

## SISTEMAS OPERATIVOS

Segundo Cuatrimestre de 2022

### Proyecto

## 1. Experimentación de Procesos y Threads con los Sistemas Operativos

### 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.

### 1.1. Procesos, threads y Comunicación

1. PLANTA DE RECICLADO. Considere una planta que gestiona el reciclado de residuos. La planta cuenta con recolectores que empaquetan y envían lo que colectan al sector de clasificación de residuos. En este punto, se identifica cada paquete y se derivan a los distintos puntos donde los recicladores los procesan. Los residuos tratados pueden ser vidrio, cartón, plástico o aluminio. Asuma 3 recolectores que empaquetan cualquiera de los 4 tipos de residuos, 2 clasificadores que reciben paquetes de cualquier tipo y envían a los recicladores correspondientes. De estos últimos funcionan uno por tipo. Si el reciclador no tiene nada que reciclar puede optar por colaborar con otro reciclador, o detenerse a tomar mate.
  - a) Resuelva el problema considerando que la forma de comunicación entre los componentes (procesos) es con pipes.
  - b) Resuelva el problema considerando que la forma de comunicación entre los componentes (procesos) es con colas de mensajes.
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. SECUENCIA. Dada la siguiente secuencia:

AB(DoE)AC(DoE)FAB(DoE)AC(DoE)FAB(DoE)AC(DoE)F

- a) Resuelva el problema utilizando hilos (`threads`) y semáforos para su sincronización. Recuerde hacer uso eficiente de los recursos como por ejemplo la cantidad de semáforos que utiliza.
  - b) Implemente el mismo problema utilizando Procesos y pipes para su sincronización.
2. PUENTE DE UNA SOLA MANO. Diferentes vehículos provenientes del norte y del sur esperan para cruzar un puente de una sola mano. Los vehículos que circulan en la misma dirección que el que se encuentra cruzando, pueden acceder al puente al mismo tiempo. Sin embargo, aquellos que lo quieren hacer en dirección opuesta, deben esperar a que no haya otro vehículo sobre el puente.
    - (I) Resuelva el problema utilizando hilos y para la sincronización semáforos..
    - (II) Explique el comportamiento que presenta su solución. ¿Encuentra algún problema?
    - (III)Amplie la solución anterior para asegurar que todo auto que se encuentre a la espera de cruzar el puente, eventualmente lo haga. Resolver el problema utilizando procesos y para la comunicación y sincronización colas de mensajes (procesos-hilos).

## 2. Problemas

### 2.1. Lectura

A partir de la lectura de alguno de los siguientes artículos, genere alguna propuesta multimedia para "promover/"vender/"presentar"la idea. Puede realizarse en formato Flyer,

Podcast, presentación o formato similar. Debe representar las ideas más importantes del artículo y ser breve (de 3 a 5 minutos).

1. artículo:  
"Allied Telesis AlliedWare Plus Operating System"
2. artículo:  
"EOS: The Next Generation Extensible Operating System"
3. artículo:  
"Virtualization Overview"

## 2.2. Problemas Conceptuales

1. Un sistema paginado tiene páginas de 512 palabras, una memoria virtual de 512 páginas numeradas de 0 a 511, y una memoria física de 16 marcos de página numerados de 0 a 15. El contenido actual de la memoria física es la siguiente:

Physical Memory	
0	free
1536	page 34
2048	page 9
	free
3072	page table
3584	page 65
	free
4608	page 10
	free

- a) Asumiendo que la tabla de páginas contiene solo números de marco de 4-bits (en lugar de la dirección completa de memoria física que es de 13-bits), muestre el contenido actual de la tabla de páginas.
- b) Muestre el contenido de la tabla de páginas después de cargar la página 49 en la ubicación 0 y la página 34 es reemplazada por la página 12.
- c) ¿Qué dirección física es referenciada por las direcciones virtuales 4608, 5119, 5120, 33300?
- d) ¿Qué sucede cuando la dirección virtual 33000 es referenciada?

2. El siguiente diagrama muestra una parte de varias estructuras de dato de un sistema de archivos. Estas están organizadas de la siguiente manera. Cada entrada en la tabla Current Directory, contiene el nombre simbólico de un archivo, junto con el índice correspondiente en la tabla de File Descriptors. Cada descriptor de archivo registra la longitud (en bytes) del archivo y el número del primer bloque en el disco. Los bloques de disco de un archivo son asignados contiguamente; Se asume la longitud de cada bloque en 256 bytes. Por ejemplo, el archivo test tiene un descriptor en el índice 19, que muestra que el archivo ocupa 280 bytes, iniciando en el bloque 8 del disco y continuando hasta el bloque 9. La tabla de archivos abiertos (OFT) lleva el control de todos los archivos abiertos. Cada entrada contiene el índice de la tabla de descriptors y la posición actual dentro del archivo (para un acceso secuencial). Actualmente el archivo abc se encuentra abierto (descriptor 16). Su posición actual es 55, lo que indica que 55 bytes de este archivo han sido leídos o escritos

*Current Directory:*

<i>symb. name</i>	<i>descr. index</i>
...	...
free	
xx	20
abc	16
test	19
free	
...	...

*File Descriptors:*

	<i>file length</i>	<i>disk block#</i>
...	...	...
15	free	
16	100	5
17	free	
18	free	
19	280	8
20	550	40
...	...	...

*Open File Table*

	<i>descr. index</i>	<i>curr. pos.</i>
...	...	...
7	free	
8	16	55
9	free	
...	...	...

*Disk:*

	5	6	7	8	9	
	xxx	free	free	xxxxxxxx	xxx	
	256 bytes					

Para cada una de estas operaciones, muestre todos los cambios a la estructura de datos:

- Abrir el archivo test: `fd=open(test)`
- Buscar la posición 60 del archivo abierto test: `seek(fd,60)`
- crear un archivo nuevo llamado new: `create(new)`
- abrir el archivo new: `fd=open(new)`
- escribir 30 bytes en el archivo new: `write(fd, buf,30)`
- cerrar el archivo new: `close(fd)`
- borrar el archivo new: `delete(new)`