Sistemas Operacionais

Comunicação entre processos.

Prof. Felipe Leite Lobo
felipe.lobo@ufrr.br

Exclusão Mútua

- Espera Ocupada;
- Primitivas *Sleep/Wakeup;*
- **Semáforos;**
- **Monitores;**
- Passagem de Mensagem.

Exclusão Mútua: Semáforos

- Idealizado pelo matemático holandês E. W. Dijkstra (1965);
- Variável inteira para contar o número de sinais de "acordar" salvos;
- Um semáforo pode ter valor 0, quando não há sinal de "acordar" armazenado ou um valor positivo se um ou mais sinais de acordar estiverem pendentes;

Exclusão Mútua: Semáforos

- Duas primitivas de chamadas de sistema: ***down (sleep) e up (wake);***

- Originalmente P (***Proberen - testar) e V (Verhogen - incrementar) em holandês.***

- Quando um processo executa uma operação ***P (down)***, o valor do semáforo é decrementado. O processo pode ser eventualmente bloqueado e inserido na fila de espera do semáforo.

Exclusão Mútua: Semáforos

- Numa operação ***V (up)***, o semáforo é incrementado e, eventualmente, um processo que aguarda na fila de espera deste semáforo é acordado.

- A operação P também é comumente referenciada como DOWN ou WAIT, e a operação V como UP ou SIGNAL.

- Semáforos que assumem somente os valores 0 e 1 são denominados semáforos binários ou mutex. Neste caso, P e V são também chamadas de LOCK e UNLOCK, respectivamente.

Exclusão Mútua: Semáforos

- Todas essas ações são chamadas de ações atômicas.

- Ações atômicas garantem que quando uma operação no semáforo está sendo executada, nenhum processo pode acessar o semáforo até que a operação seja finalizada ou bloqueada.

Exclusão Mútua: Semáforos

- Desvantagens: Embora semáforos forneçam uma abstração flexível o bastante para tratar diferentes tipos de problemas de sincronização, ele é inadequado em algumas situações.

- Semáforos são uma abstração de alto nível baseada em primitivas de baixo nível, que provêem atomicidade e mecanismo de bloqueio, com manipulação de filas de espera e de escalonamento.

- Tudo isso contribui para que a operação seja lenta. Para alguns recursos, isso pode ser tolerado; para outros esse tempo mais longo é inaceitável.

Exclusão Mútua

- Espera Ocupada;
- Primitivas *Sleep/Wakeup;*
- Semáforos;
- **Monitores;**
- Passagem de Mensagem.

Exclusão Mútua: Monitores

- Idealizado por Hoare (1974) e Brinch Hansen (1975);

- **Monitor: primitiva (unidade básica de sincronização) de alto nível para sincronizar processos:**

- Conjunto de procedimentos, variáveis e estruturas de dados agrupados em um único módulo ou pacote;

Exclusão Mútua: Monitores

- **Somente um processo pode estar ativo dentro do monitor em um mesmo instante; outros processos ficam bloqueados até que possam estar ativos no monitor;**

Exclusão Mútua: Monitores

```
monitor example
int i;
condition c;
procedure A();
.
end;
procedure B();
.
end;
end monitor;
```

Estrutura básica de um Monitor

- Dependem da linguagem de programação, o compilador é que garante a exclusão mútua.
- Todos os recursos compartilhados entre processos devem estar implementados **dentro** do **Monitor.**

Exclusão Mútua: Monitores

- Execução:

- Chamada a uma rotina do monitor;
- Instruções iniciais: teste para detectar se um outro processo está ativo dentro do monitor;
- Se positivo, o processo novo ficará bloqueado até que o outro processo deixe o monitor;
- Caso contrário, o processo novo executa as rotinas no **monitor.**

Exclusão Mútua: Monitores

- *Condition Variables (condition): variáveis que indicam uma condição;*
- Operações Básicas: *WAIT e SIGNAL*
- ***wait (condition) : bloqueia o processo;***
- ***signal (condition): "acorda" o processo que executou um wait na variável condition e foi bloqueado;***

Exclusão Mútua: Monitores

- Variáveis condicionais não são contadores, portanto, não acumulam sinais;
- Se um sinal é enviado sem ninguém (processo) estar esperando, o sinal é perdido;
- Assim, um comando WAIT deve vir antes de um comando SIGNAL.

Exclusão Mútua: Monitores

- Como evitar dois processos ativos no monitor ao mesmo tempo?

- (1) **Hoare : colocar o processo mais recente para rodar, suspendendo o outro!!! (*sinalizar e esperar)***

- (2) **B. Hansen: um processo que executa um SIGNAL deve deixar o monitor imediatamente;**

- O comando SIGNAL deve ser o último de um procedimento do monitor;

Exclusão Mútua: Monitores

- A exclusão mútua automática dos procedimentos do monitor garante que, por exemplo, se o produtor dentro de um procedimento do monitor descobrir que o buffer está cheio, esse produtor será capaz terminar a operação de WAIT sem se preocupar;

- O consumidor não estará ativo dentro do monitor até que WAIT tenha terminado e o produtor tenha sido marcado como não mais executável;

Exclusão Mútua: Monitores

- Limitações de semáforos e monitores:
- Ambos são boas soluções somente para CPU's com memória compartilhada. Não são boas soluções para sistema distribuídos;
- Nenhuma das soluções provê troca de informações entre processo que estão em diferentes máquinas;
- Monitores dependem de uma linguagem de programação – poucas linguagens suportam Monitores.

Exclusão Mútua

- Espera Ocupada;
- Primitivas *Sleep/Wakeup;*
- Semáforos;
- Monitores;
- **Passagem de Mensagem**.

Exclusão Mútua: Passagem de Mensagem

- Provê troca de mensagens entre processos rodando em máquinas diferentes;

- Utiliza-se de duas primitivas de chamadas de sistema: *send e receive;*

- Podem ser implementadas como procedimentos:

- *send (destination, &message);*

- *receive (source, &message);*

- O procedimento *send* envia para um determinado destino uma mensagem, enquanto que o procedimento *receive* recebe essa mensagem em uma determinada fonte; Se nenhuma mensagem está disponível, o procedimento *receive* é bloqueado até que uma mensagem chegue.

- Problemas desta solução:

- Mensagens são enviadas para/por máquinas conectadas em rede; assim mensagens podem se perder ao longo da transmissão;

- Mensagem especial chamada ***acknowledgement: o procedimento receive envia um acknowledgement para o procedimento send. Se esse acknowledgement não chega no procedimento send, esse procedimento retransmite a mensagem já enviada;***

Exclusão Mútua: Passagem de Mensagem

- Problemas:

- Desempenho: copiar mensagens de um processo para o outro é mais lento do que operações com semáforos e monitores;

Exclusão Mútua: Outros Mecanismos

- RPC – *Remote Procedure Call*
  - Rotinas que permitem comunicação de processos em diferentes máquinas.
  - Chamadas remotas.
- MPI – *Message-passing Interface*
  - Sistemas paralelos.
- RMI Java – *Remote Method Invocation*
  - Permite que um objeto ativo em uma máquina virtual Java possa interagir com objetos de outras máquinas virtuais Java, independentemente da localização dessas máquinas virtuais.
- Web Services
  - Permite que serviços sejam compartilhados através da Web.