



Introdução a Arquitetura de Sistemas Operacionais - Parte 01

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Prof. Edwar Saliba Júnior

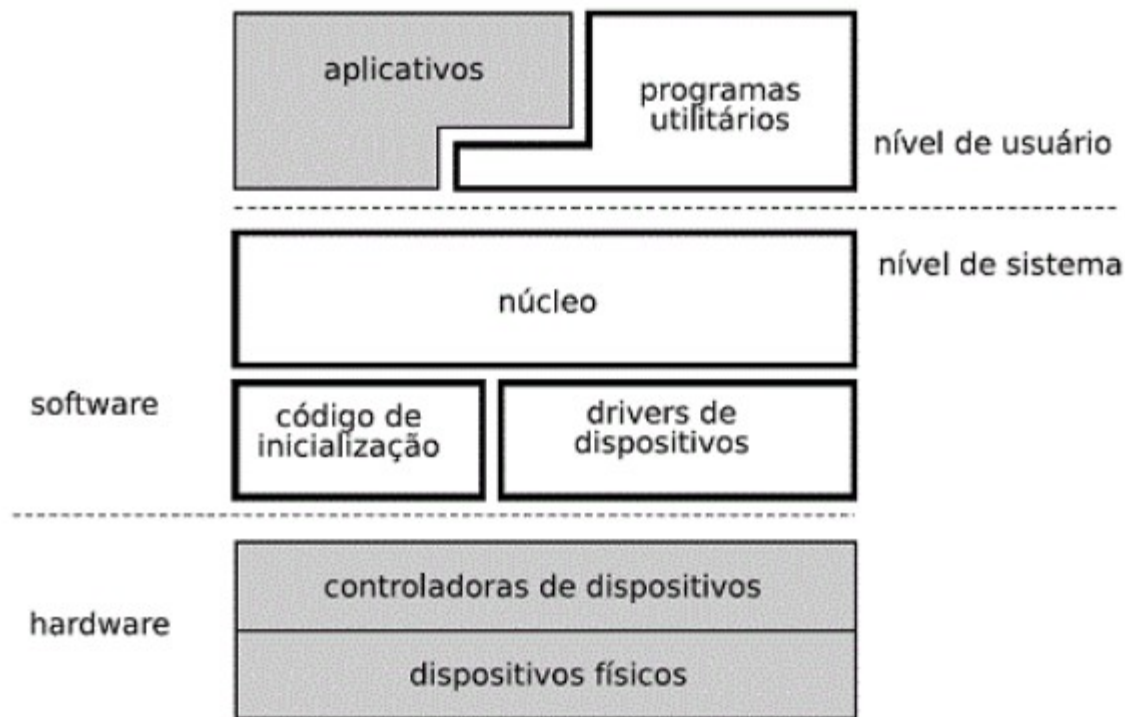
Julho / 2017



Nota

- Com exceção das figuras que possuem referência bibliográfica, todo o restante do conteúdo destes *slides* foram extraídos do livro dos autores “Francis Berenger Machado” e “Luiz Paulo Maia”. Todos referenciados no último *slide* desta apresentação.

Hardware e Software



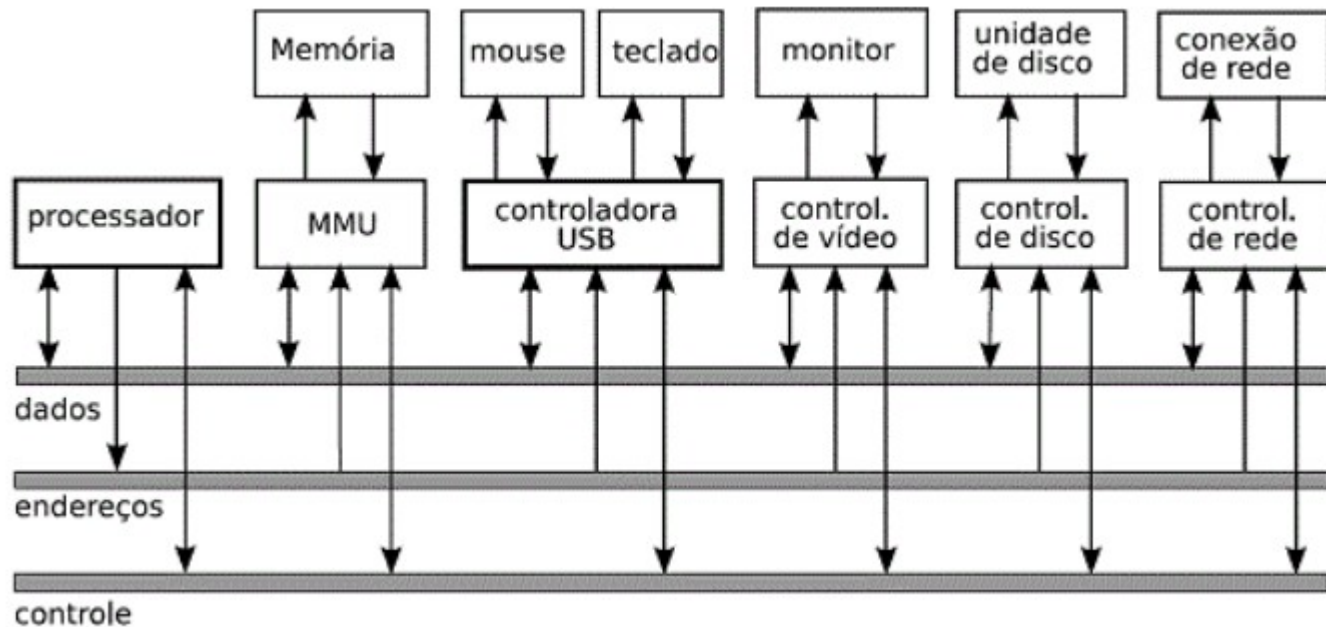


Arquitetura de Um Computador

- Todos os componentes de um sistema computacional são agrupados em três subsistemas básicos, chamados unidades funcionais: CPU (processador), memória principal e dispositivos de entrada e saída, baseado em uma arquitetura chamada de “arquitetura Von Neumann”.



Arquitetura de Um Computador



MMU – Memory Management Unit

USB – Universal Serial Bus



Concorrência

- Os sistemas operacionais podem ser vistos como um conjunto de rotinas são executadas concorrentemente de forma ordenada;
- A possibilidade do processador executar instruções em paralelo com operações de E/S, permite que diversas tarefas sejam executadas concorrentemente;
- O conceito de concorrência é o princípio básico para o projeto dos sistemas multiprogramados.



Concorrência

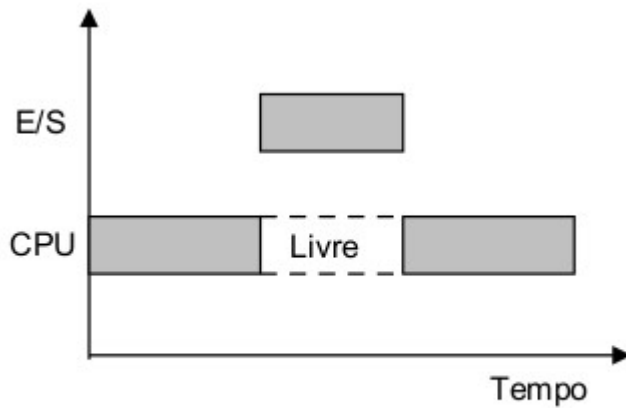


Figura 3.1: Sistema Monoprogramável

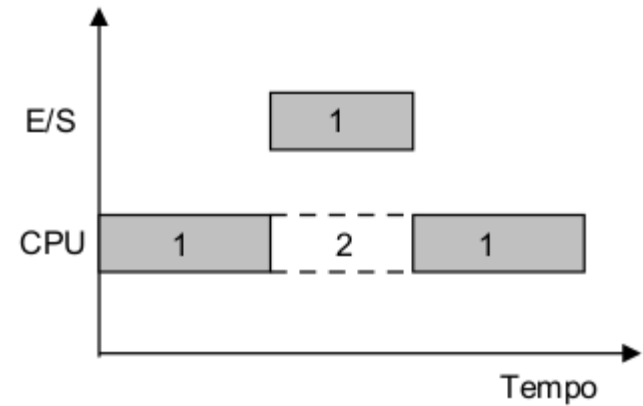


Figura 3.2: Sistema multiprogramável

Exemplo de concorrência: `$ top`



Concorrência

Leitura de um registro	0,0015	segundos
Execução de 100 instruções	0,0001	segundos
Total	0.0016	segundos
Percentual de utilização da UCP	$\frac{0,0001}{0,0015}$	$= 0,066 = 6,6\%$

Figura 3.3: Tabela com exemplo de utilização do sistema

- Na tabela da figura 3.3, vemos um exemplo de um programa que lê registros de uma arquivo e executa, em média, 100 instruções de máquina por registro lido;
- Neste caso, o processador gasta 93% do tempo esperando o dispositivo de E/S concluir a operação para continuar o processamento;
- Em um sistema monoprogramável, a UCP é utilizada em aproximadamente 30% do tempo, enquanto em sistemas multiprogramáveis o tempo de utilização sobe para até 90%.



Interrupção

- A interrupção é o mecanismo que tornou possível a implementação da concorrência nos computadores, sendo o fundamento básico dos sistemas multiprogramáveis;
- É em função desse mecanismo que o sistema operacional sincroniza a execução de todas as suas rotinas e dos programas dos usuários, além de controlar dispositivos.



Exceção

- Uma exceção é semelhante a uma interrupção, sendo a principal diferença o motivo pelo qual o evento é gerado;
- 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* em uma operação aritmética.



Interrupção ou Exceção



Figura 3.6: Mecanismos de interrupção e exceção



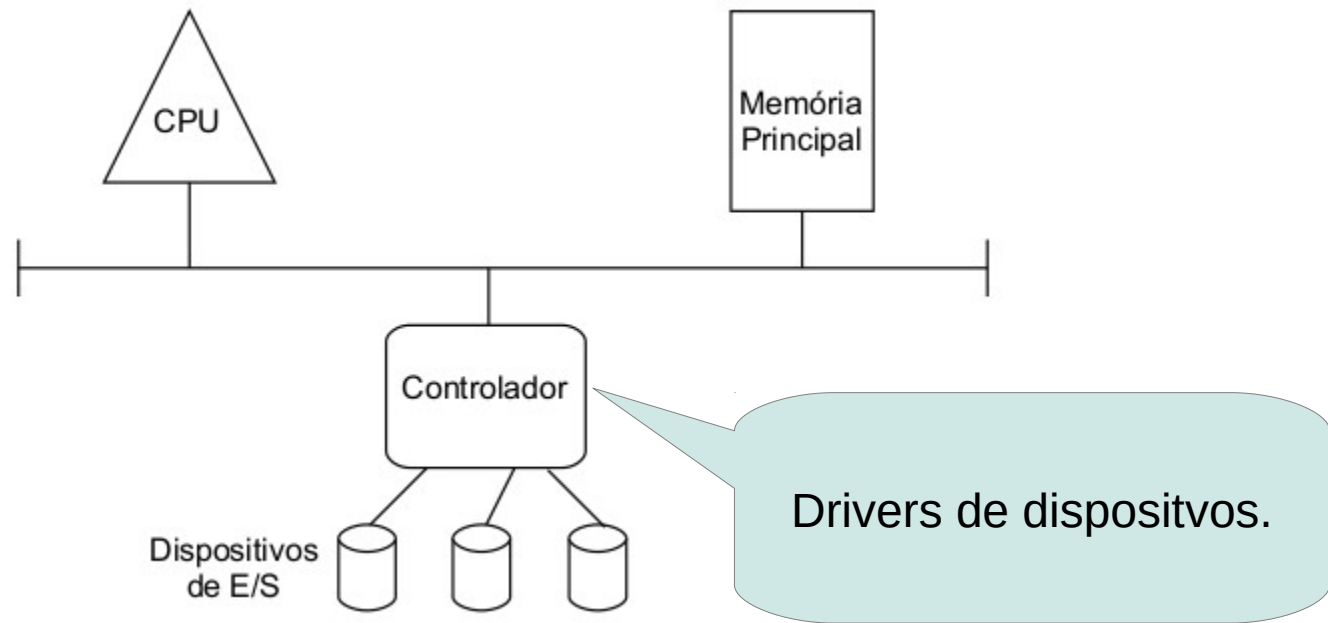
Operações de Entrada e Saída (E/S ou I/O)

- Nos primeiros sistemas computacionais, a comunicação entre o processador e os periféricos era controlada por um conjunto de instruções especiais, denominadas instruções de entrada/saída, executadas pelo próprio processador. Essas instruções continham detalhes específicos de cada periférico;
- O surgimento do controlador ou interface permitiu ao processador agir de maneira independente dos dispositivos de E/S. Com esse novo elemento, o processador não mais se comunicava diretamente com os periféricos, mas sim através do controlador.



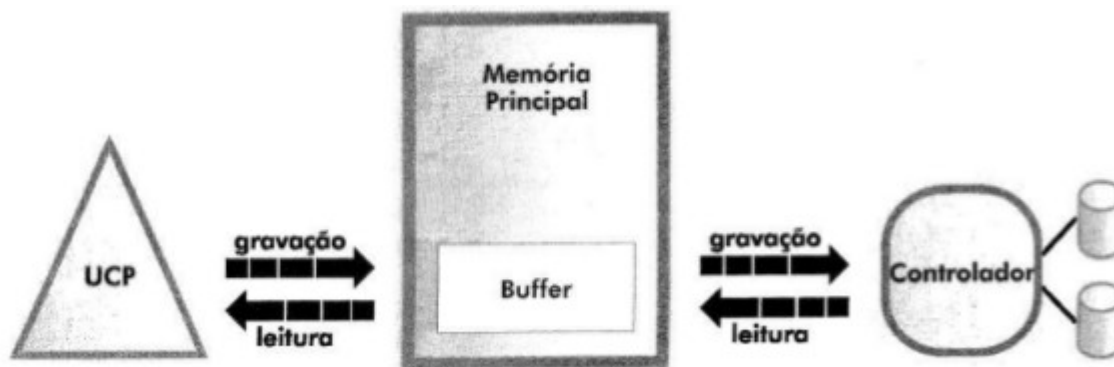
Sistema Operacional GNU/Linux

Operações de Entrada e Saída (E/S ou I/O)



Buffering

- A técnica de *buffering* é a utilização de uma área na memória principal, denominada *buffer*, para a transferência de dados entre os dispositivos de E/S e a memória;
- Esta técnica permite que em uma operação de leitura, o dado seja transferido primeiramente para o *buffer*, liberando imediatamente o dispositivo de entrada para realizar uma nova leitura.

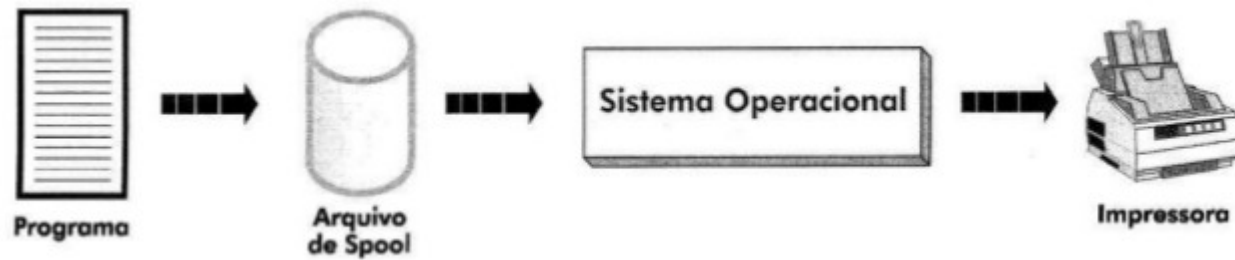




SPOOLING

- *Simultaneous Peripheral Operation On-line*
- Técnica introduzida no final dos anos 1950 para aumentar o grau de concorrência e a eficiência dos sistemas operacionais;
- Está presente na maioria dos sistemas operacionais e é utilizada no gerenciamento de impressão. No momento em que um comando de impressão é executado, as informações que serão impressas são gravadas antes em um arquivo em disco, conhecido como arquivo de *pool*, liberando imediatamente o programa para outras atividades;
- Posteriormente, o sistema operacional encarrega-se de direcionar o conteúdo do arquivo de *pool* para a impressora.

SPOOLING





Proteção do Sistema

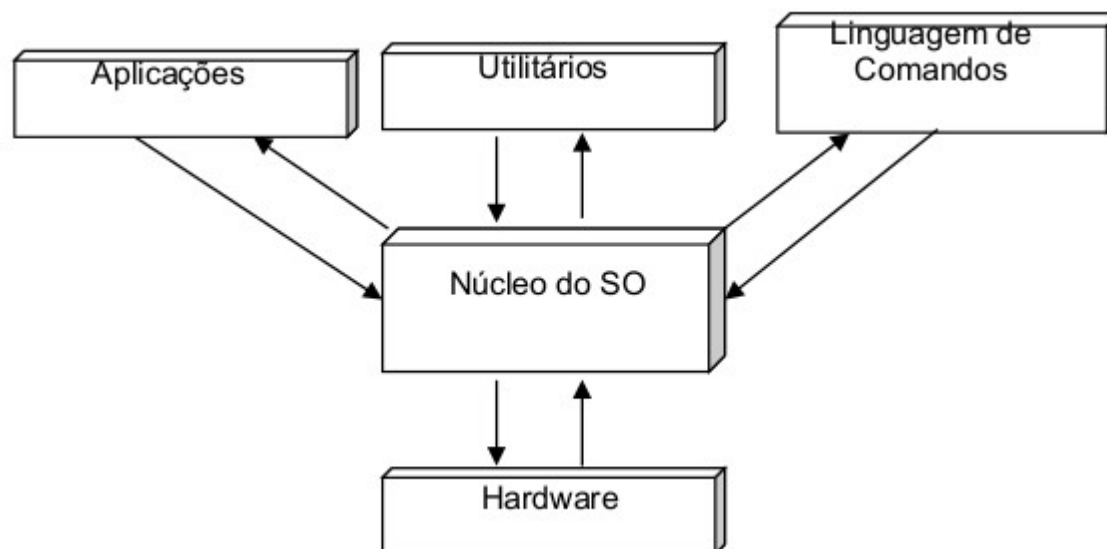
- Sistemas Multitarefa / Multiusuários
 - várias tarefas sendo executadas “simultaneamente”;
 - vários processos (programas) utilizando:
 - a mesma memória,
 - o mesmo processador e
 - o mesmo HD.
- O que garante que os dados de um processo não se misturará com os dados de outro processo? Na memória, no processador ou mesmo no HD.



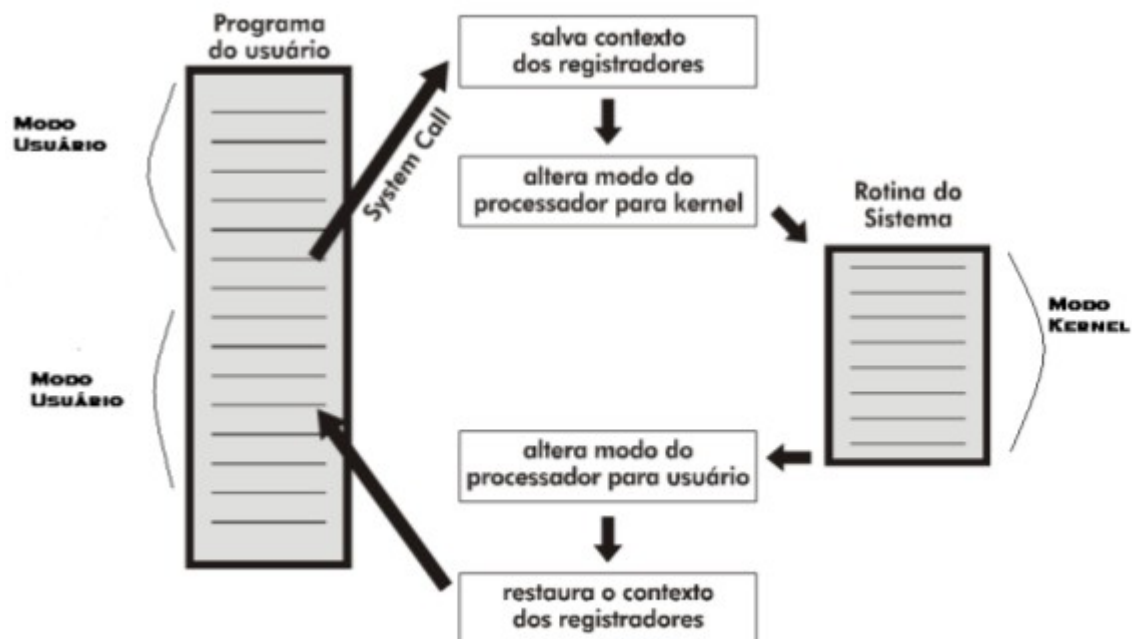
Estrutura do Sistema Operacional

- O sistema operacional é formado por um conjunto de rotinas que oferece serviços aos usuários e às aplicações;
- Esse conjunto de rotinas é denominado núcleo do sistema, ou kernel;
- A maioria dos sistemas operacionais é fornecida acompanhada de utilitários e linguagem de comandos, que são ferramentas de apoio ao usuário, porém, não são parte do núcleo do sistema.

Estrutura do Sistema Operacional



System Calls



Modo Usuário:

- Executa apenas instruções não – privilegiadas;
- Acesso a um número restrito de instruções;

Modo Kernel:

- Executa instruções privilegiadas.
- Acesso total ao conjunto de instruções do processador.



System Calls

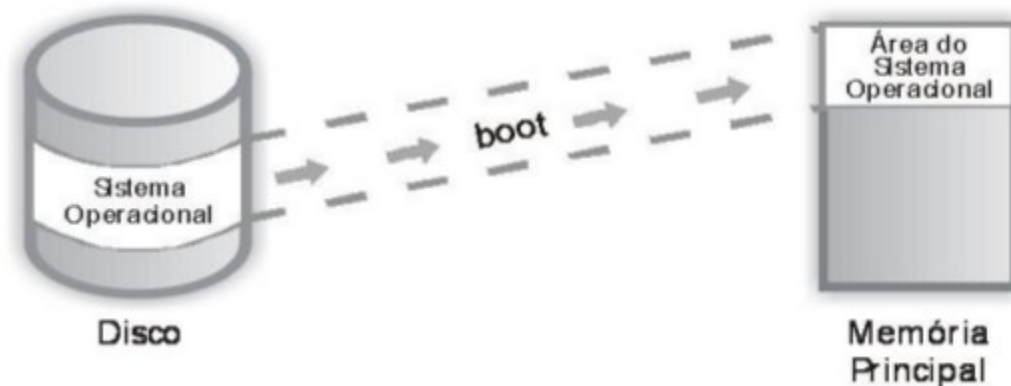
Funções	System Calls
Gerencia de processos e threads	Criação e eliminação de processos e threads Alteração das características de processos e threads Sincronização e comunicação entre processos e threads Obtenção de informações sobre processos e threads
Gerência de memória	Alocação e desalocação de memória
Gerência do sistema de arquivos	Criação e eliminação de arquivos e diretórios Alteração das características de arquivos e diretórios 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



Ativação / Desativação do S.O.

- Toda vez que um computador é ligado é necessário que programas sejam carregados da memória secundária para a memória principal. Este processo, denominado ativação do sistema (*boot*), é realizado por um programa localizado em um local específico do disco (*disk block*), sendo geralmente o primeiro bloco;
- Na maioria dos sistemas existe o processo de desativação denominado *Shutdown*. Este procedimento permite que as aplicações sejam encerradas de forma correta.

Ativação / Desativação do S.O.





Estrutura do Processo





Processo

- O conceito de processo é a base para a implementação de um sistema multiprogramado. O processador é projetado apenas para executar instruções, não sendo capaz de distinguir qual programa se encontra em execução;
- Assim, um processo pode ser entendido inicialmente como um programa em execução, só que seu conceito é mais abrangente.



Processos em Execução

- Para verificar os processos em execução:

```
$ ps aux
```

```
$ ps -ef
```

```
$ ps -f -u user1,user2
```

```
$ ps -f --pid id
```

```
$ top
```

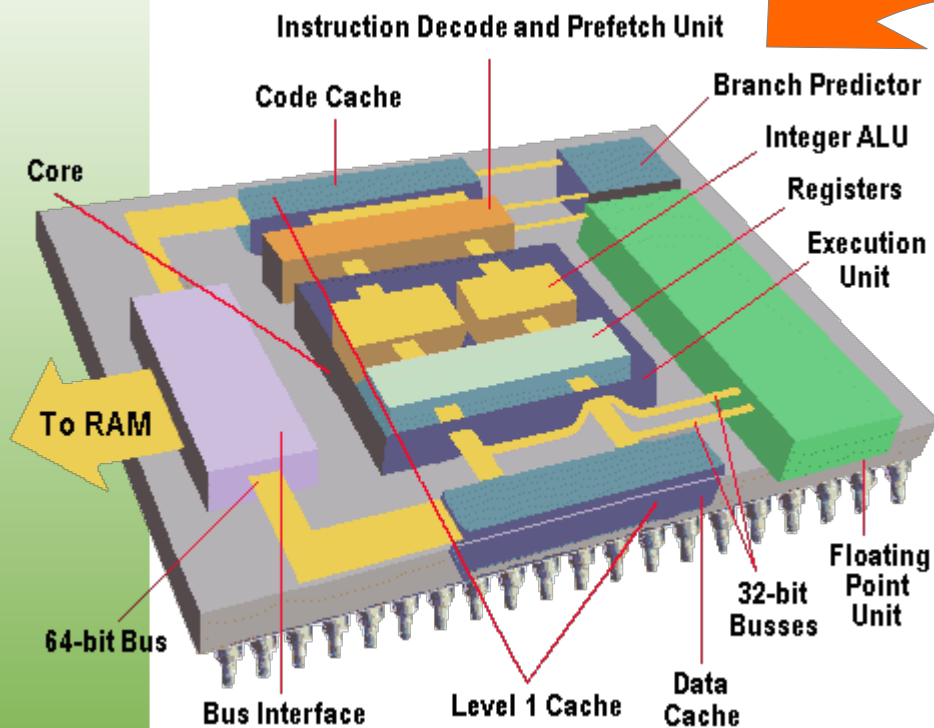
```
$ pstree
```



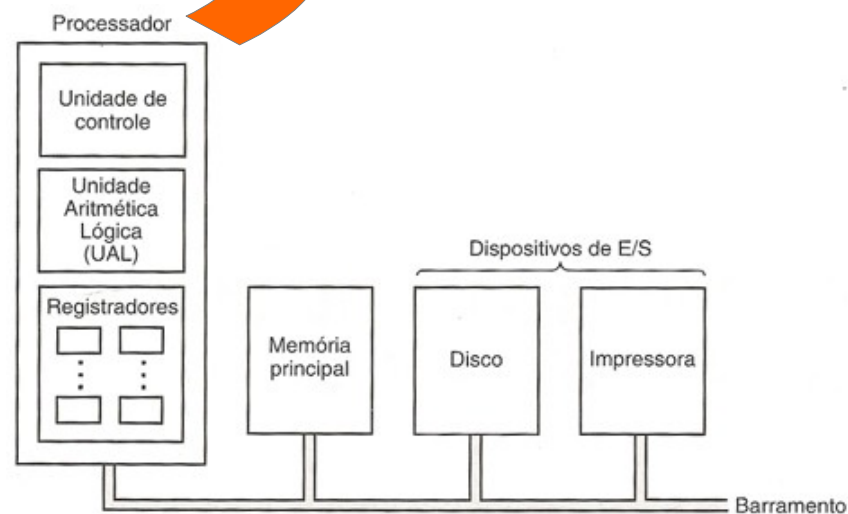
Contexto de Hardware

- armazena o conteúdo dos:
 - Registradores Gerais da CPU e
 - Registradores de uso específico:
 - *Program counter* (PC),
 - *Stack pointer* (SP) e
 - Registrador de Status.

Contexto de Hardware



(ERDEI, 2017)



(UNIVERSIDADE FEDERAL FLUMINENSE, 2017)

Contexto de Hardware - Registradores

(PANATTA *et al.*, 2017)

register encoding	zero-extended for 32-bit operands	not modified for 16-bit operands	not modified for 8-bit operands	low 8-bit	16-bit	32-bit	64-bit
0			AH*	AL	AX	EAX	RAX
3			BH*	BL	BX	EBX	RBX
1			CH*	CL	CX	ECX	RCX
2			DH*	DL	DX	EDX	RDX
6				SIL**	SI	ESI	RSI
7				DIL**	DI	EDI	RDI
5				BPL**	BP	EBP	RBP
4				SPL**	SP	ESP	RSP
8				R8B	R8W	R8D	R8
9				R9B	R9W	R9D	R9
10				R10B	R10W	R10D	R10
11				R11B	R11W	R11D	R11
12				R12B	R12W	R12D	R12
13				R13B	R13W	R13D	R13
14				R14B	R14W	R14D	R14
15				R15B	R15W	R15D	R15

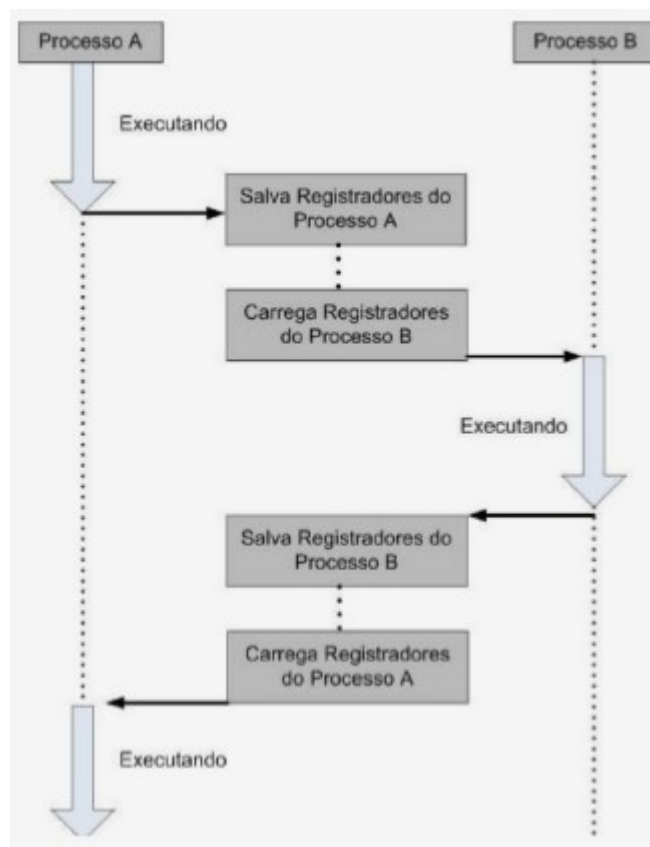
63	32	31	16	15	8	7	0
0							
63	32	31	0				

RFLAGS
RIP

* Not addressable when a REX prefix is used.
** Only addressable when a REX prefix is used.

Contexto de Hardware

- A troca de um processo por outro no processador, comandada pelo sistema operacional, é denominada de mudança de contexto;
- A mudança de contexto consiste em salvar o conteúdo dos registradores do processo que está deixando a CPU e carregá-los com os valores referentes ao novo processo que será executado.





Contexto de Software

- No contexto de software são especificadas características e limites 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.
- Muitas dessas características são determinadas no momento da criação do processo, enquanto outras podem ser alteradas durante sua existência.
- O contexto de software é composto por três grupos de informações sobre o processo: identificação, quotas e privilégios.

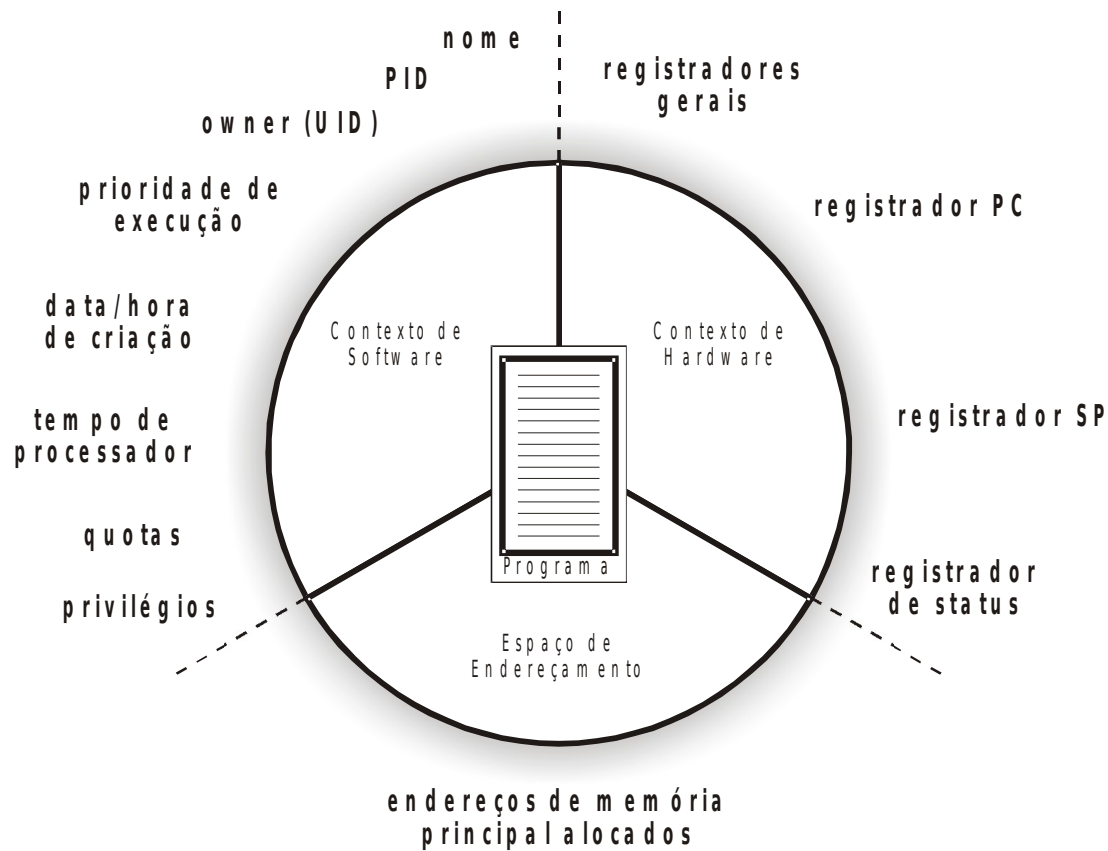


Espaço de Endereçamento

- O espaço de endereçamento é a área de memória pertencente ao processo onde as instruções e os dados do programa são armazenados para execução.
- Cada processo possui seu próprio espaço de endereçamento, que deve ser protegido do acesso dos demais processos.



Estrutura do Processo



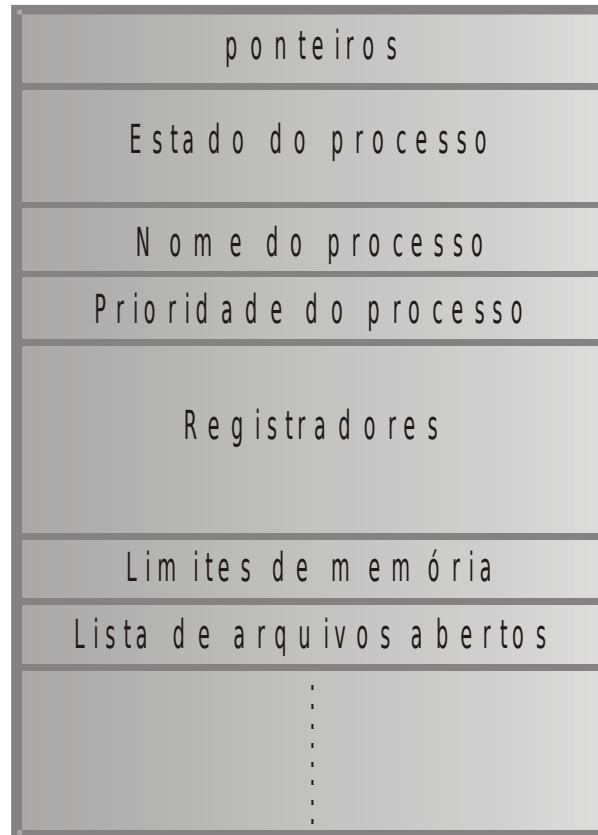


PCB

- Bloco de Controle do Processo (PCB – *Process Control Block*):
 - Estrutura utilizada para armazenar na memória principal as informações sobre os processos em execução (contexto de *hardware*, contexto de *software* e espaço de endereçamento).



PCB





PCB Algumas Informações Visíveis

- Podemos visualizar algumas informações do PCB utilizando os comandos:

```
$ top
```

```
$ ps lax
```

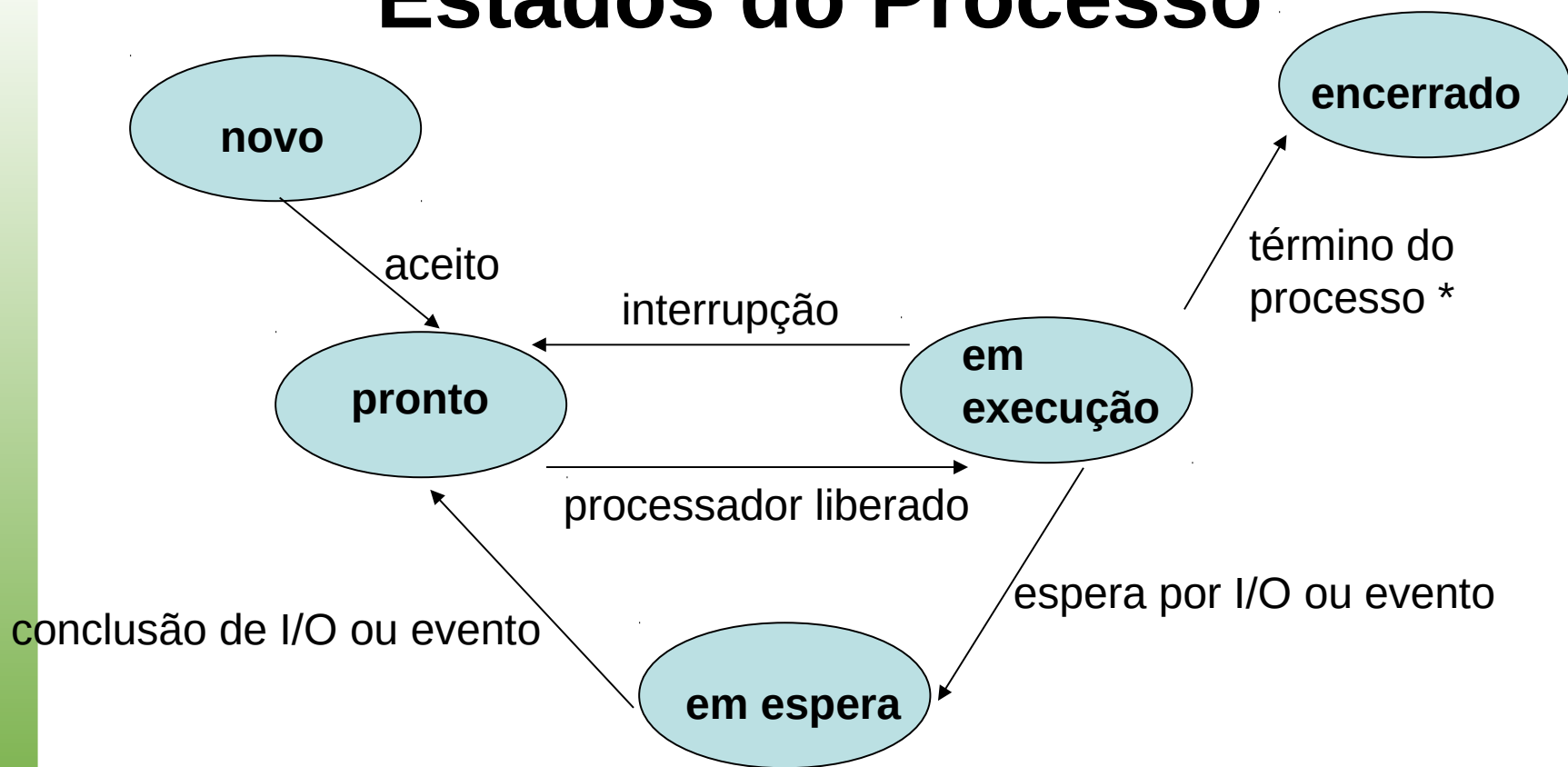
```
$ ps aux
```



Estados do Processo

- Novo (*new*): O processo está sendo criado (BCP criado, porém ainda não pode ser executado);
- Execução (*running*): As instruções estão sendo executadas:
 - Máximo de um processo em execução por processador em um dado instante;
- Pronto (*ready*): Esperando para ser atribuído a um processador;
- Espera (*wait*): Esperando a ocorrência de algum evento (conclusão de operação de I/O, por exemplo);
- Encerrado (*exit*): Terminou a execução (recursos ainda não foram liberados e BCP ainda existe).

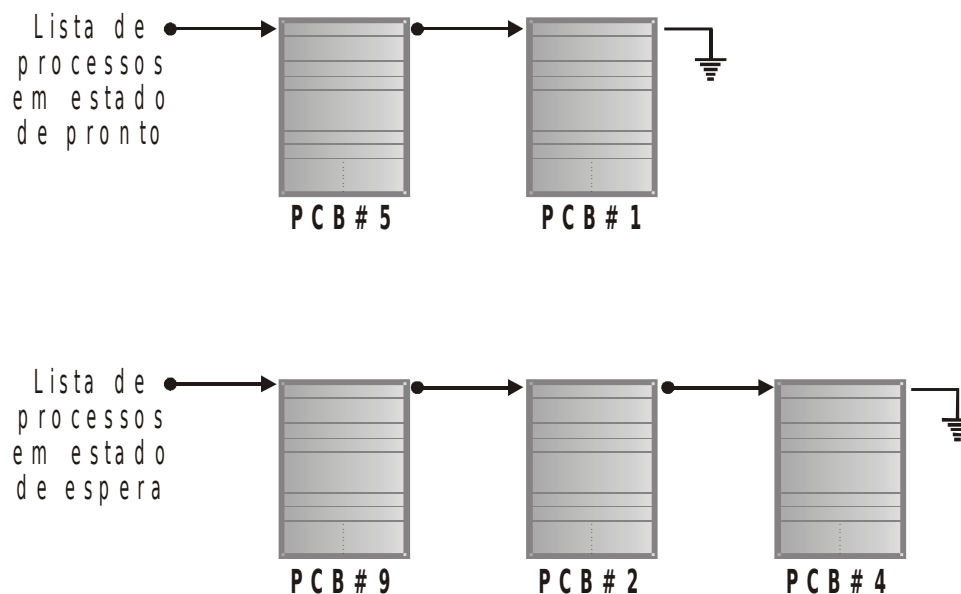
Estados do Processo



* O término do processo, na maioria das vezes, ocorre devido a: término das instruções a serem executadas, eliminação por ausência de recursos.

Estados do Processo

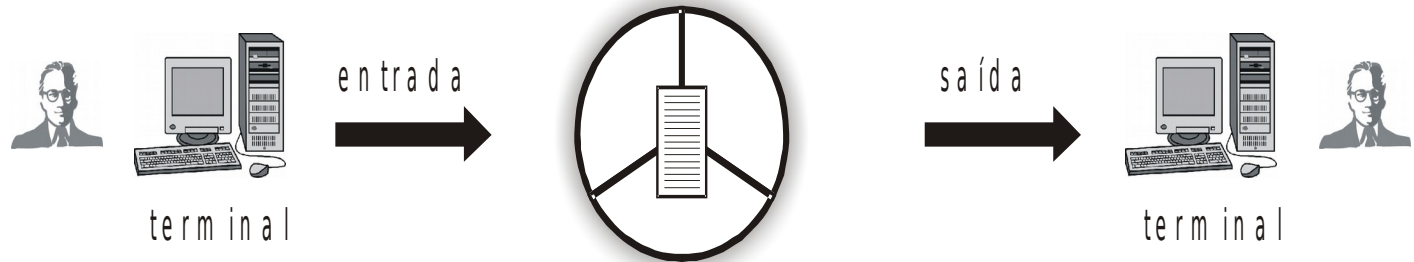
- Para cada estado do processo existe uma fila de espera. Cada uma dessas filas é gerenciada pelo escalonador.



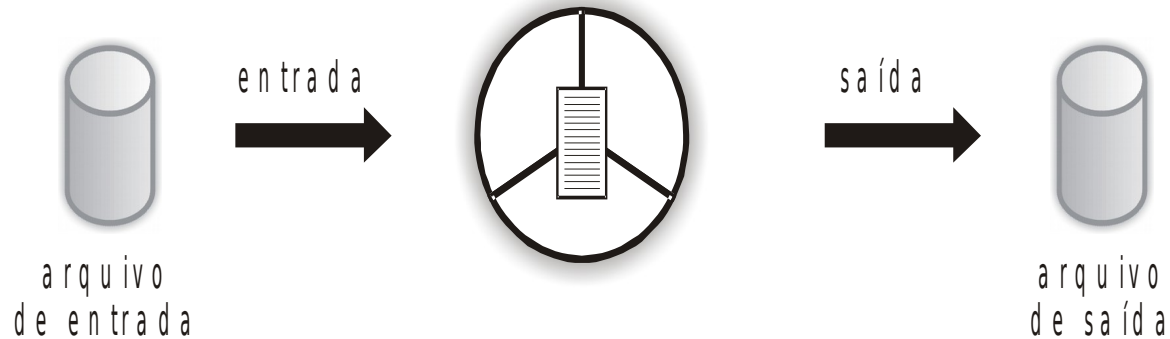


Background X Foreground

(a) Processo Foreground



(b) Processo Background





Background X Foreground

- O comando a seguir executa o ping 120 vezes na interface de loopback. E todas as saídas são direcionadas para o arquivo `/dev/null`.

```
ping -c 120 127.0.0.1 >/dev/null (executando em foreground)
^Z
[1]+ Parado ping -c 120 127.0.0.1 >/dev/null
$ jobs
[1]+ Parado ping -c 120 127.0.0.1 >/dev/null
$ bg %1
[1]+ ping -c 120 127.0.0.1 >/dev/null &
$ jobs
[1]+ Executando ping -c 120 127.0.0.1 >/dev/null &
(executando em background)
$
```



CPU-Bound X I/O-Bound

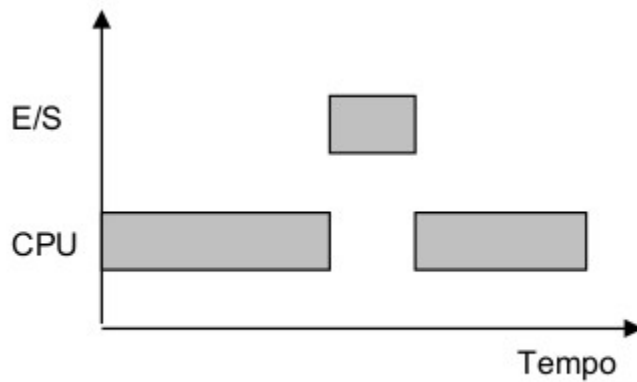


Figura 5.12: Processo CPU-Bound

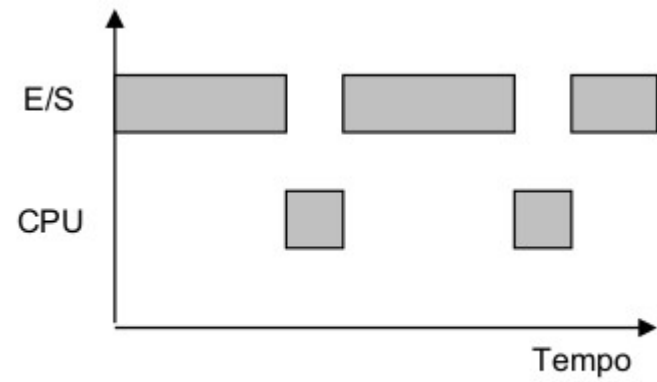


Figura 5.13: Processo I/O-Bound



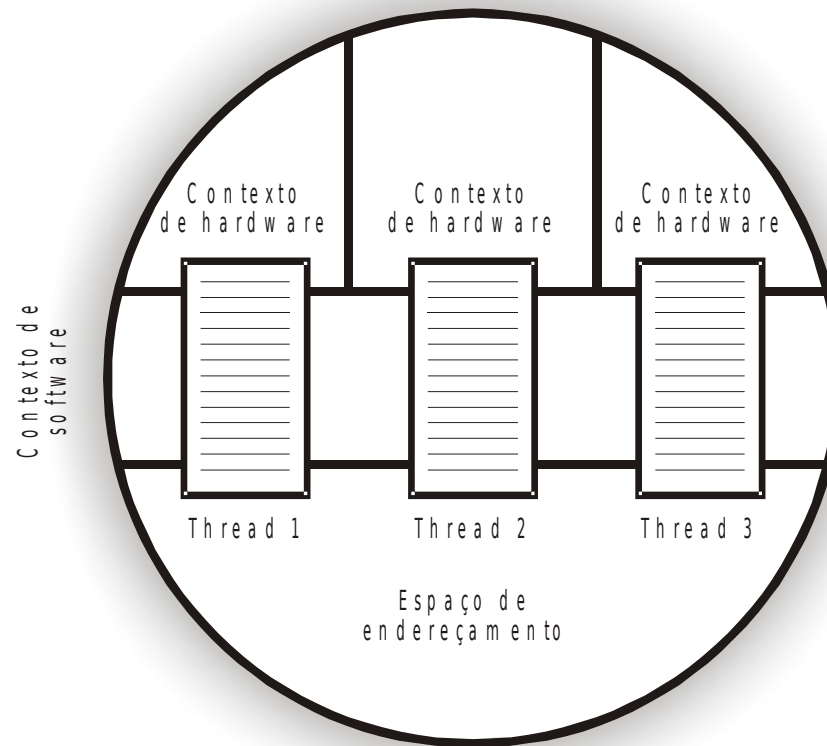
Ambiente Mono e Multithread

- Os sistemas que suportam apenas uma única *thread* de execução, são chamados de *monothread*. E aqueles que suportam múltiplas *threads*, são chamados de *multithread*;
- **Exemplos de sistemas *Monothreads*:**
 - MS-DOS e primeiras versões do Windows.





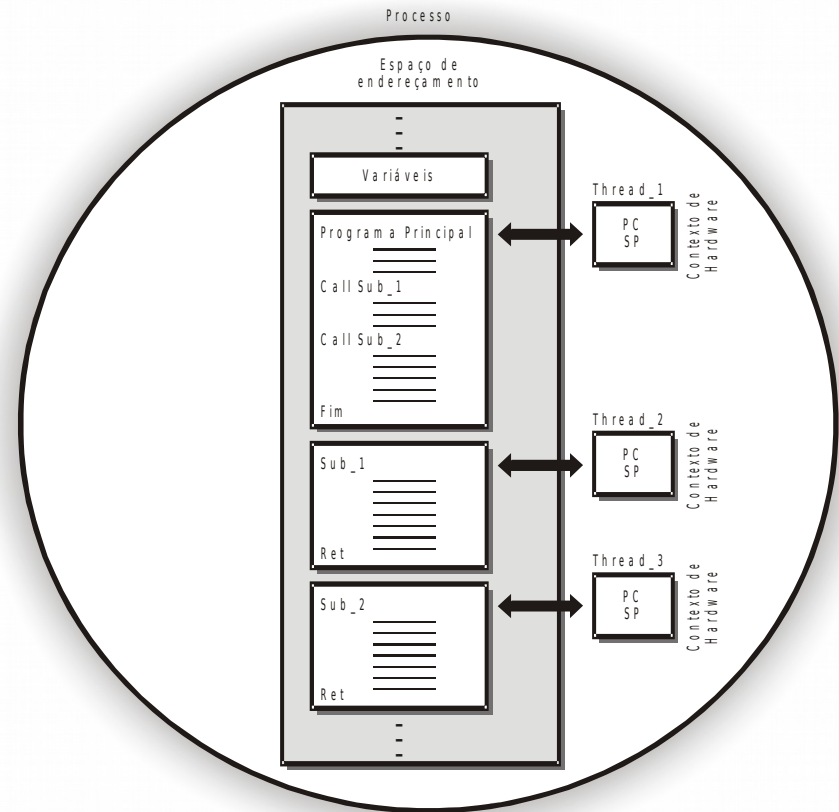
Ambiente Multithread





Ambiente Multithread

- Cada processo possui pelo menos uma *thread*
- Exemplo: programa com duas sub-rotinas independentes.





Ambiente Multithread

- Processo fica responsável pela alocação de recursos, mas a unidade escalonada é a *thread*;
- *Threads* de um mesmo processo compartilham contexto de *software* e espaço de endereçamento, mas não compartilham contexto de *hardware*;
- TCB: armazena contexto de *hardware* e informações sobre a *thread* (prioridade, estado...);
- O TCB está para a *thread* assim como o PCB está para o processo.



Referências

- ERDEI, Jacob. **Basic Structure for a Pentium Microprocessor**. Disponível em: <<https://www.pctechguide.com/cpu-architecture/basic-structure-of-a-pentium-microprocessor>>. Acesso em: 13 jul. 2017.
- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 3. ed. Rio de Janeiro: LTC, 2010.
- PANATTA, B.; SANTI, D. de; MOERSCHBACHER, G.; LIMA, L. G. de. **Esquema Geral de Funcionamento do Processador**. Disponível em: <http://sca.unioeste-foz.br/~habib/x/trabalhos/grupoa4/public_html/processador.html>. Acesso em: 14 jul. 2017.
- UNIVERSIDADE FEDERAL FLUMINENSE. **Organização de Computadores II - processadores**. Disponível em: <<http://orgcomp2.ic.uff.br/processadores.php>>. Acesso em: 14 jul. 2017.