Universidad de los Andes
Facultad de Ingeniería

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
ISIS1304
FUNDAMENTOS DE INFRAESTRUCTURA TECNOLÓGICA

Proyecto 1 - 2018-1

Este proyecto vale 15% de la nota del curso.
Debe ser elaborado en grupo (3 integrantes).
No se permite ningún tipo de consulta entre grupos.
Se debe entregar por Sicua+ a más tardar el 15 de abril a las 23:55

A. OBJETIVOS

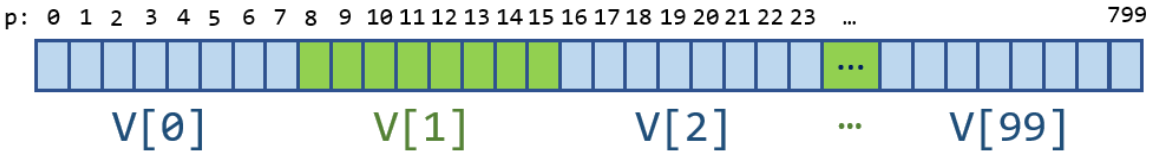
- Practicar el lenguaje C y desarrollar un programa de complejidad pequeña.
- Conocer las operaciones de C para el manejo de bits.
- Aplicar lo anterior en un editor de bits.

B. DESCRIPCIÓN DEL PROBLEMA

El objetivo de este proyecto es desarrollar un editor de bits. Este editor permitirá escribir o leer una cantidad arbitraria de bits (máximo 16) en una posición específica de un vector `V` en memoria. Las variables del problema son:

- `unsigned char V[100]`: vector de 100 chars que se inicializa en 0 al inicio del programa.
- `unsigned char *s`: cadena de caracteres de 1 y 0 que representan un número binario.
- `int p`: posición relativa del bit en el vector `V` donde se va a empezar a escribir el número binario descrito en `s`.
- `int l`: longitud de la cadena de caracteres `s`.

En el siguiente esquema se puede visualizar la posición `p` sobre el vector `V`.



El programa tienes dos modalidades: escritura y lectura.

- Modo escritura: El programa recibe del usuario una cadena de caracteres `s` y la interpreta como un número binario (la cadena solo puede estar compuesta por caracteres '0' y '1'). Luego, escribe cada valor de ese número en los bits que comienzan en la posición `p` del vector `V (posición en bits)`. La posición también es dada por el usuario (no necesariamente es múltiplo de 8)
- Modo lectura: El programa recibe del usuario la posición `p` de la cual va a empezar a leer los bits del vector `V`. También recibe la longitud `l` de caracteres que va a leer. Luego retorna la cadena de caracteres s.

Los identificadores para cada modalidad son:

```
0 = terminar
1 = escritura
2 = lectura
```

Ejemplos

- Modo escritura:

```
Escriba la modalidad deseada: 1
Ingrese la cadena s: 10110110111
Ingrese la posición p: 33
```

Entonces, lo que debe hacer el programa es escribir los bits en rojo en la posición 33:

POSICIÓN	0	1	…	33	34	35	36	37	38	39	40	41	42	43	…	798	799
V	0	0	…	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	…	0	0

Recalcamos que la posición p es en bits. En este ejemplo, la cadena se extiende desde el segundo bit de V[4] (bit 33) hasta el cuarto bit de V[5] (bit 43).

- Modo lectura:

```
Escriba la modalidad deseada: 2
Ingrese la longitud l: 4
Ingrese la posición p: 36
```

Si usamos el vector V que escribimos en el ejemplo anterior, lo que debe hacer el programa es leer los 4 bits en rojo empezando en la posición 36:

POSICIÓN	0	1	…	33	34	35	36	37	38	39	40	41	42	43	…	798	799
V	0	0	…	1	0	1	**1**	**0**	**1**	**1**	0	1	1	1	…	0	0

Y finalmente retorna:

```
La cadena deseada es: 1011
```

Restricciones y consideraciones adicionales

El programa debe tener en tener en cuenta y controlar que:

- El tamaño máximo de la cadena `s` es 16 caracteres, es decir, puede representar máximo 16 bits (2 bytes).
- El programa debe preguntar de forma iterativa la modalidad deseada hasta que se de la opción 0 de terminar.
- Si `l-1+p>799` (`s` no cabe completo en `V`) solo se escribe lo que quepa del vector `s`. Análogamente con la lectura, es decir, solo se lee hasta terminar el vector `V`, no más allá.

El programa

En el archivo adjunto ("`main.c`"), encuentra el esquema del programa. El programa se invoca por línea de comando.

El programa ya tiene definido el procedimiento `main()` que pregunta por la modalidad, y lee por consola los datos de cada una.

Los procedimientos que el grupo debe completar son los siguientes:

- `escribir(unsigned char *V, char *s, int p)`: El procedimiento de lectura recibe el apuntador al vector, la cadena de caracteres s y el entero p en el cual se empieza a escribir.
- `leer(unsigned char *V, char *s, int p, int l)`: El procedimiento de lectura recibe el apuntador al vector, la cadena de caracteres s en la cual se va a escribir lo leído, el entero p en el cual se empieza a leer, y la longitud l de la cadena a leer.

Para desarrollar el programa pueden crear las funciones adicionales que necesiten (recomendado), las cuales deben estar debidamente comentadas y documentadas.

C. ESPECIFICACIONES

- Los programas se deben escribir en C (usando el compilador de Visual C++). Nota importante: los programas se calificarán únicamente usando el compilador de Visual Studio 2015; si el programa no compila en este ambiente, se considerará que no corre (así compile en otros ambientes).
- Legibilidad del programa: indentar el programa; escribir comentarios explicando el código.
- Debe respetar la estructura del código entregado. En particular, debe usar los procedimientos y variables del esqueleto.

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 `.docx` explicando por qué cree que no funciona o qué fue lo que hizo.
- El trabajo se realiza en grupos de **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.
- **Se debe entregar por Sicua+ a más tardar el 15 de abril a las 23:55.**

E. CASOS DE PRUEBA

Se recomienda hacer pruebas usando los siguientes valores:

1. Prueba 1: `s="10101110"` `p=64` `l=8`
2. Prueba 2: `s="10101"` `p=657` `l=5`
3. Prueba 3: `s="10101"` `p=647` `l=5`
4. Prueba 4: `s="101011110110001"` `p=508` `l=15`

F. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Para las funciones propuestas se harán 8 pruebas: los 4 casos de prueba entregados y otros 4 nuevos. De cada caso, se escribirá s y luego se leerá lo que se escribió. Para cada caso, se revisará si la salida es correcta o no según los requerimientos establecidos en el enunciado. Cada prueba vale 6.25%.
- 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.