

## 4 Políticas de Comunicação

Uma analogia entre as técnicas de trabalho em grupo [Minicucci, 1992] e as políticas de comunicação foi realizada para tentar oferecer o suporte tecnológico pretendido no MC2 usando conceitos da computação.

Uma das características mais importantes de um sistema de comunicação é o conjunto de políticas de comunicação disponibilizadas por ele. De maneira geral, políticas de comunicação são responsáveis por coordenar a interação dos atores envolvidos. Normalmente a política determina quem pode escrever para os canais de comunicação, quando isto deverá acontecer e quem tem permissão para lê-los.

Em qualquer sistema o controle de acesso ao ambiente (comumente denominado “*floor control*”), que gerencia quais participantes têm direito à fala e à manipulação de documentos em um dado instante, bem como o controle do período máximo de tempo que cada participante tem, quando detém um controle específico do ambiente, é um mecanismo chave e, em geral, de difícil implementação. Um sistema ideal deve permitir a configuração completa do acesso às suas facilidades, e se encarregar de gerenciar as regras estabelecidas. [Oliveira & Soares, 1996]

É importante definir e implementar as políticas que regulamentam o acesso à palavra para garantir que o discurso de um usuário não será interrompido [Controle de Palco, 2003] Há quatro formas básicas de realizar o controle da palavra:

- Nenhum Controle: O sistema permite o acesso livre à superfície compartilhada para evitar conflitos. O resultado deste controle é satisfatório quando apenas duas pessoas estão utilizando o *chat*, mas obviamente não é prático quando o número de pessoas aumenta.

- Locking Implícito: O participante que começa a enviar informação tem controle sobre o espaço compartilhado, liberando-o quando acabar.
- Locking Explícito: O participante deve solicitar a palavra.
- Controle da Palavra: Um dos participantes é designado como mediador para a sessão colaborativa e pode escolher quem terá o controle sobre o espaço compartilhado.

A escolha das políticas de comunicação iniciou-se com a percepção da semelhança entre as políticas e as técnicas de escalonamento [Tanenbaum & Woodhull, 1997] utilizadas em Sistemas Operacionais. Normalmente, vários processos precisam utilizar o processador ao mesmo tempo. Este conflito é solucionado com o uso de técnicas de escalonamento. Fazendo uma analogia entre os processos e as mensagens enviadas pelos participantes de uma sessão de bate-papo, pode-se dizer que da mesma forma que o processo quer usar o processador, a mensagem quer usar o canal de comunicação. Outra maneira é considerar os participantes como se fossem processos que no lugar de estarem disputando o uso do processador, estão disputando o uso do canal de comunicação. Quando um ou mais participantes requisitam a palavra na sessão de bate-papo, é preciso decidir a ordem em que os pedidos serão atendidos.

A seguir serão apresentados os algoritmos de escalonamento estudados.

#### **4.1 Escalonamento FIFO (*FIFOPolicy*)**

No escalonamento FIFO [Tanenbaum & Woodhull, 1997] , como mostra a Figura 17, os processos prontos são colocados numa fila organizada por ordem de chegada. Na sua vez, cada processo recebe o uso da CPU até que seja completado, ou seja, o processo permanece em execução até que seja finalizado, de forma que os demais processos na fila fiquem esperando por sua oportunidade de processamento, assim sendo, o escalonamento FIFO é um algoritmo não preemptivo.



**Figura 17 - Escalonamento FIFO**

Devido ao fato de dar igual tratamento a todos os processos, processos de pequena duração não são favorecidos pois têm seu tempo de resposta fortemente influenciado pelos processos a serem processados primeiramente, ou seja, o tempo de resposta aumenta consideravelmente em função da quantidade de processos posicionados à frente e também pela duração destes.

Outro ponto é que processos importantes podem ficar a espera devido à execução de outros processos menos importantes dado que o escalonamento FIFO não concebe qualquer mecanismo de distinção entre processos (por exemplo, processos com diferentes níveis de prioridade).

De acordo com as considerações relativas ao escalonamento FIFO uma forma de enxergar o compartilhamento do canal de comunicação entre os participantes de uma sessão de bate-papo é organizar as mensagens enviadas por eles colocando-as numa fila ordenada por ordem de chegada. Quando chegar a sua vez, a mensagem recebe o uso do canal e é enviada para todos os participantes, de forma que as demais mensagens na fila fiquem esperando por sua oportunidade de utilizar o canal.

O fato do tratamento a todas as mensagens e a todos os participantes ser igual faz com que participantes cujas mensagens deveriam ter maior importância, não sejam favorecidos, pois eles não têm acesso imediato ao canal de comunicação e precisam esperar a sua vez na fila. Como foi dito anteriormente, o escalonamento FIFO não concebe qualquer mecanismo de distinção entre participantes, como por exemplo, participantes com diferentes níveis de prioridade.

## 4.2 Escalonamento FILO (*FILOPolicy*)

No escalonamento FILO [Tanenbaum & Woodhull, 1997] , os processos prontos são colocados numa pilha. A partir daí, o funcionamento é o mesmo do escalonamento FIFO pois cada processo recebe o uso da CPU até que seja completado, ou seja, o processo permanece em execução até que seja finalizado, de forma que os demais processos na pilha fiquem esperando por sua oportunidade de processamento. Os problemas também são os mesmos.

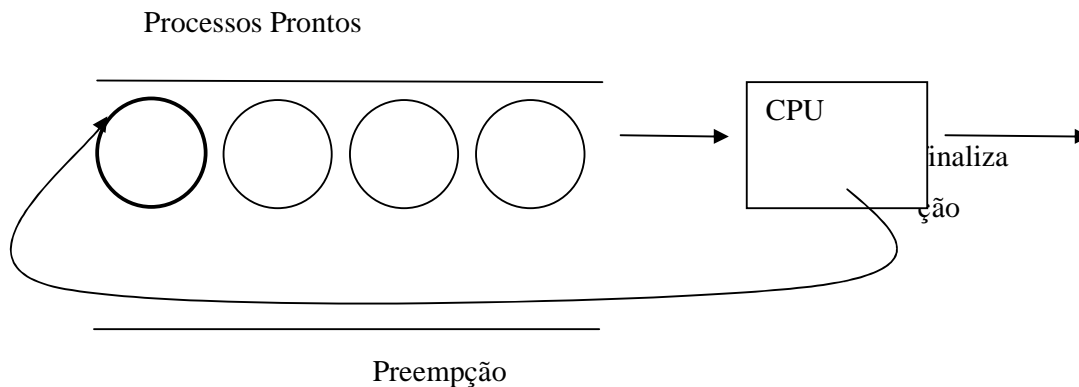
Do mesmo modo que os processos, as mensagens podem ser colocadas em um pilha e a partir daí a utilização do canal de comunicação poderá ser realizada da mesma forma que no escalonamento FIFO.

## 4.3 Escalonamento Round Robin (*RoundRobinPolicy*)

No escalonamento *Round Robin* ou circular [Tanenbaum & Woodhull, 1997] , como pode ser visto na Figura 18, os processos são organizados numa fila segundo sua ordem de chegada e então são despachados para execução. No entanto, ao invés de serem executados até o fim, a cada processo é concedido apenas um intervalo de tempo (*time-slice* ou *quantum*). Caso o processo não seja finalizado neste intervalo de tempo ocorre sua substituição pelo próximo processo na fila de processos ativos, sendo o processo interrompido colocado no fim da fila. Isto significa que ao final de seu intervalo de tempo ocorre a preempção do processador, ou seja, o processador é designado a outro processo, sendo salvo o contexto do processo interrompido para permitir a continuidade da execução quando sua vez chegar novamente.

O escalonamento *Round Robin* se baseia na utilização de temporizadores, sendo um algoritmo preemptivo e bastante adequado para ambientes interativos, ou seja, em sistemas de tempo repartido onde coexistem múltiplos usuários simultâneos sendo, portanto, necessário garantir-se tempos de resposta razoáveis. A sobrecarga imposta pela troca de contexto representa um investimento para atingir-se um bom nível de eficiência, pois com diversos processos em execução

simultânea (pseudoparalelismo) é possível manter ocupados todos os recursos do sistema.



**Figura 18 - Escalonamento Round Robin**

A determinação do tamanho do intervalo de tempo (*quantum*) é extremamente importante pois relaciona-se com a sobrecarga imposta ao sistema pelas trocas de contexto dos processos ativos. Para cada processo existe uma janela de tempo onde ocorrem:

1. a recuperação do contexto do processo (trc),
2. a execução do processo pela duração do *quantum* (q) e
3. a preservação do processo após o término de seu *quantum* (tpc).

Como o tempo tomado para a troca de contexto (tc) não é útil do ponto de vista de processamento de processos dos usuários, temos que, para cada janela de tempo concedida aos processos, a troca de contexto representa uma sobrecarga, pois somente o *quantum* de processamento é útil.

Analogamente ao tratamento do escalonamento *round robin* com processos, a forma de compartilhar o canal de comunicação entre os participantes de uma sessão de bate-papo é organizá-los numa fila segundo a ordem de solicitação do espaço compartilhado e então conceder o uso do canal a cada participante durante um intervalo de tempo pré-definido. Caso o participante não libere o canal neste intervalo de tempo, ocorre a sua substituição pelo próximo participante na fila, sendo que o participante interrompido é colocado no fim da fila. Não faz sentido

compartilhar o canal dessa forma para as mensagens enviadas pelos participantes, pois não faz sentido subdividi-las em mensagens menores.

#### 4.4 Escalonamento por Prioridades (*PriorityPolicy*)

O escalonamento *round robin* faz a suposição implícita de que todo processo é igualmente importante. Com frequência, as pessoas que possuem e operam computadores multiusuários têm idéias diferentes sobre tal assunto. Em uma universidade, a ordem de prioridade pode ser diretores primeiro, depois professores, secretários, monitores e, por fim, os alunos. A necessidade de levar em conta fatores externos conduz ao escalonamento por prioridade. A idéia básica é simples e direta: a cada processo é atribuída uma prioridade, e o processo executável com a maior prioridade recebe permissão para executar. [Tanenbaum & Woodhull, 1997]

Mesmo em um PC com um único proprietário, pode haver múltiplos processos, alguns mais importantes que outros. Por exemplo, a um processo de servidor que envia uma mensagem de correio eletrônico em segundo plano deve ser atribuído uma prioridade mais baixa que a um processo que exibe um vídeo na tela em tempo real.

Para evitar que processos de alta prioridade executem indefinidamente, o escalonador pode diminuir a prioridade do processo atualmente em execução a cada *clock*. Se esta ação fizer com que sua prioridade caia para baixo do próximo processo com maior prioridade, ocorre uma comutação de processos. Alternativamente, a cada processo pode ser atribuído um *quantum* máximo em que lhe é permitido usar a CPU continuamente. Quando esse *quantum* acaba, é dada uma chance de executar ao próximo processo com maior prioridade. As prioridades podem ser atribuídas aos processos estática ou dinamicamente.

O sistema UNIX tem um comando, *nice*, que permite que um usuário voluntariamente reduza a prioridade do seu processo, para ser gentil (*nice*) com os outros usuários. Ninguém o utiliza. [Tanenbaum & Woodhull, 1997]

Muitas vezes é conveniente agrupar processos em classes de prioridade e utilizar escalonamento por prioridade entre as classes e escalonamento *round robin* dentro de cada classe.

Considerando que cada participante pode ser considerado como um processo pelo escalonamento por prioridades, a cada participante é atribuída uma prioridade, e o participante com maior prioridade recebe permissão para utilizar o espaço compartilhado. O mesmo raciocínio pode ser usado com as mensagens. A mensagem cujo participante tem maior prioridade recebe o uso do canal e é enviada para todos os participantes.

#### **4.5 Escalonamento Aleatório (*RandomPolicy*)**

Também chamado de escalonamento por sorteio, tem como idéia básica dar bilhetes de loteria para cada processo. Sempre que uma decisão de escalonamento tiver de ser feita, um bilhete de loteria é escolhido aleatoriamente, e o processo que armazena este bilhete recebe o recurso. [Tanenbaum & Woodhull, 1997]

Parafrazeando George Orwell em [Orwell, 1945], “Todos os processos são iguais, mas alguns processos são mais iguais”. Os processos mais importantes podem receber bilhetes extras, para aumentar suas chances de ganhar.

O escalonamento por sorteio tem várias propriedades interessantes. Por exemplo, se um novo processo aparece e recebe alguns bilhetes, no próximo sorteio, ele terá uma chance de ganhar em proporção ao número de bilhetes que possui. Em outras palavras, o escalonamento por sorteio é altamente responsivo.

O principal objetivo dos algoritmos de escalonamento é designar o processador para um certo processo dentre vários processos existentes, otimizando um ou mais aspectos do comportamento geral do sistema. Stallings em [Stallings, 1995] categoriza os escalonadores como:

1. Orientados ao Usuário: quando procuram otimizar os tempos de resposta e permanência além da previsibilidade e,

2. Orientados ao Sistema: quando enfatizam a produtividade, a taxa de utilização do processador, o tratamento justo e o balanceamento do uso de recursos.

De acordo com o conceito do escalonamento aleatório, a forma de compartilhar o canal de comunicação entre os participantes de uma sessão de bate-papo é dar bilhetes de loteria para cada participante. Sempre que uma decisão de escalonamento tiver de ser tomada, um bilhete de loteria é escolhido aleatoriamente, e ao participante que armazena este bilhete é concedido o uso do canal.

Após o estudo dos algoritmos de escalonamento, verificou-se que estes não propiciam coordenação de forma satisfatória, pois os seus objetivos são muito distintos dos objetivos da coordenação. Os algoritmos de escalonamento têm como objetivo a utilização do canal com o menor desperdício possível, e dando oportunidade a todos os processos. Já no caso da mediação da conversação, o objetivo é ter uma utilização criativa, ou seja, fornecer subsídios ao mediador para que ele conduza a conversação entre os aprendizes de um debate da forma que desejar.