

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

Examen 2

(Segundo semestre de 2020)

Horario 0781: prof. V. Khlebnikov

Horario 0782: prof. F. Solari A.

Duración: 3 horas

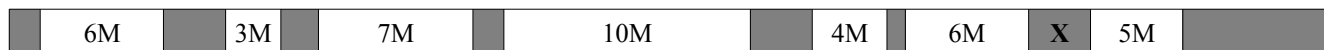
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

Pregunta 1 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **INF239-E2-Buzón 1** del **Examen 2** en PAIDEIA **antes de las 08:45**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_21.txt`. Por ejemplo, `20202912_21.txt`.

a) (3 puntos) Este diagrama muestra un ejemplo de configuración de memoria bajo el particionamiento dinámico, después de realización de unas operaciones de asignación (según uno de los 4 algoritmos: *first-fit*, *next-fit*, *best-fit*, *worst-fit*) y de intercambio (*swapping-out*). Las direcciones van de izquierda a derecha; las áreas grises indican los bloques ocupados por los procesos; las áreas blancas indican los bloques de memoria libre. El último proceso colocado en la memoria es de 4MB y está marcado con un X. Después de su colocación solamente un proceso fue intercambiado (*swapped-out*).



I) (0,5 puntos) ¿Cuál fue el tamaño máximo del proceso intercambiado (*swapped-out*)?

II) (0,5 puntos) ¿Cuál fue el tamaño del bloque libre justo antes de ser particionado por el proceso X? ¿Y qué algoritmo (de los 4 mencionados anteriormente) fue usado para la asignación?

III) (2 puntos) Una nueva solicitud de asignación de 5.1MB debe satisfacerse enseguida. Indique los intervalos de memoria en los cuales será creada una partición para el nuevo proceso usando todos los 4 algoritmos de asignación mencionados. Para cada algoritmo presente la lista de los tamaños de bloques libres ordenados por la dirección.

b) (2 puntos) In this problem you are to compare the storage needed to keep track of free memory using a bitmap versus using a linked list. The 2 GB memory is allocated in units of n bytes. For the linked list, assume that memory consists of an alternating sequence of segments and holes, each 4 MB. Also assume that each node in the linked list needs a 64-bit memory address, a 32-bit length, and a 32-bit next-node field. How many bytes of storage is required for each method? Which one is better?

Pregunta 2 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **INF239-E2-Buzón 2** del **Examen 2** en PAIDEIA **antes de las 09:30**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_22.txt`. Por ejemplo, `20202912_22.txt`.

Considere el siguiente programa en C, para Linux x86_64 o amd64:

```
$ cat ex2_p2.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```

int g_var_noi;
int g_var_i=321;
char *g_var_str="variable global string";
char *g_var_str_ptr;

int f_a(int a) {
    return (f_b(a)-a);
}

int f_b(int b) {
    return (-b);
}

int main(int argc, char *argv[],char *envp[]) {
    int i,j=123;
    char *cad;

    cad=(char *)malloc(1024);
    g_var_str_ptr=cad;
    strcpy(g_var_str_ptr,getenv("SHELL"));
    printf("pid %d g_var_noi \t dirección %p \t contenido %d\n",getpid(),&g_var_noi,g_var_noi);
    printf("pid %d g_var_i \t dirección %p \t contenido %d\n",getpid(),&g_var_i,g_var_i);
    printf("pid %d g_var_str \t dirección %p \t contenido %p \t cadena %s\n",\
        getpid(),&g_var_str,g_var_str,g_var_str);
    printf("pid %d g_var_str_ptr \t dirección %p \t contenido %p \t cadena %s\n",\
        getpid(),&g_var_str_ptr,g_var_str_ptr,g_var_str_ptr);
    printf("pid %d &main \t dirección %p\n",getpid(),main);
    printf("pid %d &f_a \t dirección %p\n",getpid(),f_a);
    printf("pid %d &f_b \t dirección %p\n",getpid(),f_b);
    printf("pid %d &malloc \t dirección %p\n",getpid(),malloc);
    printf("pid %d &printf \t dirección %p\n",getpid(),printf);
    printf("pid %d argc \t dirección %p \t contenido %d\n",getpid(),&argc,argc);
    printf("pid %d argv \t dirección %p \t contenido %p\tcadena %s\n",getpid(),&argv,argv,argv[0]);
    printf("pid %d envp \t dirección %p \t contenido %p\tcadena %s\n",getpid(),&envp,envp,envp[0]);
    printf("pid %d envp[9] \t dirección %p \t contenido %p\tcadena %s\n",getpid(),&envp[9],envp[9],envp[9]);
    printf("pid %d i \t dirección %p \t contenido %d\n",getpid(),&i,i);
    printf("pid %d j \t dirección %p \t contenido %d\n",getpid(),&j,j);
    printf("pid %d cad \t dirección %p \t contenido %p \t cadena %s\n",getpid(),&cad,cad,cad);
    printf("pid %d PATH\t dirección %p \t contenido %s\n",getpid(),getenv("PATH"),getenv("PATH"));
    printf("pid %d SHELL\t dirección %p \t contenido %s\n",getpid(),getenv("SHELL"),getenv("SHELL"));

    getchar();
    printf("-----\n");
    return(0);
}

```

La compilación del programa:

```
$ gcc -static -o ex2_p2 ex2_p2.c
```

La ejecución del programa:

```

$ ./ex2_p2
pid 11182 g_var_noi      dirección 0x6c4a98      contenido 0
pid 11182 g_var_i       dirección 0x6c1070      contenido 321
pid 11182 g_var_str     dirección 0x6c1078      contenido 0x493fa8      cadena variable global string
pid 11182 g_var_str_ptr dirección 0x6c4aa0      contenido 0x2213860     cadena /bin/bash
pid 11182 &main         dirección 0x40108b
pid 11182 &f_a          dirección 0x40105e
pid 11182 &f_b          dirección 0x40107d
pid 11182 &malloc       dirección 0x414e80
pid 11182 &printf       dirección 0x4081b0
pid 11182 argc          dirección 0x7fffd1e9ddac contenido 1
pid 11182 argv          dirección 0x7fffd1e9dda0 contenido 0x7fffd1e9dec8  cadena ./ex2_p2
pid 11182 envp          dirección 0x7fffd1e9dd98 contenido 0x7fffd1e9ded8  cadena LC_PAPER=es_PE.UTF-8
pid 11182 envp[9]       dirección 0x7fffd1e9df20 contenido 0x7fffd1e9f4a2  cadena SHELL=/bin/bash
pid 11182 i             dirección 0x7fffd1e9ddb0 contenido -773202232
pid 11182 j             dirección 0x7fffd1e9ddb4 contenido 123
pid 11182 cad           dirección 0x7fffd1e9ddb8 contenido 0x2213860      cadena /bin/bash
pid 11182 PATH          dirección 0x7fffd1e9fc05 contenido
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
pid 11182 SHELL         dirección 0x7fffd1e9f4a8 contenido /bin/bash

```

Revisando el mapa de memoria del proceso, en /proc:

```

$ cat /proc/11182/maps
00400000-004c0000 r-xp 00000000 08:06 1685242 /home/felipe/ex2/ex2_p2
006c0000-006c3000 rw-p 000c0000 08:06 1685242 /home/felipe/ex2/ex2_p2

```

```

006c3000-006c6000 rw-p 00000000 00:00 0
02212000-02235000 rw-p 00000000 00:00 0
7f95ab63d000-7f95ab63f000 rw-p 00000000 00:00 0
7fffd1e7f000-7fffd1ea0000 rw-p 00000000 00:00 0
7fffd1f6f000-7fffd1f72000 r--p 00000000 00:00 0
7fffd1f72000-7fffd1f74000 r-xp 00000000 00:00 0
fffffffff600000-fffffffff601000 r-xp 00000000 00:00 0

```

[heap]
[stack]
[vvar]
[vdso]
[vsyscall]

En el mapa de memoria, mostrado como tabla, la primera columna indica la dirección de inicio y fin (excluyente), luego los bits de permisos (rwxp), el *offset* en el *file*, el dispositivo *major:minor* del *file*, el i-nodo, y el *filename* del *file*.

a) (2 puntos) Ubique todos los elementos como variables y funciones, en el mapa de memoria, indicando además, a qué espacio del proceso corresponde (código, datos inicializados, no inicializados, *heap*, *stack*) sin considerar [vsyscall]. En su respuesta, puede copiar el mapa obtenido y (quitando lo no relevante) colocar lo solicitado apropiadamente.

```

00400000-004c0000
006c0000-006c3000
006c3000-006c6000
02212000-02235000
7f95ab63d000-7f95ab63f000
7fffd1e7f000-7fffd1ea0000
7fffd1f6f000-7fffd1f72000
7fffd1f72000-7fffd1f74000

```

b) (2 puntos) Si el tamaño de página es de 4 KiBytes, ¿cuántas páginas tiene cada una de las partes descritas, y cuántas páginas tiene el programa en total en memoria? Justifique sus cálculos.

c) (1 punto) Si, simplificando un poco, se utilizan 3 niveles de tablas de página de igual tamaño, (en realidad, Linux usa 4), ¿qué tamaño tiene el directorio de páginas y las tablas en cada nivel? ¿Cuántas tablas en realidad se usarían? Muestre cuáles serían, en forma escrita o tabular.

Pregunta 3 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **INF239-E2-Buzón 3 del Examen 2** en PAIDEIA **antes de las 10:15**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <su_código_de_8_dígitos>_23.txt. Por ejemplo, 20202912_23.txt.

Se cuenta con una memoria SD, la cuál se quiere analizar en detalle desde un sistema operativo Linux. Primero se utiliza el utilitario *fdisk* sobre el dispositivo completo:

```
$ sudo fdisk -l
```

```

Disk /dev/mmcblk0: 15.9 GB, 15931539456 bytes
255 heads, 63 sectors/track, 1936 cylinders, total 31116288 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

```

Device	Boot	Start	End	Blocks	Id	System
/dev/mmcblk0p1		8192	31116287	15554048	c	W95 FAT32 (LBA)

```
Disk /dev/sda: ***** (resto no es de interés)
```

Entonces, sabiendo que encontraremos un filesystem FAT32 en la partición 1, configuramos *mttools.conf* para que corresponda la unidad 'x:' con el dispositivo */dev/mmcblk0p1* y ejecutamos el comando *minfo*:

```
device information:
=====
filename="/dev/mmcblk0p1"
sectors per track: 63
heads: 255
cylinders: 1937

mformat command line: mformat -T 31108096 -h 255 -s 63 -H 8192 x:
```

```
bootsector information
=====
banner:"      "
sector size: 512 bytes
cluster size: 64 sectors
reserved (boot) sectors: 598
fats: 2
max available root directory slots: 0
small size: 0 sectors
media descriptor byte: 0xf8
sectors per fat: 0
sectors per track: 63
heads: 255
hidden sectors: 8192
big size: 31108096 sectors
physical drive id: 0x80
reserved=0x1
dos4=0x29
serial number: 37383362
disk label="NO NAME      "
disk type="FAT32      "
Big fatlen=3797
Extended flags=0x0000
FS version=0x0000
rootCluster=2
infoSector location=1
backup boot sector=6
```

```
Infosector:
signature=0x41615252
free clusters=295584
last allocated cluster=190355
```

(a) (1 punto) A partir de la información obtenida, obtener, en forma de tabla, lo que sería el “layout” del filesystem, con los valores propios de este caso:

Indicador	Valor	Comentario (cálculo del valor/valores o si es dato leído)
BootSector	sector 0	Dato
BootInfo	sector ?	?????
BootBackup		
BootInfoBackup		
Fat-1-FirstSector		
Fat-1-Size		
Fat-2-FirstSector		
Fat-2-Size		
RootDir cluster/sector		En FAT32, cluster y sector le que corresponde;
DataAreaSize (clusters/sectors)		En otras, sector inicial y sectores.

(b) (2 puntos) Haciendo un breve cálculo, hacemos un “dump” del directorio raíz:

```
$ sudo hexdump -C -s 4096k -n 32768 /dev/mmcblk0p1
00400000  42 20 00 49 00 6e 00 66 00 6f 00 0f 00 72 72 00 |B .I.n.f.o...rr.|
00400010  6d 00 61 00 74 00 69 00 6f 00 00 00 6e 00 00 00 |m.a.t.i.o...n...|
00400020  01 53 00 79 00 73 00 74 00 65 00 0f 00 72 6d 00 |.S.y.s.t.e...rm.|
00400030  20 00 56 00 6f 00 6c 00 75 00 00 00 6d 00 65 00 |.V.o.l.u...m.e.|
00400040  53 59 53 54 45 4d 7e 31 20 20 20 16 00 8b 77 12 |SYSTEM~1 ...w.|
00400050  35 51 35 51 00 00 78 12 35 51 03 00 00 00 00 00 |5Q5Q...x.5Q.....|
00400060  46 4f 55 4e 44 20 20 20 30 30 30 16 00 00 0f a7 |FOUND 000.....|
00400070  36 51 36 51 00 00 0f a7 36 51 06 00 00 00 00 00 |6Q6Q....6Q.....|
(algunas líneas omitidas ...)
```

```

00401300 42 63 00 61 00 63 00 68 00 65 00 0f 00 79 00 00 |Bc.a.c.h.e...y..|
00401310 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff |.....|
00401320 01 52 00 65 00 61 00 64 00 79 00 0f 00 79 42 00 |.R.e.a.d.y...yB.|
00401330 6f 00 6f 00 73 00 74 00 2e 00 00 00 73 00 66 00 |o.o.s.t....s.f..|
00401340 52 45 41 44 59 42 7e 31 53 46 43 20 00 a1 63 95 |READYB~1SFC ..c.|
00401350 93 51 93 51 00 00 64 95 93 51 08 00 00 00 e0 ff |.Q.Q..d..Q.....|
00401360 42 32 00 32 00 31 00 31 00 34 00 0f 00 cf 34 00 |B2.2.1.1.4....4.|
00401370 2e 00 6a 00 70 00 67 00 00 00 00 00 ff ff ff ff |..j.p.g.....|
00401380 01 49 00 4d 00 47 00 5f 00 32 00 0f 00 cf 30 00 |.I.M.G._.2....0.|
00401390 31 00 39 00 31 00 32 00 31 00 00 00 36 00 5f 00 |1.9.1.2.1...6..|
004013a0 49 4d 47 5f 32 30 7e 31 4a 50 47 20 00 3b 41 98 |IMG_20~1JPG .;A.|
004013b0 93 51 93 51 02 00 4c 0a 91 4f 8e e6 64 67 31 00 |.Q.Q..L..O..dg1.|
004013c0 42 32 00 32 00 31 00 31 00 32 00 0f 00 2f 39 00 |B2.2.1.1.2.../9.|
004013d0 2e 00 6a 00 70 00 67 00 00 00 00 00 ff ff ff ff |..j.p.g.....|
004013e0 01 49 00 4d 00 47 00 5f 00 32 00 0f 00 2f 30 00 |.I.M.G._.2.../0.|
004013f0 31 00 39 00 31 00 32 00 31 00 00 00 36 00 5f 00 |1.9.1.2.1...6..|
00401400 49 4d 47 5f 32 30 7e 32 4a 50 47 20 00 57 41 98 |IMG_20~2JPG .WA.|
00401410 93 51 93 51 02 00 4c 0a 91 4f f1 e6 cf 79 50 00 |.Q.Q..L..O...yP.|
00401420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|

```

Indique los archivos que describe el directorio: nombre corto y largo, su tamaño en bytes y el cluster de inicio, según

Nombre Corto	Nombre Largo	Tamaño en Bytes	1 ^{er} cluster
--------------	--------------	-----------------	-------------------------

(c) (2 puntos) Además, se ejecuta el utilitario de `mtools`, `mshowfat`, que muestra el siguiente resultado recortado y que debe completar y comentar cada línea de cada archivo, a modo de comprobación:

```

$ sudo mshowfat x:
X:/ <2>
$ sudo mshowfat x:*
X:/System Volume Information <3>
X:/_____ <6>
X:/_____ <7>
X:/_____ <8-71935> <131014-190093>
X:/_____ <190094-190192>
X:/_____ <190193-190353>

```

Pregunta 4 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **INF239-E2-Buzón 4 del Examen 2** en PAIDEIA **antes de las 11:00 hrs.** Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_24.txt`. Por ejemplo, `20202912_24.txt`.

Se tiene un sistema de archivos Ext2, cuyo diseño general es el siguiente:

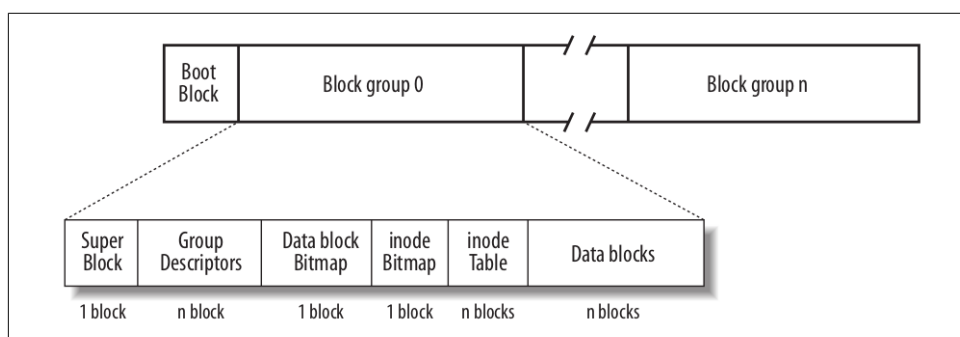


Figure 17-1. Layouts of an Ext2 partition and of an Ext2 block group

n indica que el tamaño es variable en cada caso. Tenga en cuenta además que después del bloque de descriptores se añade un grupo de bloques reservados denominados GDT que sirven solamente para cuando el sistema de archivo desea extenderse, de esta forma el grupo de descriptores podrá crecer. El siguiente sistema de archivo no tiene *Boot Block*.

Al ejecutar el siguiente comando `dumpe2fs` se obtiene la siguiente salida:

```

$ dumped2fs ext2.img
Filesystem volume name: <none>
Last mounted on: <not available>
Filesystem UUID: e3a823fa-84ae-4576-9327-569f8e86d71b
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: ext_attr resize_inode dir_index filetype sparse_super
Filesystem flags: signed_directory_hash
Default mount options: user_xattr acl
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 10016
Block count: 20000
Reserved block count: 1000
Free blocks: 19318
Free inodes: 10005
First block: 0
Block size: 2048
Fragment size: 2048
Reserved GDT blocks: 19
Blocks per group: 16384
Fragments per group: 16384
Inodes per group: 5008
Inode blocks per group: 313
Filesystem created: Tue Jul 2 08:46:00 2013
Last mount time: n/a
Last write time: Tue Jul 2 08:46:00 2013
Mount count: 0
Maximum mount count: -1
Last checked: Tue Jul 2 08:46:00 2013
Check interval: 0 (<none>)
Reserved blocks uid: 0 (user root)
Reserved blocks gid: 0 (group root)
First inode: 11
Inode size: 128
Default directory hash: half_md4
Directory Hash Seed: d1daf3a2-3324-4073-973e-4f5b574fa6cb

```

En la siguiente información se indica qué bloques ocupa cada estructura (por ejemplo, Bloques 0-¿?) y los datos que están entre paréntesis con el signo (+¿?) hacen referencia al desplazamiento relativo dentro de su grupo. Con esta descripción se necesita completar la siguiente información:

Grupo 0: (Bloques 0-¿?) (0,25 puntos)

Primario superbloque en 0, descriptores de grupo en 1-1
 Se reservaron los bloques GDT en ¿?-¿? (0,25 puntos)
 Mapa de bits de bloque en ¿? (+¿?), mapa de bits de nodo-i en ¿? (+¿?) (0,5 puntos)
 tabla de nodos-i en ¿?-¿? (+¿?) (0,5 puntos)
 16038 free blocks, 4997 free inodes, 2 directories
 Bloques libres: ¿?-¿? (0,5 puntos)
 Nodos-i libres: ¿?-¿? (0,5 puntos)

Grupo 1: (Bloques ¿?-¿?) (0,25 puntos)

Respaldo superbloque en 16384, descriptores de grupo en ¿?-¿? (0,25 puntos)
 Se reservaron los bloques GDT en ¿?-¿? (0,25 puntos)
 Mapa de bits de bloque en ¿? (+¿?), mapa de bits de nodo-i en ¿? (+¿?) (0,5 puntos)
 tabla de nodos-i en ¿?-¿? (+¿?) (0,25 puntos)
 ¿? free blocks, ¿? free inodes, 0 directories (0,5 puntos)
 Bloques libres: ¿?-¿? (0,25 puntos)
 Nodos-i libres: ¿?-¿? (0,25 puntos)



Preparado por FS (2,3) y VK (1,4)
con LibreOffice Writer en Linux Mint 20 "Ulyana"

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

Lima, 23 de diciembre de 2020