

CC41B: Sistemas Operativos
Examen–Semestre Primavera’96
Prof.: Luis Mateu
Prof. Aux.: Daniel Verdugo

Pregunta 1

1. Se tiene un archivo Unix en una partición que usa bloques de 1024 bytes. Haga el diagrama con la descomposición en inodo, bloques de indirección y bloques de datos del archivo, suponiendo que su largo es:

$(12+256+256*256)*1024+1$ bytes

2. Programe un procedimiento que calcule el número de bloques de 1024 bytes que se requieren para un archivo de tamaño n bytes. En sus cálculos debe incluir los bloques de indirección.

Pregunta 2

El siguiente procedimiento ordena un arreglo de n enteros usando *bubble sort*.

```
void bubblesort(int a[], int n)
{
    int i,j;
    for (i=n; i>1; i--)
        for (j=1; j<i; j++)
            if (a[j-1]>a[j]) swapint(&a[j-1], &a[j]);
}
```

Este procedimiento se ejecuta con un arreglo de 1 MByte (256 K palabras enteras) en un computador que posee memoria real como para almacenar sólo la mitad.

1. ¿Podría correr este procedimiento si el computador posee una arquitectura segmentada? Explique. (1 punto)
2. El procedimiento se ejecuta en una arquitectura que usa páginas de 4 KBytes (o 1024 enteros) con un sistema operativo que implementa la estrategia de reemplazo LRU (*least recently used*).

Haga una estimación del número de *page-faults* que se producirá debido a páginas de datos. Para hacer este cálculo estime para cada iteración de i el número de *page-faults* (es decir con $i = n, i = n - 1, \dots, i = n/2 + 1, i = n/2, i = n/2 - 1, \dots, i = 0$).

Además haga una estimación del número de *page-faults* que se producirá debido a páginas de código y de pila.

Concluya sobre la utilidad de un sistema de memoria virtual en este caso.
(5 puntos)

Pregunta 3

Se tiene un computador de 128 KBytes de memoria real que implementa páginas de 8 Kbytes. La tabla de páginas posee 128 filas con lo que se pueden tener procesos de hasta 1 MByte. 40 KBytes de la memoria real se emplean para el núcleo de Unix. Este S.O. tiene habilitado un disco para hacer paginamiento y además implementa la técnica *copy-on-write*.

El computador está ejecutando 3 procesos que poseen los siguientes tamaños para cada una de las áreas de memoria.

Nombre del comando	Tamaño del código	Tamaño de los datos	Tamaño de la pila
ls	14 KB	23 KB	2 KB
csch	12 KB	14 KB	5 KB
csch	12 KB	14 KB	5 KB

Ambos procesos `csch` corresponden al padre e hijo de un `fork` que se acaba de hacer. El sistema no ha ejecutado ninguna instrucción de estos dos procesos después del `fork`.

1. Haga un diagrama que muestre una posible asignación de las páginas de memoria real a estos 3 procesos (no olvide que el núcleo ocupa parte de la memoria real).
2. Considerando la asignación de páginas que Ud. hizo en la parte anterior, indique claramente el contenido de la tabla de páginas cuando se ejecuta el proceso `ls` (indicando atributos y traducción).
3. La siguiente es una traza de los números de página accedidos por el proceso `ls`:
0, 0, 1, 1023, 0, 4, 4, 1, 0, 3, 2, 1023, 3, 0, 2, 1023, 1, 2, 2, 3, 1023, 1, 2, 4.
Suponiendo que el sistema operativo calcula el `working-set` cada 8 accesos a memoria, especifique los `working-set` después de cada cálculo.