



Calificación
1
2
3
4
5

**Sistemas Operativos**  
**Examen parcial, 11 de mayo de 2002**

Nombre	Titulación

*Duración del examen: 2 horas y media*

---

**1** (2'5 puntos) Responda con brevedad y precisión a las siguientes preguntas:

- ¿Qué ventajas tiene poder declarar hilos dentro de un proceso pesado?
- El intérprete de órdenes (*command interpreter*) puede ser una aplicación independiente (ej. UNIX), o puede estar implementado dentro del núcleo del sistema operativo (ej. CP/M). ¿Qué ventajas o inconvenientes observa usted en cada alternativa?
- ¿Por qué es necesario establecer dos modos de ejecución (modo supervisor y modo usuario)?
- ¿Qué diferencia hay entre las llamadas al sistema y los programas del sistema?
- Si un sistema no es multiusuario, ¿tampoco es multitarea?

---

**2** (1'25 puntos) Se propone una nueva política de planificación de procesos, que consiste en ceder siempre la CPU al proceso que menos memoria necesita.

- a) ¿Esta política ofrece algún beneficio?
- b) ¿Y algún efecto negativo?
- c) ¿Es implementable?

---

**3** (2'5 puntos) Tenemos un sistema concurrente con una capacidad de memoria de N bytes. Cada proceso del sistema, para poder ejecutarse, debe reservar previamente un número de bytes. Si no hay cantidad suficiente de memoria, debe quedarse bloqueado hasta que otros procesos vayan liberando memoria suficiente. La ejecución del proceso debe estar regulada mediante un monitor, como se muestra a continuación:

```
moni tor. reserva (nbytes);  
    ... se ejecuta el proceso ...  
moni tor. devuelve (nbytes);
```

Implemente las operaciones de este monitor. Para escribir su solución, puede usar un monitor clásico o bien cerrojos y variables condición.

---

4 (1'25 puntos) Dada la siguiente carga de trabajo, obtener el diagrama de Gantt, el tiempo medio de retorno y el tiempo medio de espera al aplicar las siguientes políticas de planificación:

- Primero el más corto (SJF) con expropiación.
- Round-Robin con cuanto de 3 u.t.

Proceso	Tiempo de llegada	Duración
P0	0	9
P1	2	5
P2	3	2
P3	5	4

---

5 (2'5 puntos) A continuación se muestra la implementación de los semáforos en Nachos:

```
void Semaphore::P()
{
    IntStatus oldLevel = interrupt->SetLevel(IntOff);
    while (value == 0) {
        queue->Append((void *)currentThread);
        currentThread->Sleep();
    }
    value--;
    (void) interrupt->SetLevel(oldLevel);
}

void Semaphore::V()
{
    Thread *thread;
    IntStatus oldLevel = interrupt->SetLevel(IntOff);

    thread = (Thread *)queue->Remove();
    if (thread != NULL)
        scheduler->ReadyToRun(thread);
    value++;
    (void) interrupt->SetLevel(oldLevel);
}
```

- ¿Qué significa el **ReadyToRun()** de la operación V()?
- ¿Qué tipo de cola es el atributo **queue**? ¿FIFO? ¿LIFO? ¿con prioridad?
- ¿La implementación de los semáforos en Nachos utiliza espera activa?
- ¿Por qué hace falta un **while** en la implementación de la operación P()? ¿No puede sustituirse por un **if**?