



IIC 2333 — Sistemas Operativos y Redes — 1/2016
Interrogación 2

Lunes 02-Mayo-2016

Duración: 2 horas

1. [12p] Considere un sistema automático para controlar el acceso a un andén del metro. Cada estación tiene un andén con capacidad máxima N , y posee una puerta que solo permite entrar a los trenes cuando hay alguno presente en la vía. Cada pasajero es un *thread* que ejecuta un método `subirTren(Andén a)`. Cada tren es un *thread* que llega a la estación con capacidad C , y ejecuta:

```
1 TREN
2   tren.entrarEstacion(Andén a, int C)
3   tren.esperar()
4   tren.dejarEstacion(Andén a)
```

Implemente los siguientes métodos:

- `pasajero.subirTren(Andén a)`. Permite que un pasajero entre al andén y luego al tren, si hay alguno presente.
- `tren.entrarEstacion(Andén a, int C)`. Permite que un máximo de C pasajeros entren al tren.
- `tren.dejarEstacion(Andén a)`. Detiene el acceso de pasajeros al tren.

Puede utilizar semáforos, locks, variables de condición, y las variables compartidas que estime conveniente. Su solución no debe permitir más de N pasajeros en el andén, y tampoco debe permitir que más de C pasajeros suban al tren. Suponga que ningún pasajero desciende del tren.

2. [6p] En la cocina del DCC hay solamente 10 tazas, y 3 salas donde se efectúan reuniones. Nadie puede entrar a una reunión sin su taza, y una reunión no puede terminar si no se encuentran todos los miembros dentro de ella. La señora Antonieta se encarga de entregar un taza a cada persona que llega a una reunión.

A mediodía hay tres salas ocupadas: Álvaro Campos (AC), Microsoft (MS), y Consejo (C), reservadas para un máximo de 7, 4, y 9 personas respectivamente. Así mismo, en cada sala hay 2, 2, y 3 personas, cada una con su taza.

- Si llegan 3 personas: 1 para la sala C, y 2 para la sala MS, ¿qué debe hacer la señora Antonieta?

3. [12p] Un sistema posee direcciones virtuales de 48-bit, y direcciones físicas de 32-bit.

- 3.1) ¿Cuántas entradas requiere una tabla de páginas de un nivel si las páginas son 16KB?
- 3.2) ¿De qué tamaño es la tabla de páginas?
- 3.3) Describa una solución que permita disminuir la memoria física requerida por la tabla de páginas.
- 3.4) ¿Cuántas entradas necesitaría una tabla de páginas invertida para esta arquitectura?

4. [12p] Considere el siguiente programa en C:

```
1 int big[N];
2 int step = M;
3
4 for(int i=0; i<N; i+=step) {
5     big[i] = big[i] + 1;
6 }
```

- 4.1) Suponga que se ejecuta en una máquina con páginas de 4KB, y un TLB de 64 entradas. ¿Para qué valores de M y N la instrucción de la línea 5 genera un TLB-miss en cada repetición?
- 4.2) Suponga que agregamos esta instrucción en la línea 3: `for (k=0; k<2; k++)`. ¿Cómo cambia su respuesta a la pregunta anterior?
5. [6p] Considere un TLB con 1024 entradas (1K), que pueden ser accedidas en 1 nsec (10^{-9}). Una entrada de la tabla de páginas puede ser accedida en 100 nsecs. El tiempo requerido para copiar una página desde el disco a la memoria es de 6 msec.

Si el 99 % de los acceso a una página es un *TLB Hit*, y un 0,01 % de los accesos genera un *page fault*, ¿cuál es el tiempo promedio de acceso a una dirección de memoria?. Puede dejar el cálculo expresado de manera clara.

6. [12p]

- 6.1) [3p] Para las siguientes afirmaciones, indique (no necesita justificar), cual solución: paginación o segmentación, o ambas, se ajusta mejor:

- Es mejor que la asignación contigua porque no provoca fragmentación externa.
- Permite un único espacio de direcciones virtuales.
- Permite que el tamaño de la memoria virtual sea mayor que la memoria física.

- 6.2) [9p] Considere un programa con segmentos paginados. En este programa hay un segmento 0 (código) de 64 KB, y tiene acceso *read-only*; y un segmento 1 (datos) de 20 KB con acceso *read-write*. Los segmentos están divididos en páginas de 4KB. Las páginas se cargan *on-demand* y el sistema asigna un *frame* en orden secuencial (donde el primero es el 0). Inicialmente solo está cargada la página 0, segmento 0, en el frame 2. Describa una tabla que mezcle páginas y segmentos e incluyendo un bit de valid/invalid, para indicar si la página está cargada en memoria, luego de ejecutar la siguiente secuencia. Si hay instrucciones inválidas, indíquelas y continúe.

- *Read* segmento 1, página 1, offset 3
- *Read* segmento 1, página 3, offset 120
- *Read* segmento 0, página 4, offset 5
- *Write* segmento 1, página 3, offset 5
- *Read* segmento 1, página 5, offset 2047
- *Write* segmento 0, página 7, offset 10
- *Write* segmento 1, página 0, offset 1024
- *Write* segmento 1, página 6, offset 15
- *Read* segmento 0, página 1, offset 2000