambiente monothread?
2. Quais os problemas de aplicações concorrentes desenvolvidas em ambientes monothread?
3. O que é um ambiente multithread e quais as vantagens de sua utilização?
4. Qual estrutura de dados organiza as informações das threads criadas e o que ela armazena?

SINAIS



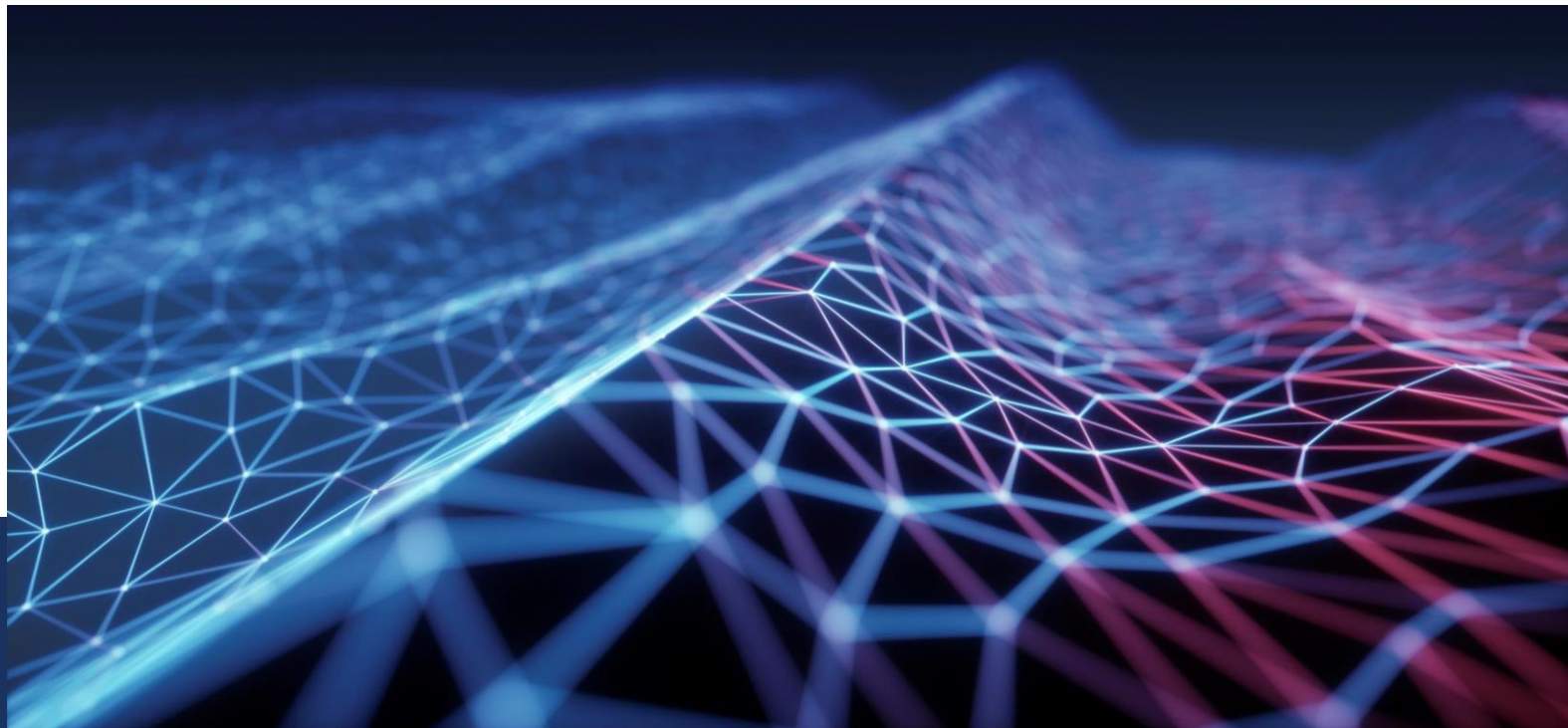
- Mecanismo que permite notificar processos sobre eventos gerados pelo sistema operacional ou por outros processos;
- O uso de sinais é fundamental para a gerência de processos, além de possibilitar a comunicação e sincronização entre os mesmos;
- Observar a sinalização abaixo das teclas CTRL+C.

O QUE FICOU NA MENTE?

-
1. Defina os cinco estados possíveis de um processo.
 2. Dê um exemplo que apresente todas as mudanças de estado de um processo, juntamente com o evento associado a cada mudança.
 3. Diferencie processos multithreads e subprocessos.
 4. Explique a diferença entre processos foreground e background.

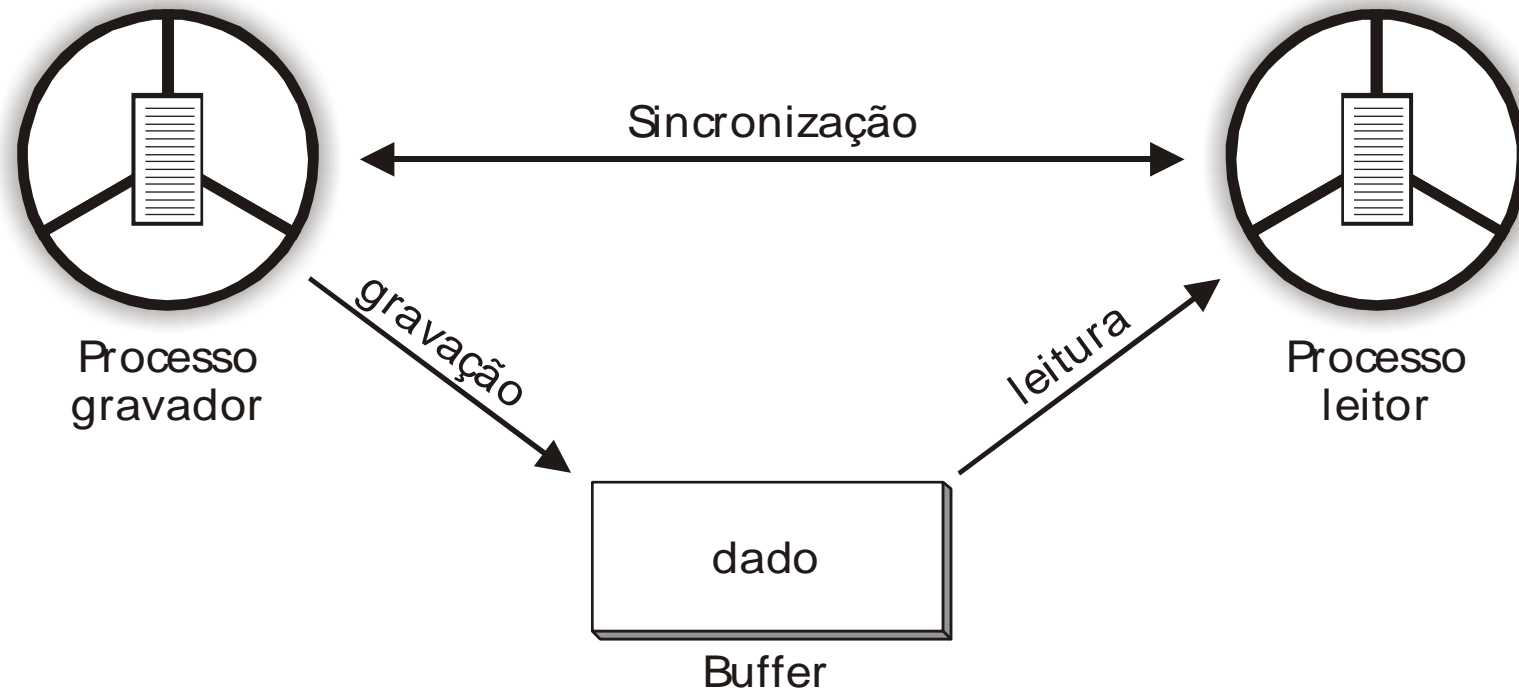
O QUE FICOU NA MENTE?

-
5. Qual a relação entre processo e arquitetura do Sistema Operacional que implementa serviços (microkernel)?
 6. Dê exemplos de aplicações CPU-bound e I/O-bound.
 7. Explique como a eliminação de um processo utiliza o mecanismo de sinais.
 8. Quais as características de um processo independente?



APLICAÇÕES CONCORRENTES

APLICAÇÕES CONCORRENTES



Garantem a comunicação entre processos em recursos compartilhados



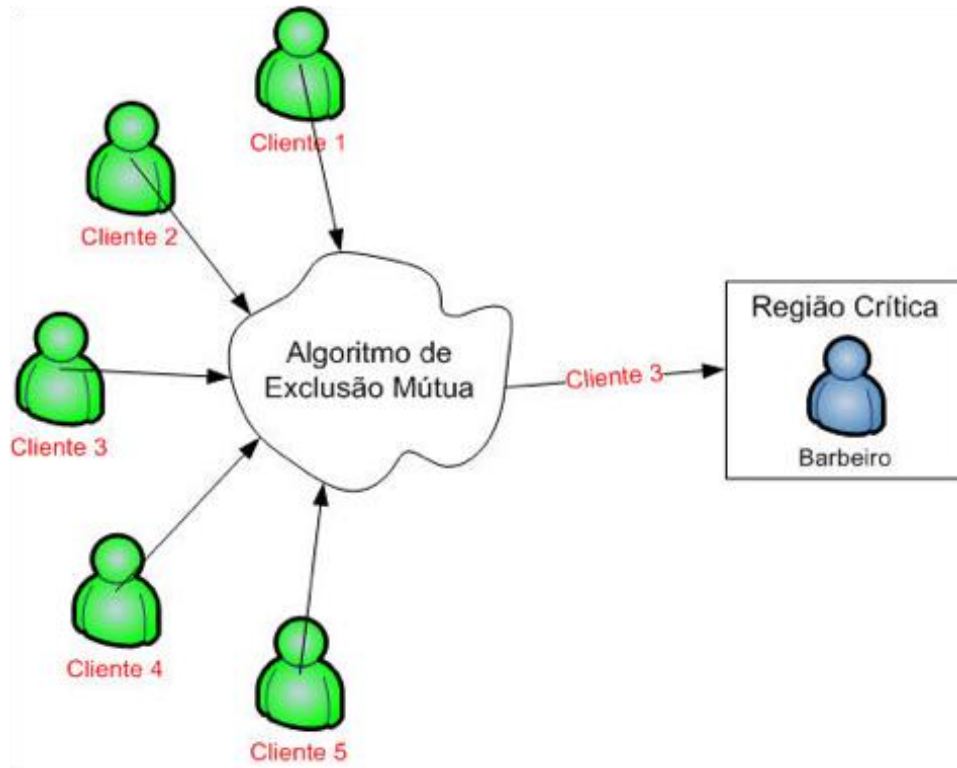
Pergunta-se??

Que tipos de problemas podem ocorrer no compartilhamento de recursos?

MECANISMOS DE
SINCRONIZAÇÃO

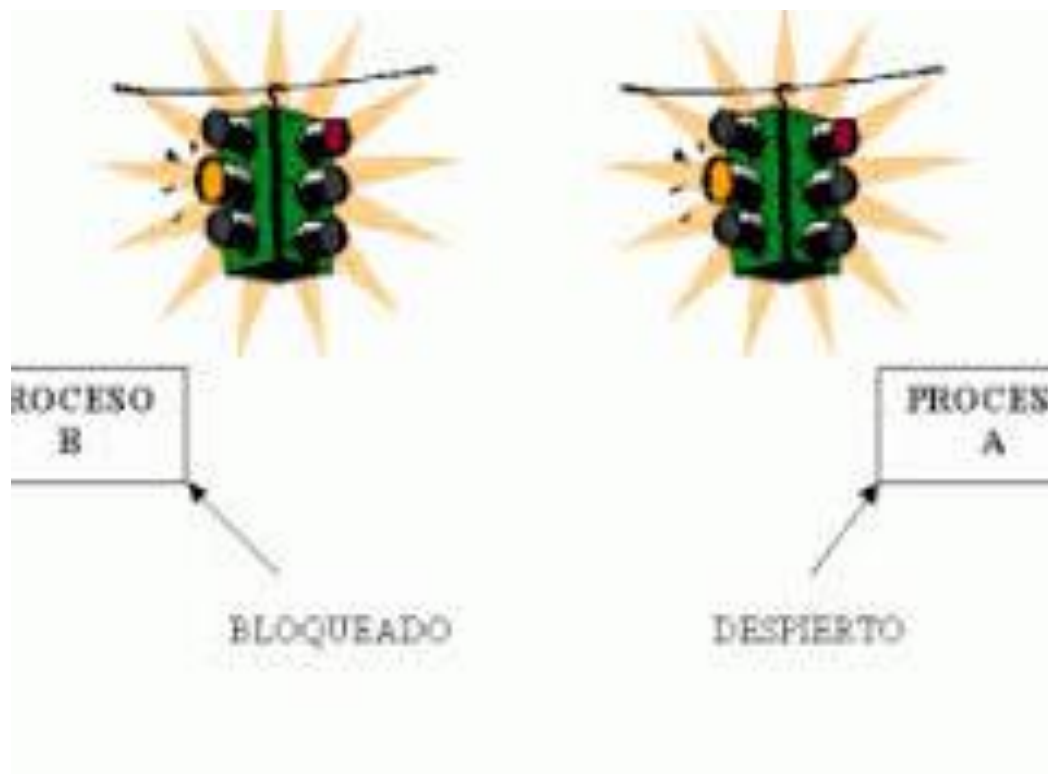
[HTTPS://WWW.MENTI.COM/](https://www.menti.com/)

EXCLUSÃO MÚTUA



- Apenas um processo acessa o recurso
- Os demais processos ficam em espera (exclusão)
- Região crítica :
 - parte do código onde existe o acesso compartilhado;
 - Exige um protocolo de entrada e saída da RC;
- Starvation – o processo nunca consegue acessar a Região Crítica.

SEMÁFOROS

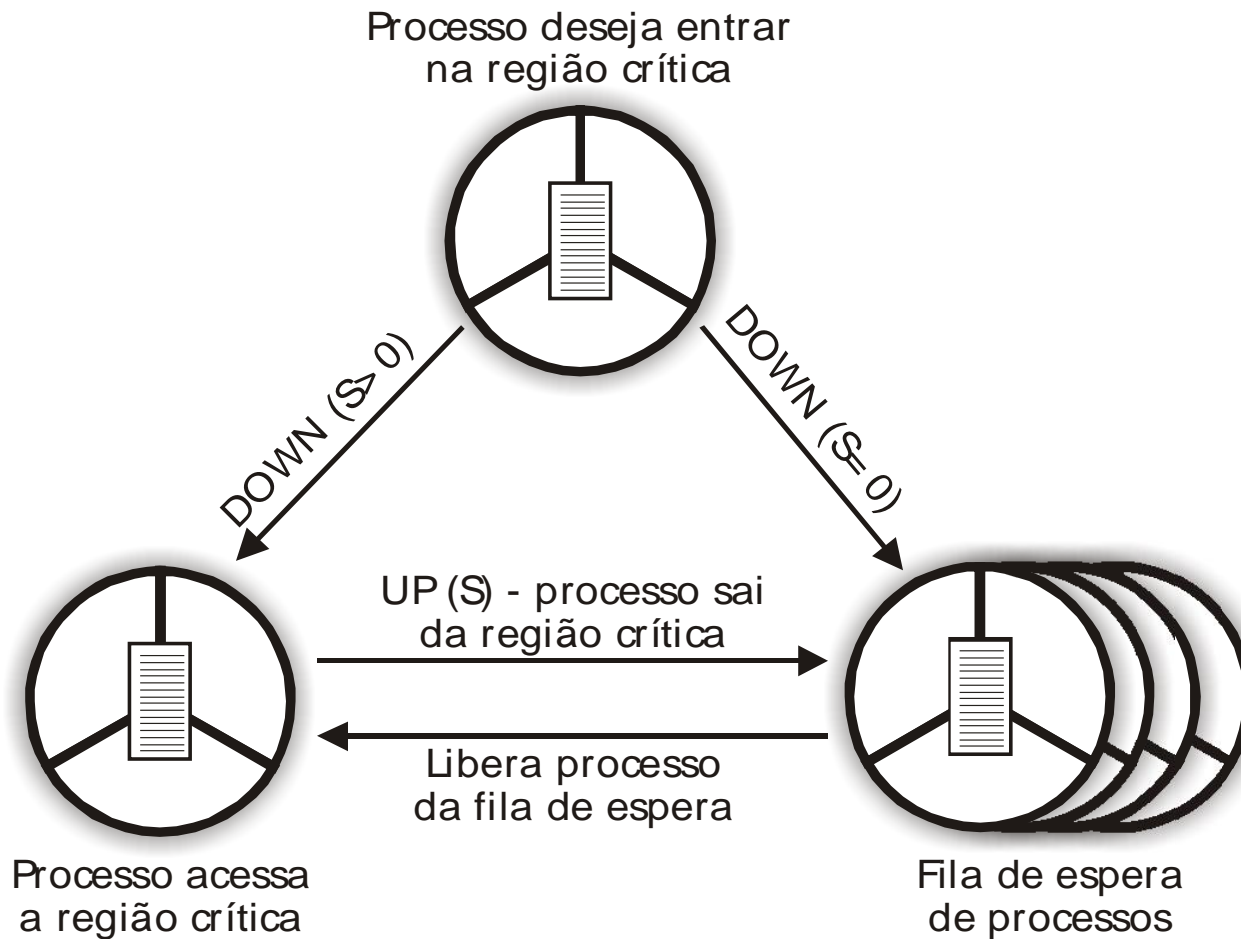


- Proposto por Dijkstra em 1965;
- É uma variável inteira, não negativa, que só pode ser manipulada por duas instruções:
 - DOWN (decrementa) e;
 - UP (incrementa);

FUNCIONAMENTO DO SEMÁFORO BINÁRIO

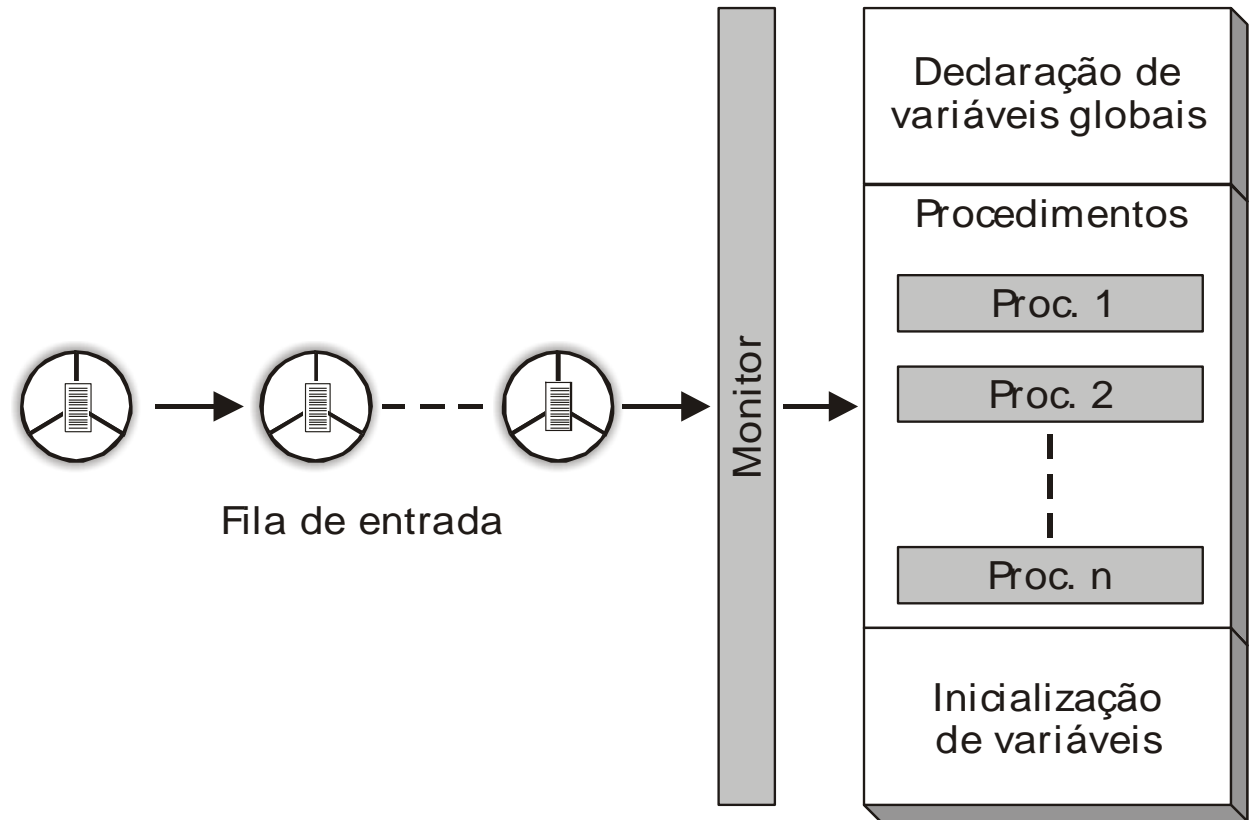
- Sempre que deseja entrar na região crítica, um processo executa uma instrução DOWN.
- Se o semáforo for igual a 1, o valor é decrementado e o processo pode acessar a região crítica.
- Se o valor for igual a 0, o processo fica impedido do acesso, permanecendo em estado de espera.

SEMÁFOROS



MONITORES

- Somente um processo deve estar ativo dentro do Monitor.





- É formado por um conjunto de procedimentos encapsulados dentro de um módulo.
- Implementação automática da exclusão mútua entre os procedimentos.
- Somente um processo pode estar executando um dos procedimentos do monitor em um determinado instante.

FUNCIONAMENTO DO MONITOR

Toda vez que algum processo faz uma chamada a um procedimento do monitor, ele verifica está sendo utilizado.

Caso positivo, o processo fica aguardando na fila de entrada.

As variáveis globais de um monitor são visíveis apenas aos procedimentos de sua estrutura, sendo invisíveis fora do contexto do monitor.



Mecanismo de comunicação e sincronização;

fx

Não utiliza variáveis compartilhadas;



Deve existir um canal de comunicação – um buffer ou link de uma rede de computadores;



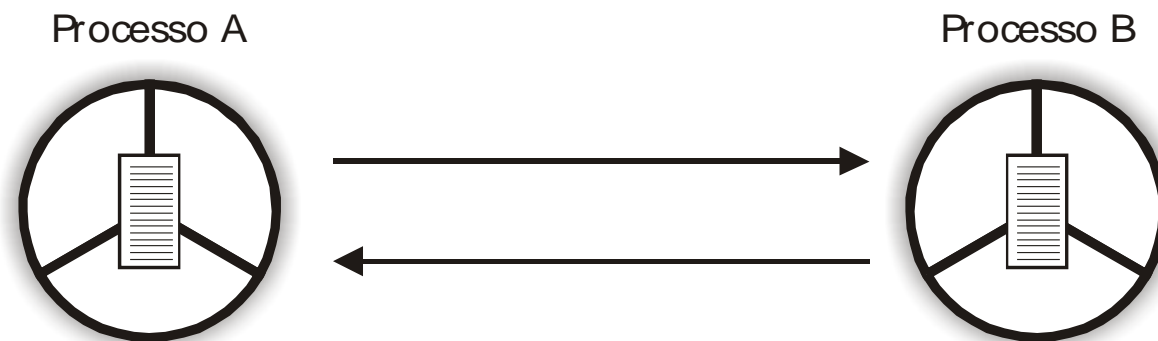
Pode-se utilizar as rotinas **send** e **receive**.

TROCA DE MENSAGENS

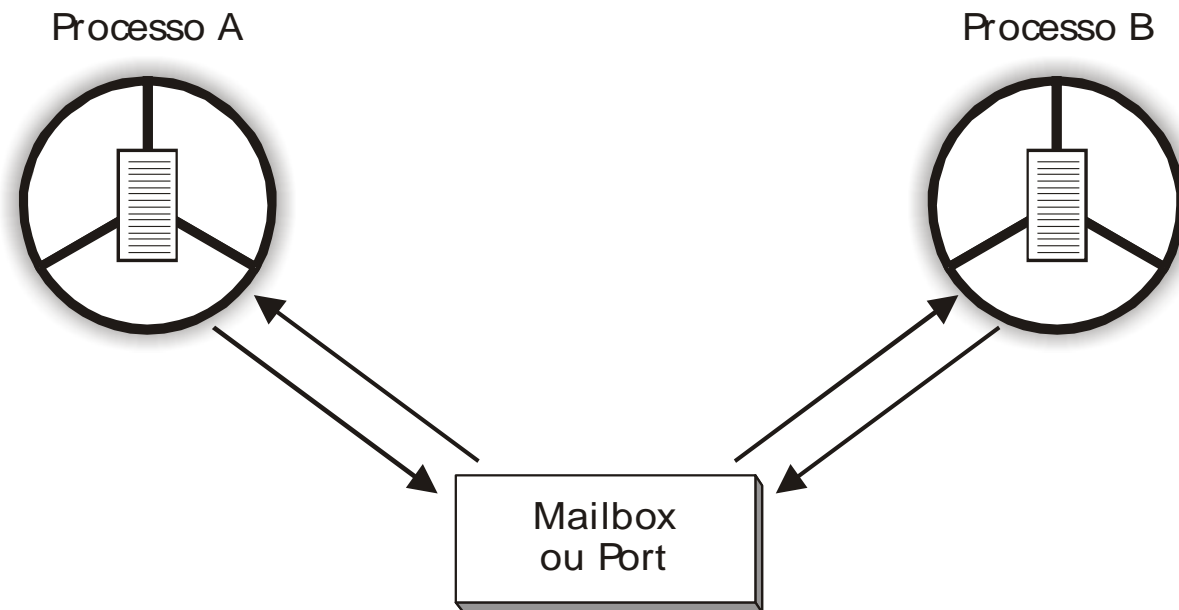


TROCA DE MENSAGENS

- Comunicação direta

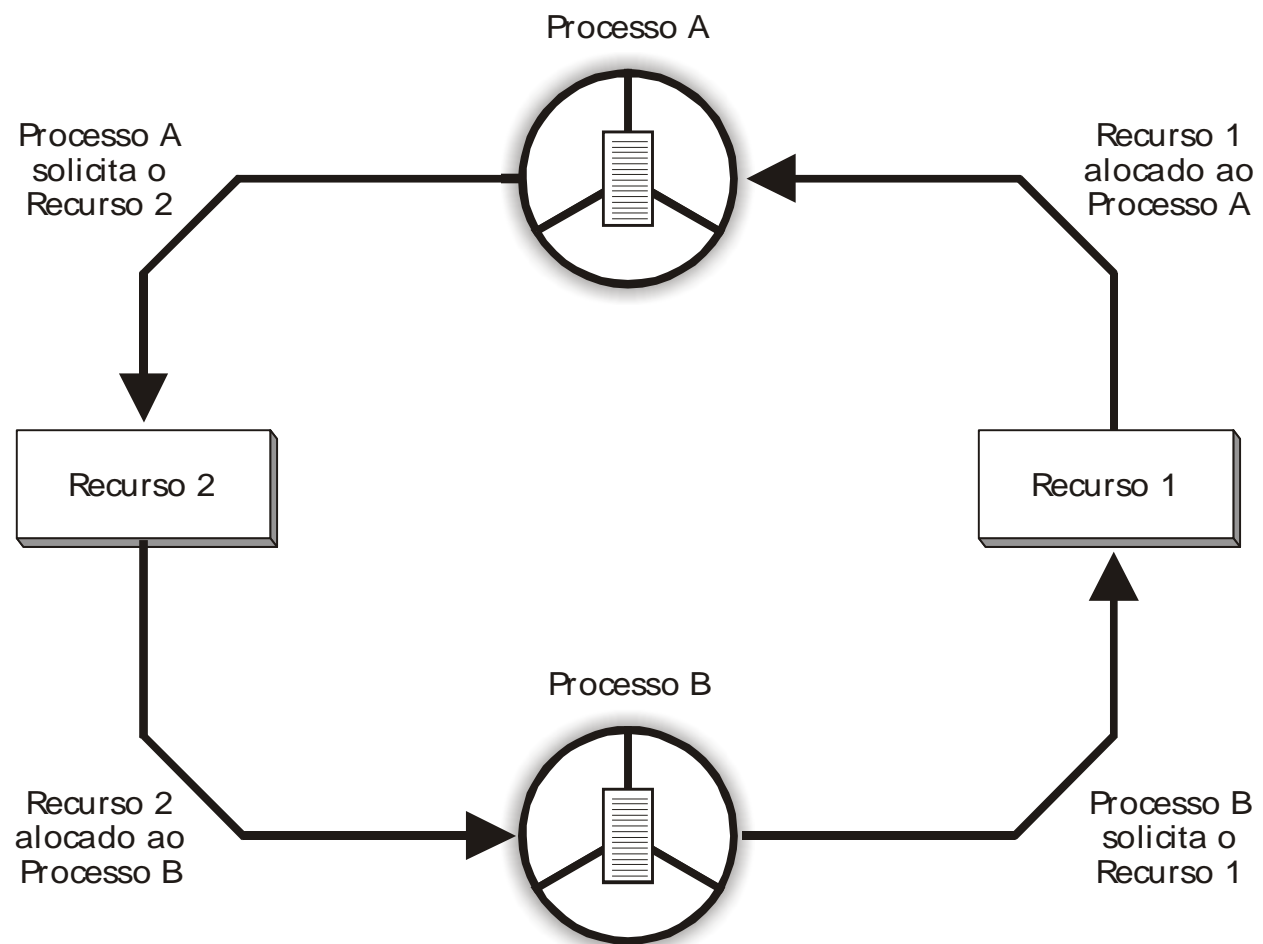


TROCA DE MENSAGENS



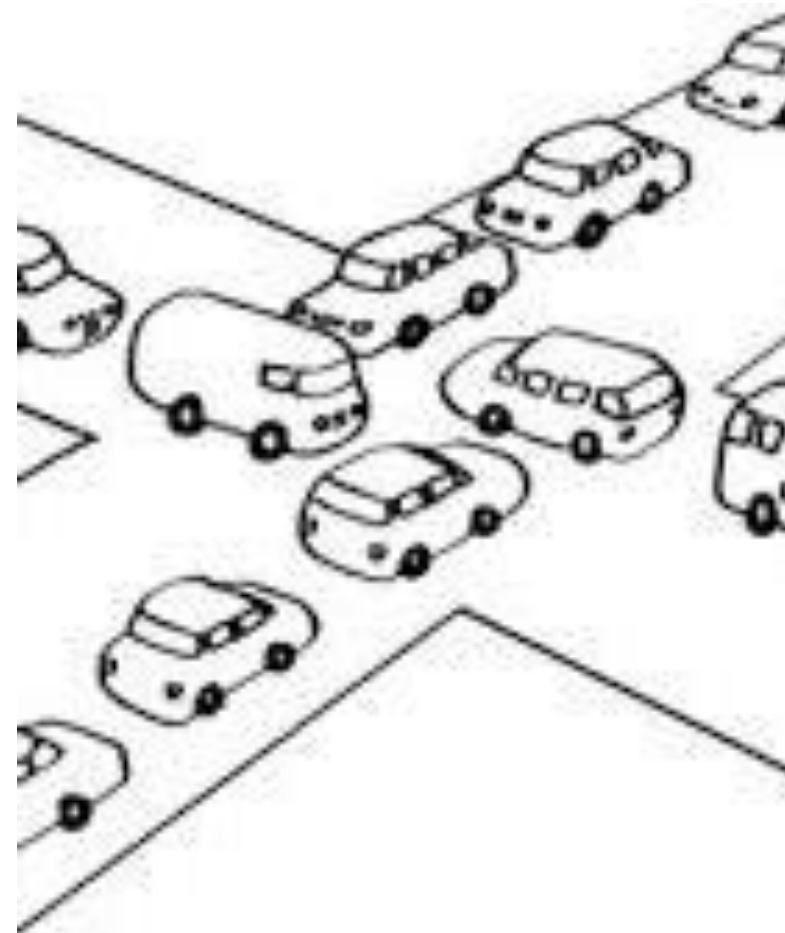
DEADLOCK

- Espera circular



DEADLOCK

- É uma situação em que um processo aguarda por um recurso que nunca estará disponível ou um evento que nunca ocorrerá.
- Consequência do compartilhamento de recursos, dispositivos, arquivos e registros entre processos concorrentes em que a exclusão mútua é exigida.



DEADLOCK



Condições necessárias

- Exclusão mútua
- Espera por recurso
- Não-preempção
- Espera circular



Mecanismos

- Prevenção
- Deteção
- Correção

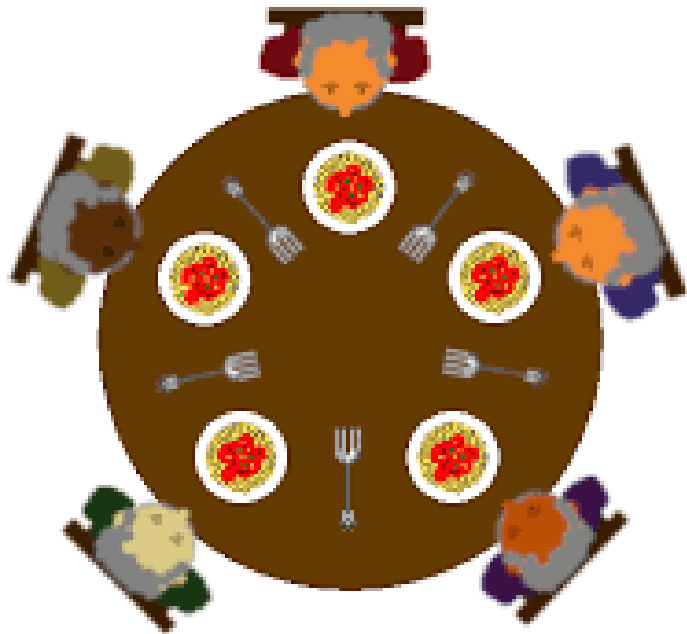
DEADLOCK

- Prevenção
 - Garantir que uma das 4 condições não exista
 - O processo deve pré-alocar seus recursos para executar
 - Permitir a retirada do recurso
 - Excluir a espera circular – o processo pode ter só 1 recurso por vez, liberar um recurso para solicitar outro recurso
- Não é utilizada na prática porque limita muito o sistema.

CORREÇÃO DE DEADLOCK

- Solução mais comum
 - Eliminar um dos processos envolvidos e desalocar seus recursos.
- Outra solução
 - Liberar apenas alguns recursos alocados pelos processos.

PROBLEMA - JANTAR DOS FILÓSOFOS



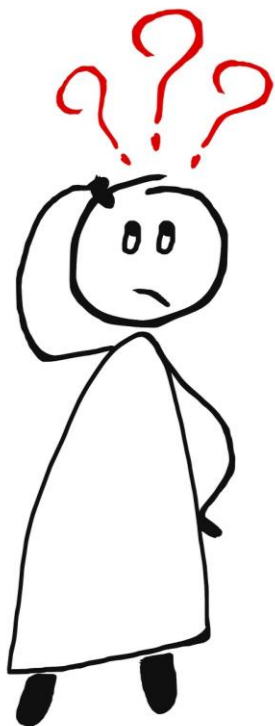
- Neste problema, há uma mesa com cinco pratos e cinco garfos onde os filósofos podem sentar, comer e pensar. Toda vez que um filósofo para de pensar e deseja comer é necessário que ele utilize dois garfos, posicionados à sua direita e à sua esquerda.

PROBLEMA DO BARBEIRO

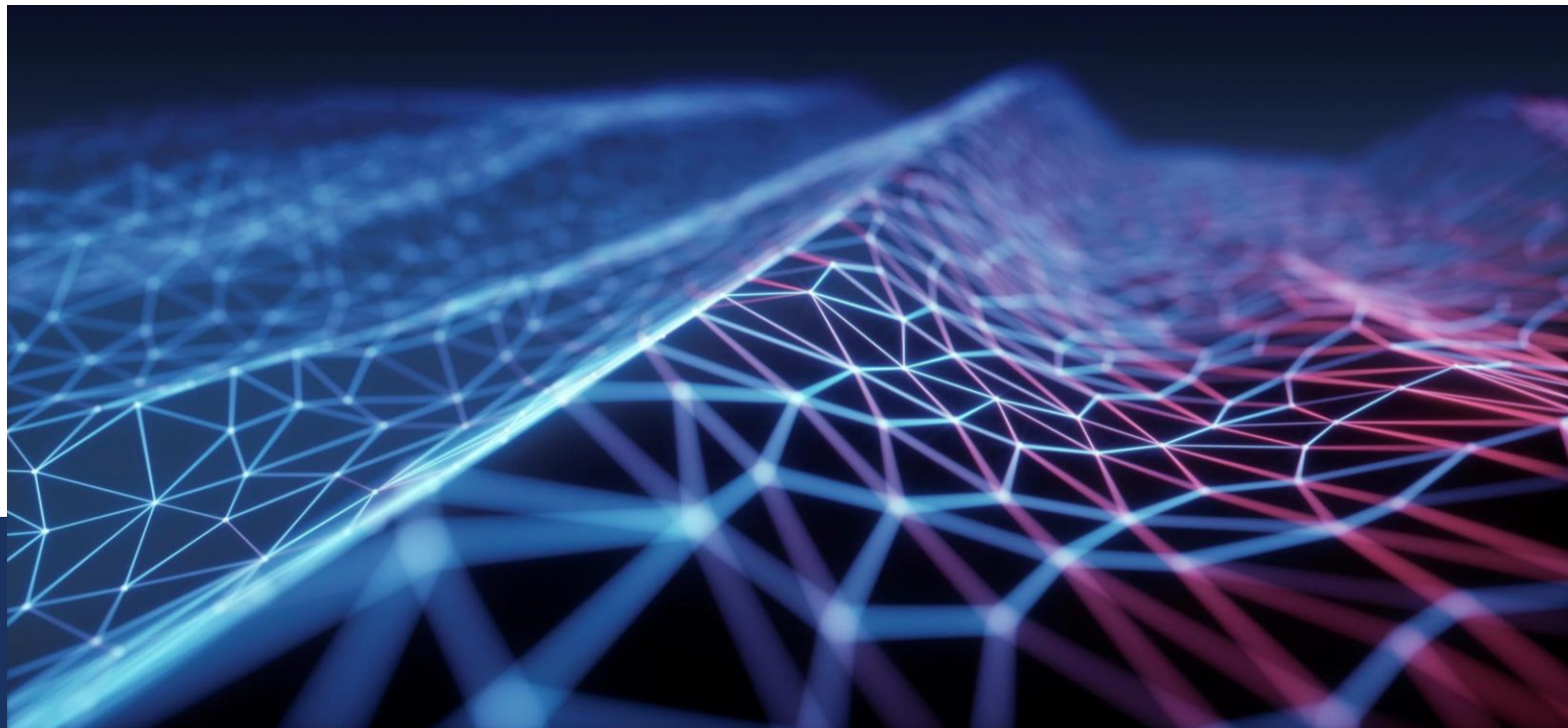
- Um barbeiro recebe clientes para cortar o cabelo. Na barbearia há uma cadeira de barbeiro e apenas cinco cadeiras para os clientes esperarem. Quando um cliente chega, caso o barbeiro esteja trabalhando ele se senta, ou vai embora se todas estiverem ocupadas. No caso de o barbeiro não ter cliente, ele senta na cadeira e dorme até que o cliente apareça.



O QUE FICA EM MENTE?



1. O que você entende por aplicação concorrente?
2. Conceitue exclusão mútua. Como pode ser implementada?
3. O que é sincronização condicional?
4. Qual o problema da solução que desabilita as interrupções na exclusão mútua?
5. Como funciona o semáforo binário?
6. Qual a diferença entre o semáforo e o monitor?
7. Quais as condições para que ocorra deadlock?
8. Tarefa do grupo para a próxima aula: pesquisar sobre o problema dos Filósofos e do Barbeiro. Apresentar uma solução que evite o starvation e deadlock.



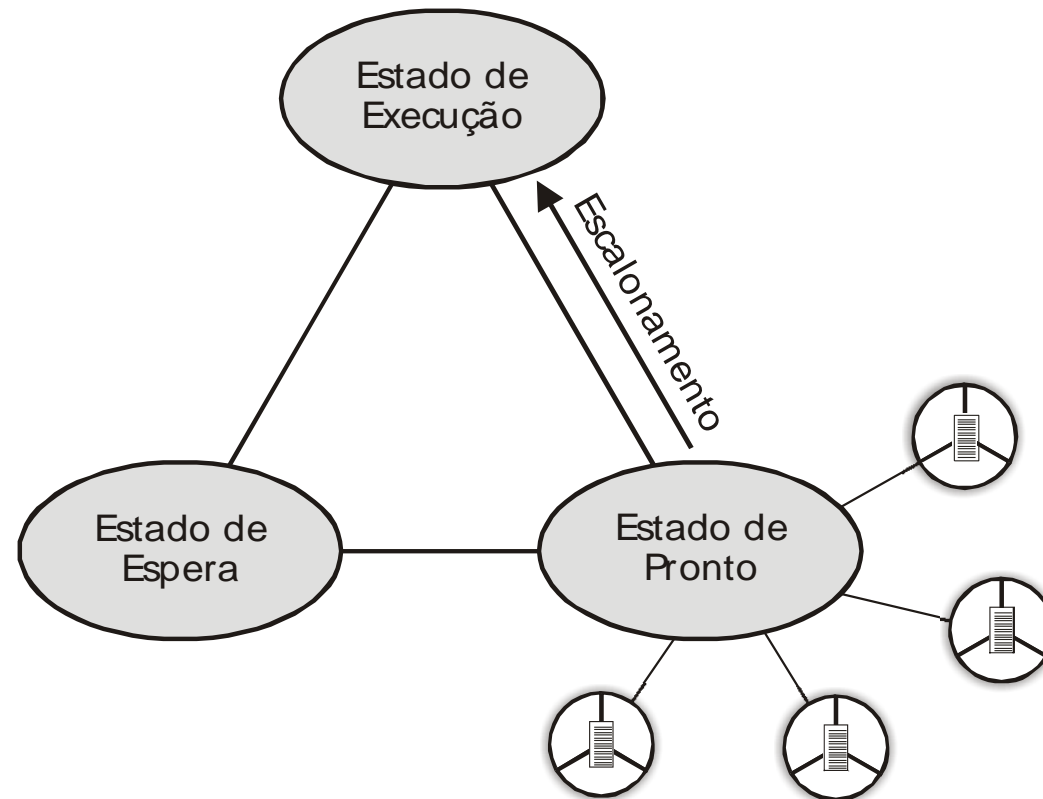
GERÊNCIA DO PROCESSADOR

OBJETIVOS

- Gerência do Processador
- Escalonador
- Algoritmos de Escalonamento

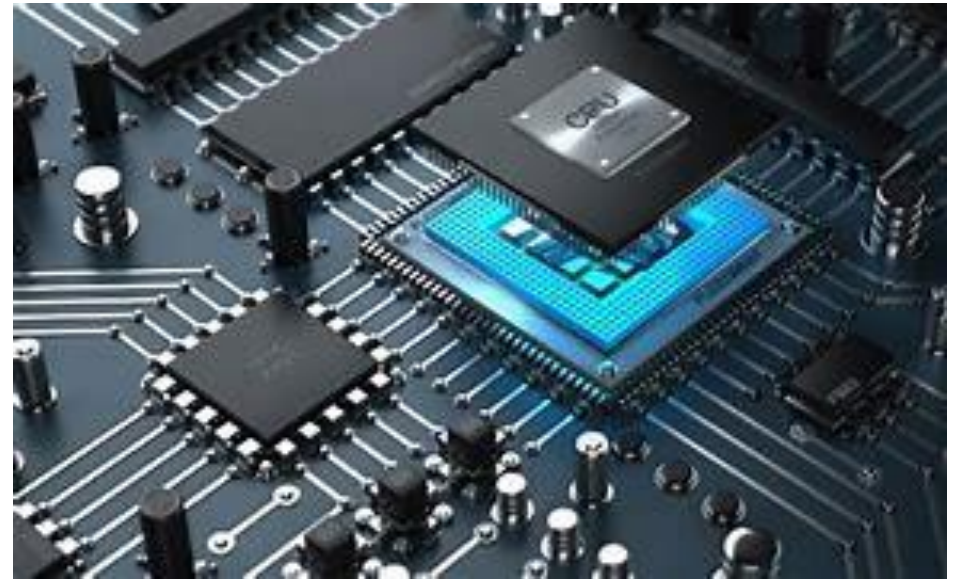
INTRODUÇÃO

- Escalonamento



FUNÇÕES BÁSICAS

- Manter a UCP ocupada a maior parte do tempo
- Balancear o uso da UCP entre processos
- Privilegiar a execução de aplicações críticas
- Maximizar o throughput – número processos
- Oferecer tempos de resposta razoáveis para usuários interativos



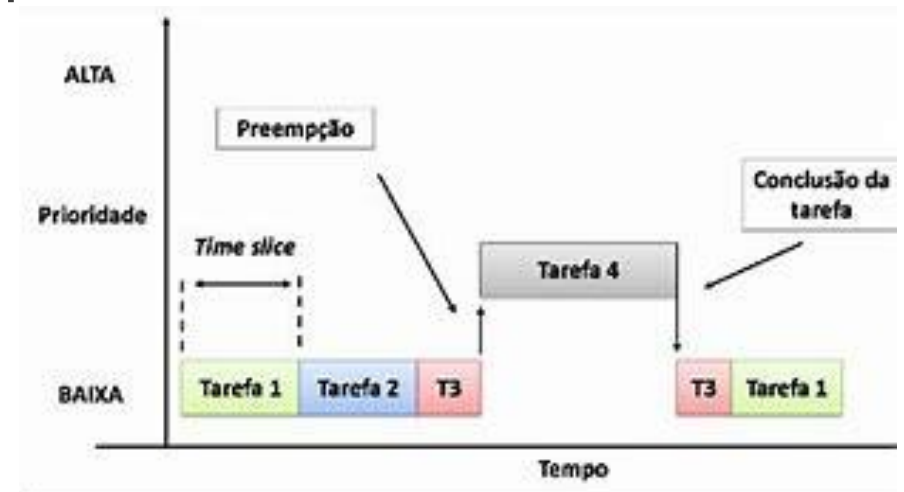
CRITÉRIOS DE ESCALONAMENTO

- Utilização do processador
- Throughput – número de processos
- Tempo de Processador / Tempo de UCP
- Tempo de Espera
- Tempo de Turnaround – tempo total do processo
- Tempo de Resposta

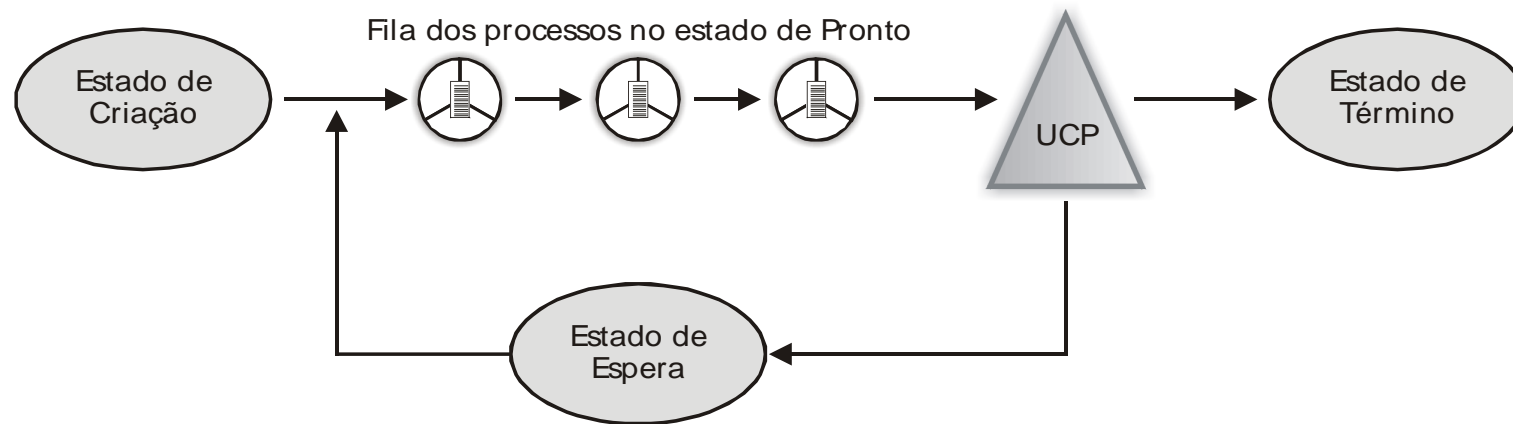


ESCALONAMENTOS NÃO-PREEMPTIVO E PREEMPTIVO

- Preempção – possibilidade do SO interromper um processo em execução e substituí-lo por outro
- Escalonamento não-preemptivo
- Escalonamento preemptivo

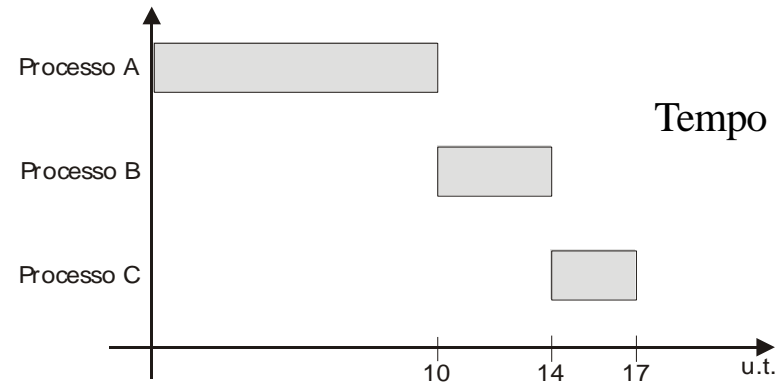


ESCALONAMENTO FIFO

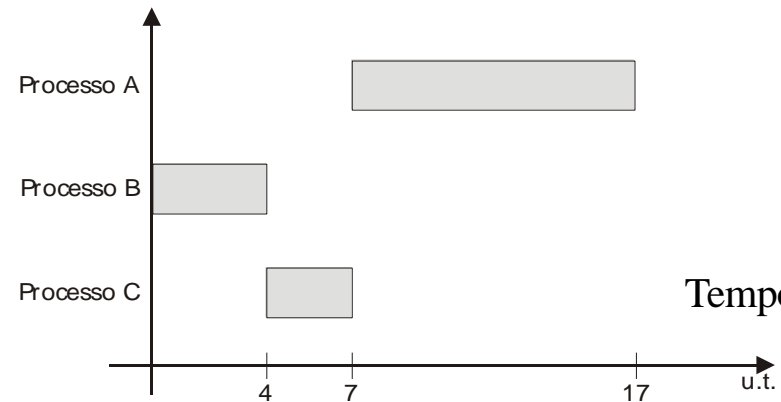


- Escalonamento First-In-First-Out

ESCALONAMENTO FIFO



■ Exemplo

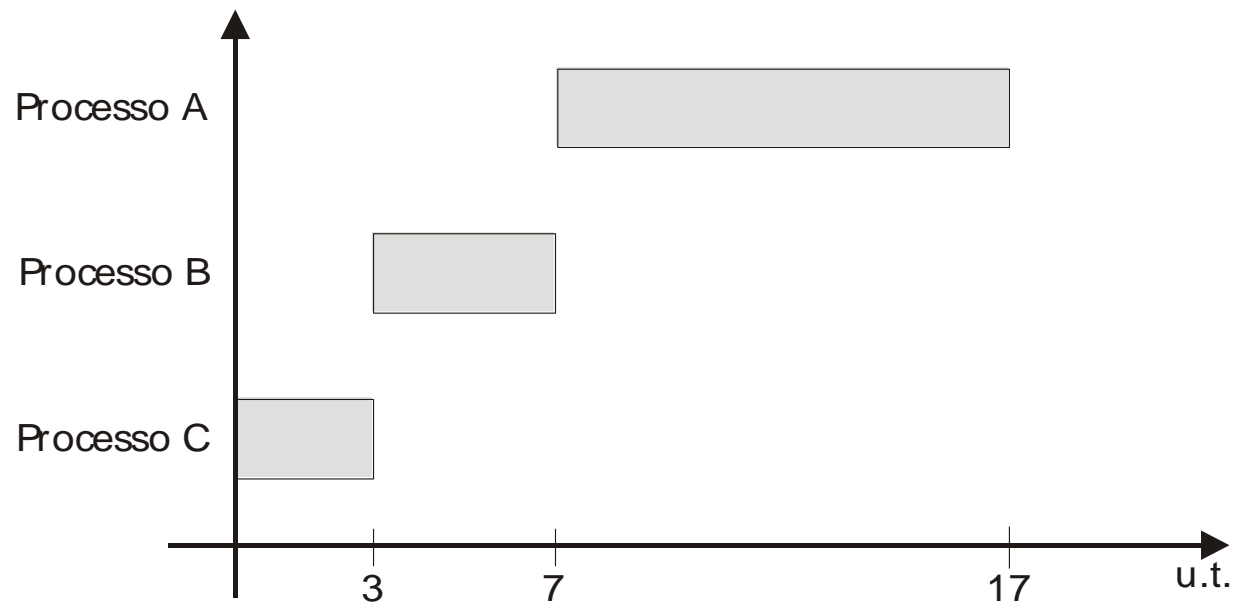


Processo	Tempo de processador (u.t.)
A	10
B	4
C	3

ESCALONAMENTO FIFO

- Deficiências:
 - Impossibilidade de prever quando um processo terá sua execução iniciada;
 - O algoritmo não se preocupa em melhorar o tempo médio de espera dos processos – tempo de turnaround.
 - Os processos CPU-bound levam vantagem no uso do processador sobre processos I/O bound.
 - Foi projetado para sistemas monoprogramáveis (processamento batch)

ESCALONAMENTO SJF

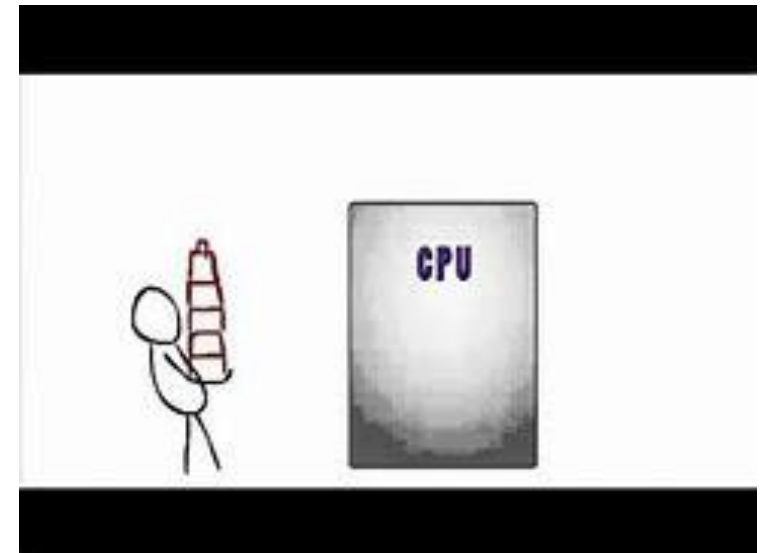


Tempo de espera = 3,3 u.t.

- Escalonamento Shortest-Job-First

ESCALONAMENTO SJF

- Seleciona o processo que tiver o menor tempo de processador ainda para executar no estado de pronto.
- Foi utilizado nos primeiros sistemas operacionais exclusivamente batch.
- Principal problema: impossibilidade de estimar o tempo para processos interativos.

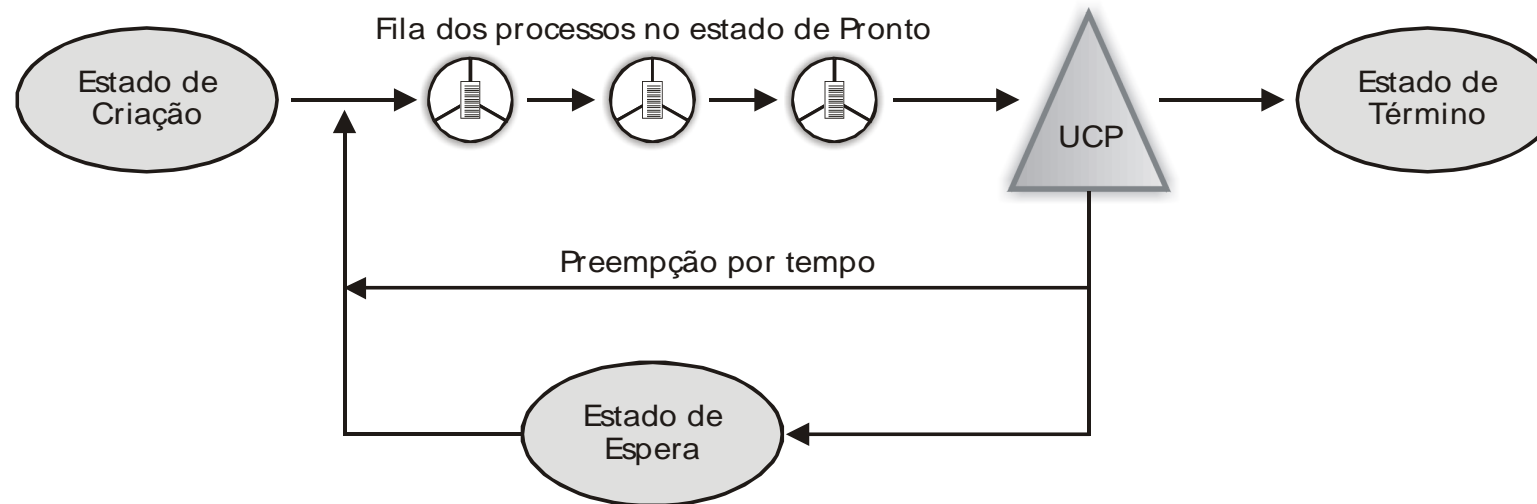


ESCALONAMENTO COOPERATIVO

- Processo em execução libera voluntariamente a UCP
- Verifica a fila de mensagens periodicamente
- Podem ocorrer problemas – o processo não verificar a fila de mensagens.
- Exemplo: primeiros sistemas

MS-Windows

ESCALONAMENTO CIRCULAR



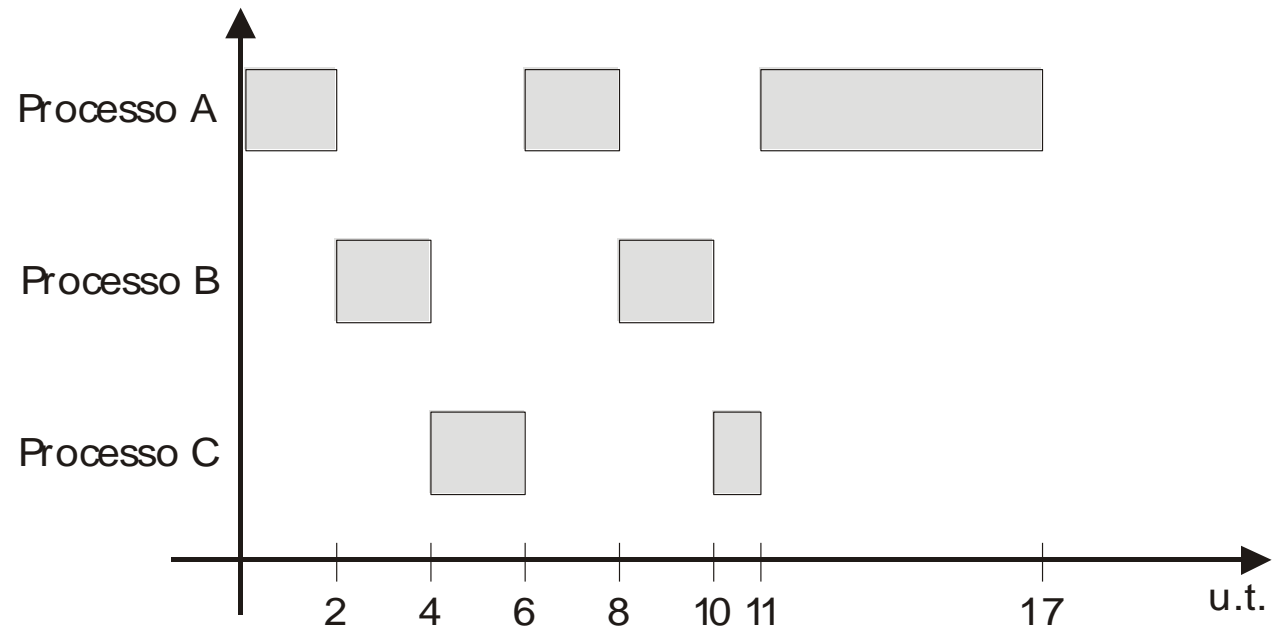
Escalonamento Preemptivo projetado para tempo compartilhado!

ESCALONAMENTO CIRCULAR

- Bastante semelhante ao FIFO, mas com fatia de tempo ou *quantum*.
- Toda vez que um processo é escalonado para execução uma nova fatia de tempo é concedida.
- Valor da fatia de tempo: muito alto, fica semelhante ao fifo. Muito baixa, grande número de preempções com excessivas mudanças de contexto. (10 a 100 milissegundos).

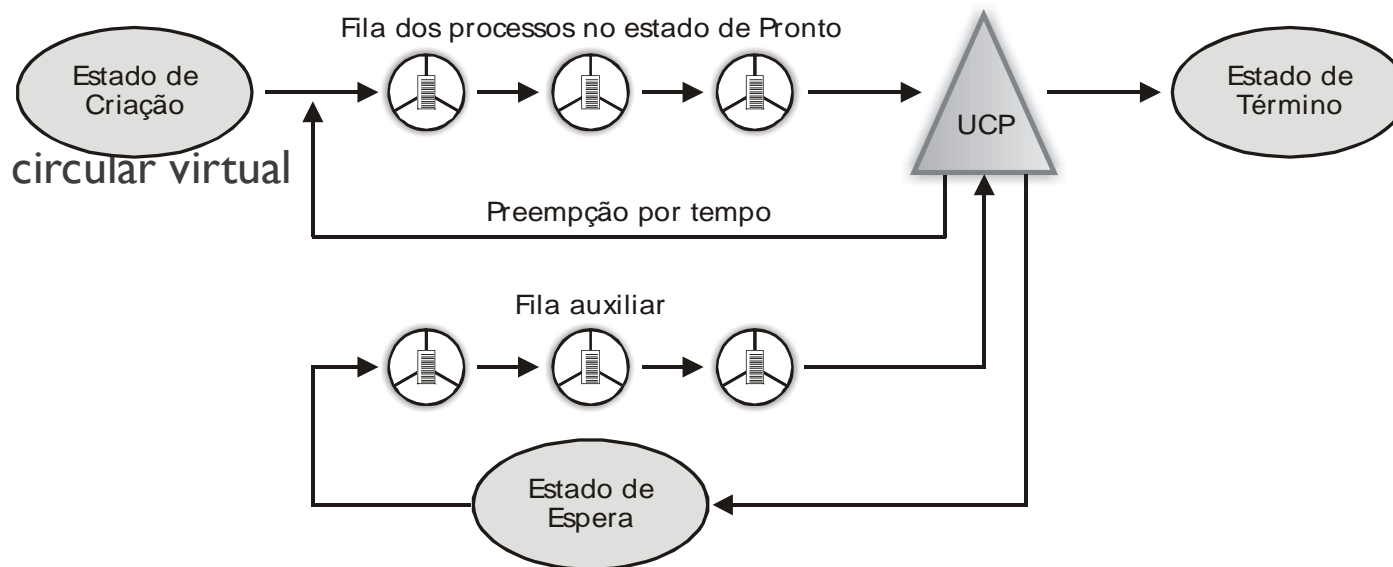
ESCALONAMENTO CIRCULAR

- Exemplo

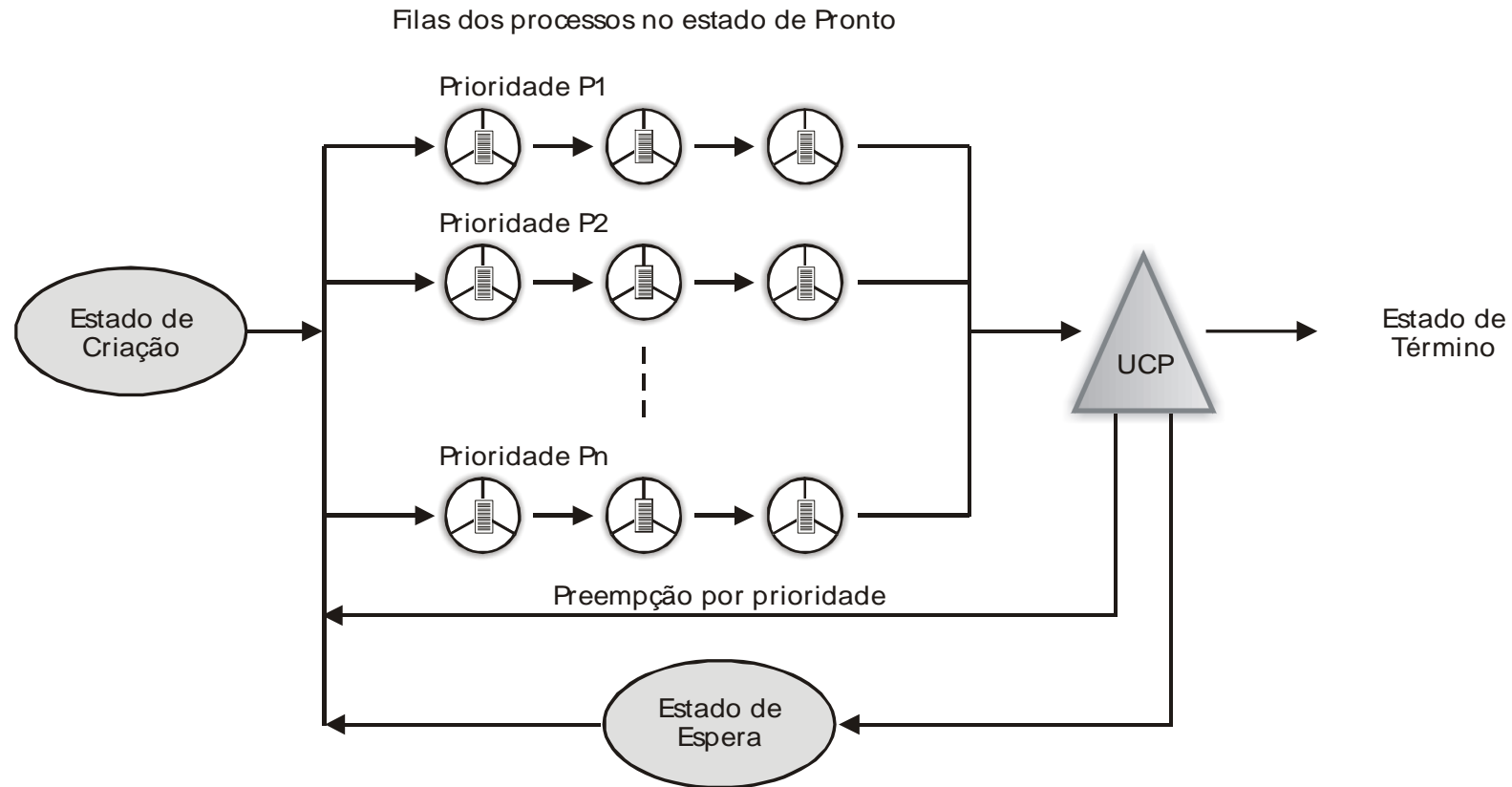


ESCALONAMENTO CIRCULAR

- Escalonamento circular virtual



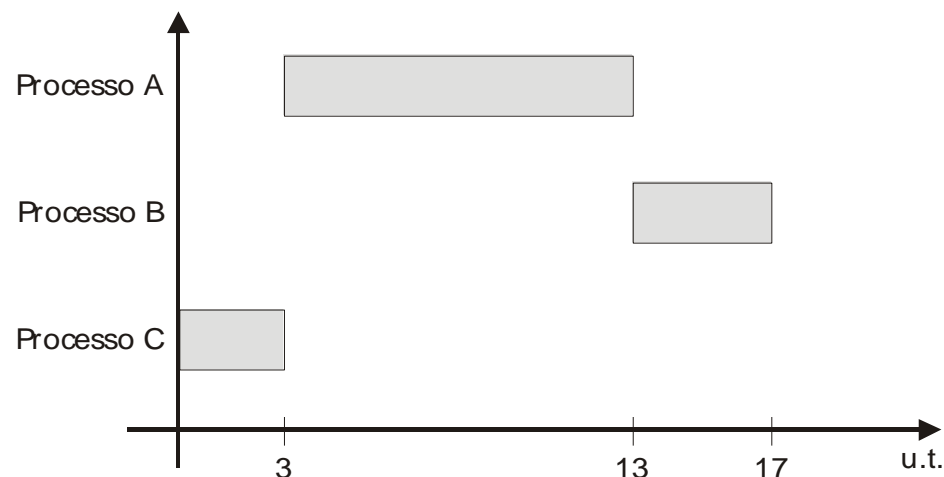
ESCALONAMENTO POR PRIORIDADES



Processo com maior prioridade é escolhido, não existe fatia de tempo.

ESCALONAMENTO POR PRIORIDADES

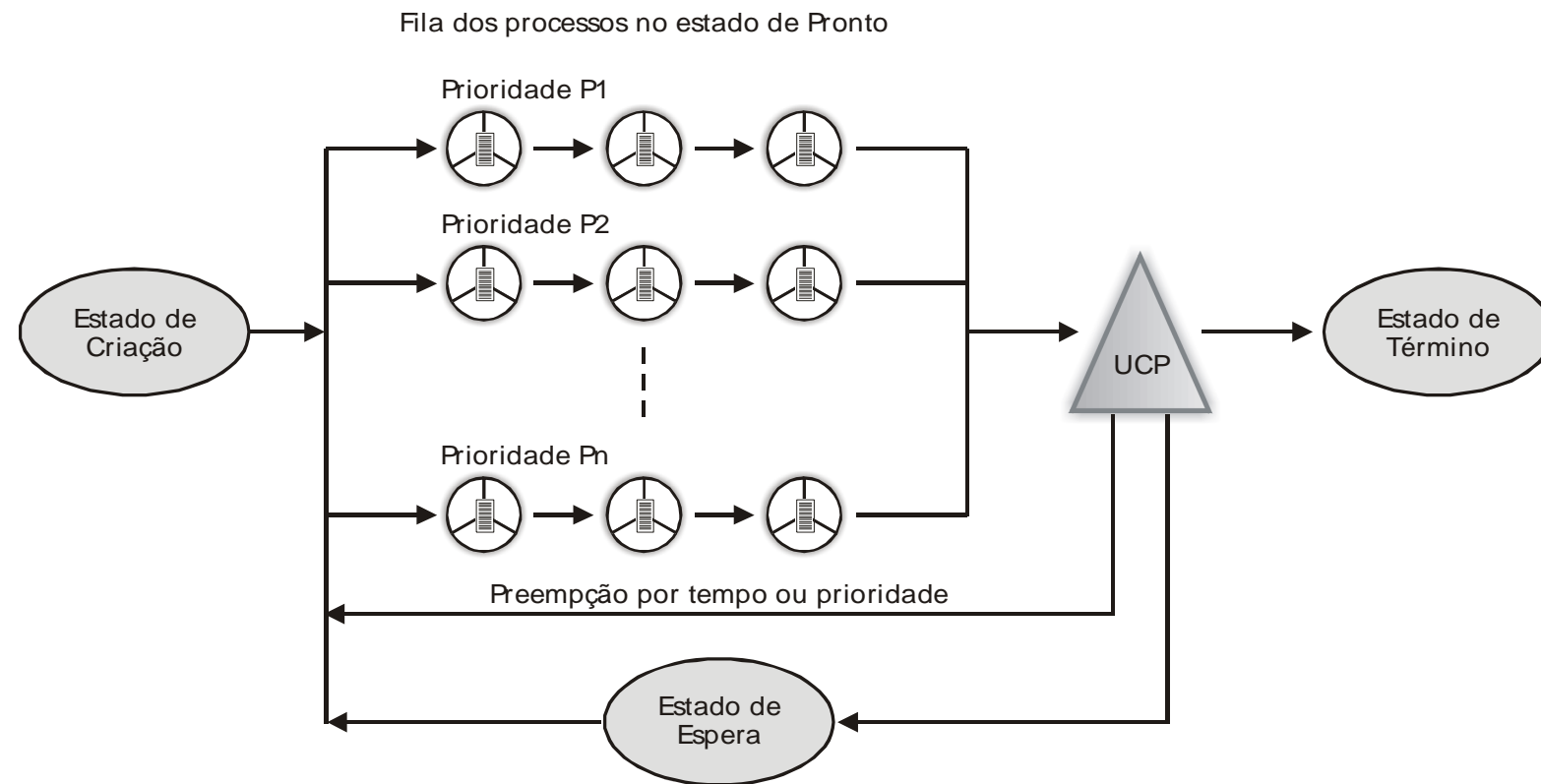
■ Exemplo



Processo	Tempo de processador (u.t.)	Prioridade
A	10	2
B	4	1
C	3	3

A perda do processador ocorrerá no caso de uma mudança para o estado de espera ou quando um processo de maior prioridade estiver em pronto.

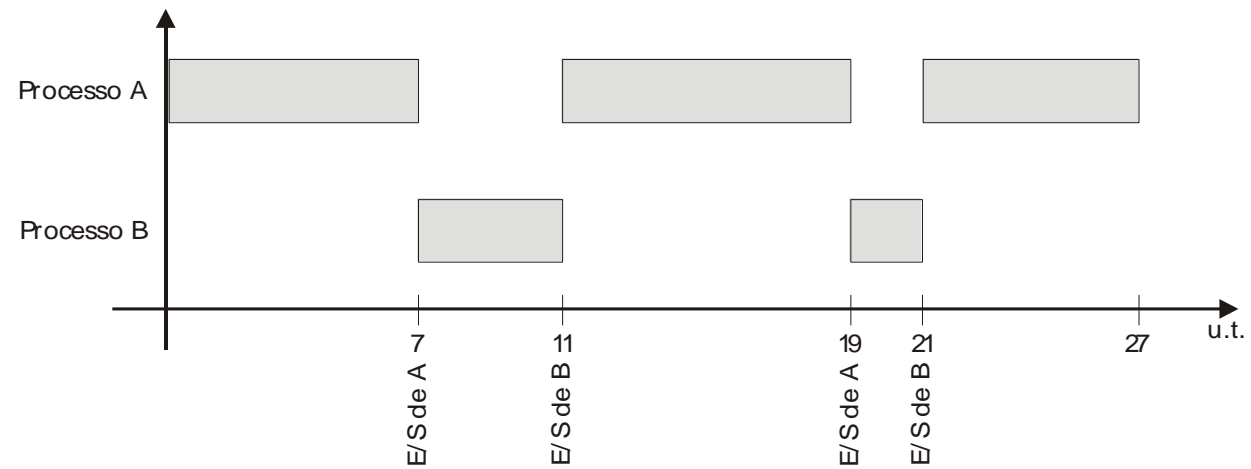
ESCALONAMENTO CIRCULAR COM PRIORIDADES



ESCALONAMENTO CIRCULAR COM PRIORIDADES

- Implementa fatia de tempo e prioridades.
- Permite um melhor balanceamento no uso do processador em sistemas de tempo compartilhado.
- Prioridade estática ou dinâmica – permanece ou não inalterada no contexto de software ao longo da existência do processo.

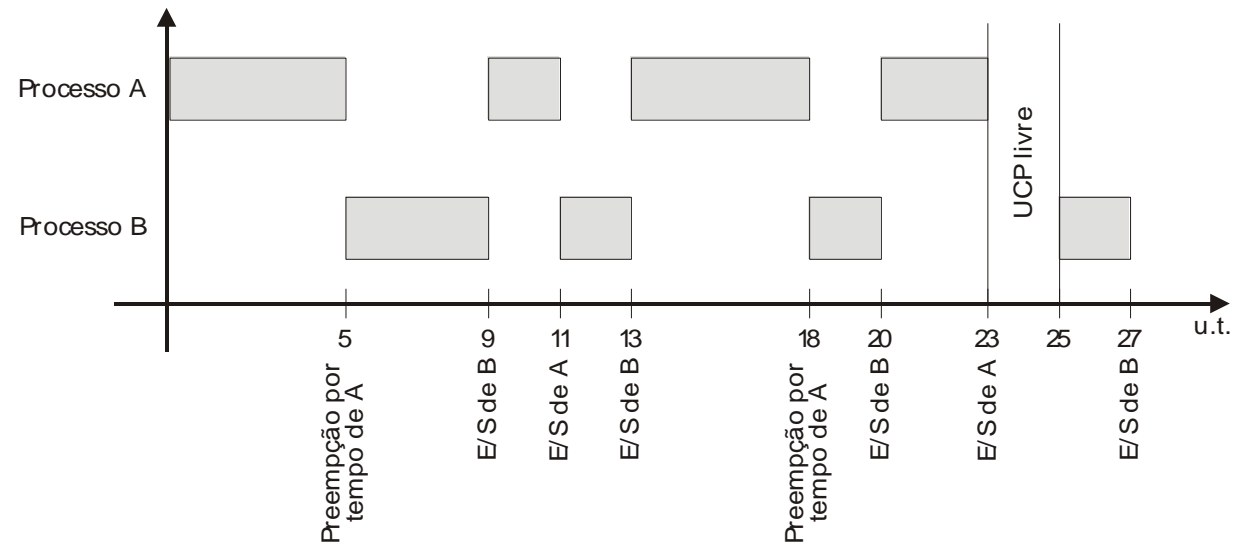
POLÍTICAS EM SISTEMAS DE TEMPO COMPARTILHADO



Processo	Tempo de processador (u.t.)	Característica
A	21	CPU-bound
B	6	I/O-bound

- Escalonamento FIFO (exemplo)

POLÍTICAS EM SISTEMAS DE TEMPO COMPARTILHADO



- Escalonamento circular (exemplo)

Processo	Tempo de processador (u.t.)	Característica
A	15	CPU-bound
B	10	I/O-bound

POLÍTICAS EM SISTEMAS DE TEMPO REAL

- Tempos de respostas rígidos
- Aplicações de controle de processos
- Utiliza prioridades estáticas
- Não utiliza fatias de tempo
- Exemplo: OpenVMS

UMA DÚVIDA...

Que algoritmo de escalonamento seria mais adequado para um sistema operacional cliente-servidor?



ATIVIDADE

- Quais recursos são necessários para a criação de um processo?
- Qual a diferença básica entre um processo e um programa?
- Por que, muitas vezes, “quebrar” um programa em mais de um processo pode ser vantajoso?
- Pesquise sobre plataformas multicore e como elas podem colaborar com o desempenho de um computador em termos de processamento.

CONCLUSÃO

- Conhecemos um pouco sobre SO.
- Conhecemos também seus tipos, funções, estrutura e arquiteturas e seus processos.
- Aprofundamos são necessários para refinar estes conhecimentos, acesse a literatura indicada.
- A necessidade vai dizer quais serão importantes.

REFERÊNCIAS

- COMER, Douglas E. Redes de Computadores e Internet. Porto Alegre: Bookman, 2016. <https://integrada.minhabiblioteca.com.br/reader/books/9788582603734/>
- TANENBAUM, Andrew. Redes de Computadores. 5.ed. São Paulo: Campus, 2011. <https://plataforma.bvirtual.com.br/Acervo/Publicacao/2610>
- MORAES, Alexandre Fernandes de; Redes de computadores. -- 1. ed. -- São Paulo : Érica, 2014. <https://integrada.minhabiblioteca.com.br/reader/books/9788536532981/>



FIM UNIDADE I