Si o si mayor o igual que cero

sem nombre y lo inicializo si o sí cuando lo declaro.

sem espera [5] = ([5] 1) → Los cinco inicializados con valor 1

A partir de que fue declarado lo único que puedo hacer con el semáforo, es llamar a una op P o V

P lo que hace es demorar el proceso hasta qeu el valor sea > 0 , de form atómica lo que hace es decrementar el valor en 1, en forma atómica.

No pueden dos procesos al mismo tiempo hacer el P, va a poder avanzar al P y se demora hasta que pueda acceder, son atómicas. P demora un proceso hasta que sea mayor a 0

P(s) → <await (s>0) s=s-1; >

V(s) → < s:= s +1; >

Sinc por condición, sinc em, y se utiliza para las dos a la vez.

Si hubiese varios proc, esperando a hacer un P sobre un semaforo, no hay orden si hubiera muchos, no es que se encolan, van a estar esperando, cuando algun otro hace el B y se incremento, cualquiera es el q va a acceder. Inconvenientes: no asegurar la eventual entrada, siempre uno podría perder.

Hay C chicos y hay una bolsa con caramelos que nunca se vacía. Los chicos de a UNO van sacando de a UN caramelo y lo comen. Los chicos deben llevar la cuenta de cuántos caramelos se han tomado de la bolsa.

estructura del programa, variables compartidas, en este caso la bolsa de caramelos y la variable del total,

no sabemos si hay dos chicos incrementando la cantidad, interferencia, necesitamos proteger el incremento, dentro de una SC, para solucionar esto, un semaforo llamado mutex para admin el acceso a la seccion critica, mutex en 1 , de entrada cuadno comienza el prog la secc critica está libre, que ocurre si mutex en 0, el problema es q ninfuno podria accceder, toos los que llegan ninguno va a lograr avanzar y se podruce un bloqueo permsnnete, l inicializamos en 1, y nos tenemos q asgu q siepre el P , sc, y luevo el v, ninguno puede hace el v si no estaba en la SC.

sc para no generar una inter sobre cant, esto nos asegura q mas de un chico no tome de la bolsa al mismo tiempo?? solo controlo cant, ais que no, solo protege el increm de cantidad, otro semaforo para excluir tomar caramelos.

MAL → Entran a SC a tomar un caramelo, lo toman, liberan V mutex dos y luego sc para el incremento. Podrian tomarse mas caramelos q ls que se han tomado.

Dos cosas relac entre si, debe pertenecer a la misma secc critica, inclur ambas cosas, tomar el caramelo e increm la cantidad, utilizamos un solo semaforo dnde hago todo.

Ahi si incluye en la misma sc las doscosas, pero que ocurre? comemos el caramel entr ode la secc critica, nadie puede tomar hasta q no termine,

comer el caramelo fuera de la sc. Maximixar la concurrencia, de lo mas impoortante, e cuales son las secciones criticas, y que es lo qeu hay que hacer dentro de la sc, si son hechos relacionados entre si, con las mismas varaibles, o no , todo dentr de la mimsa seccion critica, sino, no puede alguien modificar ese rec compartido, lo otro improtante es siempre dentro de una secc critica hay que hacer lo minimo e indispensanble,

Hay C chicos y hay una bolsa con caramelos que nunca se vacía. Los chicos de a UNO van sacando de a UN caramelo y lo comen. Los chicos deben llevar la cuenta de cuántos caramelos se han tomado de la bols Solo n caramelos, cantidad limite de caramelos.

Partiendo de la soluc, lo primero a ver es qeu modificar, los chics ya no en un while true, sino que cantidad = n , hay uno solo, y dos o mas chicos chequean la condicion del while, a todos verdadero, desp del chequeo cada uno de ls chicos va a intentar acceder a la cond critica, de todos estos el primero q etre va a tomar el ultimo, desp bolsa vacia e incrementando, entonces estamos teniedno problema porque se chequea la cond de corte sin proteccion, el chequeo de la cond dentro de una secc critica.Chequeo la cond desp cierro la sc, desp hago, dentro del mismo conj de acc atom, dentro del P y V, del semaforo, la forma mas sencilla, protegemos todo dentro de la secc critica, protegiendo el cheuqueo de la condición final, un unico chico q va a comer los n.

Protegemos todo, antes de la condicion del while y desp de la condicion, al salir, como ambas cosas estaban relacionadas, funcionaria bien, cual es el problema? uno va a comer los n caram, el resto va a esperar y no va a haber mas caramelos, cumple con lo de que sea prot el chequeo de la condición y el incremento, pero solo un chico ocme los caramelos, que deberíamos hacer? agarrar los caramelos, libero la sc, luego comer, mientras le da lugar a cada chico, corremos el v, desp de la varaible cantidad, asi cuando comienza un P de mutex, cheuqea la cond, increm la variable,

se toma el caramelo yylueogo se libera, sin proteccion solo funciona par el primero. Deberia luego de comer el caramelo, P de nuevo, comi el caramelo, termino, accedo a la sc, hago el chequeo de forma protegida,

Hay que cerrar toda la seccion critica, una ez que salgo el while debo liberrarla, para que otro pueda acceder y vea qe no hay caramelos en la bolsa , al salir V(mutex)

Tercer ejercicio, abuela que admin el acceso a la bolsa, ciando los chicos llegan se habilita la abuela para que elija e forma aleatoria y deja pasar a tomar caramelos, barrera esperando a que los chicos lleguen, variable compartida contador, llevando la cuenta de la cntidad de chicos q van llegando, cada chico incrementar el contador, si cant=0 los despierta y a la abuela, sino duermo hasta q llegue el ultimo.

necesitamos un semáforo para manejar la em, o secc crtica, como vamos a demorar esos chicos hasta qeu lleguen todos? Abuela demorada, commo la despierto?

tres semáforos, forma exclusiva ocn la variale contador, inic en 1

otro semáforo inci en 0 para dept a la abuela

y el tercero en 0 tb hasta que lleguen todos los chicos. v espera abuelaafuera del for, auqne no parece.

v mutex, antes qeu el P barrera, SIEMPRE DE ESA MANERA

Libero la secc crit y desp demoro, sino bloqueo permanente de todos lo procesos, que pasa en este for? no hay forma de ver el valor de un semáforo, apra ver si es mayor que 0, sol P y v, como voy a tener c procesos, c operaciones v sobre un mismo semáforo,

seguir en falso para avisar que no hay mas caramelos, una vez que pasa la abuela esperar hasta q lo despierte, espera chico en su semáforo privado, chequea en un while la cond seguir, si seguir es verdadero hay caramelos puedo pasar, en ningun momento estamos asegurando que haya un chico a la vez tomando caramelos en a bolsa, la abuela deberia esperar a qe el chico haya terminado, a tomar un caram, cuando el chico se fue, despertar a otro para uqe tome un caramelo, sinc el principo cuando l avisa al chico cuando puede esar el recurso, y sinc el chico con la abuela para avisar uqe termino de usar el recuro.

Para esto agrego un nuevo semaforo, porque la abuela debe demorar hasta que el chico le dice q ya tomo el caramelo, demorar en un p hasta que el chico le diga que ya tomo el caramelo, la unica manera qeu nos aseguramos q no haya ds persons tomando el caramelo, siempre q un proce admin un rec compartido cada vez q le da permiso alguien para accceder tienee q esperar para poder darselo a otoro. primer asegurar q termino de usarlo, el rec compartido aca es la bolsa, agregamos un for dnde despertamos a los c chicos para que puedan agarrar caramelos de la bolsa.

Ejercicio 4-

Para trabajar con la cola, con em, vamos a P(mutex) cola v( mutex)

para encolar y desencolar bloquea la sec critica, el proceso seridor tiene que hacer un pop del elem de salida, lo mismo bloque y libera,

poro de cola vacia? chequeo si la cola esta vacia o no. asi no se generará error.

si no esta vacia, entonces pop . partir de semáforos esta prohibido usar busr waiting, como resuelvo entonces? tener otro semáforo, qeu el proceso servidor se demore en otro semáforo hasta que la cola no esté vacia, no es un semáforo que va a tener valor 0 o 1, po cada pedido necesito increm el valor del semáforo, el que voy a uar es uno ue actua como contador de recursos,cantidad de recursos disponibles, pedidos pendentes qeu hay en la cola, cantidad de elem en la cola, espero en el semaforo demoro, hasta q haya al menos un pedido, semáforo pedidos inicia en 0, que hace cada cliente cuando termina de usar su pedido, si hubiera estado dormido el servidor con ese aviso lo despertara y asi poder acceder al servidor de la cola, demorado en el P hasta q haya uno pendiente, no preguntamos desp del pop si esta o no la cola , el viso de que hay un elem mas en la cola SI O SI desp de haber hecho el push, no se puede hacer previo, si lo hago antes del p de mutex, le estaría aislando al servidor q hay un elem y antes de dejarlo podra avisar, solicitudes encoladas, cantidad de elementos en la cola, el acceso por em, y utilizo un semáforo extra, donde el uqe hace el push avisa q deja un elem mas y el otro se asegura q hay un elem en la cola, una vez q el servidor ya tomó su pedido, lo toma y tiene q devolver el resultado al cliente que hizo el pedido, no le puedo avisar a cualquiera q esta un resultado sino que al que hizo el pedido, vamos a usar un arreglo, cada pos para un cliente en particular, arreglo de semáforos privados para indicarle a cada uno que hizo ese pedido y a su vez le devuelve algún valor, resultado del analisis para un cliente en particular, arreglo resultados, y un arreglo de semáforos espera inici en 0, donde debe demorarse el clietne esperando el resultado, demorarse en su semáforo privado hasta q el servidor lo despierta, una vez q saca el elem de la cola, aca no aviso, porque al estar en el vector cada pos es independiente, resultado de ese pedido en otro lugar interf si esta atendiendo un pedido del mismo cliente, deberá dejar el resultado en la misma pos del vector, el clietnne va mandando solic, hasta no recibir el resultado de un pedido hasta no haber mandado otro,

Alumno y maestra

Dos procesos y una interacción

LA maestra le asigna al alumno la tarea a realizar

el alumno va a esperar a que le den la tarea, cada uno espera algo propio, por medio de un semáforo, para cada alumno, entonces voy a tener uno para cada.

en 0, significa que los alumnos lo primero que haga es quedarse esperando un evento, el id de cada alumno un P sobre el semaforo que está inic en 0.

p demora a un proceso hasta que el valor sea mayor que 0 -

v incrementa el valor interno del semaforo-- no tiene demora de esperar a qeu una condi se cumpla.

si un proc tuviera qeu demorarse hasta qeu algo ocurra, P, sino avisar que algo ocurrio un V.

manejar sinc por cond, por exclusion mutua, hacer secciones critica y también una meacla de ambas cosas, si hubiese varios procesos que esperan hacer un p, si hubiese 3 o 4 no hay ningun hombre, no es que pase el primero uqe llego, NO HAY ORDEN, esto puede no asegurar la eventual entrada podria ser qe siempre uno de los proc pierda, habira qeu hacerlo a mano, si uno tuviera uqe permitir en p que pase de acuerdo al orden hay qeu implementar algo por sw, como passing the buttom .

hay c chico y una bolsa de caramelos que nunca se vacia, debe llevar ka cuanta de cuantos caramelos se han tomado, procesos uqe forman parte del prog concurrente r compartido de los chicos, los caramelos y ademas cuantos se han tomado en total.

como hacemos una seccion ciritica en lo que seria semaforos?

mutex en 1 porque la sc al rpincipio esta libre, debe poder pasar, si el semaforo lo inic en 0, ningun procesos va a poder acceder, y se produciria un bloqueo permanente,

{

P(mutex)

cant := cant +1

v(mutex)

--comer caramelo

}

siempre p , secc crítica y luego v. si mutex menor que uno el v al terminar una seccion crítica.

esto nos asegura que mas de un chico no pueda estar tomando caramelos al mismo tiempo?

esa seccion citica solo protege la cantidad no progete si hay mas de un chico, vamos oa tener dos ops,

1op otro semaforo para excluir tomar caramelos.

entran a la sc, pm2, tomar, vm2 y luego sc p inc v , esto esta mal porque el valor de canti podría no ser el que deben tomar, el inc de cant debe ir relacionado a tomar el caramelo, esta relacionado entre si cuando se saca un caramelo inc cant, la sc tienne q incluir varias cosas. solucion utiilizando un unico semafoto donde hago todo.

pero para que comeria el caramelo dentro de la seccion cirita , tomo y decrement en la sec. pero lo come fuera. no maximizo la concurrencia sino

el increm de cant y ,la condic deben estar dentro del mismo conj de acc atom, dentro del p y v, lo resolvemos la forma mas sencilla todo dentro de la sc, como ambAS cosas estaban relacionadas pertenecen a la misma sc y estaba bien, hay un unico chico qe come un caram, no maximixa la concurrencia, los caramelos de a uno,, cada chico agarra libera y desp come, mientras le da lugar a que otro entre, cada vez qeu saco un caram debo liberar la sc para q otro entre, asi cuando el proc comienza chequea la cond toma el caram, incrme y libera, sirve solo para uno, el p de mutex entonces va adentro .}

V de mutex siempre antes que el P barrera sino bloqueo permanente de todos los procesos.

**Exp Semáforos**Se inicializan al momento de la declaración.

P y V son atómicas.

**P.** Demora a un semáforo hasta que el valor sea mayor que 0, en forma atomi lo decrem en 1.

**V.** Atómicamente lo incrementa en 1. Avisar que algo ocurrió usamos un V.

Sincronización por condición, exclusión mutua y ambas.   
  
No hay orden de prioridad entre los semáforos, podría no asegurar la eventual entrada.