

PRACTICA 4: Threads, Administración de Memoria y File systems**THREADS****Ejercicio 1:**

Explique con sus palabras por que el cambio de contexto a nivel procesos es más “costoso” que a nivel de threads.

Ejercicio 2:

Un proceso posee tres threads, que ocurre si:

- El proceso realiza un fork?
- El proceso realiza un exec?

Administración de Memoria**Ejercicio 3:**

Enuncie las diferencias entre los esquemas de Particiones Estáticas o Fijas y Particiones Dinámicas

Ejercicio 4:

Considerando el uso del Sistema de Colegas y un bloque de memoria de 1 GB

Mostrar los resultados mediante un gráfico (como el de la presentación) luego de la siguiente secuencia de solicitudes y retornos: Proceso A 70MB, Proceso B 300MB, Proceso C 35MB, Retorna A, Proceso D 130MB, Retorna B, Retorna D, Retorna C.

Ejercicio 5:

¿Cual es la diferencia entre una dirección física y una virtual?

Ejercicio 6:

Explique los pasos que realiza el Administrador de Memoria cuando se produce un fallo de página.

Ejercicio 7:

¿Por que es indeseable la hiperpaginación?

Ejercicio 8:

¿En que consiste el concepto de Working Set?

Ejercicio 9:

Considerando la siguiente serie de referencias a páginas

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

¿Cuántos fallos de página ocurrirán con los algoritmos de reemplazo de página

LRU

FIFO

Óptimo

suponiendo la existencia de 3, 5 y 7 marcos de página?

Sistema de Archivos**Ejercicio 10:**

¿Para que es útil un sistema de archivos?

Ejercicio 11:

¿En que consiste el *journaling*?

Ejercicio 12:

¿Cuales son las operaciones típicas que se pueden realizar sobre archivos y directorios?

Ejercicio 13:

En un file system montado en / y otro montado en /home se pretende establecer un enlace simbólico con el siguiente comando:

```
$ ln -s /home/usuario/carta.txt /carta_de_usuario
```

Es decir, que `carta_de_usuario` referencie a `carta.txt`.

¿Esto es posible? ¿Por qué?

Ejercicio 14:

¿Cuales son los derechos de acceso típicos que se pueden conceder o denegar a un usuario sobre un archivo? Explique en que consisten las Access Control Lists y como complementan estos derechos de acceso.

Ejercicio 15:

Cual es el tamaño máximo de archivo que puede referenciar un i-nodo (tenga en cuenta los bloques de datos apuntados por los punteros simple, doble y triplemente indirectos).

PTHREADS

Ejercicio 16:

Implementar utilizando *pthread*s el siguiente algoritmo (solución de Hyman).

```
#define FALSE 0
#define TRUE 1
typedef int bool;
```

Variables compartidas:

```
bool q[3]; /*inicialmente en FALSE cada elemento, q[0] no utilizado*/
int turno = 1;
```

```
Proceso1 () {
    ...
    q[1] = TRUE;
    while(turno != 1) {
        while(q[2] == TRUE)
            /*Nada*/ ;
        turno = 1;
    }
    /*SECCION CRITICA*/
    q[1] = FALSE;
    ....
}
```

```
Proceso2 () {
    ...
    q[2] = TRUE;
    while(turno != 2) {
        while(q[1] == TRUE)
            /*Nada*/ ;
        turno = 2;
    }
    /*SECCION CRITICA*/
    q[2] = FALSE;
    ....
}
```

Use este algoritmo como se emplea en

Ejercicio 17:

Implementar semáforos N-arios utilizando elementos tipo *pthread_mutex_t*. Ver implementación dada en el material de clases.

Ejercicio 18:

Implementar el problema del productor y el consumidor utilizando *pthread*s.

Ejercicio 19:

Implementar el problema del barbero dormilón utilizando *pthread*s.

Ejercicio 20:

Implementar el problema de los lectores y escritores utilizando *pthread*s.

Realizar 3 versiones:

- (a) Favoreciendo a los lectores
- (b) Favoreciendo a los escritores
- (c) Tratando de que sea justa (equilibrado)