

Ejercicio



Compresor de Datos

Integrantes del equipo

César Mauricio Arellano Velásquez

Raúl González Portillo

Profesor

César Arturo Ángeles Ruiz

Materia

Taller de Desarrollo de Aplicaciones

Introducción

En ciencias de la computación, la compresión de datos es la reducción del volumen de datos tratables para representar una determinada información empleando una menor cantidad de espacio. Al acto de compresión de datos se denomina «**compresión**», y al contrario «**descompresión**».

El espacio que ocupa una información codificada (datos, señal digital, etc.) sin compresión es el producto entre la frecuencia de muestreo y la resolución. Por tanto, cuantos más bits se empleen mayor será el tamaño del archivo. No obstante, la resolución viene impuesta por el sistema digital con que se trabaja y no se puede alterar el número de bits a voluntad; por ello, se utiliza la compresión, para transmitir la misma cantidad de información que ocuparía una gran resolución en un número inferior de bits.

La compresión es un caso particular de la codificación, cuya característica principal es que el código resultante tiene menor tamaño que el original.

Definición del problema

El objetivo del programa era comprimir 4 números ingresados por terminal, en un int de 16 bits, es decir cada número ingresado, debía ser de 4 bits (0 al 15), para resolver el problema se utilizaron operadores Bitwise que permitieron **Comprimir** y **Extraer** los datos de la variable donde se almacenaban los datos.

Pseudocódigo

#define exit(0) -> función para finalizar la ejecución del programa.

#define sscanf -> Formatear texto de una cadena.

Principal(argc,argv)

{

Num=0,Opcion=0,Extract[4];

ContNum = argc-1;

Si(argc > 5 OR argc = 1)

{

Imprimir("Los parámetros no son los adecuados");

```

        exit(0);
    }
Si no
{
    ComprimirDatos(argv,ContNum | Num);
    ExtraerDatos(ContNum,Num, | Extract);
    Hacer
    {
        Imprimir("Compresor de datos:");
        Imprimir("¿Qué desea hacer?");
        Imprimir("1.- Imprimir Numeros.");
        Imprimir("2.- Realizar Operaciones.");
        Imprimir("3.- Salir del programa.");
        Imprimir("Opción: ");
        Leer(Opcion);
        Seleccion(Opcion)
        {
            caso 1:
                ImprimirDatos(ContNum | Extract);
                romper;
            caso 2:
                Si(ContNum>1)
                    Operaciones(ContNum | Extract);
                Si no
                    Imprimir("No es posible realizar operaciones, sólo existe un
número");
                romper;
            caso 3:
                Imprimir("Saliendo del programa...");
                exit(0);
                romper;
            Por defecto:
                Imprimir("Opción invalida, intente de nuevo");
                romper;
        }
        Imprimir("Presione Enter para continuar...");
    }
    Mientras(Opcion<>3);
}
}

```

```

ComprimirDatos(argv, ContNum | Num)
{
    Res=12;
    Desde i=0 Hasta i<ContNum; i++
    {
        sscanf(argv[i+1],ConvertirDecimal,Temp);
        Si ((ComparadorCadenas(argv[i+1],"0")==0) OR (Temp>=1 AND Temp<=15))
        {
            Temp <<= Res;
            Num |=Temp;
            Res-=4;
        }
        Si no
        {
            Imprimir("Error, los valores de los datos son incorrectos [Números
aceptados: 0 al 15]");
            exit(0);
        }
    }
}

```

```

ExtraerDatos(ContNum, Num, | Extract)
{
    Res=12;
    Temp=Num;
    Extract[0] = Temp >> Res;
    Desde i=0; Hasta i<ContNum; i++
    {
        Temp = Extract[i] << Res;
        Num = Num ^ Temp;
        Temp = Num;
        Res-=4;
        Extract[i+1]=Temp>>Res;
    }
}

```

```

ImprimirDatos(ContNum, Extract | )
{

```

```

Letra=A;
Desde i=0; Hasta i<ContNum;i++
{
    Imprimir("Número en " +Letra +“es: ”"+Extract[i]);
    Letra+=1;
}
}

```

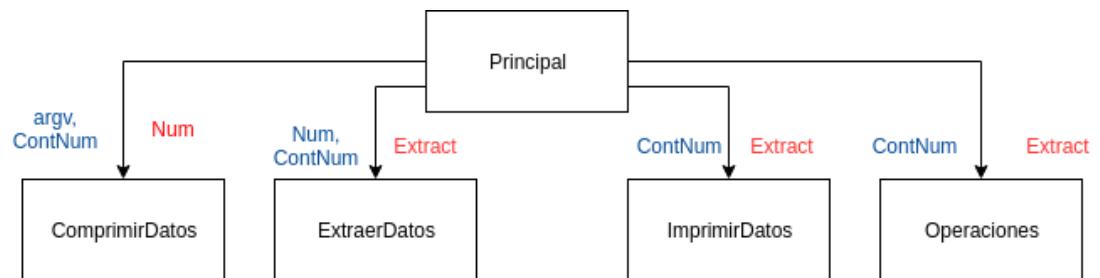
```

Operaciones(ContNum, Extract | )
{
    Resultado=Extract[0];
    Imprimir("¿Qué operación desea hacer?");
    Imprimir("Suma (+)");
    Imprimir("Resta (-)");
    Imprimir("Multiplicación (x)");
    Imprimir("División (/)");
    Imprimir("Opción: ");
    Leer(Operacion);
    Selector(Operacion)
    {
        caso '+':
            Desde i= 1; Hasta i<ContNum;i++
                Resultado+=Extract[i];
            Imprimir("El resultado es: "+Resultado);
            romper;
        caso '-':
            Desde i= 1; Hasta i<ContNum;i++
                Resultado-=Extract[i];
            Imprimir("El resultado es: "+Resultado);
            romper;
        caso 'x':
            Desde i= 1; Hasta i<ContNum;i++
                Resultado*=Extract[i];
            Imprimir("El resultado es: "+Resultado);
            romper;
        caso '/':
            Desde i= 1; Hasta i<ContNum;i++
                Resultado/=Extract[i];
            Imprimir("El resultado es: "+Resultado);
            romper;
    }
}

```

```
Por defecto:  
  Imprimir("Opción inválida...");  
  romper;  
}  
}
```

Diagrama de IPO



Variables de Entrada Variables de Salida
