

Sistemas Operativos

Certamen Recuperativo

Pauta de corrección

Wenceslao Palma <wenceslao.palma@ucv.cl>

1. (20 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?

R.:

(1) (6 ptos.) palabra en caché: $(0.9 \times 20)ns$

(2) (6 ptos.) palabra en memoria pero no en caché: $(0.6 \times 0.1) \times (60 + 20)$

(3) (6 ptos.) palabra ni en memoria ni en caché: $(0.4 \times 0.1) \times (12 \times 10^6 + 60 + 20)$

tiempo = (1)+(2)+(3) (2 ptos.)

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.

R.:

LRU (4 ptos)

2	3	1	2	4	5	2	3	1	5	6	1
2	2	2	2	2	2	2	2	2	5	5	5
	3	3	3	4	4	4	3	3	3	6	6
		1	1	1	5	5	5	1	1	1	1
				F	F		F	F	F	F	

CLOCK (5 ptos)

2	3	1	2	4	5	2	3	1	5	6	1
*2-1	*2-1	*2-1	*2-1	4-1	4-1	*4-1	3-1	3-1	*3-1	6-1	6-1
	3-1	3-1	3-1	*3-0	5-1	5-1	*5-0	1-1	1-1	*1-0	*1-1
		1-1	1-1	1-0	*1-0	2-1	2-0	2-0	5-1	5-0	5-0
				F	F	F	F	F	F	F	

LRU (4 ptos)

2	3	1	2	4	5	2	3	1	5	6	1
2	2	2	2	2	2	2	2	2	2	6	6
	3	3	3	3	5	5	5	5	5	5	5
		1	1	1	1	1	3	3	3	3	3
				4	4	4	4	1	1	1	1
					F		F	F	F	F	

CLOCK (5 ptos)

2	3	1	2	4	5	2	3	1	5	6	1	
*2	-1	*2-1	*2-1	*2-1	*2-1	5-1	5-1	5-1	*5-1	*5-1	6-1	6-1
		3-1	3-1	3-1	3-1	*3-0	2-1	2-1	2-1	2-1	*2-0	*2-0
			1-1	1-1	1-1	1-0	*1-0	3-1	3-1	3-1	3-0	3-0
					4-1	4-0	4-0	*4-0	1-1	1-1	1-0	1-1
						F	F	F	F		F	

Justificación (2 ptos.)

3. (20 pts.) Considere 2 threads, uno de los cuales ejecuta **escribirA** y el otro **escribirB**. Usando semáforos haga que la salida generada sea **BABABABABA**. Indique el valor inicial de los semáforos que utilice.

```
void *escribirA (void *p) { void *escribirB (void *p) {
    int i; i                    int i;
    for (i= 0; i< 5; i++) {    for (i= 0; i< 5; i++) {
        sem_wait(&sA);          sem_wait(&sB);
        printf ("A");          printf ("B");
        fflush(NULL);          fflush(NULL);
        sleep(random()%2);      sleep(random()%2);
        sem_post(&sB);          sem_post(&sA);
    }                            }
    pthread_exit(NULL);        pthread_exit(NULL);
}
```

R.: $sA = 0$ y $sB = 1$. Justificar con ruteo.