

SISTEMAS OPERATIVOS

Segundo Cuatrimestre de 2020

Proyecto

1. Experimentación de Procesos y Threads con los Sistemas Operativos

1.1. Procesos, threads y Comunicación

1. CONJUNTO DE TAREAS. Un proceso coordinador recibe un conjunto de tareas que debe organizar su ejecución por ciclo. La cantidad de tareas que recibe en cada uno de los ciclos oscila entre 4 y 6.
 - Cuando recibe 4 tareas: 2 corresponden con la TAREA A y las otras 2 con la TAREA B.
 - Cuando recibe 5 tareas: 2 corresponden con la TAREA A, 1 corresponde con la TAREA B y las otras 2 con la TAREA C.
 - Cuando recibe 6 tareas: 2 corresponden con la TAREA A, 2 con la TAREA B y las otras 2 con la TAREA C.

Hasta que no terminan todas las tareas recibidas en el ciclo no comienza un nuevo ciclo.

Las tareas realizan las siguientes actividades.

- TAREA A: esta tarea se encarga de realizar trabajos de pintura sobre un vehículo. Cuando se requiere una tarea de este tipo se le informa el color y el tipo de trabajo. El tipo de trabajo puede ser parcial o total. Un trabajo de tipo parcial requiere una unidad de tiempo para realizarse y en un tipo total requiere tres unidades de tiempo.
- TAREA B: esta tarea se encarga de realizar trabajos en los frenos de un vehículo. Cuando se requiere una tarea de este tipo se le informa el tipo de trabajo. El tipo puede ser verificación o reparación. En el caso de la verificación requiere de una unidad de tiempo para realizarse y una reparación requiere 2 unidades de tiempo.
- TAREA C: esta tarea se encarga de realizar trabajos sobre las ruedas de un vehículo. Cuando se requiere una tarea de este tipo se le informa el tipo de trabajo. El tipo puede ser reparación de una llanta o rotación y balanceo. En

el caso de la reparación requiere de una unidad de tiempo para cada una de las ruedas que deba reparar y una rotación y balanceo requiere 3 unidades de tiempo.

- a) Diseñe e implemente una solución que optimice el uso de los recursos. Describa la solución que ha implementado.
 - b) Considere que la forma de comunicación entre los objetos (procesos-threads) es con pipe.
2. MINI SHELL. Construir un shell que acepte un conjunto limitado de comandos de Unix. Tiene que considerar como mínimo 6 comandos. Explique las opciones de diseño que consideró al momento de implementarlo. No puede invocar los comandos mediante la función system y para la implementación de los mismos debe utilizar llamadas al sistema (system call) ó funciones de librerías.

Los mínimos comandos que debe tener son los siguientes

- a) Crear un directorio.
- b) Eliminar un directorio.
- c) Crear un archivo.
- d) Listar el contenido de un directorio.
- e) Mostrar el conenido de un archivo.
- f) Mostrar una ayuda con los comandos posibles.
- g) Modificar los permisos de un archivo. Los permisos son de lectura, escritura y ejecución.

1.2. Sincronización

Nota: Recuerden NO utilizar variables globales para la sincronización.

1. DEMASIADAS BOTELLAS DE LECHE.

El problema *Demasiadas botellas de leche* modela a dos compañeros de alojamiento que comparten un refrigerador y que, como buenos compañeros, se aseguran de que el refrigerador esté siempre bien abastecido de leche. Con compañeros de habitación tan responsables, el siguiente escenario es posible:

	Compañero 1	Compañero 2
3:00	Mira el refrigerador; no hay leche	
3:05	Va al supermercado	
3:10	Llega al supermercado	Mira el refrigerador; no hay leche
3:15	Compra leche	Va al supermercado
3:20	Llega a la casa; guarda la leche	Llega al supermercado
3:25		Compra leche
3:30		Llega a la casa; guarda la leche
3:35		oh no!

El problema surge porque se deben cumplir las siguientes propiedades:

- i. **Seguridad.** Nunca más de una persona compra leche.
 - ii. **Vivacidad.** Si se necesita leche, eventualmente alguien la compra.
- a) Resuelva el problema utilizando hilos (threads) y semáforos para su sincronización.
 - b) Considere que tiene n compañeros de alojamiento y las mismas propiedades se deben garantizar. Resuelva el problema utilizando procesos y colas de mensajes para la comunicación.
2. COMIDA RÁPIDA.

En un restaurante de comida rápida hay 30 mesas, cada una con un solo asiento. Tres cocineros preparan la comida (almuerzo fijo) y lo añaden a una cola de capacidad 10. Un camarero toma una comida de la cola y la entrega a un cliente que ha pedido una comida. Un limpiador limpia la mesa después de que un cliente se va. En el problema se tienen que respetar las siguientes condiciones:

- Inicialmente hay 50 clientes, pero solo 30 mesas disponibles. Un cliente debe esperar hasta que un cliente anterior termine la comida y su mesa esté limpia.
- Un cocinero no comenzará a preparar una comida si no queda espacio en la cola.
- Cuando un cliente pide una comida, el camarero irá a la cola, tomará una comida de la cola y se la entregará al cliente. A continuación, el cliente puede empezar a comer.
- Cuando no hay pedidos de los clientes, o cuando la cola está vacía, el camarero tiene que esperar.
- Después de que un cliente termina la comida y se levanta de la mesa, el limpiador limpiará la mesa.

(I) Utilice semáforos para sincronizar los hilos de cliente, cocinero, camarero y limpiador.

(II) Explique el comportamiento que presenta su solución. ¿Encuentra algún problema?

(III) Considere la siguiente modificación en la cual se tienen varias opciones en el menú. Un cliente puede elegir: menú con carne o menú vegetariano. Para resolver esta versión utilice colas de mensajes para la comunicación y sincronización de los elementos (procesos-hilos).

2. Problemas

2.1. Lectura

De los siguientes artículos elegir uno para leer y contestar las preguntas.

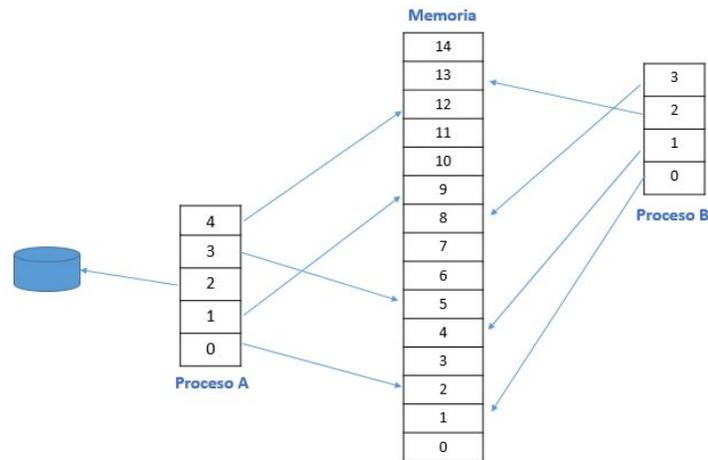


Figura 1: Asignación de Memoria

1. A partir de la lectura del artículo:

”Android Operating System Architecture”

Realice los siguientes puntos. La evaluación de este punto incluirá el orden y elaboración de las ideas como la presentación de las mismas.

- a) Describa los componentes de la arquitectura.
 - b) Identifique elementos que son representativos para este tipo de sistemas operativos.
 - c) Realice un comentario general del artículo (al menos 500 palabras).
2. A partir de la lectura del artículo:

”Evolution of Cloud Operating System: From Technology to Ecosystem”

Realice los siguientes puntos. La evaluación de este punto incluirá el orden y elaboración de las ideas como la presentación de las mismas.

- a) Describa similitudes y diferencias con los tipos de sistemas operativos presentados en clase.
- b) Realice un comentario general del artículo (al menos 500 palabras).

2.2. Problemas Conceptuales

1. La figura 1 muestra la asignación de páginas en un sistema Unix que ejecuta los procesos A y B. Las páginas son de 4 KB.
 - a) Suponga que el proceso B invoca fork y que el kernel utiliza la estrategia copy-on-write para implementar fork. Realice un nuevo diagrama para la asignación

Nombre	tipo	fecha	Nro. Bloque
Actividades.txt	F	29-09-2020	3

FAT

1		16	
2		17	
3	15	18	
4		19	
5		20	
6		21	
7		22	
8		23	
9		24	
10		25	
11		26	
12		27	
13		28	
14		29	
15	*	30	

Figura 2: Tabla de Archivos -FAT

de páginas después de invocar fork y luego de que el proceso B modificó la página 2.

- b) Construya la tabla de páginas para el padre y el hijo indicando los atributos de cada página (bits de válido y escritura).
2. En la figura 2 se representa una tabla FAT. Al borde de sus entradas se ha escrito, como ayuda de referencia, el número correspondiente al bloque en cuestión. También se ha representado la entrada de cierto directorio. Como simplificación del ejemplo, suponemos que en cada entrada del directorio se almacena: Nombre de archivo/directorio, el tipo (F=archivo, D=directorio), la fecha de creación y el número del bloque inicial. Tenga en cuenta que: - el tamaño de bloque es de 512 bytes - el asterisco indica último bloque - todo lo que está en blanco en la figura está libre.
- a) Completa la tabla FAT con los siguientes datos.
 - a) Creación del archivo Act-Labo1.txt con fecha 10-10-2020, y tamaño de 120 bytes.
 - b) Creación del archivo Act-Labo2.txt con fecha 15-10-2020, y tamaño 1200 bytes.
 - c) El archivo Actividades.txt aumenta de tamaño, necesitando 2 bloques más.
 - d) Creación del directorio Prueba, con fecha 15-10-2020, y tamaño 1 bloque.
 - e) Creación del archivo Informe con fecha 18-10-2020 y tamaño 2 kBytes.
 - b) Si utiliza un Mapa de Bits para la gestión del espacio libre, especifique la sucesión de bits que contendría respecto a los bloques dados. ¿Cuánto espacio ocuparía?
 - c) Si utiliza una lista enlazada para la gestión del espacio libre, indique cuánto espacio requerirá. Si se pierde el primer puntero de la lista de espacio libre, ¿podría el Sistema Operativo reconstruirla? ¿Cómo?

3. Suponga que un proceso, **P1**, abre el archivo *Actividad.txt* en modo lectura/escritura y otro proceso, **P2**, abre el mismo archivo y con el mismo modo, y a continuación crea un proceso hijo que abre el archivo */usr/pepe/Informe* en modo lectura/escritura. Represente toda la información relevante sobre el estado de las tablas de descriptores de archivos, tabla de archivos y tabla de i-nodos después de dichas operaciones.

Indicaciones

- Los experimentos deben realizarse en lenguaje C.
- Las pruebas deben realizarse sobre el sistema operativo especificado para las distintas actividades de laboratorio.
- Se debe entregar los fuentes realizados para cada una de las experiencias debidamente identificados y un informe con los resultados obtenidos y los problemas resueltos. Además se debe facilitar el compilado y ejecución de cada una de las experiencias por medio de un script o make y describiendo la forma de ejecución.