

Sistemas Operativos

Práctico 3

Curso 2024

Ejercicio 1 Se desea contar el número de veces que es ejecutado un determinado proceso y para esto se definen dos contadores globales:

```
var unidades, decenas : integer := 0
```

Que son usados de la siguiente forma:

```
procedure actividad is
begin
  for i in 1..12 loop
    actividad_propia_del_proceso
    unidades := unidades + 1;
    if (unidades = 10) then
      unidades := 0;
      decenas := decenas + 1;
    end if;
  end loop;
end procedure;
```

Se ejecutan **concurrentemente** dos copias de actividad:

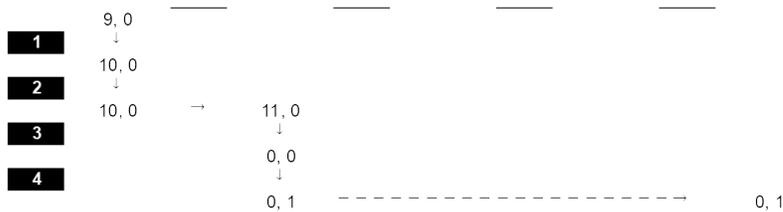
- Analizar la conducta del algoritmo e indicar algunos de los resultados posibles de los contadores.
- Indicar cómo podría obtenerse siempre el resultado correcto.

Solución: Si todo anduviera bien, al terminar de ejecutarse las dos copias del procedimiento actividad deberían quedar decenas = 2 y unidades = 4.

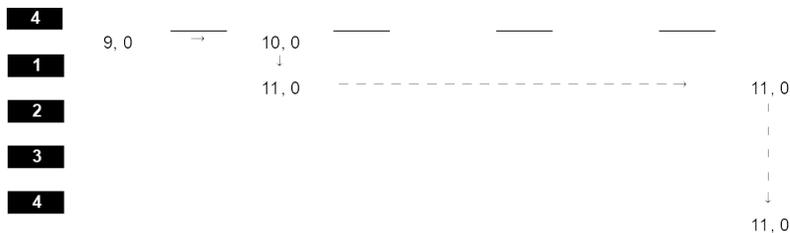
Veamos como sería el entrelazado en el punto crítico, es decir cuando las decenas = 0 y las unidades = 9 y la próxima instrucción para ambos procesos es incrementar las unidades (instrucción 1):

```
procedure actividad is
begin
  for i in 1..12 loop
    actividad_propia_del_proceso
1_ unidades := unidades + 1;
2_ if (unidades = 10) then
3_   unidades := 0;
4_   decenas := decenas + 1;
  end if;
  end loop;
end procedure;
```

El estado lo definimos como (unidades, decenas). Vamos a ver algunas ejecuciones posibles que generan resultados erróneos (no es un entrelazado completo, solo algunas alternativas):



En este caso, el proceso1 ejecuta la instrucción 1, incrementando el valor de las unidades a 10, luego ejecuta la instrucción 2 e inmediatamente después pierde el procesador. El proceso2 toma el procesador y ejecuta la instrucción 1, incrementando el valor a 11 (recordemos que el primero proceso sigue en el if de la instrucción 2). El proceso1 retoma el procesador, ejecuta la instrucción 3 y actualiza el valor de las unidades a 0 y luego con la instrucción 4 actualiza el valor de las decenas a 1. Cuando el proceso2 vuelve a ejecutar, las unidades están en 0, por lo que termina su ejecución. Esta ejecución obtiene un resultado equivocado (unidades = 0 y decenas = 1). En este fragmento de ejecución, las unidades deberían valer 1.



En este ejemplo, el proceso2 ejecuta la instrucción 1 actualizando el valor de las unidades en 10 y pierde el procesador. El proceso1 toma el procesador y ejecuta la instrucción 1, e incrementa el valor de las unidades, que ahora valen 11. El proceso2 retoma el procesador. Al ejecutar la instrucción 2, el valor de las unidades es 11, y entonces no entra a if ($11 \neq 10$) y termina su ejecución. Luego el proceso1 retoma el procesador y ejecuta la instrucción 2 donde sucede lo mismo que con el proceso2. En este caso el resultado final es unidades = 11 y decenas = 0, lo que es claramente erróneo.

En el entrelazado se ve que:

- Para la ejecución de los dos programas en distintos órdenes tenemos resultados distintos.
- Hay ejecuciones en las que no se actualiza las decenas haciendo que unidades quede en 11 y decenas en 0, no volviéndose a actualizar nunca.

El problema sucede cuando incrementó unidades a 10 y, antes de que se evalúe si es igual a 10, el otro proceso vuelve a actualizar unidades a 11. Cuando se evalúe la condición 'unidades = 10', ya será falsa.

Se da otro problema cuando

- el proceso1 evalúa y ve que es igual a 10,
- el proceso2 actualiza unidades una o más veces y
- el proceso1 pone unidades en 0, perdiéndose todas las actualizaciones del proceso2.

Otro punto crítico es cuando inicialmente tengo de decenas = 0 y unidades = 8: Después de hacer dos ejecuciones paso por la actualización de decenas. Haciendo el entrelazado se ve que ambos procesos actualizan decenas y también unidades.

Para obtener un resultado correcto, es decir obtener siempre unidades = 4 y decenas = 2, habría que no permitir que se ejecutara simultáneamente la parte de actualización de unidades y decenas en ambos procesos. Es decir forzar la mutua exclusión de esta parte de los procesos.

Una solución sería:

```
procedure actividad is
begin
  for i in 1..12 loop
    actividad_propia_del_proceso
    principio_mutua_exclusión;
    unidades = unidades + 1;
    if (unidades = 10) then
      unidades := 0;
      decenas := decenas + 1;
    end if;
    fin_mutua_exclusión;
  end loop;
end actividad;
```

Otra solución sería:

```
procedure actividad is
begin
  for i in 1..12 loop
    actividad_propia_del_proceso
    unidades = unidades + 1;
    principio_mutua_exclusión;
    if (unidades >= 10) then
      unidades := unidades - 10;
      decenas := decenas + 1;
    end if;
    fin_mutua_exclusión;
  end loop;
end actividad;
```