

Algoritmos y Programación Paralela

Dra. Ing. Ana Cori Morón

Descripción

- El sistema mantiene un número (variable global) que refleja el número de turno del cliente que se está atendiendo en el instante actual. Los clientes deben esperar en una cola hasta que llega su turno. Una vez que se acaba con un cliente, la variable global se incrementa en uno y el cliente que tenga un boleto con ese número pasa a ser atendido. Cuando termina una compra, el cliente se desprende de su boleto y se dedica a realizar cualquier otra actividad libremente (guardar el dinero en la billetera, retirarse, etc.), ya que no se encuentra más en su sección crítica. Si más tarde quiere volver a comprar, tendrá que obtener un nuevo número.

ALGORITMO PARA N-PROCESOS

```
process  $P_i$ 
repeat
   $C[i] := \text{cognum};$ 
   $\text{numero}[i] := 1 + \max(\text{numero}[0], \dots, \text{numero}[n-1]);$ 
   $C[i] := \text{nocognum};$ 
  for  $j := 0$  to  $n-1$  do
    begin
      while ( $C[j] = \text{cognum}$ ) do;
      while (( $\text{numero}[j] \neq 0$ ) and (( $\text{numero}[i], i > \text{numero}[j], j$ ))) do;
    end;
    Sección Críticai;
     $\text{numero}[i] := 0;$ 
    Restoi;
  forever
```

ALGORITMO DE LAMPORT

- Es un algoritmo que soluciona el problema de la exclusión mutua para n procesos. Conocido también con el algoritmo de la panadería, es adecuado para entornos centralizados como para entornos distribuidos.
- $S[i]_{i:1..n}$: indica si el proceso i está cogiendo su número (true) o si ya termino de hacerlo (false)
- $nt[i]_{i:1..n}$: indica el numero de ticket cogido por el proceso i
- Inicializar
- $S[i]:\text{false}$
- $nt[i]=0$
- $N=\text{numero de procesos}$

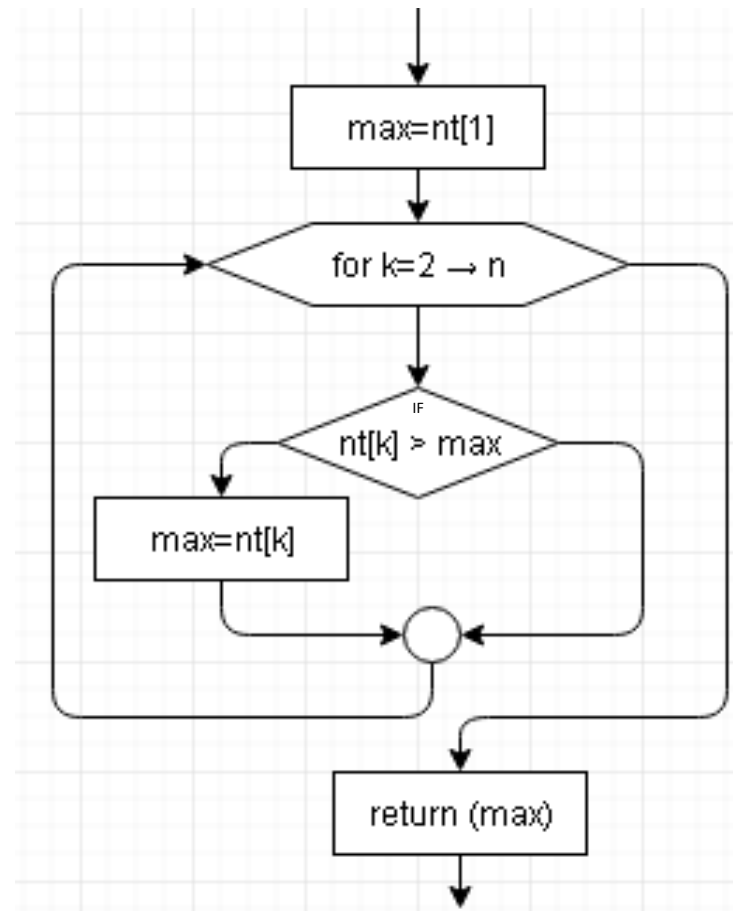
Variables en el Diagrama de flujo

- $S[i]_{i:1..n}$: indica si el proceso i está cogiendo su número (true) o si ya termino de hacerlo (false)
- $nt[i]_{i:1..n}$: indica el numero de ticket cogido por el proceso i
- Inicializar
- $S[i]:\text{false}$
- $nt[i]=0$

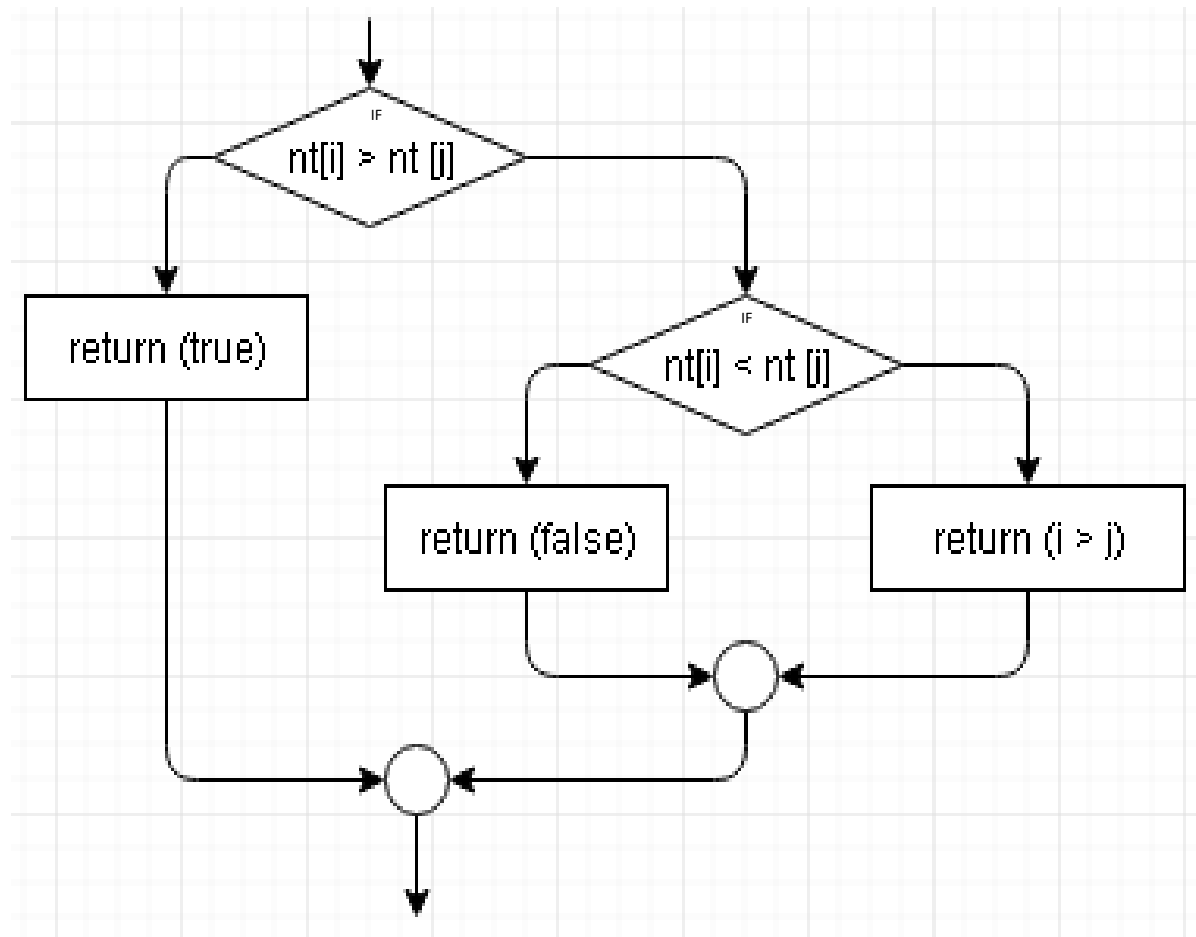
Variables en el libro

- $C[i]_{i:1..n}$: indica si el proceso i está cogiendo su número (true) o si ya termino de hacerlo (false)
- $numero[i]_{i:1..n}$: indica el numero de ticket cogido por el proceso i
- Inicializar
- $C[i]:\text{false}$
- $numero[i]=0$

FUNCIÓN ntmax()



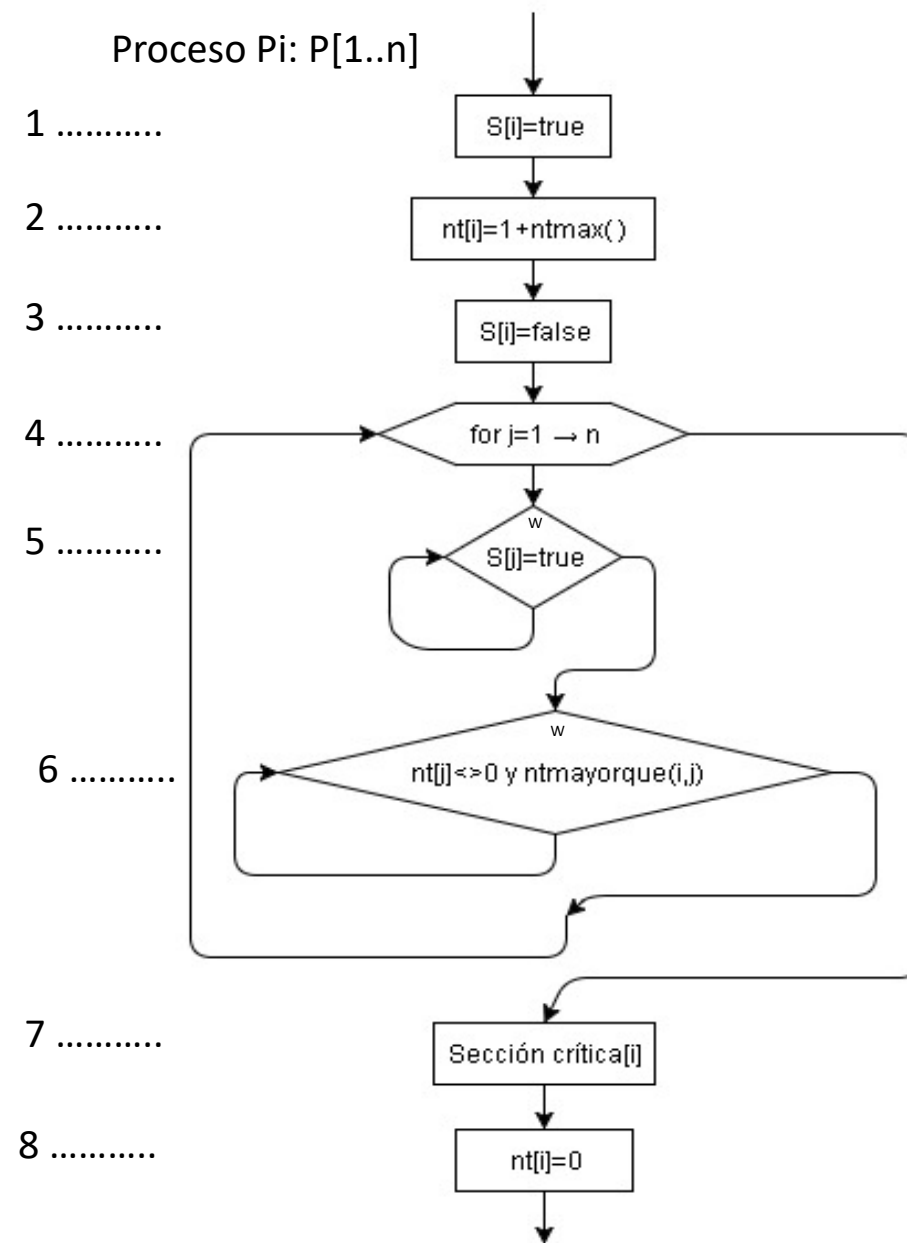
FUNCIÓN ntmayorque(i,j)

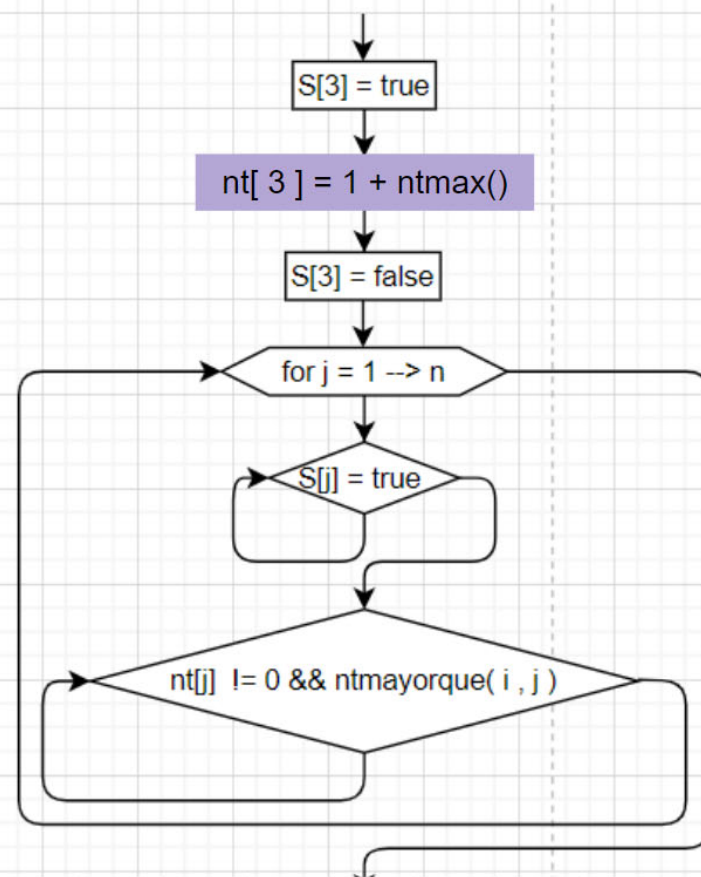
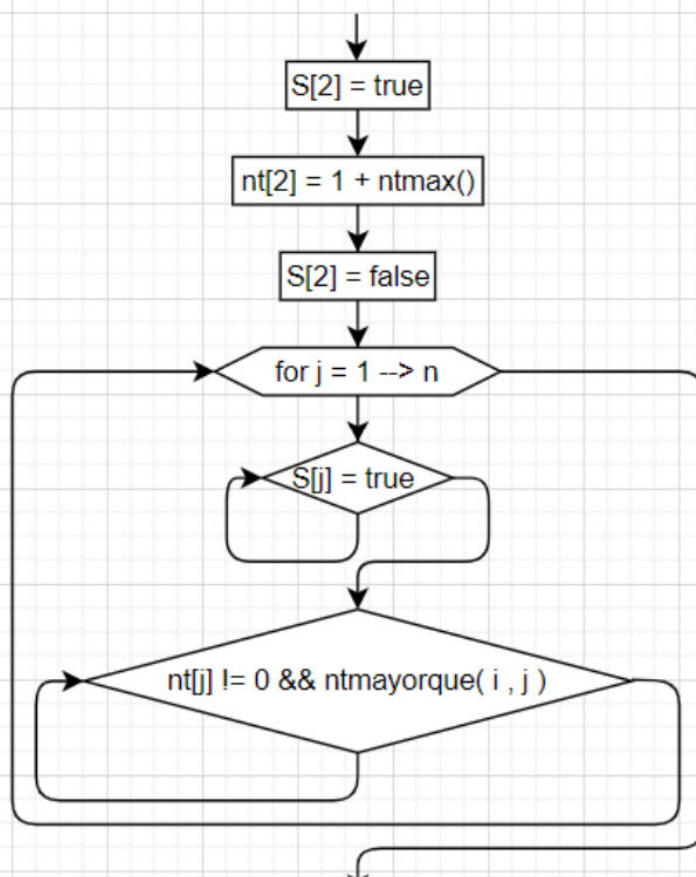
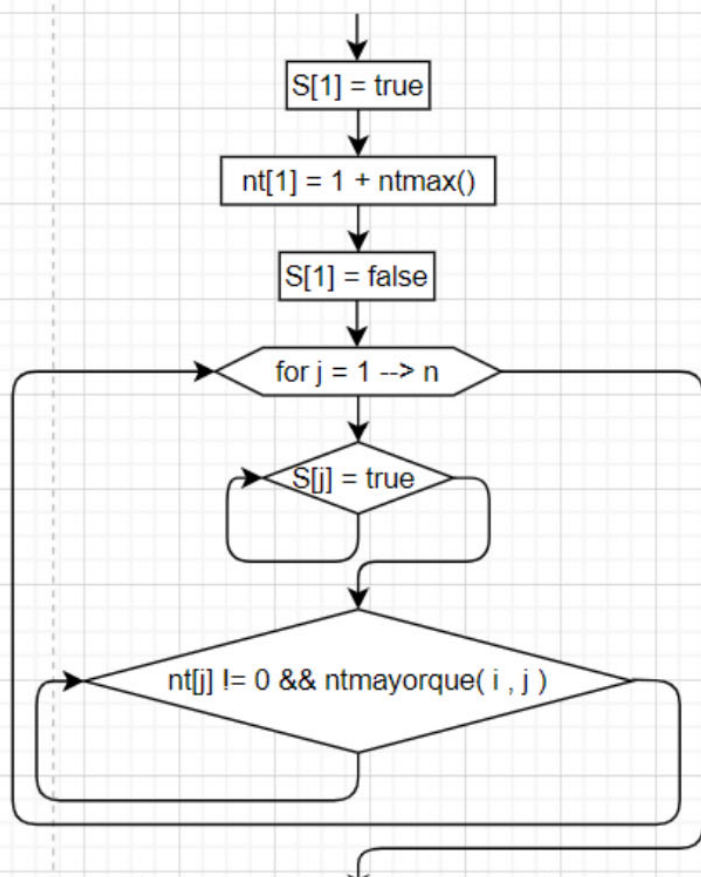


El algoritmo es correcto porque:

- 1. Satisface la condición de exclusión mutua ya que P_i solo estará en su sección crítica si se superó el segundo while.
- 2. Se satisface la condición de progreso en la ejecución debido a que cada proceso tiene un orden (nt) para ejecutarse.
- Un inconveniente es que los números que cogen los procesos pueden aumentar a lo largo de la ejecución, por lo tanto no debe ser usado en modelos de alto grado de concurrencia.

ALGORITMO DE LAMPORT





- <https://www.youtube.com/watch?v=1eynspijhFU&t=5s>
- <https://www.youtube.com/watch?v=1eynspijhFU>

Tarea

1. Algoritmo de Ricart y Agrawala, Grupos de 8 integrantes
2. Algoritmo de Berkeley, Grupo
3. Algoritmos de votación de Maekawa, Grupo
4. Algoritmo de Peterson para N procesos, Grupo
5. Algoritmo de Cristian,