PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMA DE ARCHIVO exFAT

El objetivo del presente laboratorio es explorar y reconocer la estructura (*layout*) del sistema de archivo *Extended File Allocation Table* (en forma corta *exFAT*).

"*exFAT (Extended File Allocation Table) is a proprietary file system designed especially for flash drives developed by Microsoft, which has applied for patent protection. exFAT is also supported in a umber of media devices such as modern flat panel TVs, media centers, and portable media players" (from Wikipedia, the free encyclopedia)*

El primer paso para este laboratorio es conocer los detalles de la estructura de este sistema de archivos. Para lograr este propósito usted debe leer *reverse-engineering-microsoft-exfat-file-system.pdf*, archivo que se adjunta con esta guía. Adicionalmente también se adjunta la imagen de un dispositivo con sistema de archivo *exFAT* (imagen.lab.usb.zip). Al desempaquetar este archivo obtendrá la imagen que posee un tamaño de cerca de 4MB. Así que debe tener las consideraciones de espacio respectivas.

A continuación se presenta un pequeño programa que lee el V*olume Boot Record* conteniendo información relevante acerca de este sistema de archivos.

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "exFATstruct.h"


int main(int n, char *name[]){
    int fd,exp;
    exFatBootSector boot;


    if(n!=2) {
        printf("Uso: %s <Image File System>\n",name[0]);
        exit(1);
    }
    if((fd = open(name[1],O_RDONLY))<0)
         perror("No se pudo abrir imagen del disco\n");
    if(read(fd,&boot,sizeof(boot))<0)
         perror("No se pudo leer imagen del disco\n");
    printf("\t\tDatos relevantes del disco con sistema de archivo exFat\n");
    printf("El tamaño de la imagen en sectores es de:%ld\n",boot.sizeVol);
    printf("La FAT se encuentra en el sector:%d\n",boot.FATOffset);
    printf("El tamaño de la FAT en sectores es:%d\n",boot.FATlen);
    printf("La data inicia en el sector:%d\n",boot.ClusterHeapOffset);
    printf("El tamaño del espacio para la data en cluster es:%d\n",boot.ClusterCount);
    printf("El directorio raíz se encuentra en el cluster:%d\n",boot.RootDirFirstCluster);
    exp = boot.BytePerSector;
    printf("El tamaño de un sector en bytes es de:%d\n",(int)pow(2.0,exp));
    exp = boot.SectorPerCluster;
    printf("El número de sectores por cluster es de:%d\n",(int)pow(2.0,exp));
    printf("Número de FAT's en este ssitema de archivos:%d\n",boot.NumberFats);
    exit(0);
}

```

```
alejandro@MINIX:~/Cursos dictados/Sistemas Operativos/Laboratorio/7$ ./leeBootExFat imagen.lab.usb
                Datos relevantes del disco con sistema de archivo exFat
El tamaño de la imagen en sectores es de:3932128
La FAT se encuentra en el sector:128
El tamaño de la FAT en sectores es:992
La data inicia en el sector:1152
El tamaño del espacio para la data en cluster es:122843
El directorio raíz se encuentra en el cluster:4
El tamaño de un sector en bytes es de:512
El número de sectores por cluster es de:32
Número de FAT's en este ssitema de archivos:1
```

A continuación la estructura empelada:

```
typedef struct {
    char jump[3];
    char FSName[8];
    char Chunk[53];
    long offsetPart;
    long sizeVol;
    int FATOffset;
    int FATlen;
    int ClusterHeapOffset;
    int ClusterCount;
    int RootDirFirstCluster;
    int VSN;
    short FSR;
    short FlagVol;
    char BytePerSector;
    char SectorPerCluster;
    char NumberFats;
    char DriveSelect;
    char PercentUse;
    char reserved[7];
    char BootCode[390];
    short BootSignature;
} __attribute((packed)) exFatBootSector;
```

Para compilar el programa, escriba en la línea de ordenes:

```
gcc -o leeBootExFat  leeBootExFat.c -lm
```

donde *leeBootExFat.c* es el programa fuente.

TAREA

Con esta información relevante se le solicita lo siguiente:

1) Escriba un programa en lenguaje C que reciba como argumento un número que indica el número de un *cluster*. El programa debe devolver si está libre u ocupado. La búsqueda debe ser realizada en el área de la FAT.

2) Escriba un programa en lenguaje C que reciba como argumento un número que indica el número de un *cluster*. El programa debe devolver si está libre u ocupado. La búsqueda debe ser realizada en el mapa de bits.

3) Escriba un programa en lenguaje C que lea todas las entradas válida del directorio raíz y muestre por pantalla el nombre (corto) de cada uno de ellos, el tamaño y el primer *cluster*.

4) Escriba un programa en lenguaje C que lea todas las entradas válida del directorio raíz y muestre por pantalla el nombre (corto) de cada uno de ellos y pregunte: `¿Desea mostrar la cadena de clusters?` Si la respuesta es: `Si`. El programa debe mostrar todos los *clusters* correspondientes a ese archivo. Si es: `No` se pasa a la siguiente entrada.

Prof: Alejandro T. Bello Ruiz