

Comunicação entre Processos

Problemas Clássicos

Prof. Edson Pedro Ferlin

Agradecimento ao Prof. Osmar Betazzi Dordal

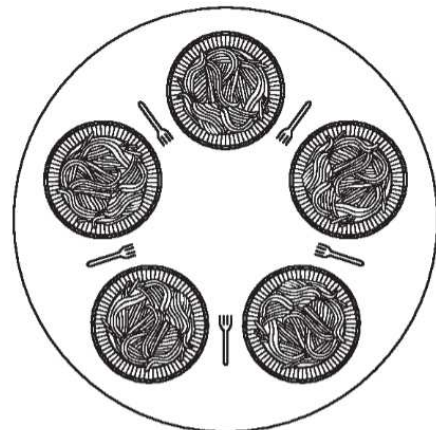
- **Objetivos**
 - Estudar os problemas clássicos envolvendo comunicação entre processos e deadlocks
- **Conteúdos**
 - Comunicação entre Processos – Problemas Clássicos
 - Deadlocks

Problemas Clássicos

- Produtor Consumidor
 - Já verificado anteriormente.
 - Muito importante para estudo.
 - Protocolo TCP/IP utiliza muito.
- Jantar dos Filósofos
 - Também utilizado na transmissão de dados em TCP/IP.
- Leitores e Escritores
 - Exemplo de bloqueios em banco de dados.

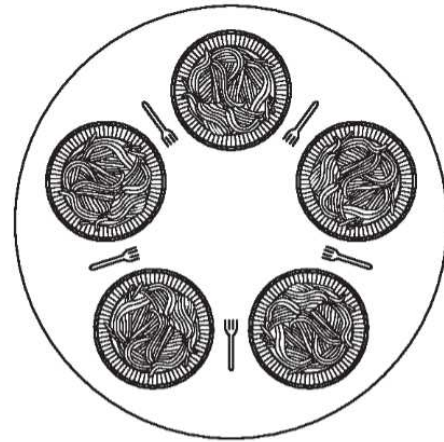
Jantar dos Filósofos

- Em uma mesa circular, 5 filósofos estão sentados para o jantar.
- Cada um dos filósofos possui um prato com macarrão.
- Cada um deles precisa utilizar 2 garfos.
- Entre os pratos existe apenas 1 garfo.
- Estes garfos precisam ser compartilhados de forma sincronizada.
- Então, são 5 pratos, 5 filósofos e 5 garfos, mas cada filósofo deve utilizar 2 garfos para comer.



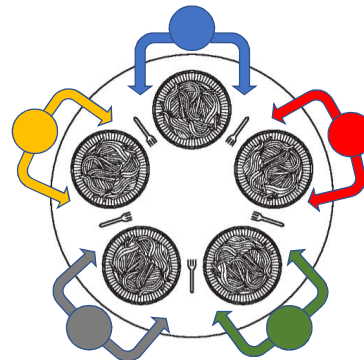
Jantar dos Filósofos


- Os filósofos comem e pensam, alternadamente.
- Além disso, vão comer, pegam apenas um garfo por vez.
- Se conseguirem pegar os dois, comem por alguns instantes e depois largam os garfos.
- **Como evitar que haja bloqueio dos filósofos?**



Jantar dos Filósofos

- Como os filósofos estão sentados a mesa.
- Lembrando que eles só comem quando estão com 2 garfos a mão:
 - Seria melhor imaginar que cada garfo seria um *rashi*. Assim, realmente eles precisariam de dois para comer.
- Outro ponto importante é que nenhum deles pode passar fome (*starvation* – Inanição).





Professor
Edson Pedro FERLIN

Sistemas Operacionais

Jantar dos Filósofos

Solução:

Alternar entre:

Pensar;


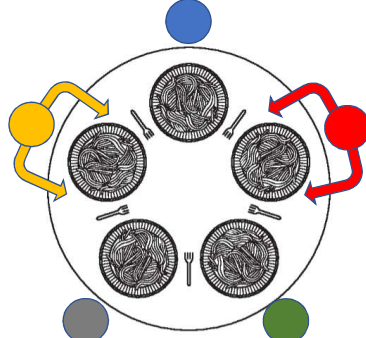
Pegar os garfos:

Bloqueio Direito;


Bloqueio Esquerdo;

Comer;

Soltar os garfos;

7
Comunicação entre Processos – Problemas Clássicos
Prof. Edson Pedro Ferlin

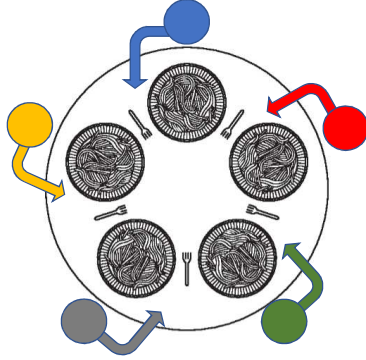


Professor
Edson Pedro FERLIN

Sistemas Operacionais

Jantar dos Filósofos

- Problemas com essa solução...
- Não possui exclusão mútua:
 - **Deadlock**
 - Todos os filósofos pegam um único garfo ao mesmo tempo;
 - Isso leva a um impasse.
 - **Starvation**
 - Os filósofos ficam indefinidamente pegando os garfos simultaneamente.



Uso dos semáforos é uma solução, mas isso faz com que apenas 1 filósofo coma de cada vez.

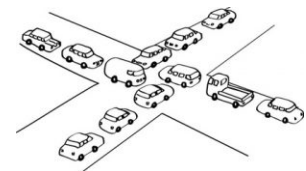
8
Comunicação entre Processos – Problemas Clássicos
Prof. Edson Pedro Ferlin

Leitores e Escritores

- Modelo utilizado em acesso em base de dados.
- Acesso **simultâneo** a base de dados por diversos utilizadores.
- Acessos de Leitores e Escritores
 - Isto leva a operações de leitura e escrita.
- Se um **Escritor** for escrever na base, nenhum **Leitor** pode ler.
- O Escritor que bloqueia a base de dados.
- Mas se a leitura for feita para o primeiro Leitor, este bloqueia a base para os Escritores, mas não para outros Leitores.
- Pode ocorrer *Starvation* com a chegada de muitos Leitores.

Deadlock

- Como os dispositivos e recursos são compartilhados, em Sistemas Operacionais ocorrem os chamados *Deadlocks*
 - Impressora, *Hard Discs*, arquivos, entre outros.
- Os Sistemas Operacionais devem prover acesso exclusivo a certos recursos.
- Um **Deadlock** pode ser traduzido como **impasse**
 - **Processos** ficam **parados** e **não continuam**;
 - Ocorrem tanto em recursos no nível de hardware quanto no nível de software.
- Processos estão em deadlocks se cada processo estiver esperando por um evento que somente outro processo no conjunto pode causar.

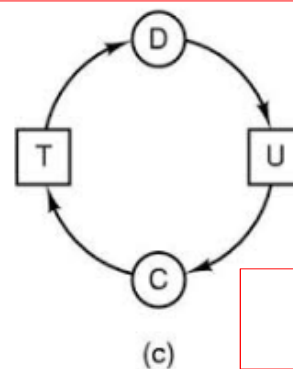


Condições para *Deadlock*

- **Exclusão Mútua**
 - Um recurso estará unicamente alocado para um processo em um determinado momento.
- **Uso e Espera** (*hold and wait*)
 - Processos que já possuem algum recurso podem querer outros.
- **Não Preempção**
 - Recursos já alocados não podem ser retirados; só o próprio processo pode liberá-los.
- **Espera Circular**
 - Um processo espera por recursos alocados a outro processo, em uma cadeia circular.

Condições para *Deadlock*

1. Processo **C** está solicitando o recurso **T**
2. Recurso **T** está alocado para o processo **D**
3. Processo **D** está solicitando o recurso **U**
4. Recurso **U** está alocado para o processo **C**



Estratégias para *Deadlock*

- Ignorar o problema.
- Detectar e recuperar o problema.
- Evitar dinamicamente o problema
 - Alocação de recursos de forma cuidadosa.
- Prevenir o problema por meio da não satisfação de uma das quatro condições citadas anteriormente
 - Exclusão Mútua; Uso e Espera (*hold and wait*); Não Preempção; e Espera Circular.

Contato



eferlin@live.com



(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) ProfEdsonPedroFerlin