

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

4ta práctica (tipo a)
(Primer semestre de 2012)

Horario 0781: prof. V.Khlebnikov

Horario 0782: prof. A.Bello R.

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.

Puntaje total: 20 puntos

Pregunta 1 (10 puntos)

a) (2 puntos) En un programa en C se tiene la variable `sBlock` declarada de la siguiente forma:

```
struct super_block {
    ino_t      s_ninodes;      /* # usable inodes on the minor device */
    zone1_t    s_nzones;      /* total device size, including bit maps etc */
    short      s_imap_blocks;  /* # of blocks used by inode bit map */
    short      s_zmap_blocks;  /* # of blocks used by zone bit map */
    zone1_t    s_firstdatazone; /* number of first data zone */
    short      s_log_zone_size; /* log2 of blocks/zone */
    short      s_pad;          /* try to avoid compiler-dependent padding */
    off_t      s_max_size;     /* maximum file size on this device */
    zone_t     s_zones;        /* number of zones (replaces s_nzones in V2) */
    short      s_magic;        /* magic number to recognize super-blocks */
    /* The following items are valid on disk only for V3 and above */
    /* The block size in bytes. Minimum MIN_BLOCK_SIZE. SECTOR_SIZE
     * multiple. If V1 or V2 filesystem, this should be
     * initialised to STATIC_BLOCK_SIZE. Maximum MAX_BLOCK_SIZE.
     */
    short      s_pad2;         /* try to avoid compiler-dependent padding */
    unsigned short s_block_size; /* block size in bytes. */
    char       s_disk_version; /* filesystem format sub-version */
} sBlock;
```

Sabiendo que en MINIX el primer mapa de bits siempre se alcanza avanzando 2 bloques. Indique una formula en lenguaje C para obtener el número de bytes que se necesitan desplazar desde el inicio del sistema de archivos para ubicar el *n-ésimo* i-nodo. Recuerde que el tamaño de un i-nodo es de 64 bytes.

b) (2 puntos) En una terminal se ejecutan las siguientes ordenes:

```
alulab@infor:~$ ls -l emacs
lrwxrwxrwx 1 alulab alulab 10 jun  6 12:49 emacs -> emacs.conf
```

```
alulab@infor:~$ stat emacs
  File: «emacs» -> «emacs.conf»
  Size: 10          Blocks: 0          IO Block: 4096   enlace simbólico
Device: 802h/2050d Inode: 3334988      Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/alulab)   Gid: ( 1000/alulab)
Access: 2012-06-06 12:49:34.000000000 -0500
Modify: 2012-06-06 12:49:33.000000000 -0500
Change: 2012-06-06 12:49:33.000000000 -0500
```

```
alulab@infor:~$ stat -L emacs
  File: «emacs»
  Size: 3545        Blocks: 8          IO Block: 4096   fichero regular
Device: 802h/2050d Inode: 3336542      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/alulab)   Gid: ( 1000/alulab)
Access: 2012-06-06 10:40:02.000000000 -0500
Modify: 2012-04-24 13:22:36.000000000 -0500
Change: 2012-04-24 13:22:36.000000000 -0500
```

donde **Size** es el tamaño del archivo en bytes, **Blocks** es el tamaño del archivo en bloques de 512 bytes. ¿Qué tienen en común y en qué se diferencian los enlaces simbólicos con los accesos directos del sistema operativo Windows?

c) (4 puntos) Se tiene el siguiente *dump* de un sistema de archivo de MINIX:

```
00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00000400  00 03 00 00 00 00 01 00 01 00 10 00 00 00 00  |.....|
00000410  ff ff ff 7f 68 01 00 00 5a 4d 00 00 00 10 00  |...h...ZM....|
00000420  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00  |.....|
00000430  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00002000  07 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
00002010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00003000  ff 1f 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
00003010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00004000  ff 41 02 00 02 00 02 00 c0 00 00 00 ec 73 f3 4d  |.A.....s.M|
00004010  e7 73 f3 4d e7 73 f3 4d 10 00 00 00 00 00 00 00  |.s.M.s.M.....|
00004020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00004040  a4 81 01 00 00 00 00 00 f8 96 00 00 97 ba f3 4d  |.....M|
00004050  e0 73 f3 4d e0 73 f3 4d 11 00 00 00 12 00 00 00  |.s.M.s.M.....|
00004060  13 00 00 00 14 00 00 00 15 00 00 00 16 00 00 00  |.....|
00004070  17 00 00 00 19 00 00 00 00 00 00 00 00 00 00  |.....|
00004080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
00010000  01 00 00 00 2e 00 00 00 00 00 00 00 00 00 00  |.....|
00010010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |.....|
*
```

Considerando la estructura de la parte a) y las definiciones de tipo dadas abajo, diga, ¿cuántas zonas libres tiene este sistema de archivos?

```

/* Types used in disk, inode, etc. data structures. */
typedef short      dev_t;      /* holds (major|minor) device pair */
typedef char       gid_t;      /* group id */
typedef unsigned long ino_t;    /* i-node number (V3 filesystem) */
typedef unsigned short mode_t; /* file type and permissions bits */
typedef short      nlink_t;    /* number of links to a file */
typedef unsigned long off_t;    /* offset within a file */
typedef int        pid_t;      /* process id (must be signed) */
typedef short      uid_t;      /* user id */
typedef unsigned long zone_t;   /* zone number */
typedef unsigned long block_t;  /* block number */
typedef unsigned long bit_t;    /* bit number in a bit map */
typedef unsigned short zone1_t; /* zone number for V1 file systems */
typedef unsigned short bithub_t; /* collection of bits in a bitmap */

```

- d) (2 puntos) Considere un i-nodo de un sistema de archivos con 7 punteros directos y 1 simplemente indirecto. Además cada puntero necesita 4 bytes. Este sistema de archivos cuenta con 1031 zonas libres y suficientes i-nodos disponibles. ¿Se puede copiar un archivo de 4222970 bytes en este sistema de archivos? Explique su respuesta. El bloque es de 4KB.

Pregunta 2 (1 punto) (*A. S. Tanenbaum, Modern Operating Systems, 3rd ed.*) The first version of MS-DOS used FAT-12 with 512-byte blocks, giving a maximum partition size of $2^{12} \times 512$ bytes, about 2 MB. This system worked well for floppy disks, but when hard disks came out, it became a problem. Microsoft solved the problem with a change. This change preserved the structure and size of the FAT-12 table, but allowed disk partitions of up to 16 MB. What change was it?



Pregunta 3 (1 punto) Why FAT is "fat" and i-node is "thin"?

Pregunta 4 (8 puntos) (*A. Casillas, L. Iglesias*) El sector de arranque de un disco MS-DOS de 64 MiB contiene la siguiente información: 512 bytes/sector, 4 sectores/*cluster*, 2 FAT. Determine:

- (2 puntos) Tipo de FAT necesaria (FAT-12, FAT-16 o FAT-32) para poder gestionar todos los *clusters* del disco.
- (2 puntos) La cantidad de sectores que ocupa cada tabla FAT.
- (2 puntos) Número máximo de archivos que puede albergar el directorio raíz si ocupa 64 sectores, y la entrada del directorio contiene el nombre de archivo en 11 bytes más su tamaño en 4 bytes más el número del 1er cluster en 2 bytes más otros atributos.
- (2 puntos) El número del primer sector del *cluster* 2.



La práctica fue preparada por AB(1) y VK(2,3,4)

Profesores del curso: V.Khlebnikov
A.Bello R.

Pando, 6 de junio de 2012