Universidad de los Andes
Facultad de Ingeniería

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
ISIS1304
FUNDAMENTOS DE INFRAESTRUCTURA TECNOLÓGICA

Proyecto 1 - 2016-2

Este proyecto vale 15% de la nota del curso.
Debe ser elaborado en grupo (mínimo 2, máximo 3 integrantes).
No se permite ningún tipo de consulta entre grupos.
Se debe entregar por Sicua a más tardar el 23 de octubre a las 23:50

A. OBJETIVOS

- Practicar el lenguaje C y desarrollar un programa de complejidad pequeña.
- Entender la estructura de la memoria en bytes y bits, y las operaciones de C para el manejo de bits.
- Comprender el propósito y el procedimiento de codificación de un archivo binario a uno de texto

B. DESCRIPCIÓN DEL PROBLEMA

En los inicios del correo electrónico, el único tipo de archivos soportado para ser enviado era archivos de texto, por lo cual, para otros tipos de archivos binarios era necesario codificarlos, de alguna manera, en ASCII. Esta codificación consistía en transformar cualquier archivo binario a un archivo de texto, el cual estaba compuesto de todas las letras mayúsculas y minúsculas del alfabeto en inglés, junto con algunos caracteres adicionales (ver la Figura 1).

USASCII code chart

b7 b6 b5 / b4 b3 b2 b1 / Row	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	0	1	2	3	4	5	6	7
0 0 0 0 0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0 2	STX	DC2	"	2	B	R	b	r
0 0 1 1 3	ETX	DC3	#	3	C	S	c	s
0 1 0 0 4	EOT	DC4	$	4	D	T	d	t
0 1 0 1 5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0 6	ACK	SYN	&	6	F	V	f	v
0 1 1 1 7	BEL	ETB	'	7	G	W	g	w
1 0 0 0 8	BS	CAN	(	8	H	X	h	x
1 0 0 1 9	HT	EM	)	9	I	Y	i	y
1 0 1 0 10	LF	SUB	*	:	J	Z	j	z
1 0 1 1 11	VT	ESC	+	;	K	[	k	{
1 1 0 0 12	FF	FS	,	<	L	\	l	|
1 1 0 1 13	CR	GS	-	=	M	]	m	}
1 1 1 0 14	SO	RS	.	>	N	^	n	~
1 1 1 1 15	SI	US	/	?	O	_	o	DEL

Figura 1: Tabla de codificación ASCII

El propósito de este proyecto es ejemplificar el proceso de codificación y decodificación de archivos binarios a archivos de texto en una versión simplificada y con algunas modificaciones.

Codificación

La codificación que se va a realizar en el proyecto consiste en tomar un archivo binario con cualquier número de bytes, dividirlo en grupos de 5 bits, y a cada grupo asignarle un carácter en código ASCII según la codificación mostrada en la tabla 1.

Binario	Codificación	Caracter
0 0 0 0 0	0 1 0 0 0 0 0 0	@
0 0 0 0 1	0 1 0 0 0 0 0 1	A
0 0 0 1 0	0 1 0 0 0 0 1 0	B
0 0 0 1 1	0 1 0 0 0 0 1 1	C
0 0 1 0 0	0 1 0 0 0 1 0 0	D
0 0 1 0 1	0 1 0 0 0 1 0 1	E
0 0 1 1 0	0 1 0 0 0 1 1 0	F
0 0 1 1 1	0 1 0 0 0 1 1 1	G
0 1 0 0 0	0 1 0 0 1 0 0 0	H
0 1 0 0 1	0 1 0 0 1 0 0 1	I
0 1 0 1 0	0 1 0 0 1 0 1 0	J
0 1 0 1 1	0 1 0 0 1 0 1 1	K
0 1 1 0 0	0 1 0 0 1 1 0 0	L
0 1 1 0 1	0 1 0 0 1 1 0 1	M
0 1 1 1 0	0 1 0 0 1 1 1 0	N
0 1 1 1 1	0 1 0 0 1 1 1 1	O
1 0 0 0 0	0 0 1 1 0 0 0 0	0
1 0 0 0 1	0 0 1 1 0 0 0 1	1
1 0 0 1 0	0 0 1 1 0 0 1 0	2
1 0 0 1 1	0 0 1 1 0 0 1 1	3
1 0 1 0 0	0 0 1 1 0 1 0 0	4
1 0 1 0 1	0 0 1 1 0 1 0 1	5
1 0 1 1 0	0 0 1 1 0 1 1 0	6
1 0 1 1 1	0 0 1 1 0 1 1 1	7
1 1 0 0 0	0 0 1 1 1 0 0 0	8
1 1 0 0 1	0 0 1 1 1 0 0 1	9
1 1 0 1 0	0 0 1 1 1 0 1 0	:
1 1 0 1 1	0 0 1 1 1 0 1 1	;
1 1 1 0 0	0 0 1 1 1 1 0 0	<
1 1 1 0 1	0 0 1 1 1 1 0 1	=
1 1 1 1 0	0 0 1 1 1 1 1 0	>
1 1 1 1 1	0 0 1 1 1 1 1 1	?

Tabla 1: Tabla de codificación binario a ASCII

Note que la codificación consiste en tomar los cinco bits originales y adicionarles un encabezado de tres bits, completando de esta manera los 8 bits del carácter codificado. Como se puede apreciar en la tercera columna, el código resultante es siempre un carácter ASCII válido.

En la figura 2 se muestra gráficamente el proceso que se va a seguir con un archivo binario de entrada. Cada cuadrado representa un bit, y los bytes están enmarcados en rojo. El archivo binario se divide en grupos de a 5 bits (representados en la figura por grupos de colores diferentes). Por lo tanto el archivo binario de la figura tiene 5 Bytes, los cuales se dividen en 8 grupos de a 5 bits. El archivo codificado en ASCII tendrá por lo tanto 8 Bytes, cada uno correspondiente a uno de los grupo de a 5 del archivo original. Como se mencionó anteriormente, los caracteres del archivo ASCII se componen de un

encabezado determinado por la tabla 1 en los tres bits más significativos y de los cinco bits originales en los bits menos significativos.

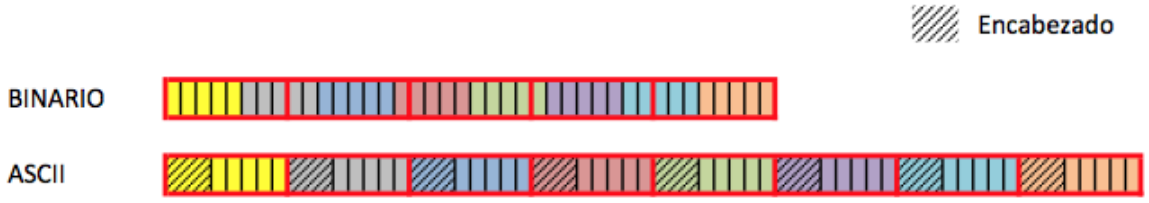

Figura 2: Representación gráfica de codificación

Puede ocurrir que al dividir el archivo, el último grupo tenga menos de 5 bits, en cuyo caso se le adjuntan al final bits en cero hasta llegar a 5. A este grupo de 5 posteriormente se le aplica la misma transformación de la tabla 1. Un ejemplo se muestra en la figura 3, donde en el archivo original se tienen 4 Bytes los cuales se dividen en 6 grupos de a 5 bits y uno de 2 bits. Se puede apreciar que en el archivo ASCII codificado se incluyen estos bits, completándolos con una secuencia de ceros para que formen un grupo de 5 y aplicándoles el encabezado correspondiente.

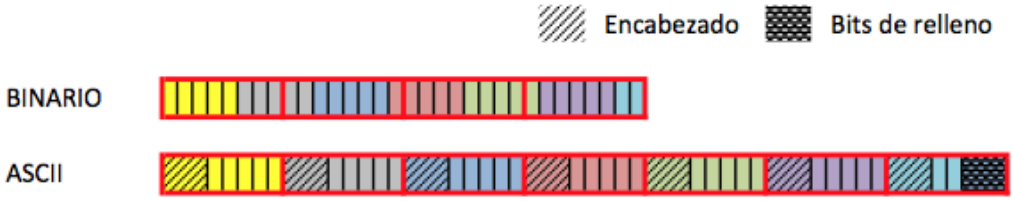

Figura 3: Representación gráfica de codificación

Decodificación

Para decodificar el archivo de texto (para recuperar el original), se deben tomar todos los Bytes, eliminarles el encabezado de 3 bits y juntar los grupos de 5 bits. Dado caso que después de juntar todos los grupos de bits al final no se complete un número entero de Bytes, el Byte incompleto no se debe escribir (pues el hecho de que no se completen los 8 bits implica que éstos fueron los bits de relleno adicionales que se agregaron al hacer el primer procedimiento, y, en consecuencia, no hacen parte del mensaje original).

Programa

En el archivo adjunto (main.c) se encuentra el esqueleto del programa. El programa recibe dos parámetros por línea de comando: el primero corresponde al archivo de entrada y el segundo al archivo donde se va a guardar el resultado. Si se va a realizar una codificación, el archivo de entrada es el binario, y el archivo del resultado es uno de texto. En el caso de la decodificación el archivo de entrada es de texto mientras que el de salida debe ser del mismo tipo que el archivo codificado: por ejemplo, si se está decodificando una codificación de un archivo bmp, el archivo de salida debe ser bmp. Al ejecutar el programa se escoge la opción que se quiere realizar, 1 si se quiere codificar o 2 si se quiere decodificar. Se creó un tipo de datos `ARCHIVO` con la siguiente estructura:

```
typedef struct archivo
{
	int tamanho;
	unsigned char *informacion;
} ARCHIVO;
```

El esqueleto entregado se encarga de cargar el archivo de entrada en la variable `arch` de tipo `ARCHIVO` y de guardar en el archivo de salida la información guardada en la variable `resultado`. El primer campo de esta variable, `tamanho`, corresponde al tamaño en Bytes del archivo, y el segundo campo, `informacion`, corresponde al arreglo de caracteres que representan la información guardada en el archivo. El campo que se debe modificar en el proyecto es el de `informacion` del archivo `resultado`, el otro campo (`tamanho`) ya se encuentra modificado en el esqueleto entregado.

Nota: Recuerden que si quieren acceder al Byte `i` de la información del archivo `arch` se puede lograr con la instrucción `arch->informacion[i]` donde si `i` es igual a 0 se accede al primer byte del vector (por ejemplo, en el caso de la figura 2, al Byte que contiene las casillas amarillas).

Hay unos procedimientos que no se deben modificar, lo cual se indica en los comentarios. Los demás procedimientos se deben implementar según la documentación adjunta a cada uno.

C. ESPECIFICACIONES

- Los programas se deben escribir en C (en Visual Studio). Nota importante: los programas se calificarán únicamente usando el ambiente de visual; si el programa no compila en este ambiente, se considerará que no corre (así compile en otros ambientes).
- Legibilidad del programa: sangrar el programa; escribir comentarios explicando el código; nombres dicientes de variables.
- Debe respetar la estructura y requerimientos del código entregado. En particular, debe usar los procedimientos y variables del esqueleto, y no pueden crear procedimientos adicionales.

D. CONDICIONES DE ENTREGA

- Entregar el código fuente junto con el ejecutable en un archivo *.zip. **Al comienzo del archivo fuente escriba los nombres de los miembros, sus códigos y correos, de lo contrario no será evaluado (ver esqueleto)**. Si su programa no funciona o si su solución tiene particularidades, puede enviar un archivo .doc explicando por qué cree que no funciona o qué fue lo que hizo.
- El trabajo se realiza en grupos de máximo **3** personas. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).
- Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación puede afectar la nota de todos los miembros.
- El proyecto debe ser entregado por Sicua por uno solo de los integrantes del grupo.
- **Se debe entregar por Sicua a más tardar el 23 de octubre a las 23:50.**

E. CASOS DE PRUEBA

Para hacer pruebas se adjuntan 3 casos de prueba: dos archivo de texto (`caso1.txt` y `caso2.txt`) y un archivo de mapa de bits (`casoImagen.bmp`). Además, se adjuntan los

archivos de texto resultantes de la codificación de cada uno de los dos casos de prueba con el objetivo de poder verificar el funcionamiento de la codificación o decodificación por separado.

F. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Para las funciones propuestas se harán cuatro pruebas: dos de los casos de prueba entregados (seleccionados al azar) y otros dos nuevos.
- Inspección del código (50%). Se consideran tres aspectos:
  - ✓ 10% - legibilidad (nombres dicientes para variables, comentarios e indentación)
  - ✓ 20% - manejo de bits (uso de los operadores de bits de c: >>, &, etc.)
  - ✓ 20% - manejo de la estructura de datos (recorrido, manejo de los elementos).

G. RECOMENDACIONES

- Recuerden que los trabajos hechos en grupo y entregados individualmente (o en grupos diferentes al original) son una forma de fraude académico.