

# SISTEMAS OPERATIVOS

## CERTAMEN #2

Wenceslao Palma <wenceslao.palma@pucv.cl>

1. Responda las sgtes preguntas. Justifique cada una de sus respuestas.
  - (a) (5 ptos.) En un sistema que administra su memoria mediante paginación se puede direccionar como máximo 1GB de memoria y el tamaño de página es de 16KB. Determine el formato de la dirección lógica.
  - (b) (10 ptos.) Un computador tiene una memoria caché, una memoria principal y un disco usado para memoria virtual. Si una palabra referenciada está en memoria caché, se necesitan 20ns para acceder a ella. Si está en memoria principal pero no en memoria caché, se necesitan 60ns para cargarla allí, y la referencia comienza de nuevo. Si la palabra no está en memoria principal se necesitan 12ms para cargarla desde disco, seguidos de 60ns para copiarla en memoria caché, y la referencia comienza de nuevo. La tasa de aciertos en la memoria caché es de 0.9 y la tasa de aciertos de la memoria principal es de 0.6. Cuál es el tiempo, en ns, necesario para acceder a una palabra referenciada en este sistema?
  - (c) (5 ptos.) Cuál es el número mínimo de procesos y recursos necesarios para que se produzca deadlock?
2. (20 ptos.) Considere un sistema con memoria virtual donde el conjunto residente es de tamaño 3. Muestre el funcionamiento de los algoritmos de reemplazo de página LRU y CLOCK para la siguiente secuencia de referencias a páginas: 2 3 1 2 4 5 2 3 1 5 6 1. Muestre si es aconsejable aumentar a 4 el tamaño del conjunto residente. Justifique.
3. (20 ptos.) Considere 2 procesos, uno de los cuales ejecuta **escribirA** y el otro **escribirB**. Indique los valores iniciales de los semáforos **sA** y **sB** para que la salida generada sea **BABABABABA**. Fundamente su respuesta.

```
void *escribirA () {
    int i; i
    for (i= 0; i< 5; i++) {
        wait(sA);
        printf ("A");
        sleep(random()%2);
        signal(sB);
    }
}
```

```
void *escribirB () {
    int i;
    for (i= 0;i< 5; i++) {
        wait(sB);
        printf ("B");
        sleep(random()%2);
        signal(sA);
    }
}
```