CC4302 Sistemas Operativos – Control 3 – Semestre Primavera 2014 – Prof.: Luis Mateu

Pregunta 1

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

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

void clockStrategy() { /* un proceso daemon */
  Process *p;
  int page= *(int*)receive(&p);
  Iterator *it= processIterator();
  for (;;) {
    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);
        else {

          int *ptab= p->pageTable;
          savePage(q, i); /* retoma otro proceso */
          setBitV(&qtab[i], 0);
          setBitS(&qtab[i], 1);
          setRealPage(&ptab[page],
                      realPage(qtab[i]));
          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);
  }
}
```

Parte I.- Suponga que la MMU implementa el bit D (*dirty*). Modifique la función *clockStrategy* para que haga un uso eficiente del bit D.

Parte II.- La implementación de más arriba no funciona cuando 2 procesos comparten su área de código. (a) Dé un ejemplo de ejecución incorrecta. (b) ¿Qué cambios haría para evitar que las páginas compartidas se fuesen a disco?

Parte III.- ¿En qué tipo de sistema el *shell* de comandos es más eficiente? ¿En un sistema que usa segmentación o en uno basado en paginamiento? Explique por qué.

Parte 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, …).

página							
6		r	r r	w w	r	w	
5	r w	r		r r r		r	r
4		r					r
3	r		r r r		w w	ww	
2	r r r	r	r	r r	w r	w w	r
1	r r	r r	r	r w			
0		w w	r		r	w	
	A	B	C	D	E	F	G

Tiempo

(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

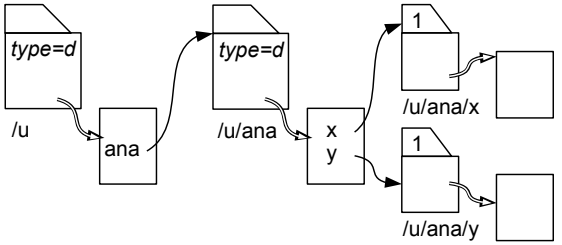
Parte a.- Suponga que en un sistema existen solo 2 procesos en ejecución. Ambos leen secuencialmente archivos de gran tamaño en la misma partición con bloques de 1 KB. El primero de los archivos se encuentra localizado en las pistas externas del disco, mientras que el segundo en las pistas internas. Ambos procesos leen los archivos en trozos de 1 KB usando *read*. Estime la velocidad de lectura de estos archivos considerando los siguientes tipos de almacenamiento secundario:

- Un disco moderno con tiempo de acceso y velocidad de transferencia razonables, el núcleo no implementa *read-ahead*.
- El mismo disco moderno, el núcleo sí implementa *read-ahead* de 100 bloques.
- Un SSD (*solid state drive*), el núcleo no implementa *read-ahead*.

Parte b.- Suponga que en el sistema de archivos de Linux se usó *memcopy* en vez de *copy_from_user* para implementar la operación de escritura (*write*). Explique (i) cómo aprovecharía esto para leer datos del núcleo de Linux sin ser el administrador (*root*), y (ii) si es posible usar esta brecha para alterar el código del núcleo.

Parte c.- Se tiene un archivo que no requiere bloques de indirección doble en una partición Unix con bloques de 2 KB. Se agrega un byte a este archivo y se crea el bloque de indirección doble. Haga un diagrama mostrando inodo, bloques de datos y de indirección. ¿De qué tamaño es el archivo?

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



La usuaria *ana* ejecuta los siguientes comandos:

```
% mkdir /u/ana/tmp
% cd /u/ana
% ln x tmp/z
% cp y tmp/w
% cd tmp
% ln -s ../x v
```

Rehaga la figura de acuerdo a los cambios realizados.