

# INE 5410 - Laboratório

## AULA 05 – UTILIZANDO MONITORES

---

### 1. INTRODUÇÃO

O uso de semáforos efetivamente permite efetuar o controle da concorrência, impondo restrições à execução de trechos de código de threads ou processos. As restrições impostas podem ser aplicadas a diversos cenários de execução em sistemas concorrentes, fazendo deste mecanismo um aliado importante para a programação destes.

### 2. OBJETIVO DA AULA

Nesta aula prática, será utilizado o mecanismo de monitor para verificar sua contribuição em diferentes cenários de execução de programas concorrentes.

### 3. EXEMPLOS

#### 3.1. Controle de exclusão mútua no acesso a uma variável

Este é o mesmo problema apresentado na aula prática 2. Em sua versão utilizando *monitores*, é realizada a declaração de um único *monitor* denominado *Contagem*, provido de duas operações exportadas. A primeira denominada *inc*, responsável pelo acesso concorrente à variável de contagem; a segunda a operação *prn*, utilizada para imprimir os identificadores dos processos a cada vez que causam um incremento na variável *Conta*. Nos processos que acessam concorrentemente a variável de contagem, a garantia de exclusão mútua é feita pela chamada, no corpo dos dois processos, da operação *inc* do monitor *Contagem*.

```
PROGRAM INCREMENTO;
VAR Conta : INTEGER;

MONITOR Contagem;
EXPORT inc, prn;

PROCEDURE inc (VAR n : INTEGER);
BEGIN
    n := n + 1;
END; (* inc *)

PROCEDURE prn (VAR n : INTEGER);
BEGIN
    WRITELN(n:3);
END; (* prn *)
```

```

BEGIN
END; (* Contagem *)

PROCESS Type ProcType (pid : INTEGER);
VAR I : INTEGER;
BEGIN
  FOR I := 1 TO 25
  DO
    BEGIN
      Contagem.inc(Conta);
      Contagem.prn(pid);
    END;
  END; (* PROCTYPE *)

VAR
  P : ARRAY[1..8] OF ProcType;
  I : INTEGER;

BEGIN (* main *)
  COBEGIN
    FOR I:= 1 TO 8
      DO P[I](I);
    COEND;
  WRITELN('Contagem Total +++> ', Conta:3);
END.

```

### 3.2. Pombos Correio

Considere o problema de comunicação onde diversos pombos correio transportam cartas de uma cidade A para uma cidade B. As pessoas da cidade A depositam continuamente cartas na Caixa Postal. A cada vez que uma pessoa deposita uma carta, verifica se gerou um número de cartas que seja múltiplo de 3. Em caso positivo, acorda um pombo para que ele leve 3 cartas para a cidade B. Ao chegar na cidade, cada pombo entra numa gaiola e aguarda a chegada dos demais pombos. Quando todos os pombos chegarem em B, estes saem da gaiola e retornam para a cidade A.

```

program pombos;

monitor cxpostal;
export coloca,retira;
var
  ncartas : integer;
  cond : condition;

procedure coloca;
begin
  ncartas := ncartas + 1;
  writeln('Carta colocada --> Cartas = ', ncartas:2);
  resume(cond);
end; (* coloca *)

procedure retira (pid : integer);
begin
  while ncartas < 5
  do delay(cond);
  ncartas := ncartas - 5;

```

```
writeln('Pombo ',pid:2,' retirou -> Cartas = ',ncartas:2);
resume(cond);
end; (* retira *)

begin
  ncartas := 0;
end; (* monitor cxpostal *)

monitor gaiola;
export entra;

var
  npombos : integer;
  cond : condition;

procedure entra (pid : integer);
var i : INTEGER;
begin
  npombos := npombos + 1;
  if (npombos < 5)
  then
    begin
      writeln('Pombo ',pid,' na gaiola -> Pombos = ', npombos:2);
      delay(cond);
    end
  else
    begin
      writeln('Liberando pombos');
      for i:= 1 to 4
      do resume(cond);
      end;
    end;
end;

begin
  npombos := 0;
end; (* monitor gaiola *)

process type tpessoa ( pid : integer );
begin
  repeat
    sleep(7);
    cxpostal.coloca;
  forever
end;

process type tppombo ( pid: integer );
var i: integer;
begin
  repeat
    sleep(4);
    cxpostal.retira(pid);
    sleep(5);
    gaiola.entra(pid);
  forever;
end;
```

```

var
    p1, p2, p3, p4, p5 : tppombo;
    pessoa : array [ 1.. 8 ] of tppessoa;
    i: integer;
begin
    cobegin
        p1( 1 ); p2 ( 2 ); p3 ( 3 ); p4 ( 4 ); p5 ( 5 );
        for i := 1 to 8 do pessoa [ i ] ( i + 10 );
    coend;
end.

```

#### 4. EXERCÍCIOS

Desenvolver e executar as soluções dos seguintes problemas.

##### 4.1 Acesso a mesas num restaurante

Desenvolver um programa que represente o controle de acesso a mesas de um restaurante. Os clientes devem ser representados por processos que disputam as 12 mesas do estabelecimento. Utilize comandos de impressão para representar o estado dos clientes, os quais poderão estar caminhando ou comendo no restaurante. Ao fim da refeição, o cliente deverá liberar a mesa para permitir que outros clientes iniciem suas refeições.

##### 4.2 Controle de Acesso a um Arquivo por vários processos

Escrever e executar um programa que faça o uso de monitores para controlar o acesso a um arquivo da parte de diversos processos. A limitação para o acesso a este arquivo é que a soma dos identificadores dos processos não deve ultrapassar o valor 40.

##### 4.3. Produtor-Consumidor

O desafio é permitir que processos *produtores* e *consumidores* possam ter acesso, respectivamente, em escrita e leitura, num buffer de pequena capacidade de armazenamento.

Nesta solução com monitor, um único monitor, aqui denominado *Buffer*, resolve a questão através de duas operações, uma de escrita e a outra de leitura, denominadas *put* e *take*. A novidade neste exemplo é o uso de variáveis condição e das operações de bloqueio e desbloqueio de processos, *wait* e *signal*, sobre as variáveis condição *notfull* e *notempty*. Essas variáveis permitem controlar o acesso em escrita e leitura pelos processos, de modo a impedir, de um lado, que processos *produtor* armazenem mensagens no buffer se o mesmo estiver cheio e, por outro lado, que processos *consumidor*, tentem ler mensagem caso o buffer esteja vazio. Segue a listagem abaixo implementando esta solução.

##### 4.4. Gerenciamento de uma piscina

Neste primeiro programa, a preocupação é a alocação de recursos em exclusão mútua numa piscina onde um banhista só pode entrar para nadar após ter conseguido acesso a uma cabine e, antes disso, a um cesto para guardar suas roupas. Na área da piscina, estão disponíveis apenas 8 cestos e três cabines de modo que haverá competição pelos mesmos. A seqüência de execução de um banhista é chegar na piscina, acessar um cesto, entrar na cabine para trocar de roupa e entrar na piscina pra nadar; ao terminar de nadar, o banhista acessa uma cabine para se trocar e libera a cabine e o cesto de roupa.

