



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

Jueves 29-Septiembre-2016

Duración: 2 horas

Una HOJA por pregunta. Para cada pregunta se corregirá **máximo una hoja por pregunta**. (4 hojas en total)

1. [15p] Respecto a los algoritmos de *scheduling*, responda **brevemente**

- 1.1) FCFS, en general, no es un buen algoritmo de *scheduling*. ¿Bajo qué condiciones FCFS puede entregar el mejor tiempo promedio de espera?
- 1.2) ¿Bajo qué condiciones el algoritmo *Round-Robin* puede comportarse peor que FCFS?
- 1.3) Considere un sistema con N procesos que toman igual tiempo L , y se usan *Round-Robin* con *quantum* Q . ¿Cuál es el tiempo promedio de espera, y cómo se afecta al reducir el *quántum*?

2. [15p] Sincronización y *deadlocks*

- 2.1) Soluciones de sincronización como la de Peterson pueden ser extendidas a N procesos, y cumplen con las condiciones de acceso a una sección crítica. ¿Por qué aún así son necesarias instrucciones de *hardware* para implementar primitivas de sincronización?
- 2.2) Considere una instrucción *compare-and-swap* (*cas*) cuya ejecución es atómica y su comportamiento (descrito en C) equivale a:

```
int cas (int *value, int expected, int new_value) {  
    int previous = *value;  
    if (*value == expected)  
        *value = new_value;  
    return previous;  
}
```

Utilice esta instrucción para construir un semáforo, incluyendo sus métodos $P()$ (*wait*) y $V()$ (*signal*)

- 2.3) Considere un sistema con cuatro procesos y dos tipos de recursos. Las siguientes matrices indican la asignación actual (*alloc*) y las solicitudes actuales de cada proceso (*request*).

Alloc	P_1	1	3
	P_2	4	1
	P_3	1	2
	P_4	2	0

,

Request	P_1	1	2
	P_2	4	3
	P_3	1	7
	P_4	5	1

Para los siguientes valores del vector de disponibilidad (*available*), indique si el sistema se encuentra o no en *deadlock*. Si lo está, debe indicar cuáles son los procesos involucrados y la cantidad mínima de recursos que se puede asignar para ya no esté en *deadlock*. Si no lo está, debe indicar por qué.

- a) $Available = \{1, 4\}$
- b) $Available = \{2, 5\}$

3. **[15p]** Un sistema de paginación para una arquitectura Intel x86 utiliza 48 bit para las direcciones virtuales. El esquema utiliza 4 niveles de paginación, donde cada nivel utiliza 9-bit para el número de página. La dirección física utiliza 52 bit. Cada entrada en una tabla de páginas utiliza 8 byte.
- 3.1) Dibuje el esquema de traducción especificando la cantidad de bit usados para cada paso, y el espacio ocupado por cada tabla de páginas. Suponga que el último nivel no utiliza bits adicionales (*dirty*, *valid*, etc...).
- 3.2) Si se incrementa el tamaño de página, es posible reducir el número de niveles. ¿Cuán grande deben ser las páginas para utilizar solamente dos niveles? Ambos niveles deben tener el mismo tamaño. Justifique su cálculo y dibuje el sistema de traducción modificado.
4. **[15p]**
- 4.1) Un acceso normal a memoria toma 100ns. El tiempo de lectura desde el disco toma 10ms. Calcule el tiempo promedio de acceso a memoria (incluyendo el tiempo incurrido por la *page fault*, si las *page faults* ocurren en el 0,1 % de los accesos. Puede suponer que no hay tiempo asociado a un TLB *miss*.
- 4.2) Un acceso normal a memoria toma 100ns. El tiempo de lectura desde el disco toma 10ms. Calcule el máximo valor de la tasa de *page faults* que se puede considerar para que el rendimiento del acceso a memoria no sea más allá de un 10 %. Puede suponer que no hay tiempo asociado a un TLB *miss*.
- 4.3) En un sistema de memoria virtual se duplica el tamaño de página. ¿Implica esto que el *working set* del proceso se incrementa, se reduce, o se mantiene igual? Justifique su respuesta.