

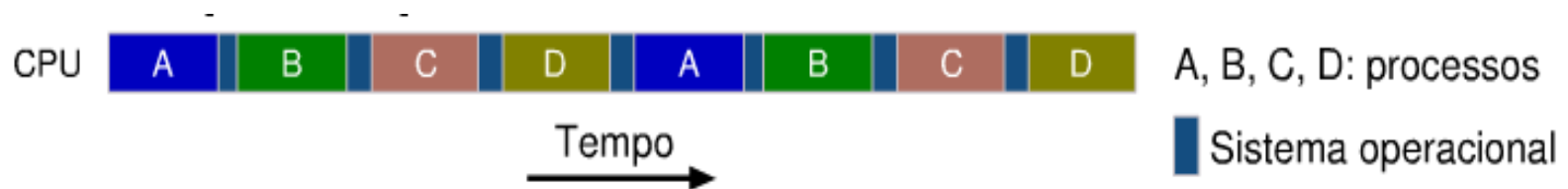
INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

Gerencia de processos

Prof. Msc Denival A. dos Santos

Pseudo paralelismo

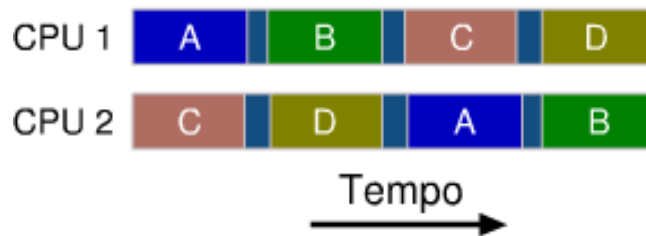
- Se o computador possui apenas um processador, então teremos um **pseudo paralelismo**.
- O termo pseudo paralelismo é portanto o termo empregado no contexto no qual uma CPU é compartilhada por diversas aplicações.
- Num instante de tempo qualquer, o processador estará executando apenas um dado programa, mas durante um intervalo de tempo maior ele poderá ter executado trechos de muitos programas criando a ilusão de paralelismo.



Neste exemplo temos um pseudoparalelismo, pois temos somente um processador (CPU) no computador, e temos quatro processos executando.

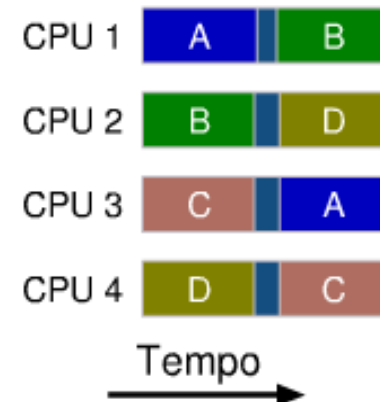
Paralelismo Real

- Teremos um **paralelismo real** quando o computador possuir mais de um processador.
- Paralelismo real só ocorre em máquinas multiprocessadas.



Acima temos algum grau de pseudoparalelismo, pois somente dois processos podem executar ao mesmo tempo, mas ao lado não temos um pseudoparalelismo, pois os quatro processos podem executar simultaneamente.

A, B, C, D: processos
Sistema operacional

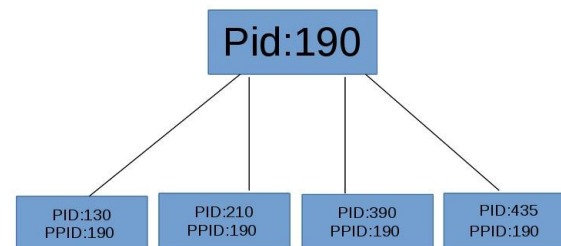
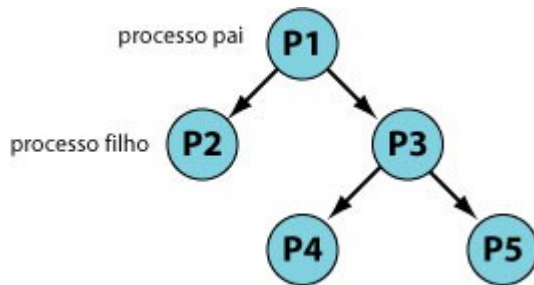


Introdução

- Um processo computacional ou simplesmente **processo** pode ser entendido como uma atividade que ocorre em meio computacional, usualmente possuindo um objetivo definido, tendo duração finita e utilizando uma quantidade limitada de recursos computacionais.
- Simplificando, podemos entender um processo como um programa em execução, o que envolve o código do programa, os dados em uso, os registradores do processador, sua pilha (stack) e o contador de programa além de outras informações relacionadas a sua execução.
- Os computadores atuais são capazes de realizar diversas tarefas ao mesmo tempo, onde cada uma destas pode representar um ou mais processos.
- A administração de vários diferentes programas em execução concomitante é o que permite o funcionamento eficiente dos computadores modernos

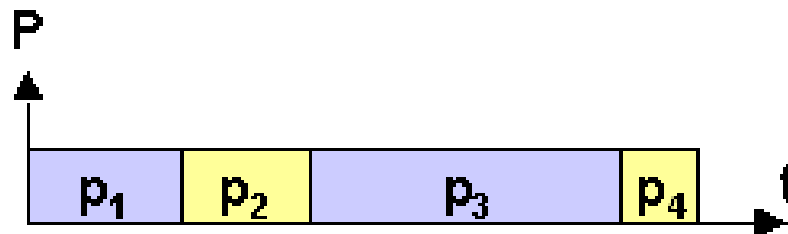
Subdivisão dos processos

- Um ponto importante é que os processos computacionais podem ser divididos em subprocessos, ou seja, podem ser decompostos em processos componentes mais simples que o processo como um todo, o que permite um detalhamento da realização de sua tarefa ou do seu modo de operação.
- Processos tipicamente também podem criar novos processos.
 - O processo criador é chamado de **processo pai** (parent process);
 - Enquanto os processos criados são denominados de **processos filhos** (child process).

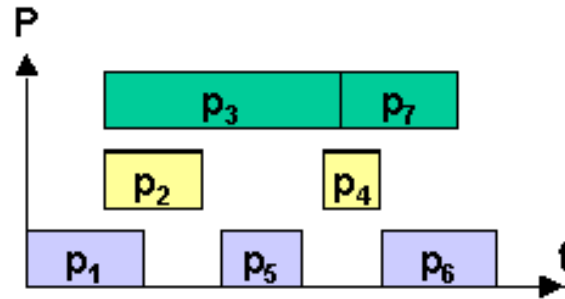


Ocorrência de processos

- Um critério muito importante de análise dos processos computacionais é aquele que considera os processos segundo sua ocorrência, isto é, a observação de seu comportamento considerando o tempo. Neste caso teríamos os seguintes tipos de processos: **sequenciais** e **paralelos**.
 - **Sequenciais**: Os processos sequenciais são aqueles que ocorrem um de cada vez no tempo, como numa série de eventos, temos que para um dado processo, todos os recursos computacionais estão disponíveis, ou seja, como só ocorre um processo de cada vez, os recursos computacionais podem ser usados livremente pelos processos, não sendo disputados entre processos diferentes, mas apenas utilizados da maneira necessária por cada processo.
 - No geral, com a execução de um único processo, temos que a ociosidade dos diversos recursos computacionais é muito alta, sugerindo que sua utilização é pouco efetiva, ou, em outros termos, inviável economicamente.



Ocorrência de processos



- **Paralelos:** Os processos paralelos são aqueles que, durante um certo intervalo de tempo, ocorrem simultaneamente.
- Se consideramos a existência de processos paralelos, então estamos admitindo a possibilidade de que dois ou mais destes processos passem, a partir de um dado momento, a disputar o uso de um recurso computacional particular.
- Considerando tal possibilidade de disputa por recursos e também sua natureza, os processos paralelos podem ser classificados nos seguintes tipos: **Independentes**, **Concorrentes** e **Cooperantes**.

Ocorrência de processos paralelos

- **Independentes** - Quando utilizam recursos completamente distintos, não se envolvendo em disputas com outros processos.
- **Concorrentes** - Quando pretendem utilizar um mesmo recurso, dependendo de uma ação do sistema operacional para definir a ordem na qual os processos usarão o recurso.
- **Cooperantes** - Quando dois ou mais processos utilizam em conjunto um mesmo recurso para completarem uma dada tarefa.
- Observações:
 - Apesar da maior complexidade, a existência de processos paralelos permitem o melhor aproveitamento dos sistemas computacionais e mais, através do suporte oferecido pelo sistema operacional passa a ser possível a exploração do processamento paralelo e da computação distribuída.

Estrutura do processo

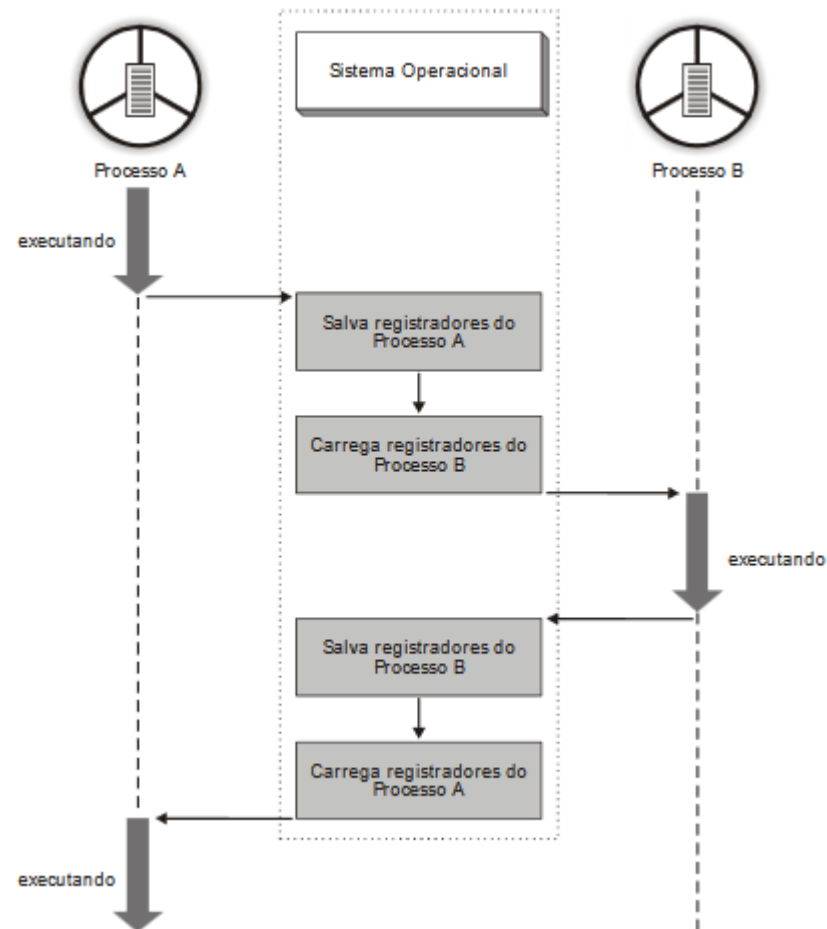
- Um processo é formado em sua essência por três partes, sendo eles: contexto de *hardware*, contexto de *software* e espaço de endereçamento. Estas três partes são responsáveis por manter todas as informações necessárias à execução de um programa.
 - **Espaço de Endereçamento:** Área da memória do processo onde o programa será executado e para dados utilizados por ele. Deve ser protegido do espaço de endereçamento dos demais processos.



Estrutura do processo

▪ Contexto de hardware

- O contexto de *hardware*, relativo a um processo, armazena informações sobre: registradores gerais do processador, registradores de uso específico como o contador de programa (PC - *Program Counter*), *Stack Pointer* (SP) e registrador de *status*.
- A troca de um processo por outro na CPU, pelo sistema operacional, é denominada mudança de contexto.
- Mudança de Contexto - salva o conteúdo dos registradores da CPU e carregá-los com os valores referente ao do processo que está ganhando a utilização do processador.



Estrutura do processo

▪ Contexto de software

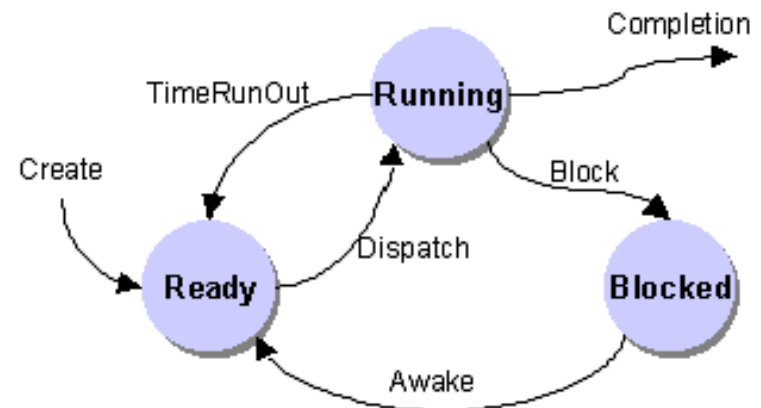
- O contexto de *software* de um processo é composto por três grupos de informações, que são: **identificação**, **quotas** e **privilégios**.
- **Identificação:** Para que um processo seja identificado junto ao sistema operacional utiliza-se um número, chamado de identificação do processo (PID - *Process Identification*). É através deste número que o sistema operacional e outros processos podem fazer referência a este processo.
- **Quotas:** Representam os limites de cada recurso existente no sistema que um processo pode alocar. São exemplos de quotas existentes nos principais sistemas operacionais usuais: número máximo de arquivos abertos simultaneamente; tamanho máximo de memória que o processo pode alocar; número máximo de operações de entrada e saída; número máximo de processos, subprocessos e *threads* que podem ser criados.
- **Privilégios:** Os privilégios definem basicamente o que um processo pode fazer frente ao sistema operacional, aos demais processos ou a ele mesmo. Estes privilégios variam baseados onde serão aplicados.

Estrutura do processo



Estados dos processos

- Quando solicitamos a execução de um programa, o sistema operacional cria (Create) um processo atribuindo a este um número de identificação ou seu PID (Process Identifier), um valor inteiro que servirá para distinguir este processo dos demais. Os três estados básicos de existência de um processo:
 - **Pronto (Ready)** - Situação em que o processo está apto a utilizar o processador quando este estiver disponível. Isto significa que o processo pode ser executado quando o processador estiver disponível.
 - **Execução (Running)** - Quando o processo está utilizando um processador para seu processamento. Neste estado o processo tem suas instruções efetivamente executadas pelo processador.
 - **Bloqueado (Blocked)** - Quando o processo está esperando ou utilizando um recurso qualquer de E/S (entrada e saída). Como o processo deverá aguardar o resultado da operação de entrada ou saída, seu processamento fica suspenso até que tal operação seja concluída.



Estados dos processos

- Num sistema em tempo repartido a entidade que coordena a utilização do processador por parte dos processos é o **escalonador (scheduler)**. O scheduler é uma função de baixo nível, que se utiliza de um temporizador (timer) do sistema para efetuar a divisão de processamento que, em última instância é uma mera divisão de tempo.
- Regularmente, a cada intervalo de tempo fixo ou variável, este temporizador dispara uma interrupção (um mecanismo especial de chamada de rotinas) que ativa uma rotina que corresponde ao escalonador do sistema.
- Devemos destacar que a transição do estado Execução (Running) para Bloqueado (Blocked) é a única causada pelo próprio processo, isto é, voluntária, enquanto as demais são causadas por agentes externos (entidades do sistema operacional). Outro ponto importante é que os estados Pronto (Ready) e Execução (Running) são estados considerados ativos enquanto que o estado Bloqueado (Blocked) é tido como inativo.

Estados dos processos

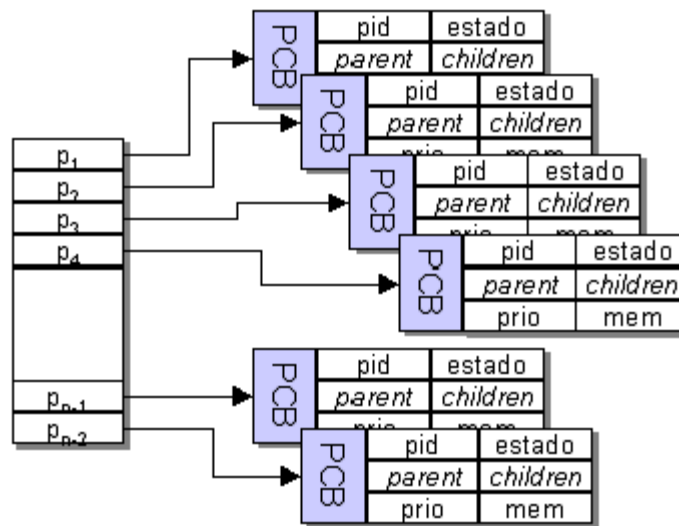
- Os processos criados podem ser de dois tipos:
 - Em primeiro plano (**foreground**): é aquele que permite a comunicação direta do usuário com o processo durante o seu processamento. O processamento interativo tem como base processos foreground.
 - Em segundo plano (**background**): é aquele onde não existe a comunicação com o usuário durante o seu processamento.
- Um **daemon** é um processo em *background* que recebe requisições de outros processos para tratar de suas tarefas específicas.
 - Daemons podem ser bastante eficientes, pois eles só funcionam quando necessário.
 - Muitos daemons são iniciados quando o sistema é ligado.
 - Alguns exemplos incluem httpd e udevd.
 - No Linux, ao usar sysvinit, scripts no diretório /etc/init.d iniciam vários daemons do sistema. Esses scripts iniciam daemons usando malabarismos do shell que são definidos no arquivo /etc/init.d/functions .

Daemon na inicialização do Linux

```
DHCPACK from 10.0.2.2
Reloading /etc/samba/smb.conf: smbd onlyNo process in pidfile '/var/run/samba/smbd.pid' found running; none killed.
.
bound to 10.0.2.15 -- renewal in 33463 seconds.
done.
Starting portmap daemon....
Starting NFS common utilities: statd.
Cleaning up temporary files....
Setting up ALSA...done.
Setting console screen modes.
Skipping font and keymap setup (handled by console-setup).
Setting up console font and keymap...done.
Setting sensors limits.
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting portmap daemon...Already running..
Starting NFS common utilities: statd.
Enabling additional executable binary formats: binfmt-support.
Starting enhanced syslogd: rsyslogd.
Starting system message bus: dbusStarting ACPI services....
.
Starting anac(h)ronistic cron: anacron.
Starting deferred execution scheduler: atd.
Starting network connection manager: NetworkManager_
```


Bloco de controle de processo - PCB

- É uma estrutura de dados que mantém a representação de um processo para o sistema operacional. O PCB contém todas as informações necessárias para a execução do mesmo possa ser iniciada, interrompida e retomada conforme determinação do sistema operacional, sem prejuízo para o processo.



Bloco de controle de processo - PCB

- Apesar de ser dependente do projeto e implementação particulares do sistema operacional, geralmente o PCB contém as seguintes informações:
 - Identificação do processo (PID)
 - Estado corrente do processo
 - Ponteiro para o processo pai (parent process)
 - Lista de ponteiros para os processos filho (child processes)
 - Prioridade do processo
 - Lista de ponteiros para as regiões alocadas de memória
 - Informações sobre horário de início, tempo utilizado do processador
 - Identificador do processador sendo utilizado
 - Permissões e direitos
 - Cópia do conteúdo dos registradores do processador
 - Entre outras

Bloco de controle de processo - PCB

Aplicativos Locais Sistema

Monitor do sistema

Monitor Editar Ver Ajuda

Sistema Processos Recursos Sistemas de arquivos

Cargas médias para os últimos 1, 5, 15 minutos: 1,10, 1,12, 1,19

Nome do processo	Estado	% CPU	Nice	ID	Memória	Aguardando canal
nm-applet	Dormindo	0	0	1760	3,5 MiB	poll_schedule_timeout
notification-area-applet	Dormindo	0	0	1840	2,0 MiB	poll_schedule_timeout
notify-osd	Dormindo	0	0	1799	2,7 MiB	poll_schedule_timeout
plugin-container	Dormindo	4	0	2515	9,3 MiB	poll_schedule_timeout
polkit-gnome-authenticati	Dormindo	0	0	1762	1,3 MiB	poll_schedule_timeout
pulseaudio	Dormindo	0	-11	1709	1,4 MiB	poll_schedule_timeout
sh	Dormindo	0	0	1752	64,0 KiB	do_wait
ssh-agent	Dormindo	0	0	1660	192,0 KiB	poll_schedule_timeout
syndaemon	Dormindo	0	0	1729	240,0 KiB	poll_schedule_timeout
trashapplet	Dormindo	0	0	1836	2,7 MiB	poll_schedule_timeout
ubuntuone-syncdaemon	Dormindo	0	0	1993	20,1 MiB	poll_schedule_timeout
unity-window-decorator	Dormindo	0	0	1753	3,7 MiB	poll_schedule_timeout
update-notifier	Dormindo	0	0	2018	2,8 MiB	poll_schedule_timeout
VBoxNetDHCP	Zumbi	0	0	2390	N/A	do_exit
VBoxSVC	Dormindo	0	0	2242	4,1 MiB	futex_wait_queue_me
VBoxXPCOMIPCD	Dormindo	0	0	2236	1,5 MiB	poll_schedule_timeout
VirtualBox	Dormindo	0	0	2441	328,3 MiB	poll_schedule_timeout
VirtualBox	Dormindo	0	0	2375	401,4 MiB	poll_schedule_timeout
VirtualBox	Dormindo	0	0	2227	40,8 MiB	poll_schedule_timeout
wnck-applet	Dormindo	0	0	1834	3,2 MiB	poll_schedule_timeout
zeitgeist-daemon	Dormindo	0	0	1770	8,0 MiB	poll_schedule_timeout

Terminar processo

[VirtualBox OSE Gere... [server_windows (20... [Rede virtual no Virtu... [XP [Executando] - Vir... Monitor do sistema