

INE5410 – Programação Concorrente

Unidade II – Fundamentos de Programação Concorrente

Prof. Frank Siqueira
frank.siqueira@ufsc.br



Conteúdo

- Processos
- Threads
- Exclusão Mútua
- Semáforos
- Deadlocks

Processos

- Definição
 - Um programa em execução em uma máquina
 - Identificado pelo seu PID (*Process Identifier*)
- Execução dos Processos
 - Um núcleo do processador executa somente um processo a cada instante
 - Se o computador possuir N núcleos, N processos podem ser executados simultaneamente
 - Em um S.O. multitarefa, processos se alternam na utilização dos núcleos; cada processo é executado durante um *quantum* de tempo

Processos

- Processo == Programa??
 - Um **programa** é uma sequência de instruções
 - É uma entidade estática
 - Um **processo** corresponde a um programa em execução no computador
 - É uma entidade dinâmica, que altera seu estado ao longo do tempo
 - Além do programa, contém uma área de dados e informações de contexto
 - Um mesmo programa pode ser executado por vários processos, em um mesmo instante ou em momentos diferentes

Processos

- Escalonamento de Processos
 - O escalonador do S.O. seleciona o(s) processo(s) que deve(m) ser executado(s) pelo(s) núcleo(s)
- Algoritmo de Escalonamento
 - Define a ordem de execução de processos com base em uma fila, prioridade, deadline, ...
 - Em geral, os sistemas adotam uma política de melhor esforço para atender a todos os processos de maneira justa e igualitária
 - Processos do sistema e aplicações críticas (um alarme, por exemplo) exigem maior prioridade

Processos

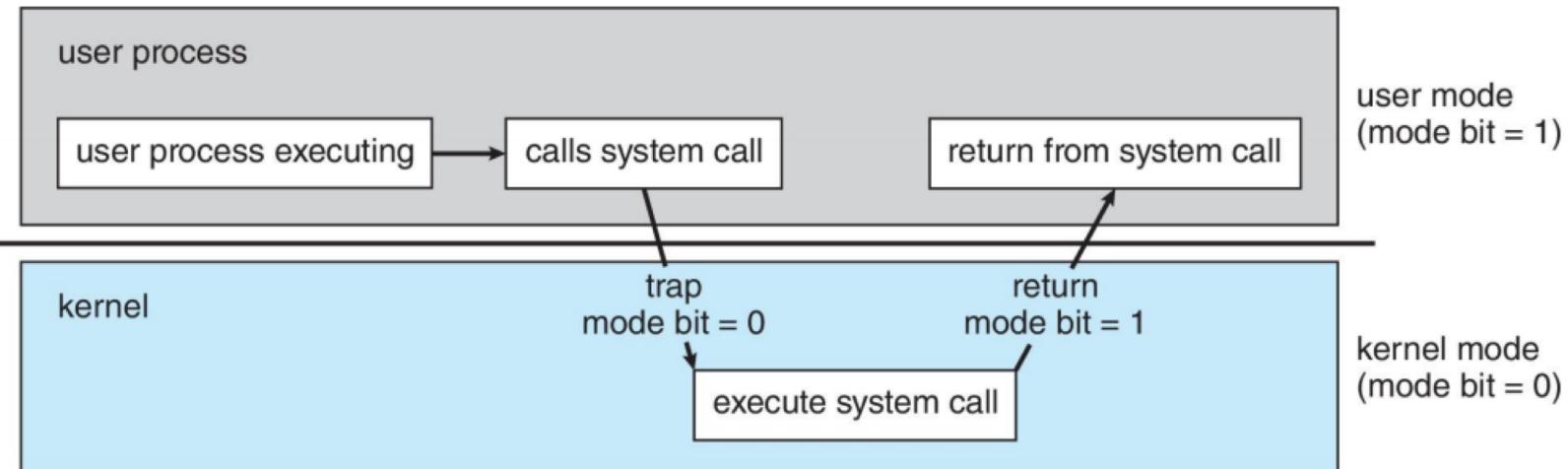
- Espaço de Endereçamento
 - É a área de memória usada pelo processo
 - Contém o programa executável, os dados do programa e informações de contexto como contador de programa, pilha de execução, registradores, alarmes pendentes, lista de arquivos e portas de comunicação abertas, etc.
 - Em geral, sistemas multitarefa protegem o espaço de endereçamento de cada processo, de modo a evitar que um processo interfira ou até mesmo ocasione falhas em outro processo

Processos

- Processos e Usuários
 - Cada processo é associado ao identificador do usuário que o criou
 - Em geral, o processo herda todas as permissões do usuário a ele associado, podendo por exemplo alterar arquivos criados pelo usuário
 - Usuários especiais podem ser criados para determinadas finalidades, como executar os serviços/*daemons* do sistema
 - Processos do usuário root (Linux) ou administrador (Windows) têm acesso irrestrito

Processos

- Modos de execução:
 - Modo usuário: execução normal do processo
 - Modo núcleo (*kernel*): acesso privilegiado
- Transições de modo ocorrem devido a:
 - Chamadas de sistema (*traps*) ou Exceções
 - Interrupções (ex.: operações de I/O)

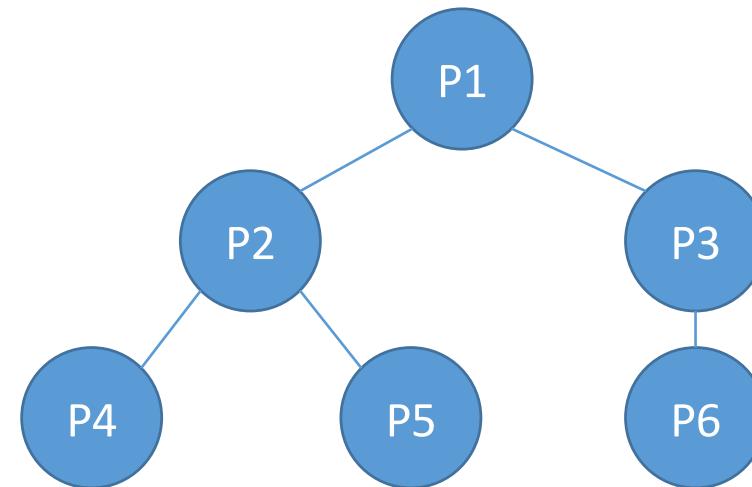


Processos

- Processos são iniciados de diversas formas:
 - Ao carregar o sistema operacional
 - Ao executar um programa
 - Ao executar um comando no terminal
 - Quando um processo explicitamente cria outro processo usando chamadas de sistema
 - Quando uma tarefa previamente agendada é executada pelo sistema

Processos

- Hierarquia de Processos
 - Quando um processo cria outro processo, é estabelecida uma relação hierárquica entre o processo pai e o processo filho
 - Um processo pai pode ter vários filhos
 - Um processo tem apenas um pai
 - Os filhos podem criar seus próprios filhos, e assim por diante
 - Com isso temos uma hierarquia de processos



Processos

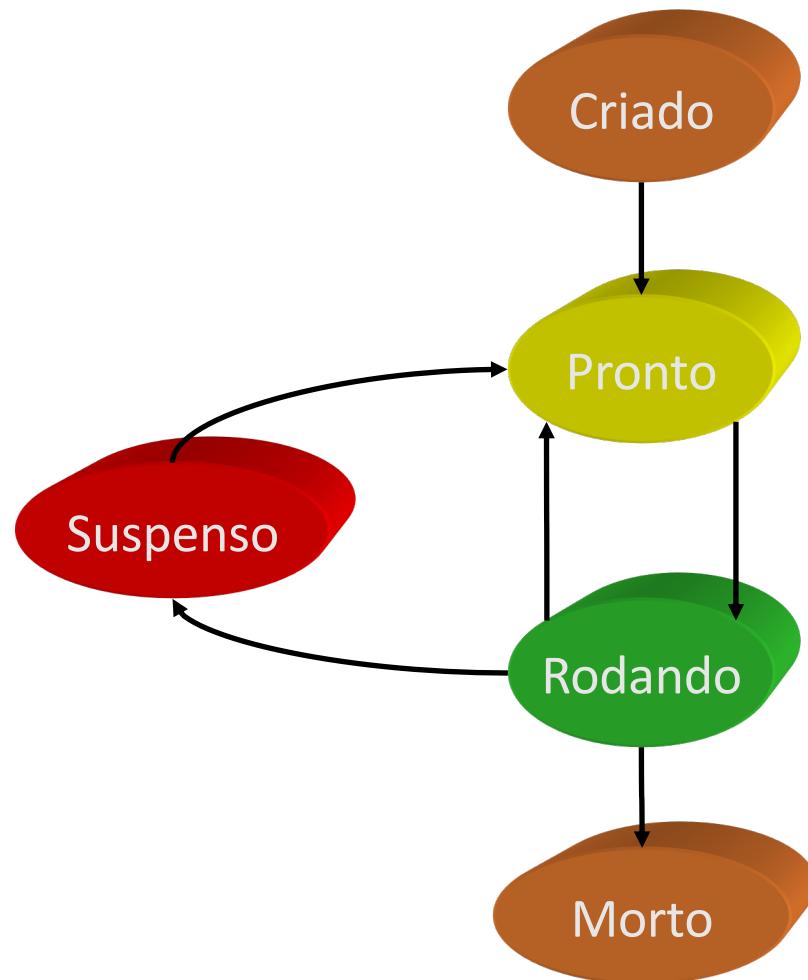
- Um Processo pode ser executado em:
 - **Foreground**: o processo é mostrado na tela, de modo que o usuário pode interagir e fornecer dados necessários para a execução do programa
Ex.: editor de texto, navegador Web, etc.
 - **Background**: executado em plano de fundo, sem precisar interagir diretamente com o usuário, podendo executar tarefas de forma autônoma
Ex.: baixar atualizações para serem instaladas, verificar se há novos e-mails, atualizar arquivos em uma pasta compartilhada, etc.

Processos

- Término do Processo
 - Voluntário: por ação do próprio processo
 - Saída normal: programa conclui a execução e encerra o processo
 - Saída por erro: o programa identifica um erro e encerra a execução
 - Involuntário: por uma ação externa ao processo
 - Erro fatal: erro causado pelo processo, como uma instrução ilegal, divisão por zero, referência a um endereço de memória inexistente ou nulo, etc.
 - Cancelamento: o processo é encerrado por solicitação de outro processo do mesmo usuário ou de usuário privilegiado (*root/administrador*)

Processos

- Ciclo de vida simplificado de um processo



Processos

- Estados de um Processo
 - **Pronto**: processo pronto para ser executado, mas sem o direito de usar o processador
 - **Rodando**: sendo executado pelo processador
 - **Suspenso**: aguarda operação de I/O, liberação de um recurso ou fim de tempo de espera
- Processos trocam de estado de acordo com:
 - Algoritmo de escalonamento
 - Troca de mensagens
 - Interrupções de hardware ou software

Processos

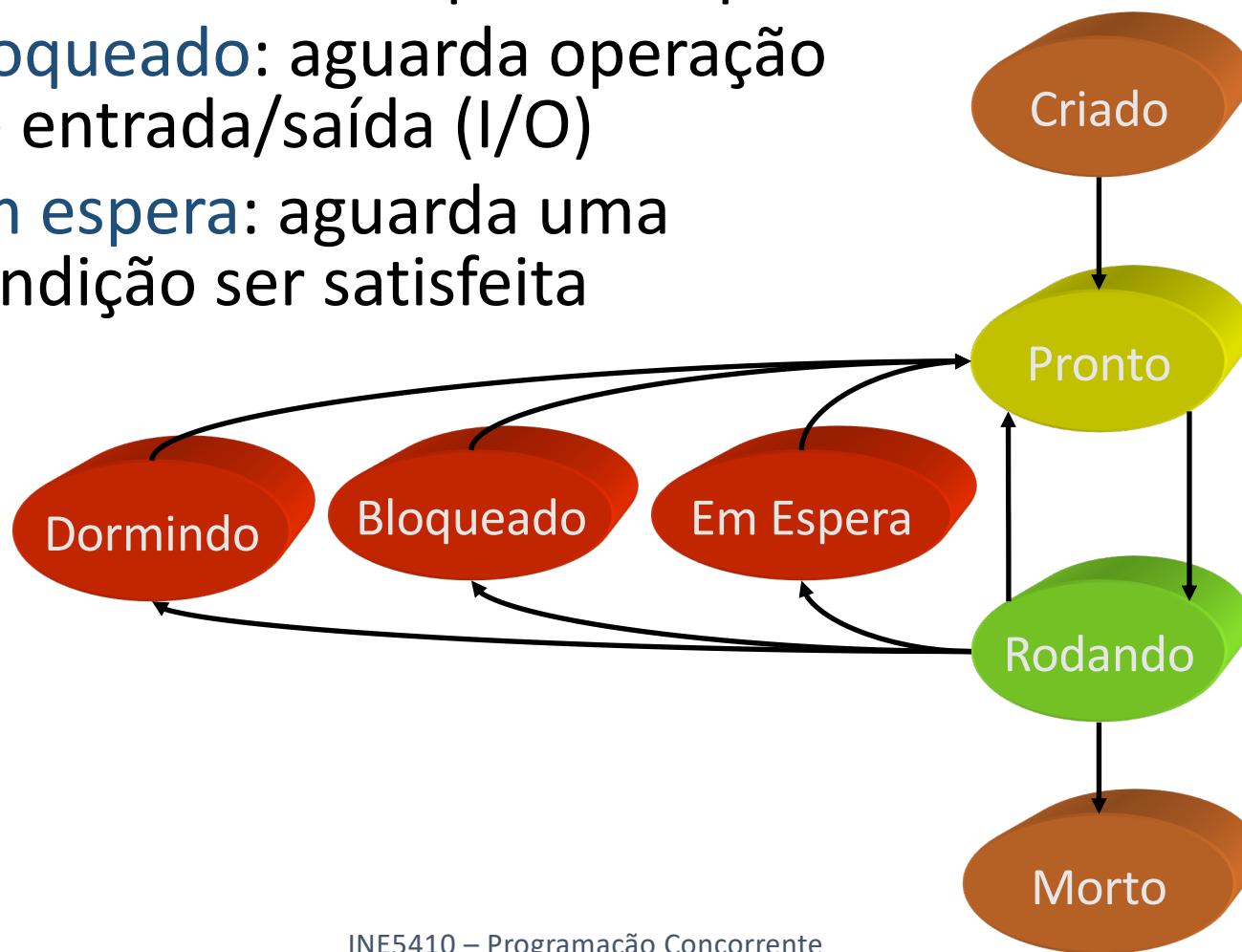
- Troca de Contexto / Preempção
 - O processo em execução é suspenso, e um outro processo passa a ser executado
 - Ocorre por determinação do escalonador ou quando o processo que estava sendo executado é suspenso
 - O contexto do processo suspenso deve ser salvo para retomar a execução posteriormente

Processos

- O contexto de um processo compreende:
 - O estado do processo
 - Informações para escalonamento
 - Dados para contabilização de uso
 - Um segmento de código
 - Um segmento de dados
 - Os valores dos registradores
 - O contador de programa
 - Uma pilha de execução
 - Arquivos, portas e outros recursos alocados

Processos

- É possível distinguir o motivo da suspensão
 - **Dormindo**: em espera temporizada
 - **Bloqueado**: aguarda operação de entrada/saída (I/O)
 - **Em espera**: aguarda uma condição ser satisfeita



Processos

- Chamadas do Sistema Operacional Linux
 - Criar um Processo:
 - `fork()` cria uma cópia do processo que a executa, alocando um novo espaço de endereçamento
 - Retorno: PID do filho para o pai; valor 0 para o filho; valor negativo se houve erro na criação do processo

```
...
int pid = fork();           // cria cópia do processo
if (pid > 0) {              // é o processo pai
    ...
} else if (pid == 0) {       // é o processo filho
    ...
} else {                     // houve erro na criação do processo
    ...
}
...
...
```

Processos

- Chamadas do Sistema Operacional Linux
 - Trocar o código binário de um processo: `exec()`, `execle()`, `execlp()`, `execv()`, `execve()`, `execvp()`
 - Obter PID do processo: `getpid()`
 - Obter PID do processo pai: `getppid()`
 - Suspender a execução: `sleep(<tempo>)`
 - Aguardar processo filho: `wait(<status>)`
 - Reiniciar processo suspenso: `kill(<pid>,SIGWAIT)`
 - Finalizar o processo: `exit(<retorno>)`
 - Destruir um processo: `kill(<pid>,SIGKILL)`

Processos

- Chamadas do Sistema Operacional Linux
 - Exemplo: uso de getpid(), fork(), wait() e execl()

```
...
printf("PID do pai: %d\n", getpid()); // Processo pai imprime seu PID
int pid = fork(); // Cria processo filho
if ( pid > 0 ) { // Se é o pai e o filho foi criado
    int status; // Variável para status do filho
    wait(&status); // Pai aguarda o filho terminar
    printf("Filho encerrou com status: %d\n", status);
} else if ( pid == 0 ) { // É o processo filho
    printf("PID do filho: %d\n", getpid()); // Filho imprime seu PID
    execl( "/bin/ls", "ls", "-l", NULL ); // Substitui o código do filho
} else { // Ocorreu erro ao criar o filho
    printf("Erro %d na criação do processo filho\n", pid);
}
...
```

Processos

- Gerenciamento de Processos no Linux
 - O sistema operacional mantém uma tabela com informações sobre cada processo em execução
 - A tabela de processos armazena o estado de cada processo e informações de contexto

Gerenciamento de processos	Gerenciamento de memória	Gerenciamento de arquivos
Registradores	Ponteiro para o segmento de texto	Diretório raiz
Contador de programa (PC)	Ponteiro para o segmento de dados	Diretório de trabalho
Ponteiro de pilha		Descritores de arquivo
Estado do processo		
Tempo em que o processo iniciou		
Tempo de CPU utilizado		
Tempo de CPU dos filhos		
ID do processo (PID)		

Processos

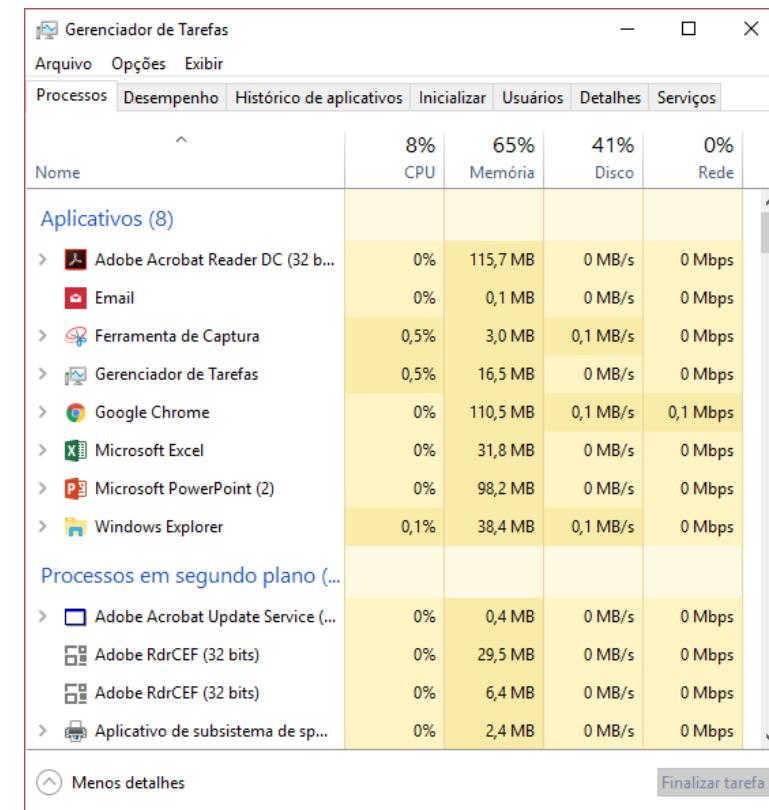
- Gerenciamento de Processos no Linux
 - Processos são criados pelo usuário através da interface gráfica ou de comandos no terminal
 - Comandos bloqueiam o terminal; para rodar em background devem ser seguidos de um &
 - Os processos em execução são listados com o comando `ps`
 - O comando `top` lista os processos que mais consomem recursos
 - Comando `pstree` mostra hierarquia de processos
 - Processos são destruídos com `kill -9 <pid>`

Processos

- Chamadas da API do Windows
 - Criar um Processo:
`CreateProcess(<nome>, <comando>, ...)` ou
`CreateProcessAsUser(<usuário>, <nome>, ...)`
 - Obter o Identificador do Processo:
`GetCurrentProcessId()`
 - Suspender a Execução: `Sleep(<tempo>)`
 - Finalizar o Processo: `ExitProcess(<retorno>)`
 - Destruir um Processo: `TerminateProcess(<pid>, <retorno>)`

Processos

- Gerenciamento de Processos no Windows
 - Processos são criados pelo usuário através do ambiente gráfico ou de comandos no *Prompt*
 - O *Prompt* de Comando não bloqueia quando é aberta uma aplicação Windows
 - Processos são listados e destruídos usando o Gerenciador de Tarefas ou executando os comandos *tasklist* e *taskkill* no *Prompt*



The screenshot shows the Windows Task Manager window titled "Gerenciador de Tarefas". The window has a tab bar with "Processos" selected, followed by "Desempenho", "Histórico de aplicativos", "Inicializar", "Usuários", "Detalhes", and "Serviços". Below the tabs is a table with columns: Nome, 8% CPU, 65% Memória, 41% Disco, and 0% Rede. The table lists several processes under the heading "Aplicativos (8)".

Nome	8% CPU	65% Memória	41% Disco	0% Rede
Adobe Acrobat Reader DC (32 b...)	0%	115,7 MB	0 MB/s	0 Mbps
Email	0%	0,1 MB	0 MB/s	0 Mbps
Ferramenta de Captura	0,5%	3,0 MB	0,1 MB/s	0 Mbps
Gerenciador de Tarefas	0,5%	16,5 MB	0 MB/s	0 Mbps
Google Chrome	0%	110,5 MB	0,1 MB/s	0,1 Mbps
Microsoft Excel	0%	31,8 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (2)	0%	98,2 MB	0 MB/s	0 Mbps
Windows Explorer	0,1%	38,4 MB	0,1 MB/s	0 Mbps
Processos em segundo plano (...)				
Adobe Acrobat Update Service (...)	0%	0,4 MB	0 MB/s	0 Mbps
Adobe RdrCEF (32 bits)	0%	29,5 MB	0 MB/s	0 Mbps
Adobe RdrCEF (32 bits)	0%	6,4 MB	0 MB/s	0 Mbps
Aplicativo de subsistema de sp...	0%	2,4 MB	0 MB/s	0 Mbps

At the bottom left is a button labeled "Menos detalhes" and at the bottom right is a button labeled "Finalizar tarefa".