Sistemas Operacionais

CCP/SIF
UNISUL – Tubarão

Cassio Brodbeck Caporal
cassio{NOSPAM}ostec.com.br

Agenda

- Revisão;
- Comunicação interprocessos;
- *Race conditions* (condições de disputa);
- Regiões críticas;
- Exclusão mútua;
- Semáforos, *mutexes*, monitores;

Agenda

- Passagem/troca de mensagens;
- Problemas clássicos de IPC;
  - Jantar dos filósofos;
  - Leitores e escritores;
  - Barbeiro sonolento;
- Exercício.

Comunicação interprocessos

- Ambiente concorrido:
  - Compartilhamento de recursos computacionais;
  - É necessário ter acordo e sincronismo da execução de processos concorrentes;
  - Garantia de processamento correto de *softwares*;

Comunicação interprocessos

- Um processo precisa se comunicar com outro:
  - Muito comum em *NIX: utilização de *pipes*;
  - A saída de um programa serve para a entrada do outro;

Comunicação interprocessos

- Objetivos:
  - Como um processo passa a informação para outro;
  - Como garantir que processos não invadam espaço de outro quando envolvidos em atividades “críticas”;
  - Como garantir o sequenciamento correto de atividades em processos dependentes;

Condições de disputa

- Compartilhamento de uma mesma *área* que pode ser lida ou escrita por processos diferentes;
  - Armazenamento compartilhado:
    - Memória principal;
    - Memória secundária;

Condições de disputa

- Lei de Murphy: “se algo puder dar errado, certamente dará”;
  - Imagine operadores de banco (caixas) fazendo operações de crédito e débito ao mesmo tempo em sua conta...
- Resultado final depende das informações de quem e quando executa **precisamente**;

Regiões críticas

- Como evitar *race conditions*?
  - Modelo usual: impedir que mais de um processo leia e/ou escreva ao mesmo tempo em 'locais' compartilhados;
- Terminologia: exclusão mútua (*mutual exclusion*);
  - Impedir o acesso a um dado compartilhado se já tiver um processo utilizando.

Regiões críticas

- Problema com os caixas do banco:
  - Um caixa iniciou a operação usando uma variável compartilhada **antes** que o outro caixa terminasse sua operação;
- Região crítica ou seção crítica:
  - Parte do programa em que existe acesso a memória ou recurso compartilhado;

Regiões críticas

- Como resolver?
- Modelo básico:
  - Dois processos nunca podem estar em suas regiões críticas ao mesmo tempo;
- Problema? Ineficiência de compartilhamento de dados entre processos;
- Soluções...

Regiões críticas

- Desabilitar interrupções:
  - Logo após entrar na região crítica, desabilita interrupções e a CPU não é alternada de processo até que saia da região crítica;
  - Sistemas SMP: *disable*;
  - *DANGEROUS!*

Regiões críticas

- Usar a instrução TSL (*test and set lock*):
  - Permite ler uma variável, armazená-la em outra área e atribuir um novo valor a **mesma** variável;
  - Presente em grande parte dos processadores;
  -

Regiões críticas

- Variáveis de *locking* (travas):
  - Solução de *software*;
  - Utilização de variável compartilhada;
  - Mesmo problema que o *spool:* dois processos podem ler a variável ao mesmo tempo e antes que seja alterado seu valor, entra na região crítica;

Regiões críticas

- Para soluções de software muitos algoritmos foram propostos;
- O primeiro verdadeiramente aceito, foi do holândes Dekker, conhecido como Algoritmo de Dekker;
- Muuuito complexo, sendo minimizado pelo Algoritmo de Peterson: variável *turn*;

Sincronização condicional

- Um recurso deve estar disponível para ser consumido;
- Exemplo: leitura e gravação em um *buffer*;
- Processos produtores;
  - Não pode gravar dados em um *buffer* cheio;
- Processos consumidores;
  - Não deve ler dados de um *buffer* vazio;

Semáforos

- Proposto por Dijkstra em 1965;
- Permite implementar exclusão mútua E sincronização condicional;
- Variável inteira, não negativa e indivisível (não há interrupções);
- Manipulação através de duas instruções: DOWN e UP;

Semáforos

- A instrução UP incrementa em 1 o valor do semáforo, a DOWN decrementa em 1;
- Atuam como protocolos de entrada e saída para que processos possam entrar e sair de regiões críticas;
- Valor 1 significa que nenhum processo está usando, 0 que o recurso está em uso;

Semáforos

- Ao entrar em uma região crítica, DOWN é executado:
  - Se o resultado for 0, o processo solicitante acessa a sua região crítica;
  - Se outro processo chamar DOWN, é impedido o acesso a região crítica;

Semáforos

- Ao sair de uma região crítica, executa UP, incrementando o valor do semáforo:
  - Acesso ao recurso é liberado;
  - Se tiverem vários processos pendentes, o SO selecionará um deles e o colocará em estado de pronto.

Monitores

- Proposto por Brinch Hansen em 1972 e desenvolvido por Hoare 2 anos depois;
- São mecanismos para sincronização em alto nível, tornando mais simples o desenvolvimento de aplicações;
- Geralmente são implementados pelo próprio compilador;

Passagem de mensagens

- Ooooutro mecanismo de comunicação e sincronização de processos;
- Não há necessidade de variáveis compartilhadas (ótimo!!!);
- Geralmente utilizam-se *UNIX Domain Sockets* ou *pipes*;

Passagem de mensagens

- O formato de mensagens, na literatura, é SEND e RECEIVE;
- Comunicação direta:
  - Endereçamento explícito do processo envolvido na mensagem;
  - Grande dependência da não alternância de nomes e identificadores de processos;

Passagem de mensagens

- Comunicação indireta:
  - Utiliza-se de um *buffer* para colocar dados (transmissor) onde o receptor pode coletar estes dados;
  - Exemplo: portas de protocolo TCP ou UDP;

Deadlock

- Hmmm isso não é bom...
  - Um processo aguarda por um evento ou recurso que nunca estará disponível;
  - Isso torna-se cada vez mais comum em SO's modernos, devido ao aumento de paralelismo e alocação dinâmica de recursos;
  - Exemplo com 2 processos e 2 'recursos'.

Problemas clássicos com IPC

- O jantar dos filósofos:
  - Formulado e resolvido por Dijsktra, em 1965;
  - Cinco filósofos sentados em uma mesa circular, cada qual com seu prato de espaguete;
  - Existem só 4 garfos, ihhh!!
  - Filósofos pensam e comem;

Problemas clássicos com IPC

- O barbeiro sonolento:
  - Há um barbeiro, uma cadeira de barbeiro e cadeiras para os clientes;
  - Quando não tem cliente, o barbeiro dorme;
  - Quando um cliente chega, acorda o barbeiro;
  - Quando as cadeiras estão cheias, o cliente não pode entrar.