Strtock(String,”separador de token“)

Teniendo en cuenta ese string y ese separador te da el primer token, si le llamas sucesivamente en un bucle, va obteniendo los sucesivos tokens

**Semáforo**

Dijkstra: Implementación de una  **estructura de datos de tipo abstracto** o una clase, que incluían una variable entera positiva compartida y dos primitivas de sincronización, P -Wait(No tiene que ver con la llamada al sistema) y V -signal

La variable solo se puede asignar a través de V y P, no puede ser asignada, saber su valor durante la ejecución…Se vio necesaria una variable de inicialización para darle un valor tras su creación.

Especificación wait:

Wait(s) {

While s <=0; { //No hacer nada

s--; //Si la variable es mayor que cero se deecrementa

}

Especificación signal

Signal(s) //Variable compartida, por lo que al asignarla para esto un proceso bloqueado por el while anterior pudiera desbloquearse

S++;}

-Posible implementación:

S-> estructura de datos que contiene

-Variable

-Lista de procesos bloqeuados

-Wait

-Signal

-Init

Wait(semáforo s) {

s.variable--;

if(s.varaible<0){

s.ListaBloqueados.InsertarProceso(P);

block(P);}

}

Signal(semáforo s){

s.variable++;

if(s.variable <= 0){

s.ListaBloqeuados.ExtraerProceso(P);

wakeup(P);

}

**Interacciones sobre semáforos**

Las operaciones wait y signal son atómicas e indivisibles

Las operaciones wait y signal se excluyen mutuamente

-Suceden secuencialmente en un orden desconocido

No se especifica cómo es la espera cuando un proceso se bloquea por un wait

No es posible saber si un proceso se bloqueará al invocar a wait

Si hay varios procesos bloqueados en un semáforo, no es posible saber qué proceso se desbloqueará tras un signal sobre ese semáforo

Tipos de semáforos

Binarios (mutex(Exclusión mutua))

Su variable tomará valores 0 o 1 (Los que usaremos)

Generales o contadores

Su variable no tiene restricciones en cuanto a los valores que puede tomar

Señalizadores de eventos

Problema de selcción mutua

Semáforo binario/mutex como cerrojoo

Init(mutex,1){

… (Lo perdí)

Gestión de acceso concurrente a las instancias de un tipo de recurso:

Semáforo general inicializado al número de instancias del recurso disponible

Wait(contador);

UsoDelRecurso();

Signal(contador);

Sincronización de eventos

El semáforo se inicializa a 0

Se realiza signal sobre el semáforo para señalizar/indicar que se ha producido el evento

Se realiza wait para esperar que se produzca el evento

Observaciones adicionales

Nunca es posible saber si un proceso se bloqueará en un wait

Nunca es posible saber qué proceso se desbloqueará tras un signal, si es que hay varios procesos bloqueados