

**BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACION
PROGRAMACION CONCURRENTE Y PARALELA
OTOÑO 2022 – NRC: 60898**

MANUAL I DE EJERCICIOS PRÁCTICOS

Dr. Mario Rossainz López

H. Puebla de Z. a 8 de agosto de 2022

Para cada uno de los siguientes programas secuenciales, aplique las condiciones de Bernstein para encontrar su correspondiente versión paralela usando el par cobegin-coend, dibujando el grafo de precedencia. Posteriormente prográmelos en algún lenguaje de alto nivel.

```
1) begin
    cuad = x*x;
    m1= a*cuad;
    m2=b*x;
    z=m1+m2;
    y=z+c;
end.
```

$$2) \frac{a}{b} + \frac{c}{d} = \frac{(a*d) + (b*c)}{b*d}$$

3. Dado el siguiente pseudocódigo en paralelo, obtenga el grafo de precedencia correspondiente:

```
begin
  S0;
  cobegin
    S1;
    begin
      S2;
      cobegin
        S3; S4;
      coend;
    end;
  S5;
end;
S6;
coend;
S7;
end.
```

4. Reescriba en algún lenguaje de alto nivel (Java o C) el siguiente pseudocódigo paralelo que utiliza el par cobegin-coend para copiar un archivo de texto *f* en un archivo *g* carácter por carácter.

```
begin
  var
    f,g: file of txt;
    r,s: char;

  abrirLectura(f);
  abrirEscritura(g);
  leer(f,r);
  while not FinArchivo(f) do
    begin
      s=r;
      cobegin
        escribir(g,s);
        leer(f,r);
      coend;
    end;
    escribir(g,r);
  end.
```

5. Escriba un programa concurrente que lance 5 hilos en ejecución paralela. Inicialmente se deberá crear un arreglo de enteros de 100 casillas. Cada casilla almacenará un número entero generado aleatoriamente en un rango de 1 a 1000. A continuación, se dividirá el arreglo en 5 partes iguales y se asignará una de esas partes a cada hilo. Los hilos calcularán la suma parcial correspondiente la cual imprimirán en pantalla y una vez finalizados todos los hilos el main obtendrá la suma total sumando las sumas parciales de todos los hilos y la mostrará en pantalla.

6. Escribir un programa concurrente que lance 3 hilos en ejecución paralela de forma cada uno ordene un arreglo de números enteros (todos del mismo tamaño) y que se haya generado previamente, pero con un algoritmo de ordenación distinto. Un hilo deberá utilizar el método de la Burbuja, mientras que otro hilo utilizará el método de ordenación por inserción y el último hilo deberá ejecutar el algoritmo de Quicksort. Se deberá visualizar en pantalla la ordenación de cada hilo mostrando también el tiempo que tardo cada uno en ordenar dichos números.

7. En el siguiente programa paralelo escrito en Pascal Concurrente, se espera que como resultado de su ejecución se imprima 110 ó 50, sin embargo, dicho programa no es correcto en su ejecución. Reescriba el programa en algún lenguaje de alto nivel como Java o C para comprobar dicha aseveración.

```

program verbatim;
  var x: integer;

  process P1; (* PROCESO *)
  begin
    x:=x+10;
  end;
  process P2; (* PROCESO *)
  begin
    if x>100
    then write(x);
    else write(x-50);
  end;
begin (* PROGRAMA PRINCIPAL *)
  x:=100;
  cobegin
    P1;P2;
  coend
end.

```

8. Hacer un programa concurrente que muestre la ejecución de dos hilos independientes con prioridades distintas. Un hilo se deberá de ejecutar con una prioridad más baja que el otro. Los hilos deberán de incrementar sus contadores, hasta que el hilo que tiene la prioridad más alta alcance al contador que corresponde al hilo que tiene la prioridad más baja. SUGERENCIA: Inicialice el contador del hilo con prioridad baja con un número mucho mayor que el valor del contador del hilo con prioridad alta.