

**Universidade de Brasília**  
**Departamento de Ciência da Computação**  
**Disciplina de Tópicos Avançados em Computadores - Programação Concorrente – Turma E**  
**Lista de Exercícios**

1. Defina processos e threads, abordando suas principais diferenças.
2. Defina processos concorrentes.
3. Dê as definições de condições de corrida, região crítica e exclusão mútua.
4. Defina deadlock e starvation.
5. Explique como várias threads conseguem acessar um mesmo dado compartilhado e indique como esta funcionalidade pode ser fornecida para processos.
6. Os algoritmos abaixo representam dois processos que acessam duas variáveis compartilhadas (x e y) e utilizam dois locks para controlar este acesso. Identifique os possíveis problemas presentes nestes algoritmos e proponha novos algoritmos que façam as mesmas tarefas de forma correta.

variáveis compartilhadas:

```
int x = 0;
int y = 0;
lock lx;
lock ly;
```

```
P1() {
    lock(lx);
    x=x+1;
    lock(ly);
    x=x+y;
    unlock(lx);
    unlock(ly);
}

P2(){
    lock(ly);
    y=y+1;
    lock(lx);
    y=y+x;
    unlock(ly);
    unlock(lx);
}
```

7. O problema dos fumantes - Resolva usando semáforos: “Três fumantes se encontram em uma sala com um vendedor de suprimentos para fumantes. Para preparar e usar um cigarro, cada fumante precisa de três ingredientes: tabaco, papel e fósforo, coisas que o vendedor tem à vontade no estoque. Um dos fumantes tem o seu próprio tabaco, o segundo tem seu próprio papel, e o outro tem seu próprio fósforo. A ação se inicia quando o vendedor coloca à venda dois ingredientes na mesa, de forma a permitir que um dos fumantes execute esta prática dita como não muito saudável. Quando o tal fumante termina, ele avisa o vendedor, que escolhe então outros dois ingredientes (aleatoriamente) e coloca a venda, portanto desbloqueando outro fumante.”
8. Dado o algoritmo de chaveamento obrigatório (ver abaixo) para implementação de exclusão mútua (lock) entre dois processos, responda as seguintes questões.
  - a) (1.0 ponto) - Cite e explique um problema relacionado com este algoritmo.
  - b) (1.0 ponto) - Explique como o mesmo poderia ser estendido para implementar um lock entre três ou mais processos.

```
while (TRUE) {
    while (turn != 0) /*espera*/;
    regioao critica ( );
    turn = 1;
    regioao não-critica ( );
} (a) Processo 0
```

```
while (TRUE) {
    while (turn != 1) /*espera */;
    regioao critica ( );
    turn = 0;
    regioao não-critica ( );
} (b) Processo 1
```

9. Resolva o seguinte problema utilizando qualquer mecanismo estudado: “Existe um galpão com comida (considere que a comida é infinita) para cães e gatos. Considere que um número qualquer (maior do que zero) de cães e somente um gato utiliza este galpão para se alimentar. Os cães entram no galpão e ficam lá comendo até que o gato desejar comer. Neste caso, os cães devem sair do galpão de forma que o gato entra no galpão apenas quando o mesmo estiver sem cães em seu interior. Diferente dos cães, o gato permanece comendo por um tempo determinado e depois sai do galpão. Enquanto o gato estiver se alimentando, os cães não podem entrar no galpão.”