



4 PROCESSOS

Os primeiros sistemas operacionais permitiam que apenas um processo fosse executado por vez. Dessa maneira, este processo tinha todo o sistema computacional a sua disposição. Os atuais sistemas operacionais permitem que vários processos sejam carregados na memória e executados de forma concorrente pelo processador.

Esta evolução visava a maior eficiência do sistema computacional e a diminuição de desperdício de recursos do mesmo. Embora tenha sido bem sucedido em alcançar essas metas, a multitarefa trouxe uma série de complicações para a implementação do sistema operacional, cujo projeto tornou-se bastante complexo. Via de regra, quanto mais complexa for a estrutura de um sistema operacional, maior será a quantidade de serviços oferecidos por ele.

A concorrência por recursos, o que inclui o próprio processador, fez com que fossem tomadas um conjunto de mecanismos para controle do compartilhamento dos recursos. Os recursos são compartilhados por processos, os quais são programas em execução.

4.1 CONCEITO DE PROCESSO

Mesmo em sistemas monousuários é possível a execução de várias tarefas em paralelo. Até quando o não nenhum programa de usuário sendo executado há processos sendo executados em paralelo.

EXEMPLO:

- **Processo para atualização da hora**
- **Processo para monitoramento da necessidade de proteção de tela**

Figura 4-1 - Exemplo

É comum que os termos *job* e processo sejam usados para a mesma finalidade. Atualmente, é mais comum o uso do termo processo, pois *job* foi muito utilizado em sistemas *batch*.

4.1.1 O PROCESSO

De forma geral, **um processo é um programa em execução**. O código do programa nada mais é do que uma seção de texto. Para ele se tornar um processo é preciso acrescentar o apontador de programas, conteúdo de registradores, área de



variáveis globais e uma pilha com dados temporários como, por exemplo, parâmetros de métodos ou funções, endereços de retorno e variáveis locais.

Pode-se definir um programa como uma entidade passiva e o processo uma entidade ativa. Um outro ponto importante a ser firmado é o fato de que a execução de duas cópias do mesmo programa gera dois processos.

4.1.2 ESTADOS DE UM PROCESSO

Um processo pode assumir uma série de estados entre o início e o fim de sua execução. A transição não é seqüencial, embora tenha uma ordem básica, pois ela pode assumir ciclos de estados. A existência e a duração de ciclos de execução depende do tempo de execução, dos recursos que ele irá utilizar e dos processos que estão em execução juntamente com ele. Os estados de execução de um processo são:

- ❑ **Novo** – o processo está sendo criado;
- ❑ **Em execução** (*Run*) – as instruções estão sendo executadas;
- ❑ **Em espera** (*Blocked*) – o processo está bloqueado esperando a ocorrência de um evento para poder voltar a ser executado. Muitas vezes ele fica esperando o término de uma operação de I/O ou um sinal de algum processo;
- ❑ **Pronto** (*Ready*) – o processo está aguardando ser selecionado para uso do processador;
- ❑ **Encerrado** – o programa terminou sua execução.

Estas terminologias para estes estados variam de acordo com a bibliografia. A Figura 4.2 ilustra o diagrama de transição de estados de um processo.

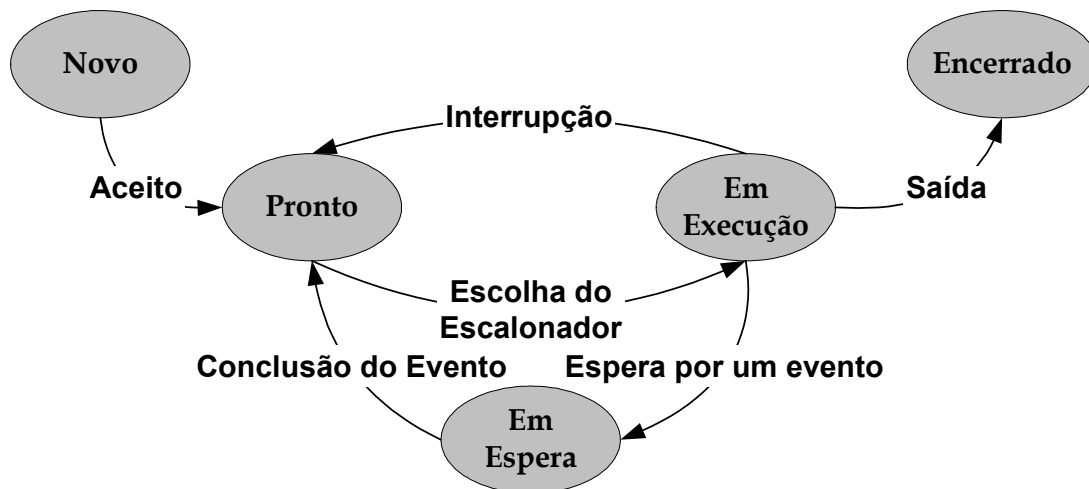


Figura 4-2 - Diagrama de Transição de Estados de um Processo



4.1.3 BLOCO DE CONTROLE DE PROCESSO

Cada processo é representado no sistema operacional por um **bloco de controle de processo** (PCB - *Process Control Block*). Ele contém uma série de informações, das quais destacam-se e são ilustradas na Figura 4.3:

- ❑ **Estado do Processo** – informa o estado atual do processo;
- ❑ **Contador de Programa** – indica o endereço da próxima instrução a ser executada para este processo;
- ❑ **Registradores de CPU** – informa os registradores a serem utilizados pelo sistema. A quantidade e categorias de registradores variam de acordo com a arquitetura do computador em uso. As informações contidas nos registradores devem ser salvas quando ocorrer uma interrupção;
- ❑ **Informações de Escalonamento de CPU** – indica a prioridade do processo, ponteiros para filas de escalonamento e demais informações do algoritmo de escalonamento;
- ❑ **Informações de Gerência de Memória** – essas informações incluem dados de valores dos registradores de base e limite, tabelas de páginas e de segmentos, dependendo da estrutura de memória utilizada;
- ❑ **Informações de Contabilização** – são alocadas informações sobre a capacidade da CPU e tempo real utilizados, número de *jobs*;
- ❑ **Informações de Status de I/O** – são os dispositivos alocados para o processo.

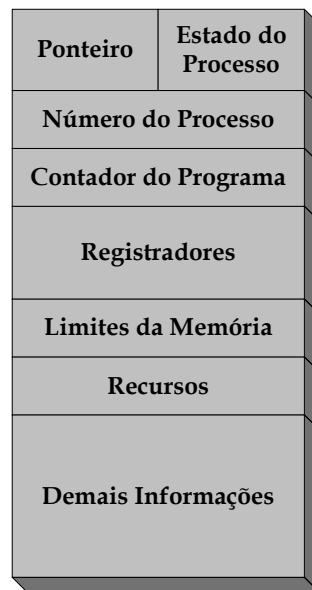


Figura 4-3 - Bloco de Controle de Processo



4.1.4 THREADS

É comum a discussão de processos que possuem um único fluxo de execução, o que possibilita o processo executar apenas uma tarefa por vez. A título de exemplificação, se um programa de processamento de textos está sendo utilizado pelo usuário, onde ele está digitando um texto, o corretor ortográfico não poderá ser utilizado, pois é um outro fluxo de controle.

Entretanto, como é sabido, é possível usar o corretor ortográfico em paralelo a edição do texto. Em outras palavras, é possível a execução de vários fluxos do programa, os quais são mais comumente chamados de *threads*.

4.2 ESCALONAMENTO DE PROCESSOS

O objetivo da multiprogramação é ter processos em execução o tempo todo, visando maximizar o uso do processador. Para isto o tempo de uso do processador é compartilhado entre todos os processos em execução. A viabilidade está no fato de que nenhum processo usa os mesmos recursos ao mesmo tempo, ou mesmo que algumas operações de I/O demoradas deixam o processador ocioso.

4.2.1 FILAS DE ESCALONAMENTO

Os processos que estão aptos a serem executados são alocados em uma fila que é denominada **Fila de Jobs**, a qual contém todos os processos do sistema que estão em execução. Esta fila possui divisões de acordo com os possíveis estados de execução de um processo (Ver Figura 4.2). Logicamente não há filas para os estados: Novo, Terminado e Em Execução.

Se o tempo de uso do processador pelo processo terminou, ele deve ser alocado na fila de processos prontos e esperar ser selecionado novamente. Já quando ele está bloqueado esperando um evento, ele fica na fila de processos Em Espera, até o evento ocorrer e, então, ele é realocado para a Fila de Prontos.

Além dessas filas, existem aquelas para espera para uso de dispositivos. Primeiro eles são alocados na Fila de Processos Em Espera e é identificado o dispositivo requerido. Em seguida, uma chamada de sistema é feita para indicar que o processo está na fila do referido dispositivo, caso um outro processo já esteja usando o dispositivo. O esquema básico das filas de escalonamento de processos é ilustrado na Figura 4.4.

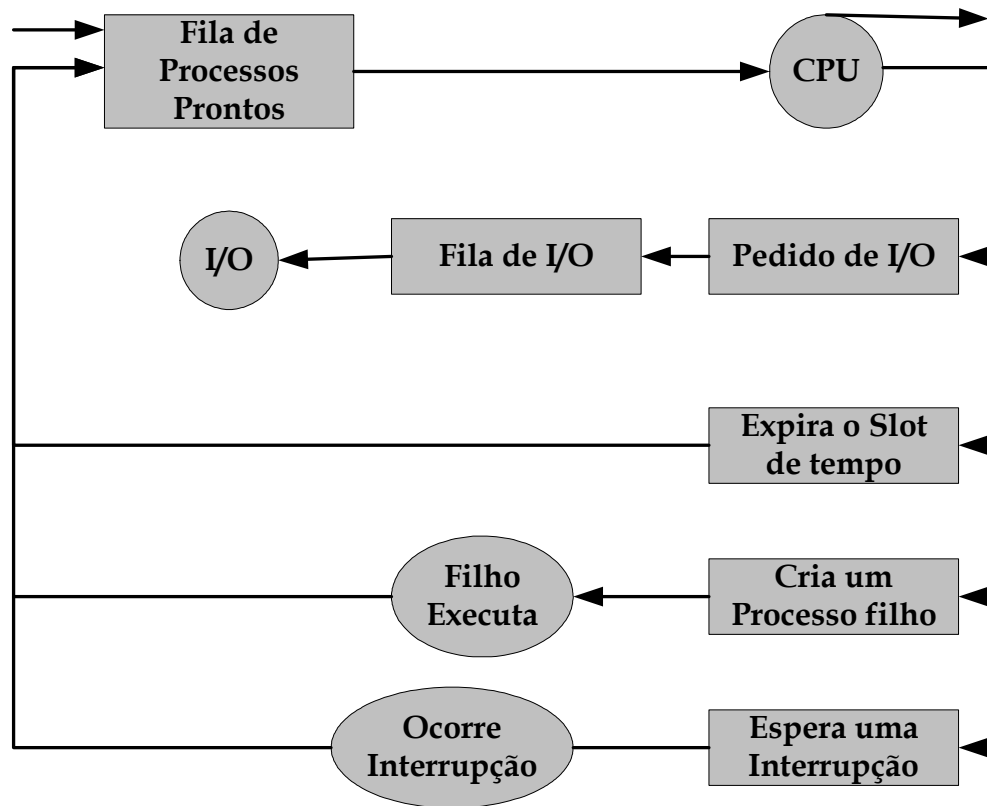


Figura 4-4 - Diagrama de Filas de Escalonamento

Cada caixa retangular representa uma fila com duas categorias: Processos Prontos e de Dispositivos. Os círculos são os recursos e as elipses eventos. As setas representam os fluxos. Todo novo processo é alocado na Fila de Processos Prontos, onde aguarda ser selecionado para execução. Podem ocorrer os seguintes eventos:

- ☐ O processo emite uma requisição de uso de um dispositivo de I/O e é alocado na fila do mesmo;
- ☐ O processo pode criar um outro filho, chamado processo filho, e espera o seu término;
- ☐ O processo pode ser removido a força da CPU como resultado de uma interrupção e ser alocado na fila de processos prontos;

4.2.2 ESCALONADORES

A Figura 4.4 ilustrou que um processo pode migrar entre várias filas de escalonamento. É tarefa do gerenciador de processos do Sistema Operacional selecionar os processos dessas filas que usarão os recursos. A parte do gerente de processos que é responsável pela seleção é chamado de **escalonador de processos** (*scheduler process*).



Nos sistemas multitarefa, vários processos são submetidos à execução. Como existe apenas um único processador, aqueles que estão aptos a serem executados imediatamente são colocados em um *spool* (espaço de armazenagem em disco), onde são mantidos até serem executados. O escalonador de longo prazo seleciona alguns processos para ocuparem a memória. O escalonador de curto prazo seleciona um dos processos da memória para utilizar o processador, por isso ele também é chamado de escalonador de CPU.

A diferença entre esses escalonadores está na frequência de execução. Enquanto o segundo é executado com altíssima frequência devido ao curto de período de tempo de uso de um processo por vez, o primeiro é executado sem muita frequência. É tarefa do escalonador de longo prazo formar o banco de processos a serem escalonados. Conseqüentemente, é importante que ele faça uma cuidadosa seleção dos processos. Para fazer esta seleção, foi feita uma nova divisão de categoria de processos:

- ❑ **Processos Limitados por I/O** – são processos que predominantemente fazem operações de I/O durante a sua execução;
- ❑ **Processos Limitados por CPU** – predominantemente, eles fazem operações com a CPU.

É altamente recomendado que o escalonador de longo prazo faça uma combinação com processos de ambas categorias. Esta recomendação tem por objetivo maximizar o uso dos recursos computacionais. A título de exemplificação, toma-se a hipótese de que há uma grande quantidade de processos limitados por I/O na memória. Existe uma grande probabilidade de não haver processos na fila de processos prontos para execução, conseqüentemente o processador estará sendo subutilizado.

Por outro lado, se houver muitos processos limitados por CPU, a probabilidade dos recursos ficarem ociosos é grande. Além disso, pode haver processos que poderiam utilizá-los e não o fazem por não estarem na memória.

4.2.3 SUBSTITUIÇÃO DE PROCESSOS

O procedimento de substituição de um processo no processador requer uma série de cuidados, pois se ele voltará a utilizar o processador, é importante que ele recomece exatamente de onde ele parou.

É importante a mudança de contexto de processo que é representado pelo Bloco de Controle de Processo. Outro ponto a ser considerado é que a troca de processos é um processo, e consome recursos da CPU, os quais acabam não sendo contabilizados.



4.3 OPERAÇÕES EM PROCESSOS

Os processos em um Sistema multitarefa executam de forma concorrente. Dessa maneira, é muito importante que haja um rigoroso controle sobre as operações realizadas pelos processos.

4.3.1 CRIAÇÃO DE PROCESSOS

Um processo é criado a partir de uma chamada de sistema e pode criar outros processos através de novas chamadas de sistemas. O processo criador é chamado processo-pai, enquanto os processos gerados por ele são chamados por processos-filho. Como cada processo pode criar novos processos, pode ser feita uma analogia com uma árvore de processos, conforme é ilustrado na Figura 4.5.

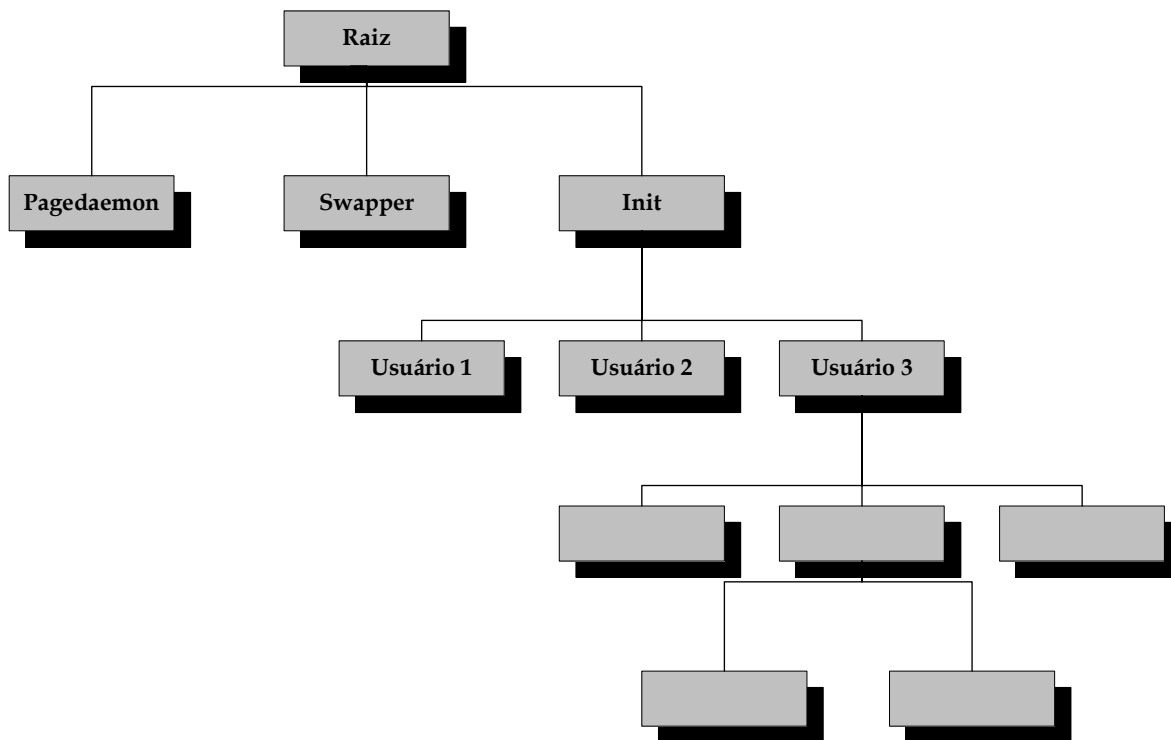


Figura 4-5 - Árvore de Processo

Para realizar a sua tarefa, os processos geralmente precisam de recursos. Um processo filho pode tanto requerer os seus recursos junto ao Sistema Operacional ou usar os recursos do processo-pai que o gerou. Quando isto ocorre, o processo-pai deve distribuir os recursos que estão em sua posse entre seus processos-filho.

Restringir um processo-filho a usar apenas os recursos que estão com o seu processo-pai possibilita um maior controle sobre os recursos e evita que um deles sobrecarregue o sistema. O processo pai também pode passar os dados de



entrada para os seus processos filhos. Quando um processo gera um outro processo podem ocorrer duas situações:

1. O processo-pai continua a executar de forma concorrente com os processos-filho;
2. O processo-pai fica bloqueado até que um ou todos os seus processos-filho tenham terminado a execução.

Além disso, um processo filho pode ser:

- ❑ Uma duplicata do processo-pai;
- ❑ Um outro programa carregado na memória.

A Figura 4.6 ilustra um exemplo em fluxograma da geração de um processo.

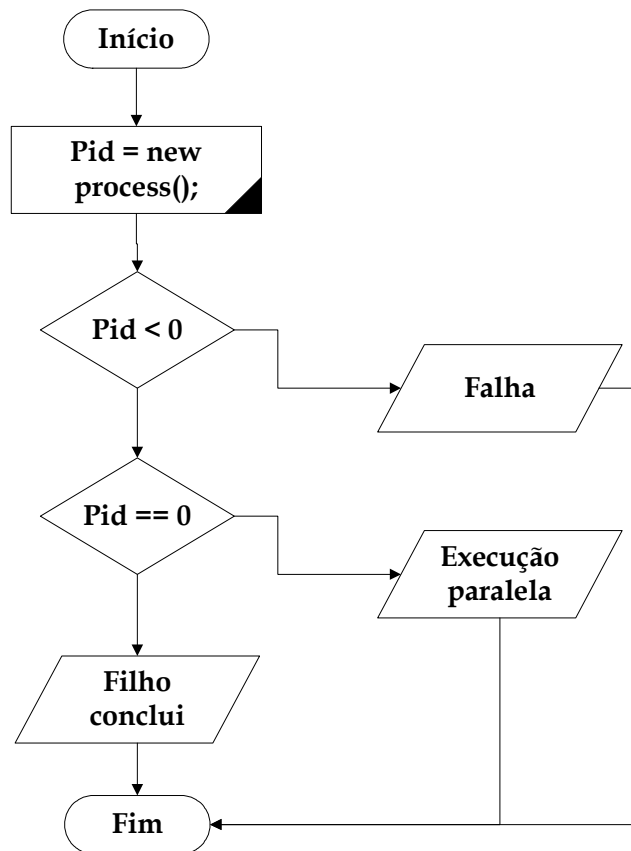


Figura 4-6 - Criação de Processo

4.3.2 TÉRMINO DE UM PROCESSO

O término de um processo é marcado pela execução de sua instrução final que pede que o sistema operacional o exclua da lista de processos. Todos os recursos alocados pelo processo são liberados.



4.4 PROCESSOS COOPERATIVOS

Os processos concorrentes que estão executando no sistema operacional podem ser concorrentes ou cooperativos. Um processo é independente se não afeta a execução de outros processos. Basicamente, qualquer processo que não compartilha dados ou recursos é independente. Caso um processo afete outro, então ele é cooperativo. Existem várias razões para construir um ambiente favorável aos processos cooperativos:

- ❑ *Compartilhamento de Informações* – vários usuários podem estar interessados na mesma informação. É preciso fornecer mecanismos para que possam utilizá-la sem causar danos uns aos outros ou à própria informação;
- ❑ *Velocidade de Computação* – dividir uma tarefa em partes menores pode fazer com que ela seja executada mais rápida. É importante que cada parte não utilize os mesmos recursos;
- ❑ *Modularidade* – a construção do sistema em forma modular facilita a manutenção do mesmo e até a gerência de memória e processos do sistema operacional;

A questão dos conflitos dos processos cooperativos é foco constante de pesquisas. Uma primeira análise de concorrência de processos pode ser feita através da analogia do problema do produtor-consumidor. Nele há dois processos e uma memória compartilhada:

- ❑ **Produtor** – insere uma série de itens continuamente na memória;
- ❑ **Consumidor** – retira os itens continuamente da memória compartilhada;

A memória pode ser limitada ou ilimitada. Ou seja, pode ou não ter uma capacidade máxima de armazenamento. Além disso, a taxa de criação dos itens do processo Produtor pode ser diferente da taxa de retirada dos itens do processo consumidor. Este problema é a analogia do uso do *buffer* de qualquer dispositivo.

4.5 COMUNICAÇÃO ENTRE PROCESSOS

O compartilhamento de recursos torna necessário que haja mecanismos para prover esta cooperação entre eles. Os mecanismos também devem assegurar a integridade dos recursos e que eles não sejam tomados arbitrariamente de algum processo. Toda cooperação é baseada na comunicação entre os membros, no caso os processos.

A melhor forma para prover esta comunicação entre os processos é a troca de mensagens entre eles, os quais podem ser definidos de várias formas.