

PARADIGMAS Y LENGUAJES DE  
PROGRAMACIÓN  
GUÍA TP - UNIDAD 1  
*Diapositiva*

Ulises C. Ramirez

12 de Septiembre, 2018

## Versionado

Para el corriente documento se está llevando un versionado a fin de mantener un respaldo del trabajo y además proveer a la cátedra o a cualquier interesado la posibilidad de leer el material en la última versión disponible.

REPOSITORIO: *<https://github.com/ulisescolina/UC-PYLP>*

## Índice de Contenido

# 1 Ejercicio 1

## 1.1 ¿Qué diferencia existe entre la multiprogramación y el multiproceso?

Ambos describen una forma de compartir el tiempo de cómputo de un sistema por programas, que en última instancia ayuda a dar una explicación conceptual de lo que significa concurrencia. La diferencia la expuesta en [?] es la siguiente: La **Multiprogramación**, se da cuando los programas se ejecutan en un único procesador disponible y sus procesos internos comparten el tiempo de cómputo del procesador mencionado. Se habla de **Multiproceso**, cuando el sistema en el cual se ejecuta el programa posee múltiples procesadores, entonces es posible asignar diferentes procesos a diferentes procesadores.

## 1.2 ¿Qué es una instrucción atómica y qué es la intercalación?

Se considera una *instrucción atómica* [?] a aquella que se ejecuta completamente antes de que se ejecute ninguna otra instrucción de cualquier otro proceso del programa, una *intercalación* en un programa es una secuencia de ejecución de las instrucciones atómicas con las que cuente dicho programa, en el caso de que hayan 2 instrucciones atómicas ejecutadas por 2 procesos concurrentes, existieran 4 posibles intercalaciones.

## 1.3 ¿Considera que $a := a + 1$ una instrucción atómica? Justifique la respuesta

Para dar respuesta a ésta pregunta es necesario saber, como se especifica en [?], *qué instrucciones del lenguaje son realmente instrucciones atómicas*, para esto debemos saber primero de qué lenguaje de programación se está hablando a la hora de responder a la consigna, para propósitos prácticos se asume que se trata de PascalFC. La siguiente tabla, extraída de la cita realizada en ésta sección, indica cuáles son las instrucciones que debe realizar el lenguaje para realizar  $a := a + 1$ .

1	load a, R1
2	add R1, 1
3	store R1, a

Tabla 3.4: Instrucciones atómicas correspondientes a la instrucción  $a := a + 1$

Primero se carga el valor de  $a$  en el registro  $R1$ , acto seguido se procede a realizar la suma de lo que se encuentra cargado en el registro y la constante proveída, en este caso el valor 1 y se procede a guardar el resultado en el registro  $R1$ , y en última instancia se asigna lo que se tiene en  $R1$  a  $a$ .

## 1.4 ¿Qué podría pasar si dos procesos ejecutaran en paralelo la instrucción $a:=a+1$ ?

Las instrucciones atómicas que se discutieron en la Subsección ?? se pueden aplicar al problema  $a:=a+1$ , de esta manera las instrucciones atómicas que conformarían la operación serían las mencionadas en la Figura ??, entonces lo que puede suceder si dos procesos ejecutan de manera concurrente el mismo bloque con la variable compartida  $a$  los resultados podrían ser imprevistos, dado al fenómeno de la intercalación.

## 2 Ejercicio 2

CONSIGNA: investigar como instalar el lenguaje PascalFC en su sistema operativo. Lectura recomendada por la cátedra: página del Ing. John Coppens <http://jcoppens.com/soft/pfc2>

## 3 Ejercicio 3

CONSIGNA:

- Implemente el ejercicio de la diapositiva 10 en PascalFC.
- Al ejecutarlo varias veces, ¿Qué sucede con la salida del mismo?

## 4 Ejercicio 4

CONSIGNA: investigar:

- ¿Qué son los semáforos en la programación paralela?
- ¿Para qué sirven?
- ¿Cuáles son sus instrucciones y para qué se utiliza cada una?

### 4.1 ¿Qué son los semáforos en la programación paralela?

Los semáforos son una herramienta que se destina para la sincronización de procesos, en PascalFC estos son un tipo abstracto de datos, y como tal tiene operaciones y estructuras de datos internos. Todo semáforo tiene un contador que toma valores positivos, y una lista de procesos asociados.

### 4.2 ¿Para qué sirven?

Estos sirven para garantizar que recursos en el sistema que deben ser utilizados por un proceso a la vez, efectivamente sean utilizados por un proceso a la vez, este conjunto de instrucciones que acceden a las áreas mencionadas son denominados *sección crítica*.

### 4.3 ¿Cuales son sus instrucciones y para que se utiliza cada una?

Las operaciones que se pueden invocar sobre una variable de tipo semáforo son:

- **initial(s,v)** inicializa el contador del semáforo **s** al valor **v**. Este procedimiento solo se puede invocar una vez y debe ser llamado desde el programa principal. Debe ser inicializado antes de utilizarse.
- **wait(s)** Este procedimiento solo se puede invocar desde un proceso. Funcionamiento:
  - Si el contador **s** tiene un valor mayor que cero, el proceso continúa su ejecución y el valor del contador se decrementa en uno.
  - Si el contador del semáforo tiene el contador igual a cero, el proceso se queda bloqueado y se añade a la lista de procesos bloqueados del semáforo.
- **signal(s)** Este procedimiento solo se puede invocar desde un proceso. Funcionamiento:
  - Si no hay procesos bloqueados en el semáforo **s**, se incrementa el valor de contador en una unidad.
  - Si hay procesos bloqueados en el semáforo, se elige aleatoriamente a uno de ellos y se desbloquea para que continúe su ejecución.

## 5 Ejercicio 5

CONSIGNA: **investigue e implemente una solución en PascalFC con semáforos al problema de la salida detectada en el Ejercicio ??.**

## Referencias

[Gortázar Bellas, et al, 2012] GORTÁZAR BELLAS, FRANCISCO; MARTÍNEZ UNANUE, RAQUEL; FRESNO FERNÁNDEZ, VÍCTOR. *Lenguajes de Programación y Procesadores - Capítulo 3.5*. Editorial Universitaria Ramon Areces, Madrid, 2012. ISBN: 9788499610702.