

Proyecto Final

Sistemas Operativos

15 de abril de 2020

1. Editor Hexadecimal de Archivos

El proyecto consiste en programar un editor de archivos que permita ver su contenido en Hexadecimal y texto. El editor debe permitir manejar archivos tan grandes como la memoria virtual de la maquina lo permita.

1.1. Segunda sesión

En esta segunda sesión se va a leer el archivo, seleccionado en la primera sesión, que se desea editar. Para ello se debe hacer un programa que permita leer un archivo por bloques y mostrarlo en la pantalla. La presentación debe mostrar tanto los códigos como del texto así como la dirección relativa del archivo. En la Figura 1 se muestra como debe verse.

00000000	eb 44 90 29 70 54 2c 55 49 48 43 00 02 01 01 00	.D.)pT,UIHC.....
00000010	02 e0 00 40 0b f0 09 00 12 00 02 00 00 00 00 00	...@.....
00000020	00 00 00 00 00 00 00 00 00 00 a3 4e 4f 20 4e 41NO NA
00000030	4d 45 20 20 20 20 46 41 54 20 20 20 20 00 00 00	ME FAT ..
00000040	00 10 00 00 00 00 fa 33 db 8e d3 bc ff 7b fb ba3.....{..
00000050	c0 07 8e da 00 3e 24 00 00 75 3d 1e b8 40 00 8e>\$...u=..@..
00000060	c0 26 ff 0e 13 00 cd 12 c1 e0 06 8e c0 33 ff 33	..&.....3.3
00000070	c0 8e d8 c5 36 78 00 fc b9 0b 00 f3 a4 1f a1 186x.....
00000080	00 26 a2 04 00 1e 33 c0 8e d8 a3 78 00 8c 06 7a	..&...3....x...z
00000090	00 1f 8a 16 24 00 cd 13 a0 10 00 98 f7 26 16 00\$......&..
000000a0	03 06 0e 00 50 91 b8 20 00 f7 26 11 00 8b 1e 0bP... ..&.....
000000b0	00 03 c3 48 f7 f3 50 03 c1 a3 3e 00 b8 00 10 8e	...H..P...>.....
000000c0	c0 33 ff 59 89 0e 44 00 58 a3 42 00 33 d2 e8 73	.3.Y..D.X.B.3..s
000000d0	00 33 db 8b 0e 11 00 8b fb 51 b9 0b 00 be d5 01	.3.....Q.....
000000e0	f3 a6 59 74 05 83 c3 20 e2 ed e3 35 26 8b 47 1c	..Yt... ..5&.6.
000000f0	26 8b 57 1e f7 36 0b 00 fe c0 8a c8 26 8b 57 1a	&.W..6.....&.W.
00000100	4a 4a a0 0d 00 32 e4 f7 e2 03 06 3e 00 83 d2 00	JJ...2.....>....
00000110	bb 00 08 8e c3 33 ff 06 57 e8 28 00 8d 36 0b 003..W.(.6..
00000120	cb be 98 01 eb 03 be ad 01 e8 09 00 be c2 01 e8
00000130	03 00 fb eb fe ac 0a c0 74 09 b4 0e bb 07 00 cdt.....
00000140	10 eb f2 c3 50 52 51 03 06 1c 00 13 16 1e 00 f7PRQ.....
00000150	36 18 00 fe c2 8a da 33 d2 f7 36 1a 00 8a fa 8b	6.....3..6.....
00000160	d0 a1 18 00 2a c3 40 50 b4 02 b1 06 d2 e6 0a f3*.@P.....
00000170	8b ca 86 e9 8a 16 24 00 8a f7 8b df cd 13 72 a6\$......r.
00000180	5b 59 8b c3 f7 26 0b 00 03 f8 5a 58 03 c3 83 d2	IY...&....ZX....

Figura 1: Vista del Editor Hexadecimal

1.2. Archivo, código ejemplo y objetivo de la sesión

En el espacio de Brighspace se encuentra un archivo de ejemplo (un disco de arranque para OS2). En la Figura 1 se muestra el inicio de dicho archivo.

A continuación se tiene el código con el que se mostro cada una de las líneas. La función regresa una cadena y tiene como parámetros el inicio del archivo y su *offset*.

```
char *hazLinea(char *base, int dir) {
    char linea[100]; // La linea que se crea es menor
    int o=0;
    // Muestra 16 caracteres por cada linea
    o += sprintf(linea, "%08x ", dir); // offset en hexadecimal
    for(int i=0; i < 4; i++) {
        unsigned char a,b,c,d;
        a = base[dir+4*i+0];
        b = base[dir+4*i+1];
        c = base[dir+4*i+2];
        d = base[dir+4*i+3];
        o += sprintf(&linea[o], "%02x %02x %02x %02x|", a, b, c, d);
    }
    for(int i=0; i < 16; i++) {
        if (isprint(base[dir+i])) {
            o += sprintf(&linea[o], "%c", base[dir+i]);
        }
        else {
            o += sprintf(&linea[o], ".");
        }
    }
    sprintf(&linea[o], "\n");

    return(strdup(linea));
}
```