

Pregunta 1

I. La siguiente es la implementación concreta de la estrategia del reloj:

```
void pagefault(int page) {
    Process *p= currentProcess;
    int *ptab= p->pageTable;
    if (bitS(ptab[page])) /* residente en disco */
        send(clock, &page);
    else
        segfault(page);
}

void clockStrategy() { /* un proceso daemon */
    Process *p; /* proceso que gatilla el page fault */
    int page= *(int*)receive(&p);
    Iterator *it= processIterator();
    for (;;) {
        Process *q= nextProcess(it);
        int *qtab= q->pageTable;
        for (i=q->firstPage; i<q->lastPage; i++) {
            if (bitV(qtab[i])) {
                if (bitR(qtab[i]))
                    setBitR(&qtab[i], 0); /* referenciada */
                else {
                    /* q: proceso que cede su página */
                    int real_page= realPage(qtab[i]);
                    savePage(q, i); /* retoma otro proceso */
                    setBitV(&qtab[i], 0);
                    setBitS(&qtab[i], 1);
                    int *ptab= p->pageTable;
                    setRealPage(&ptab[page], real_page);
                    setBitV(&ptab[page], 1);
                    loadPage(p, page); /* retoma otro proceso */
                    setBitS(&ptab[page], 0);
                    purgeTlb(); /* invalida la TLB */
                    purgeL1(); /* invalida cache L1 */
                    reply(p);

                    page= *(int*)receive(&p);
                }
            }
        }
        if (!hasNext(it))
            reset(it);
    }
}
```

Suponga que la MMU *no implementa* el bit D (*dirty*). Modifique el código de más arriba de manera que no se grabe una página en disco si esa página ya había sido grabada previamente.

Ayuda: suponga que existe un bit de software W2 que es el que realmente indica si una página se puede escribir o no. Use el bit W para emular el

bit D (*dirty*).

II. Suponga que un proceso recorre 100 veces un arreglo de 4 MB secuencialmente en un procesador Intel x86 (con páginas de 4 KB). El arreglo se encuentra completamente residente en la memoria real. Estime el número de desaciertos en la TLB (translation look-aside buffer).

III. Explique si tiene o no sentido usar la estrategia del working set en un sistema operativo que ofrece solo procesos livianos. ¿Cuál otra estrategia sería más eficaz y por qué?

IV. Considere el siguiente diagrama de lecturas y escrituras (r, w) en memoria de un proceso Unix en un sistema que usa la estrategia del *working set*. Las filas corresponden a las páginas (0, 1, 2, ...) y las columnas a los intervalos de cálculo del working set (A, B, C, ...).

<i>página</i>	6		r	rr	ww	r	w		
5		r w	r		rrr		r	r	
4			r					r	
3		r		rrr		ww	ww		
2		rrr	r	r	rr	wr	ww	r	
1		rr	rr	r	rw				
0			ww	r		r	w		
		A	B	C	D	E	F	G	
									<i>Tiempo</i>

(a) Indique el valor del atributo *Referenced* para todas las páginas al inicio del intervalo E y al final de ese intervalo. (b) Indique el valor del atributo *Dirty* para cada página al inicio del intervalo E y al final de ese intervalo, suponiendo que no hubo ningún *page-fault* en el período A-E. (c) Indique para los períodos C a F qué accesos pueden producir *page-faults*. Utilice coordenadas del estilo (G, 4, *1er. acceso*).

Pregunta 2

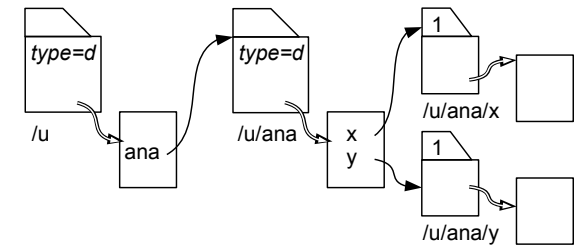
a.- Se requiere implementar un driver para coordinar la cena de los filósofos. El driver controla 5 dispositivos: /dev/f0, ..., /dev/f4 con números *minors* 0, ..., 4. La cena del filósofo i es equivalente a leer o escribir en /dev/fi. Como es

usual, el filósofo i necesita los palitos i e (i+1)%5 para poder cenar y por lo tanto 2 filósofos adyacentes en la mesa no puede cenar simultáneamente.

Programa las funciones correspondientes a *open* y *release* del driver requerido. Ud. debe usar los semáforos del núcleo de Linux y los filósofos no deben sufrir hambruna. Declare las variables globales que necesite e indique cómo se deben inicializar. Recuerde que Ud. puede obtener el número *minor* con *iminor(filp->f_inode)*. Puede ignorar las señales.

b.- Se tiene un archivo de 98 KB en una partición Unix con bloques de 4 KB. Haga un diagrama mostrando inodo, bloques de datos y de indirección. Conteste además: ¿Cuanto espacio en disco se ocupa realmente para almacenar este archivo?

c.- La figura muestra varios archivos y directorios de la partición /u en un sistema Unix:



La usuaria *ana* ejecuta los siguientes comandos:

```
% cd /u/ana
% cp x z
% mkdir bin
% ln y bin/w
% ln -s bin binarios
```

Rehaga la figura de acuerdo a los cambios realizados. No olvide indicar el contador de links para el caso de archivos normales.