

CC4302 Sistemas Operativos

Control 3 – Semestre Primavera 2012

Prof.: Luis Mateu

Pregunta 1

(a) Explique para qué optimización se usa el bit *Dirty*. Reescriba el pseudo-código de la estrategia del reloj de modo que implemente la misma optimización, pero en una arquitectura que no implementa el bit *Dirty*. Considere el siguiente pseudo-código de la estrategia del reloj sin ninguna optimización:

```

pageFault(pag_en_disco) {
    while (bitR(cursor()))==1) {
        bitR(cursor())= 0;
        avanzarCursor();
    }
    grabarPagina(cursor());
    bitV(cursor())= 0;
    leerPagina(pag_en_disco);
    bitV(pag_en_disco)= 1;
    avanzarCursor();
}
    
```

Hint: Ud. puede emular el bit Dirty a partir del bit W.

(b) Suponga que el espacio de direcciones de un proceso ocupa el rango [4KB,12KB[para el código, [12KB, 24KB[para los datos y [60KB, 64KB[para la pila. Este proceso invoca **fork**, y luego el hijo invoca **sbrk** pidiendo 6KB adicionales. Haga un diagrama con una posible asignación final de las páginas en una máquina que dispone de 16 páginas reales. Luego confecciones las tablas de páginas del padre y el hijo indicando los campos página real, atributo V y atributo W.

(c) Suponga que un proceso recorre secuencialmente un arreglo de 400 MB 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*).

(d) Considere un núcleo de Unix que usa la estrategia del *working set* y un proceso cuyo espacio de direcciones virtuales posee solo 6 páginas. Inicialmente todas las páginas están en el *working set*. Haga una tabla de accesos a memoria que considere 4 intervalos de cálculo del *working set*, de tal forma que sea posible que se produzcan 3 page-faults, en intervalos distintos,

en páginas distintas y no más de 3 page-faults.

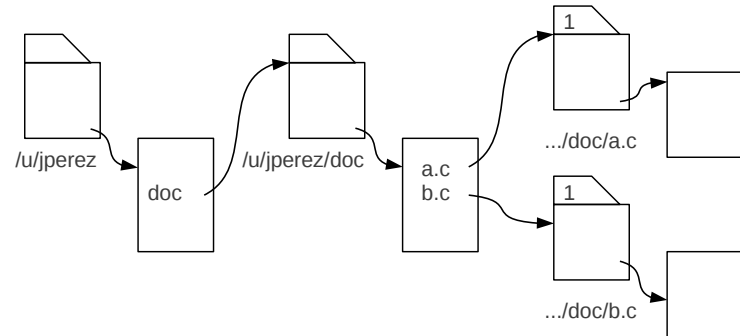
Pregunta 2

i.- Suponga que un usuario se encuentra editando un documento confidencial en la misma máquina que usa Ud. La implementación de **write** en el núcleo del sistema operativo no valida que el *buffer* recibido como argumento apunte a una dirección aceptable. Discuta cómo haría un programa en C que permitiese revelar el documento confidencial. ¿Se podría usar esta brecha en la seguridad para tomar el control de la máquina? Explique.

ii.- 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?

iii.- Un sistema Unix tiene un disco dedicado únicamente a paginamiento. El fabricante del disco anuncia que es capaz de leer/escribir a 50 MB/seg. Explique por qué el administrador del sistema mide solamente 400 KB/seg.

iv.- La figura muestra el sistema de archivos montado en la partición /u.



El usuario *jperez* ejecuta los siguientes comandos:

```

% cd doc
% ln a.c ../x.c
% rm a.c
% ln -s /home/jperez/doc/b.c ../y.c
% rm b.c
    
```

Rehaga la figura de acuerdo a los cambios realizados y complete la figura con las entradas “.” y “..” para el directorio /u/jperez/doc.