

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

4ta práctica (tipo a)
(Segundo semestre de 2018)

Horario 0781: prof. V. Khlebnikov

Horario 0782: prof. F. Solari A.

Duración: 1 h. 50 min.

Nota: No se puede usar ningún material de consulta.

La presentación, la ortografía y la gramática influirán en la calificación.

La práctica debe ser desarrollada en el cuadernillo usando lapicero.

Lo escrito con lápiz NO será evaluado.

Puntaje total: 20 puntos

Pregunta 1 (4 puntos – 20 min.) En la práctica pasada hemos usado los programas `maxsizes.c` y `maxsizes_2.c`. Modificaremos el último programa declarando ahora un arreglo de enteros:

`$ cat -n maxsizes_3.c | expand`

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4
5  #define K 1024
6  #define M K*K
7  #define G M*K
8  #define S G
9
10 int ar_bss[K][K];
11
12 int main(void) {
13     int i,j;
14     char path_buf[0x100] = {};
15
16     printf("int size = %ld\n",sizeof(int));
17     printf("ar_bss[0][0] adr = %p\n",&ar_bss[0][0]);
18     printf("ar_bss[0][1] adr = %p\n",&ar_bss[0][1]);
19     printf("ar_bss[1][0] adr = %p\n",&ar_bss[1][0]);
20     printf("path_buf adr = %p\n",path_buf);
21
22     printf("Page size: %d\n\n",getpagesize());
23     sprintf(path_buf,"cat /proc/%u/maps",getpid());
24     system(path_buf); printf("\n");
25     return 0;
26 }
```

`$ gcc maxsizes_3.c -o maxsizes_3`

`$./maxsizes_3`

```

int size = 4
ar_bss[0][0] adr = 0x559fee60c040
ar_bss[0][1] adr = 0x559fee60c044
ar_bss[1][0] adr = 0x559fee60d040
path_buf adr = 0x7ffd6b12fc10
Page size: 4096
```

```

559fee40b000-559fee40c000 r-xp 00000000 08:08 13505149 /home/vk/clases/so/progs/maxsizes_3
559fee60b000-559fee60c000 r--p 00000000 08:08 13505149 /home/vk/clases/so/progs/maxsizes_3
559fee60c000-559fee60d000 rw-p 00001000 08:08 13505149 /home/vk/clases/so/progs/maxsizes_3
559fee60d000-559fee60e000 rw-p 00000000 00:00 0
559ff048a000-559ff04ab000 rw-p 00000000 00:00 0 [heap]
...
```

Indique la longitud y los valores (use solamente 3 últimos dígitos hexadecimales del número de la página) de las cadenas de referencia (*reference strings*) que se generarían durante la ejecución de cada uno de los siguientes códigos (considerando que las variables `i` y `j` se mantienen en los registros):

```

for (i=0; i<K; i++)
    for (j=0; j<K; j++)
        ar_bss[i][j]= 0;
```

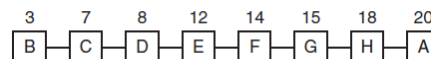
y

```

for (j=0; j<K; j++)
    for (i=0; i<K; i++)
        ar_bss[i][j]= 0;
```

Pregunta 2 (2 puntos – 10 min.) If FIFO page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string 1207341757 if the four frames are initially empty? Now repeat this problem for LRU.

Pregunta 3 (1 punto – 5 min.) Consider the following page sequence. Suppose that the R bits for the pages B through A are 11101011, respectively. Which page will second chance remove?



Pregunta 4 (2 puntos – 10 min.) A small computer on a smart card has four page frames. At the first clock tick, the R bits are 1010 (pages 0 and 2 are 1, the rest are 0). At subsequent clock ticks, the values are 0001, 1011, 0010, 1100, 0101, 0011, and 1101. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last tick.

Pregunta 5 (1 punto – 5 min.) A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	405	620	1	0
1	512	542	0	0
2	431	560	0	0
3	387	614	0	1

Which page will NRU replace?

Pregunta 6 (5 puntos – 25 min.) FAT32 es un sistema de archivos que toma su nombre del método de asignación *File Allocation Table* y el tamaño de cada entrada de la tabla: 32 bits. El siguiente *dump*, o volcado, de la parte inicial de una de estas tablas de un *filesystem* dado nos permite interpretar el contenido, siguiendo las cadenas de bloques, en este caso de tamaño 4 kibi-Bytes. Para numerar los bloques, que en FAT32 se denominan *clusters* o grupos de sectores contiguos, sólo se utilizan 28 bits, siendo entonces 0x0ffffff el valor “-1”, o *last cluster*, o *cluster* final del archivo.

```

0x00  f8 ff ff 0f ff ff ff 0f 1a 00 00 00 04 00 00 00
0x10  07 00 00 00 06 00 00 00 ff ff ff 0f 0d 00 00 00
0x20  ff ff ff 0f 05 00 00 00 00 00 00 00 09 00 00 00
0x30  18 00 00 00 00 00 00 00 0c 00 00 00 14 00 00 00
0x40  17 00 00 00 0d 00 00 00 0f 00 00 00 0e 00 00 00
0x50  19 00 00 00 11 00 00 00 12 00 00 00 0e 00 00 00
0x60  ff ff ff 0f ff ff ff 0f ff ff ff 0f 00 00 00
0x70  13 00 00 00 0b 00 00 00 08 00 00 00 15 00 00 00
0x80  00 00 00 00 00 00 00 00 00 00 00 00 .. .. ..

```

a) (1 punto – 5 min) Arme la tabla FAT a partir del *dump*.

b) (3 puntos – 15 min) La entrada 0 y 1 no se utilizan para los archivos. Establezca los archivos existentes, los bloques físicos y lógicos que les corresponden, dándoles nombres como A, B, C,

c) (1 punto – 5 min) Puede decirse que este *filesystem* tiene un “problema” en su contenido. ¿Cuál es este problema? Indique qué efecto podría tener el eliminar uno de los archivos que tiene este problema.

Pregunta 7 (5 puntos – 25 min.) Un sistema de archivos tipo Unix, que utilizan índices, o nodos- i , o también llamados *i-nodes*, tiene una estructura de *i-node* de 256 bytes, de los cuales se utilizan 40 bytes para índices directos de 32 bits, 4 bytes para el índice simplemente indirecto y 4 bytes para el índice doble indirecto, reservando algunos bytes para lo que podría usarse como índice triple indirecto.

a) (3 puntos – 15 min) Si cada bloque es de 1 kibi-Byte, muestre cómo se asignarían bloques a un archivo de 8 kibi-bytes, a otro de 10 kibi-bytes (1 punto), y a uno de poco más de 10 kibi-bytes y menos de 11 kibi-bytes (2 puntos). Considere bloques libres a partir del bloque # 8192d.

b) (1 punto – 5 min) Calcule, ¿a partir de qué tamaño se requiere usar el índice doble indirecto?

c) (1 punto – 5 min) ¿Cuál sería el tamaño máximo de un archivo con esta estructura de nodo índice?



La práctica ha sido preparada por FS (6,7) y VK (1-5)
en Linux Mint 19 Tara con LibreOffice Writer.

Profesores del curso: (0781) V. Khlebnikov
(0782) F. Solari A.

Pando, 16 de noviembre de 2018