



Universidad de Las Palmas de Gran Canaria  
Facultad de Informática

Sistemas Operativos  
Convocatoria de junio, 26 de junio de 2002

## SOLUCIONES

Calificación
1
2
3
4
5
6

**Nombre**

**Titulación**

--	--

**IMPORTANTE:** Las seis preguntas suman 12 puntos. Usted deberá descartar al menos UNA de las preguntas 3, 4, 5 y 6. Es decir, usted puede optar por contestar estos grupos de preguntas: 12345, 12346, 12356, 12456. No se admitirá ninguna otra combinación.

Dispone de tres horas y media para completar el examen.

**1** (2 puntos) - Responda, justificando sus respuestas, a las siguientes cuestiones sobre Nachos:

A. La ejecución de la función  $Y_i$  en  $d()$ , ¿provoca siempre el cambio de estado del hilo que la invoca?

No. Sólo provoca un cambio de estado (de RUN a READY) si la cola de hilos en estado READY no está vacía.

B. ¿Cómo está estructurado internamente un fichero que contiene un programa de usuario ejecutable de Nachos?

Contiene tres partes. Primero un bloque cabecera que describe como están estructuradas las dos bloques que le siguen que son: bloque de código y bloque de datos con valores iniciales. El bloque cabecera dice que tamaño y dónde empieza en el archivo cada uno de estos bloques.

C. ¿Por qué debemos utilizar un compilador cruzado para compilar un programa de usuario de Nachos y un compilador normal (no cruzado) para compilar los distintos módulos de núcleo?

Debido a que los módulos del núcleo se ejecutan en el sistema nativo de la máquina (sistema Linux, por tanto requieren un compilador "normal") y los programas de usuario se ejecutan en una máquina basada en un microprocesador MIPS (que se simula, por tanto se requiere un compilador "cruzado").

**2** (2 puntos) Diseñe una solución al primer problema de los lectores y escritores (prioridad para los lectores), con el requisito añadido de que no puede haber más de N lectores simultáneos. Como herramienta de sincronización, utilice monitores.

La solución consistirá en el algoritmo que ha de ejecutar un proceso lector, y el que ha de ejecutar un proceso escritor. El código puede estar escrito en cualquier especificación algorítmica.

```

type Lectores_Escritores = monitor
var
  Lect, Esc: condition;
  nlec: entero;
  escribiendo: booleana;

procedure entry Empezar_Lectura()
begin
  nlec:=nlec+1;
  while escribiendo or nlec>N do
    Lect.Wait;
end;

procedure entry Terminar_Lectura()
begin
  nlec:=nlec-1;
  if nlec>=N then
    Lect.Signal;
  else if nlec=0 then
    Esc.Signal;
  end if;
end;

procedure entry Empezar_Escritura()
begin
  while escribiendo or nlec>0 do
    Esc.Wait;
  Escribiendo:=verdadero;
end;

procedure entry Terminar_Escritura()
begin
  escribiendo:=falso;
  if nlec>0 then
    Lect.Broadcast;
  else
    Esc.Signal;
  end if;
end;

begin
  nlec:=0;
  escribiendo:=falso;
end;

end.

```

---

3 (2 puntos) Responda con brevedad a las siguientes cuestiones.

A. ¿Qué significa que una política de planificación de CPU sea expulsiva?

Nombre

- Que el planificador puede desalojar al proceso que está en CPU. Por ejemplo, para implementar tiempo compartido y tiempo real, es necesaria una planificación expulsiva.

B. ¿Con qué finalidad se emplea la técnica de envejecimiento en la planificación del uso del procesador?

- Para evitar el riesgo de inanición existente cuando se lleva a cabo una planificación por prioridades.

C. Un sistema de memoria virtual ha de escoger a una página víctima. Existen tres páginas candidatas, que sólo se diferencian en el estado del proceso que las posee: hay una página cuyo proceso está en estado de ejecución; otra cuyo proceso está en estado de bloqueo; y el proceso de la tercera página está en estado de preparado. ¿Qué página sería más recomendable escoger como víctima?

- Es conocido que la política óptima consiste en escoger como víctima aquella página que más tiempo tarde en usarse. De las tres candidatas, la página que más probablemente tarde en accederse es la del proceso que está en estado de bloqueo. Las otras dos páginas pertenecen a procesos que muy probablemente hagan accesos a memoria en un futuro inmediato, mientras que el proceso bloqueado no hará ningún acceso en tanto persista su situación de bloqueo. Así que esa es la mejor apuesta.

4 (2 puntos) Considere un sistema de archivos en un disco en el que el tamaño de bloque de sistema de archivos es 512 bytes. Suponga que toda la información de control de cada archivo ya se encuentra en memoria principal. Para las estrategias de asignación de espacio de disco contigua e indexada, especifique los pasos y cálculos a realizar para transformar una dirección de dato de archivo, expresada en forma de desplazamiento desde el comienzo del archivo y longitud del dato, a bloque de sistema de archivos.

Sea:

P la posición del primer byte del dato a acceder del fichero con respecto al origen de éste.

L la longitud del dato

Entonces:

1. Obtenemos el bloque relativo que contiene al primer byte del dato:

$$B0 = \text{Parte entera}(P/512)$$

2. Obtenemos la posición del primer byte del dato en el bloque B0:

$$D0 = \text{Resto}(P/512)$$

3. Obtenemos el bloque relativo que contiene el último byte del dato a acceder.

$$B1 = \text{Parte entera}((P+D-1)/512)$$

4. Obtenemos la posición de este último byte en el bloque B1

$$D1 = \text{Resto}((P+D-1)/512)$$

5. Obtenemos el número de bloque de sistema de archivo de los bloques B0 y B1:

- Para la política contigua  $DB0 = B0$  y  $DB1 = B1$

- Para la política indexada  $DB0 = \text{INDICES}[B0]$  y  $DB1 = \text{INDICES}[B1]$ . Siendo

INDICE el bloque de índices del archivo.

5 (2 puntos) Considere un sistema de memoria segmentada, sin memoria virtual. Especifique de forma algorítmica los pasos que se deben ejecutar en la traducción de una dirección lógica a dirección física. En su especificación deberá contemplar el uso de los distintos recursos físicos y lógicos requeridos.

El formato de la dirección contiene dos campos: segmento (SEG) y desplazamiento en el segmento (DES). Al no disponer de memoria virtual se asume que todo el espacio direccionable está en memoria.

1. Traducción mediante el hardware especial de traducción (que consiste en una serie de registros que representan a un subconjunto de segmentos del espacio direccionable del programa): si el segmento está representado en dicho subconjunto, entonces se obtiene la dirección sumando la dirección base de la zona de memoria que contiene el segmento con el desplazamiento.

2.1. Obtener el comienzo de la tabla de segmentos (registro origen de la tabla de segmentos)

2.2. Verificar que se trata de un segmento válido. Esto se comprueba mediante dos verificaciones. Primero, comparando SEG con el número de entradas de la tabla, si SEG lo supera entonces error por dirección inválida (número de segmento inválido). Segundo que el campo de desplazamiento no supera el tamaño del segmento, si es así entonces error por dirección inválida (desplazamiento en el segmento inválido).

2.3. Acceder a la entrada SEG de la tabla de segmentos y obtener la dirección base del área de memoria que lo contiene. El resultado de la traducción será la suma de esta dirección base con el campo DES.

6 (2 puntos) Suponga un sistema con 5 procesos, P0 a P4, y 3 tipos de recursos A, B, y C. El tipo de recurso A tiene 2 ejemplares, B tiene 10 ejemplares y C tiene 6 ejemplares. En cierto instante, el sistema se encuentra en el siguiente estado:

	Asignados			Solicitudes		
	A	B	C	A	B	C
P0	1	0	0	0	0	0
P1	0	2	0	0	2	2
P2	0	3	3	0	0	0
P3	1	2	1	0	1	0
P4	0	0	2	0	0	2

¿Hay interbloqueo en el sistema? Justifique su respuesta.

El sistema no está en un estado de interbloqueo. Para demostrarlo aplicamos el algoritmo de detección visto en clase y vemos que podemos encontrar una secuencia ordenada de procesos <p0,p2,p4,p1,p3> en la que lo que pide cada proceso puede ser satisfecho por lo disponible más lo asignado a los procesos que están antes que él en la secuencia.