INFORME PROYECTO SISTEMAS OPERATIVOS-2023

Aravena Joaquín – De Battista Agustín – Comisión 15

1.1 Procesos, threads y Comunicación

Banco: la implementación del banco simula la llegada de clientes de diferentes tipos, cuya elección es al azar. En primer lugar pasan por una mesa de entrada, la cuál tiene una capacidad máxima, y a partir de allí son distribuidos hacia 3 filas, una por cada tipo las cuáles también cuentan con capacidad limitada, en la que esperan a ser atendidos. Al finalizar su atención, se retiran. Además, se encuentran disponibles empleados, todos ellos pueden atender a clientes de tipo político, que además deben ser tratados con prioridad, y luego cada uno se dedica a un tipo específico, ya sea clientes comunes o empresas. Estos empleados verifican si tienen trabajo y sino descansan hasta ser despertados por el arribo de un nuevo cliente.

Cada una de estas capacidades fue modulada con constantes definidas para mantener un código flexible.

1. La implementación de los clientes y empleados es con un hilo por cada uno, la exclusión mutua en la mesa de entrada es modelada con un mutex y cada una de las 3 filas es implementada con un semáforo. Los empleados también son sincronizados con un mutex para no permitir que un cliente sea atendido por más de un empleado.
2. C
3. C

Minishell:

1.2 Sincronización:

Secuencias: fueron implementadas en sus respectivos archivos, cabe aclarar que la secuencia “ABABCABCD” podría haber sido modelada con la utilización de un único “print(C)”, pero llevaría a una secuencia de waits y signal entre los 4 semáforos que no creemos que sea justificada por la simplicidad de la solución propuesta.

Reserva de aulas: Resolvimos el problema utilizando un hilo para cada alumno, el cuál ejecuta la función “decidir” cuatro veces, modelando la elección aleatoria de las 4 operaciones. Para el acceso a la tabla de reservas se garantiza exclusión mutua utilizando un semáforo binario. Las reservas son siempre de 1 hora, es decir que la primera reserva posible es a las 9hs, y la última es a las 20hs.

Operaciones:

* Reservar: utiliza una hora generada aleatoriamente, pide acceso a la tabla y cuando es obtenido verifica si la hora está disponible, en el caso de estarlo la reserva usando su número de alumno.
* Cancelar: busca en la tabla si el alumno tiene reservada alguna hora, en el caso de tenerla pide acceso a la tabla y cuando es obtenido borra la reserva más temprana del día. Si no tenía reservada ninguna hora esto es mostrado por pantalla
* Consultar: genera una hora aleatoria, accede a la tabla sin necesidad de pedir acceso y muestra por pantalla si dicha hora está libre o fue reservada.

Constantes:

* NUM\_HORAS: utilizado para modular cuantas horas del día a partir de la primera hora pueden ser reservadas
* PRIMERA\_HORA: utilizado para modular a qué hora del día arrancan las reservas
* LIBRE: representa que una hora no está reservada.
* NUM\_HILOS: utilizado para modular la cantidad de hilos que son creados, cada hilo representando a un alumno.

2.2. Problemas Conceptuales

1. Considere un sistema de gestión de memoria basado en paginación. El tamaño total de la memoria física es de 2 GB, distribuido en páginas de tamaño 8 KB. El espacio de direcciones lógicas de cada proceso se ha limitado a 256 MB.

Tamaño de página = Tamaño frame: 8KB, offset: log2(8KB) = 13

Cantidad de páginas: 2GB / 8KB = 262.144 = 2^18, 18 bits para direccionar la página

a) Determine el número total de bits en la dirección física.

Log2(2GB) = 31 bits, 18 bits para direccionar la página y 13 bits para direccionar dentro de la página.

b) Determine el número de bits que especifican la sustitución de página y el número de bits para el número de marco de página.

Sustitución: dirty y referencia

Cantidad de páginas: 2GB / 8KB = 262.144 = 2^18, 18 bits para direccionar la página

Cantidad de frames: 256MB / 8KB = 32.768 = 2^15, 15 bits para direccionar al marco de página.

c) Determine el número de marcos de página.

Número de marcos de página: 256MB / 8KB = 32.768

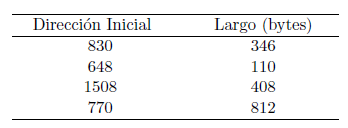
d) Determine el formato de la dirección lógica.

Log2(256MB)= 28 bits de dirección lógica.

Cantidad de frames = 2^15, por lo que se necesitan 15 bits para direccionar al frame

Tamaño de frame= 8KB = 2^13, por lo que se necesitan 13 bits para direccionar dentro del frame

Dirección lógica: |15b (direccionar frame)|13b (offset dentro del frame)|

2. Para cada una de las siguientes direcciones lógicas, determina la dirección física o indica si se produce un fallo de segmento:

a) 0, 228: segmento 0, 228 < 346, dirección física: 1058

b) 2, 648: segmento 2, 648 > 408, hay Segmentation Fault

c) 3, 776: segmento 3, 776 < 812, dirección física: 1546

d) 1, 98: segmento 1, 98 < 110, dirección física: 746

e) 1, 240: segmento 1, 240 > 110, hay Segmentation Fault