

# Sistemas Operacionais

Alisson Linhares  
CAMPINAS,  
2021/1

# O problema produtor-consumidor.

- ▶ Dois processos compartilham um buffer de tamanho fixo;
- ▶ O produtor coloca informações no buffer e o consumidor,retira;
- ▶ Caso o buffer esteja cheio, o produtor deve ser colocado para dormir;
- ▶ Se o buffer estiver vazio, o consumidor deve dormir, sendo acordado quando o produtor colocar algo no buffer;
- ▶ Pode levar a condições de corrida;
- ▶ Variável (count) usada para controlar número de itens no buffer;
- ▶ Número máximo de itens no buffer =  $N$ ;

# Detectar deadlocks.

- ▶ Quando usada, o sistema não tenta prevenir a ocorrência do deadlock.
- ▶ O sistema deixará que deadlock ocorra e tentará detectá-los à medida que isso acontecer.
- ▶ Agirá, então, de alguma maneira para se recuperar após o deadlock.
- ▶ Detecção de deadlocks com um recurso de cada tipo x Múltiplos recurso de cada tipo.

- São conhecidos muitos algoritmos para detecção de ciclos em grafos dirigidos. A seguir um algoritmo simples:

1. Para cada nodo —  $N$  no grafo — execute os cinco passos seguintes, em que  $N$  é o nodo inicial.
2. Inicialize  $L$  como uma lista vazia e assinale todos os arcos como desmarcados.
3. Insira o nodo atual no final da lista  $L$  e verifique se o nodo agora aparece em  $L$  duas vezes. Em caso afirmativo, o grafo contém um ciclo (assinalado em  $L$ ) e o algoritmo termina.
4. A partir do referido nodo, verifique se existe algum arco de saída desmarcado. Em caso afirmativo, vá para o passo 5; do contrário, vá para o passo 6.
5. Escolha aleatoriamente um arco de saída desmarcado e marque-o. Então, siga esse arco para obter o novo nodo atual e vá para o passo 3.
6. O final foi alcançado. Remova-o e volte para o nodo anterior — isto é, aquele que era atual antes desse —, marque-o como atual e vá para o passo 3. Se esse nodo for o nodo inicial, o grafo não conterá ciclo algum e o algoritmo terminará.

# LAB06

## Lab 06:

- ▶ Escreva um programa com 2 threads - uma produtora e outra consumidora. A thread produtora deve construir grafo de alocação de recursos de forma aleatória, inserindo em um buffer de tamanho 10.
- ▶ Use esse algoritmo para gerar os gráficos:
  - ▶ [https://www.ime.usp.br/~pf/algoritmos\\_para\\_grafos/aulas/random.html](https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/random.html)
- ▶ A cada rodada do algoritmo produtor, gere dois número aleatórios entre 2 - 100, uma para V e outro para A. Onde A deve ser um número menor e igual a V,  $A \leq V$ . A thread consumidora devem ler do buffer os grafos gerados pela thread produtora e imprimir os grafos que possuem deadlocks.