

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

4ta práctica (tipo a)
(Segundo semestre de 2020)

Horario 0781: prof. V. Khlebnikov

Horario 0782: prof. F. Solari A.

Duración: 1 h. 50 min.

Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

Pregunta 1 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 1** de la **Práctica 4** en PAIDEIA **antes de las 11:40**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_41.txt`. Por ejemplo, `20171903_41.txt`.

Below is an execution trace of a program fragment for a computer with **256-byte pages**.

The program is located at address **0x05fc**, and its stack pointer at **0x2200** (the stack grows toward 0 before every push operation).

a) (4 puntos – 20 min.) Give the page reference string generated by this program. Each instruction occupies 4 bytes (1 word) including immediate (instruction inside) constants. Both instruction and data references count in the reference string.

Load word at **0x1a00** into register **0**

Push register **0** onto the stack

Call a procedure at **0x1600**, stacking the return address

Make base pointer equal to the stack pointer minus **0x10**

Compare the actual parameter (pointed by base pointer) to the immediate constant **4**

Jump if equal to **0x1620**; (*consider it true*)

Make base pointer equal to stack pointer minus **0x08**

Save **0x0001** as return value (pointed by base pointer)

Return from procedure

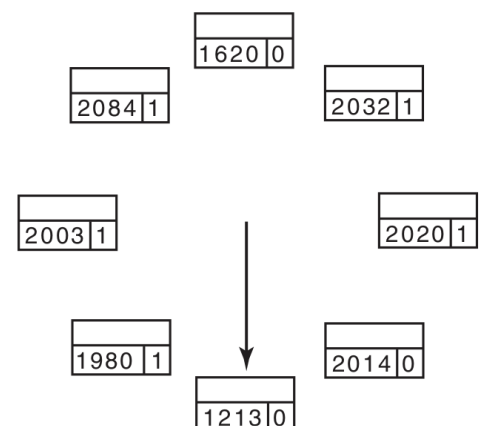
b) (2 puntos – 5 min.) Si ninguna página de este programa está cargada en la memoria, ¿cuántos fallos de página sucederían para 3 marcos con el algoritmo óptimo? ¿Y con FIFO?

Pregunta 2 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 2** de la **Práctica 4** en PAIDEIA **antes de las 12:20**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_42.txt`. Por ejemplo, `20171903_42.txt`.

2204 Current virtual time

a) (2 puntos) En la figura se presenta el algoritmo WSClock. Si sucede un fallo de página en la situación presentada, ¿será eliminada la página indicada? ¿O no será eliminada? ¿O es posible que será eliminada? Explique con detalles.



b) (4 puntos) Supongamos que el algoritmo de reemplazo de páginas WSClock usa τ de dos clics y el estado del sistema es el siguiente, donde los tres bits de banderas *V*, *R* y *M* corresponden a *Valid*, *Referenced* y *Modified*, respectivamente:

| Page | Time stamp | V | R | M |
|------|------------|---|---|---|
| 0 | 14 | 0 | 0 | 0 |
| 1 | 19 | 1 | 1 | 1 |
| 2 | 16 | 1 | 0 | 1 |
| 3 | 19 | 1 | 1 | 0 |
| 4 | 17 | 1 | 0 | 0 |

(1) (1 punto) Si una interrupción del reloj sucede en el tiempo 20, indique si habrán o no algunas modificaciones en la tabla?

(2) (3 puntos) Si, en vez de una interrupción del reloj, en el tiempo 20 sucede un fallo de página como consecuencia de una solicitud de lectura en la página 0, explique las modificaciones que sucederán en la tabla en todos los casos posibles.

Pregunta 3 (8 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 3** de la **Práctica 4** en PAIDEIA **antes de las 13:00**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_43.txt`. Por ejemplo, `20171903_43.txt`.

a) (4 puntos) El siguiente *dump*, o volcado, de la parte inicial de la tabla FAT de un *filesystem* *FAT32*, nos permite interpretar el contenido, siguiendo las cadenas de bloques o *clusters*, 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 `0xffffffff` 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 ff ff ff 0f 0c 00 00 00 14 00 00 00
0x40 17 00 00 00 0d 00 00 00 0f 00 00 00 ff ff ff 0f
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 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 .. .. ..
```

(i) (1 punto) Construya la tabla *FAT32* a partir del *dump*.

(ii) (2 puntos) Determine que archivos se tienen, nombrándolos como A, B, C,... de acuerdo a la cadena de *clusters*.

(iii) (1 punto) Puede decirse que este *filesystem* tiene un problema con sus archivos. ¿Cuál es ese problema?

b) (4 puntos) Un marco de fotos digital (*Digital Photo Frame*) viene con una tarjeta SD de 32 GB (*32GB SD card*). La tarjeta tiene el formato de *FAT32* con el *cluster* de 32KB ($0x8000 = 2^{15}$, que es 1 con 15 ceros seguidos tal como $1000000 = 10^6$ es 1 con 6 ceros seguidos). Se presenta el *dump* del directorio raíz:

```
$ sudo hd -s 0x800000 /dev/sdf1 | less
```

```
00800000 31 20 20 20 20 20 20 20 4a 50 47 20 10 81 d2 85 |1      JPG ....|
00800010 8d 4f 48 51 00 00 1b 60 f3 4e 03 00 32 3f 04 00 |.OHQ...`.N..?..|
00800020 32 20 20 20 20 20 20 20 4a 50 47 20 10 89 d2 85 |2      JPG ....|
00800030 8d 4f 48 51 00 00 1d 60 f3 4e 0c 00 b9 0b 04 00 |.OHQ...`.N.....|
00800040 33 20 20 20 20 20 20 20 4a 50 47 20 10 8e d2 85 |3      JPG ....|
00800050 8d 4f 48 51 00 00 22 60 f3 4e 15 00 99 23 03 00 |.OHQ..."`.N...#..|
00800060 34 20 20 20 20 20 20 20 4a 50 47 20 10 91 d2 85 |4      JPG ....|
00800070 8d 4f 48 51 00 00 24 60 f3 4e 1c 00 4a 6b 04 00 |.OHQ...$`.N...Jk..|
00800080 35 20 20 20 20 20 20 20 4a 50 47 20 10 97 d2 85 |5      JPG ....|
00800090 8d 4f 48 51 00 00 78 5d f2 4a 25 00 f2 eb 12 00 |.OHQ...].J%.....|
008000a0 36 20 20 20 20 20 20 20 4a 50 47 20 10 a0 d2 85 |6      JPG ....|
008000b0 8d 4f 48 51 00 00 0c 8e dd 4c 4b 00 1b 2d 02 00 |.OHQ.....LK....|
008000c0 37 20 20 20 20 20 20 20 4a 50 47 20 10 a4 d2 85 |7      JPG ....|
008000d0 8d 4f 48 51 00 00 03 65 8a 4c 50 00 f0 f1 01 00 |.OHQ...e.LP.....|
008000e0 38 20 20 20 20 20 20 20 4a 50 47 20 10 a5 d2 85 |8      JPG ....|
008000f0 8d 4f 48 51 00 00 92 65 8a 4c 54 00 fe 72 09 00 |.OHQ...e.LT...r..|
00800100 31 30 38 30 50 20 20 20 4d 50 34 20 18 ab d2 85 |1080P  MP4 ....|
00800110 8d 4f 48 51 00 00 17 96 b2 4e 67 00 46 81 0f 06 |.OHQ....Ng.F...|
00800120 43 41 52 20 20 20 20 20 4d 50 34 20 18 74 d5 85 |CAR     MP4 .t..|
00800130 8d 4f 48 51 00 00 8c 6b c9 4c 87 0c 5d c1 c6 00 |.OHQ...k.L...]|...|
00800140 e5 66 00 54 00 65 00 73 00 74 00 0f 00 49 2e 00 |.f.T.e.s.t...I..|
00800150 74 00 6d 00 70 00 00 00 ff ff 00 00 ff ff ff ff |t.m.p.....|
```

```

00800160 e5 52 00 65 00 61 00 64 00 79 00 0f 00 49 42 00 |.R.e.a.d.y...IB.|
00800170 6f 00 6f 00 73 00 74 00 50 00 00 00 65 00 72 00 |o.o.s.t.P...e.r.|
00800180 e5 45 41 44 59 42 7e 31 54 4d 50 20 00 95 d9 85 |.EADYB~1TMP ....|
00800190 8d 4f 8d 4f 00 00 db 85 8d 4f 15 0e 00 00 80 00 |.O.O.....O.....|
008001a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
```

(1) Indique el tamaño **en clusters** y el número (en hex) del primer *cluster* de cada uno de los primeros 4 archivos. Use la división entera de los números hexadecimales entre 0x8000. Por ejemplo, 0x12ebf2 ÷ 0x8000 es quitar 15 bits menos significativos de 0x12ebf2 donde cada dígito hexadecimal es de 4 bits. Quitando 12 bits nos queda 0x12e y quitando 3 bits más nos queda 0001 0010 ~~1110~~ = 0010 0101 = 0x25 = 2*16 + 5 = 37 *clusters*. Más 1 *cluster* (que será el último) para el resto de la división que es diferente de 0. En total, serán 38 *clusters*.

(2) En el directorio raíz hay un archivo eliminado. ¿Cuál fue su nombre corto y largo? ¿Cuál fue el número (en hex) de su primer *cluster* y cuántos *clusters* (en hex, binario y decimal) fueron ocupados por él?



La práctica ha sido preparada por FS (1,3a) y VK (2,3b)
con LibreOffice Writer en Linux Mint 20 “Ulyana”

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

Lima, 11 de diciembre de 2020