Necesitamos herrami mas potentes qeu nos permi implementar la sincro de forma eficiente

Los semaforos, nos perm realizar sincro entre proc concu

Semáforo ⇒ instancia de un tipo de datos abstracto (o un objeto) con sólo 2 operaciones (métodos) atómicas:P y V

Internamente el valor de un semáforo es un entero no negativo:

• V → Señala la ocurrencia de un evento (incrementa).

• P → Se usa para demorar un proceso hasta que ocurra un evento (decrementa).

decrementa de forma atom

esto perminte protegeer secciones criticas, tb sinc por condicion, estas dos op que tiene la herram nos permiten los dos tipos de sincro.

podemos declarar arreglos matrices, de estossemanforos, sem siempre debe inicializarse. No podemos ni ver ni modif el valor q tenga el semafoto, si no lo inici podria tomar cualquier valor.

Solo lo inicializo en la declaracion!

• Semáforo general (o counting semaphore)

P(s): 〈 await (s > 0) s = s-1; 〉

V(s): 〈 s = s+1; 〉

La o P va a demorar hasta que el semaforo se indique que ha ocurrido algun evento, si es igual a cero dormido hasta q logra pasar, lograr que ese sea mayor a 0. Y luego de forma atom, decremento el cont del semáforo.

V solo decremento el valor, son atómica porque trabajan con el valor interno del seaforo.

• Semáforo binario

P(b): 〈 await (b > 0) b = b-1; 〉

V(b): 〈 await (b < 1) b = b+1; 〉

Pueden tomar solo valores 0 o 1, cambia la opV, no es bloqueante solo incrementa, no debe esperar, ahora si!

Semáforos generales no hay ningun orden en la espera sobre la operacion P, cuando se increm en 1 y toma valor 1 cualquiera de los que estaba esperando puede pasar, esto a veces hace algunas cosas mas complejas porque el orden debe hacerlo a mano para que se mantenga, adeams no se puede asegurar si uno utiliza el semáforo sin tener en ucneta eso podria ser que la oluc no cumple lo de eventual entrada,

Tecnicas basicas para resovler problemas de conc con semaforos, problema en la seccion critica, usar los sem para hacer e.m

en vez de marar si la secc crit esta ocupada marcado si esta libre, ahroa esperamo a que este libre, vamos a tener una funcion free, en vez de loop con valor falso, en las sentencias await esperamos a que free verdadero, cuando terminamos de usar la secc crit como q esta libre para qeu otro proc la utilce .

free con valor true, espero a qeu free true y atom falso, que esta ocupada-

bool free = true;

process SC[i=1 to n]

{ while (true)

{ 〈await (free) free = false;〉

sección crítica;

free = true;

sección no crítica;

}

}

0 falso

1 verdadero

int free = 1;

process SC[i=1 to n]

{ while (true)

{ 〈await (free==1) free = 0;〉

sección crítica;

free = 1;

sección no crítica;

}

}

Numero enteros, 1 ver, 0 falso,

Nuevamente cambiar esa expresion y conseguir una expresión equivalente

Tengo que esperar hasta que free tenga el valor 1, lo pongo en 0 y

free = a 1 es decir que free es mayor a 1 indicando que esta libre la secc crit

Los unicos posbles valores son 0 y 1, lo primero que cambio es la sent await decir q free es igual a 1 es lo mismo que decir uqe free es mayor que 0 y y como vamos a tomar 0 o 1, decir qeu free = 0 si logre psar esa op entonces lo decremento.

int free = 1;

process SC[i=1 to n]

{ while (true)

{ <await (free > 0) free = free - 1;>

sección crítica;

<free = free + 1>;

sección no crítica;

}

}

sem free= 1;

process SC[i=1 to n]

{ while (true)

{ P(free);

sección crítica;

V(free);

sección no crítica;

}

}

Entonces donde poniamos free en 1, es decir que increm en 1, equivale a la expresion de los semaforos, semanforo free con valor 1, el priemr awat donde demoro, es lo mismo uqe hacer un P sobre el semaforo

cuando necesitamos e.m o resolv un pro en secc crti, semaforo q lo inicializamos en 1, hacemos un p para demorar cuando este libre y luego un v para liberar la seccion critica.

Si huebiera estao en 0, nadie entrada ni liberraba la secc crti. En genral cuando es e.m el semaforo para la exclusion tiene q ser inici en 1.

El segundo problema es la de usar una barrera.

Barrera para los dos procesos.

Semaforos de inicialiacion, semaforos inic en 0 donde un proc señala que un evento ocurrio, v sobre ese semaforo y otr proc espera a qeu haya ocurrido con P,

en particular, tiene que aisar que llego hasta esa barrera, ocurrio un evento en su prop semaforo, señalizar q ha llegado hasta la barrera, y luego esperar a que el otro tb haya marcado que llego, con una op P, sobre el semafoto del otro proceso.

En general los sem, inic en 0 porqu ni bien comienza seguro q el evento q se espera no ha ocurrido, tb hay uqe tener en cuenta el orden del v y el p , si hacemos v y luego p , o primero p ambos quedan bloqueados.

Productor consumidor, semaforos binarios divididos

b1 a bn esos semaforos forman uno si s cuemple: la suma de todos estos semaforos es mayor que 0 y menor q 1.

Que la suma de entre 0 y 1 me va a permitir manejar la e.m, pero no que todos compiten por acc a una secc critica, c u espera en un em dif, prcoeso o grupo de proc. voy a poder acc a usar un recc con em pero esta compet no se da al mismo tiempo.

P sobre un sem que necesito y el v sobre otro, d est manera me va a permitit alternar el uso de la s.c un produc para dejar un elem y leugo un prod para dejar un elem.

alternar entre consu y prod

lo primero que debe ocurrir es q un produc deje un elem en el buffer, y un prod juntarlo, no de otra forma. Alterno entre calq de ls consu y prod, alternan los tiposs de procesos, entre todos los prod van a competir a ver quien entra, no va a haber una competencia, dos semáforos, vacio qeu va a indicar si el buffer esta vacio el buff lleno, vacio =1 y lleno =0

dividido poorque nunca los dos con un valor =1, jamas vacio y lleno al mismo tiempo.

competir con el resto de los prod, pero ademas esperar, se demora, cuando el prod logra pasar el p, deposita el elem q genero en el buffer luego avisa con la op V, del lado de los consumidores de forma semejante, para tomar un elemen esperar q el buffer lleno , desp acceso exclusivo a ese buff, post lo consumo. Desp un v sobre el buff vacio.

Lo unico q hacen dnetro del la secc crit es tomar o dejar el elem, jamas dentro genera el elem o consume el elem. El resto lo dejo afuera de la secc.

35.26

Jamas dentro de las s.c consume elem o consume, tengo que limitar la sec critica a lo que pueda generar interferencia. El resto lo tengo que dejar fuera para no reduir a interferencia. Esos dos semaforos

buffer lilmitado usar el semaforo como contador de recursos,admin el uso de recurso d multiples actividdades distintasposc del buffer limitado.

Determ tam maximo, un unico productor y un unico consumidor. Los deja en el buffer q actua como cla y el consumidor retira los elem y los consume, buffer actua como uan cola,

La sig vez qeu se deposita algo es en la pos 0, se mantiene como una cola circular.

Arreglo de n elem, dos enteros que son punteros a la pos a la primera ocupada, y cual es la primer lirbre donde dejo el sig elem, y lo qeu vams a necesitar, es saber si hay alguna pos de ese vector que este vacia, para esto vamos a tener dos semaforos, uno llamado vacio, va a contar la cnt de pos lbres, en el inicio n libres entonces inic en n , y otro semafoto con los llenos asi que ucnado arranca en 0 porque no hay ninguno lleno.

a que en el buffer haya al menos una pos vacia, donde espera haciendo un P en la pos vacia,

una vez que dejo los elem en el buffer ahora avisa que ha quedad una pos, contando un nuevo elemento.

Tiene q esperar, lgra pasar cuadno es mayor que cero, toma el dato, cuantan la cantidad de pos libres u ocupadas en ese vector, libres o vacias en el vector.

El problema es cuando en vez de un unico prod y un consumidor tenemos mas, para uno funciona bien pero si huiera mas, para mas prod y mas consumidores no me sirve.

si hay más el proc de retirar un elem lo hacemos con em dentro de una secc critica, con quien compite? con los del mismo tipo, tanto un consum como un prod al mismo tiempo si se puede dar, la em la manejamos por separado dentro de cada tipo de proceso.

Donde el prod dejaba un elem o tomaba un elem dentro de sec crit, dos nuevos semáforos,

en el caso de los 4 procesos podría ser que se genere deadlock, cuando trabajamos con problemas de este tipo eso hay q evitarlo.

Exc mutua selectiva, prob de los filosofos, un problema de em entre proces cada uno adquirir un conj que se comparten de alguna manera, 5 filósofos y 5 tenedores recurso independiente, compite solo con el de du derecha y su isquierda poor el tenedor por eso em selectiva.

Arreglo de semaforos para acceder a cada tenedor, 5 arreglos 5 sem cuando un filo necesita levantar uno de los tenedores llamara a P sobre el semaforo que le corresponde a l tenedor. SE podria generar deadlock si todos adquieren primero el derecho y desp el izqueierdo, cuando cada uno quiera adquirir el otro no va a poder porque esta ocupado. ¿Como podemos hacer la soluc usando un semaforo sin el deadlock?

espera circular, si yo rompo ese atributo entonces no va a haer posib de que haya deadlock

todo los fil toman en primer intancia el tenedor izq y desp el derecho, alcanza con q el primero lo haga al reves para evitar el deadlock. El que no logra ganar el primer tenedor, seguro quede algun filosofo que tome ambos, entonces evitamos deadlock enalgun momemtno alun filosofo podra tomar los dos tenedores.

**Lectores y escritores qeu comparten una BD**

Cada escritor de forma exclusiva sobre la Bd en cambio lector mientras no haya un escritor, los procesos trabajan de forma diferente, es mas complejo resoler este tipo de problema,

soluc con em : los escrit necesitan acc a la bd, pero los lect tb como grupoo acceso exclusivo, tanto los lectores como los escritores deben accder a través de un P, leugo de pasar el p , la bd esta libre,, la utilzan y luego la liberan haciendo un P sobre ese mismo semaforo, una vez que uno entro el rsto puedee entrar sin necesidad de trabajar con ese semaforo. El primer lector qeu se va, no debe hacer un V, el unico que deberia hacer ese V para liberar la seccion critica es el ultimo lector, para que cualquier escritor pueda acceder de forma exclusiva.

Sincron por condicion

Ver a quien dejamos pasar en cada caso, passing de buttom nos brinda em en acceso a las variables q controlan el acceso a las bd, ademas despertar los procesos demorador de acuerdo a las cond q se den, await q hemos usado, cual es el proceso q puede seer despertado dos vrriables q indican la cantidad de lectores o escrit. desp restringir la cantdiad, llos lectores se demoran hasta q no haya ningun escrito para increm la cantidd, cuando sale decrementa la cantidad.

tecc passing de botón

hay muchos casos en los que las sentencias await pueden implementarse usando semáforos, las condiciones de espera de los distintos await son condic dif y además superpuestas.