
 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Examen de la convocatoria ordinaria 18 de julio de 2011</p>	
---	---	---

**ATENCIÓN:**

- Lea atentamente todo el enunciado antes de comenzar a contestar.
- Dispone de 2:30 horas para realizar la prueba.
- No se podrán utilizar libros ni apuntes, ni calculadoras de ningún tipo.
- Los teléfonos móviles deberán permanecer desconectados durante la prueba (apagados, no silenciados).
- Solamente se corregirán los ejercicios contestados con bolígrafo. Por favor no utilice lápiz.

**APELLIDOS:**

**NOMBRE:**

**NIA:**

**GRUPO:**

**Teoría [2 puntos]**

a) Indique que diferencias existen entre los siguientes mecanismos de comunicación:

- Tuberías
- Memoria compartida

Tuberías	Memoria compartida
<ul style="list-style-type: none"> <li>• Semi duplex (comunicación en un solo sentido)</li> <li>• Solo se pueden utilizar entre procesos emparentados, que tengan un ancestro en común.</li> <li>• Los datos son escritos por un proceso en un extremo de la tubería y leídos por otro proceso desde el otro extremo del mismo.</li> <li>• La lectura es destructiva.</li> </ul>	<ul style="list-style-type: none"> <li>• Comunicación asíncrona</li> <li>• No hace uso del sistema en las operaciones de E/S</li> </ul>

b)

Enumere las ventajas e inconvenientes de las bibliotecas dinámicas.

- Ventajas:
  - i. Menor tamaño ejecutables
  - ii. Código de rutinas de biblioteca sólo en archivo de biblioteca
  - iii. Procesos pueden compartir código de biblioteca
  - iv. Actualización automática de bibliotecas: Uso de versiones
- Inconvenientes:
  - i. Mayor tiempo de ejecución debido a carga y montaje
    - 1. Tolerable: Compensa el resto de las ventajas
  - ii. Ejecutable no autocontenido
- Uso de bibliotecas dinámicas es transparente
  - i. Mandatos de compilación y montaje igual que con estáticas

c)

Defina brevemente los siguientes conceptos

- exclusión mutua
- condiciones de carrera
- sección crítica

Concepto de exclusión mutua.

Consiste en que un solo proceso excluye temporalmente a todos los demás para usar un recurso compartido de forma que garantice la integridad del sistema.

Concepto de condiciones de carrera:

Situaciones en las que dos o más procesos leen o escriben en un área de memoria compartida y el resultado final depende de los instantes de ejecución de cada uno.

Esto se soluciona impidiendo que más de un proceso acceda simultáneamente a las variables compartidas. Se soluciona garantizando la exclusión mutua.

Concepto de sección crítica.

Es la parte del programa con un comienzo y un final claramente marcados que generalmente contiene la actualización de una o más variables compartidas.

Para que una solución al problema de la exclusión mutua sea válida, se tienen que cumplir una serie de condiciones:

- Hay que garantizar la exclusión mutua entre los diferentes procesos a la hora de acceder al recurso compartido. No puede haber en ningún momento dos procesos dentro de sus respectivas secciones críticas.
- No se deben hacer suposiciones en cuanto a la velocidad relativa de los procesos en conflicto.
- Ningún proceso que esté fuera de su sección crítica debe interrumpir a otro para el acceso a la sección crítica.
- Cuando más de un proceso desee entrar en su sección crítica, se le debe conceder la entrada en un tiempo finito, es decir, que nunca se le tendrá esperando en un bucle que no tenga final.

d)

Enumere las ventajas del uso de procesos ligeros en un SO

- Capacidad de respuesta.
  - i. Mayor interactividad al separar las interacciones con el usuario de las tareas de procesamiento en distintos hilos.
- Compartición de recursos.
  - i. Los hilos comparten la mayor parte de los recursos de forma automática.
- Economía de recursos.
  - i. Crear un proceso consume mucho más tiempo que crear un hilo ( Ejemplo: en Solaris relación 30 a 1).
- Utilización sobre arquitecturas multiprocesador.
  - i. Mayor nivel de concurrencia asignando distintos hilos a distintos procesadores.
  - ii. La mayoría de los sistemas operativos modernos usan el hilo como unidad de planificación.

## Ejercicio 1 [2 punto]

Un sistema operativo utiliza un planificador cíclico (round-robin). En un instante determinado no hay ningún trabajo en ejecución y se desean ejecutar trabajos cuyos tiempos de llegada al sistema son los siguientes:

Proceso	Tiempo de llegada al sistema	Tiempo de ejecución
A	0	3
B	1	7
C	2	5
D	4	3
E	6	4

Las prioridades son inversas al valor que tienen. Así un proceso con prioridad 1 es prioritario respecto a otro con prioridad 2 o 3.

Se pide rellenar las siguientes tablas en los siguientes casos:

- a) Política de planificación SJF (Shortest Job First).
- b) Política de planificación round-robin con rodaja de 4.

NOTA: Si la rodaja de ejecución de un proceso termina en el mismo instante que llega un nuevo proceso al sistema, entonces el nuevo proceso se coloca en la cola de listos para ejecutar antes que el proceso que le expira la rodaja.

a)

Política de planificación SJF (Shortest Job First)

A																									
B																									
C																									
D																									
E																									
Tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

b) Política de planificación round-robin con rodaja de 4

A																									
B																									
C																									
D																									
E																									
Tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

## Ejercicio 2 [2 puntos]

Codifique usando procesos y tuberías la siguiente línea de mandatos:

```
cat foo.txt | wc -l > foo2.txt
```

```
int main (int argc, char *argv[]) {
    int fd[2];
    pipe(fd);
    if (fork() == 0) {
        close(0);
        dup(fd[0]);
        close (fd[1]);
        close (fd[0]);
        close(1);
        open("foo2.txt", O_RDONLY);
        execlp ("wc", "wc", "-l", NULL);
    }
    else {
        close(1);
        dup(fd[1]);
        close (fd[0]);
        close (fd[1]);
        execlp ("cat", "cat", "foo.txt", NULL);
    }
}
```



### Ejercicio 3 [2 puntos]

Suponiendo que estamos en un sistema de ficheros ext3 (Linux). El i-nodo dispone de:

- Punteros directos a bloques (4).
- Puntero indirecto simple.
- Puntero indirecto doble.
- Puntero indirecto triple.

Además, el tamaño de bloque es de 1024 bytes y el tamaño de un índice es de 4 bytes.

Suponga para todos los casos que sólo se encuentra precargada en memoria la información del i-nodo.

Responda a las siguientes preguntas de forma justificada:

- a) Escribir un programa que lea 20 KB con acceso a nivel de 1 KB del un fichero foo desde la posición 4 KB.

```
int main (int argc, char *argv[]) {
    int size = 0;

    char buffer[1024];

    int s;

    int fd = open(foo, O_RDONLY)

    lseek(fd, 4*1024, 0);

    while(size != (20*1024)){
        s = read(fd, buffer, 1024);
        size += s;
    }
}
```

- b)

¿Cuántos accesos a disco son necesarios para leer el bloque que se encuentra en la posición 10 KB?

c) ¿Cuántos bloques de índices y de datos usa un fichero que ocupe 1MB?

d) ¿Cuál es el tamaño máximo de un fichero en este sistema de ficheros?

#### Ejercicio 4 [2 puntos]