



Centro de Enseñanza Técnica Industrial

Plantel Colomos

Nombre Alumno: José Rafael Ruiz Gudiño

Registro: 20110374

Arquitectura de Computadoras

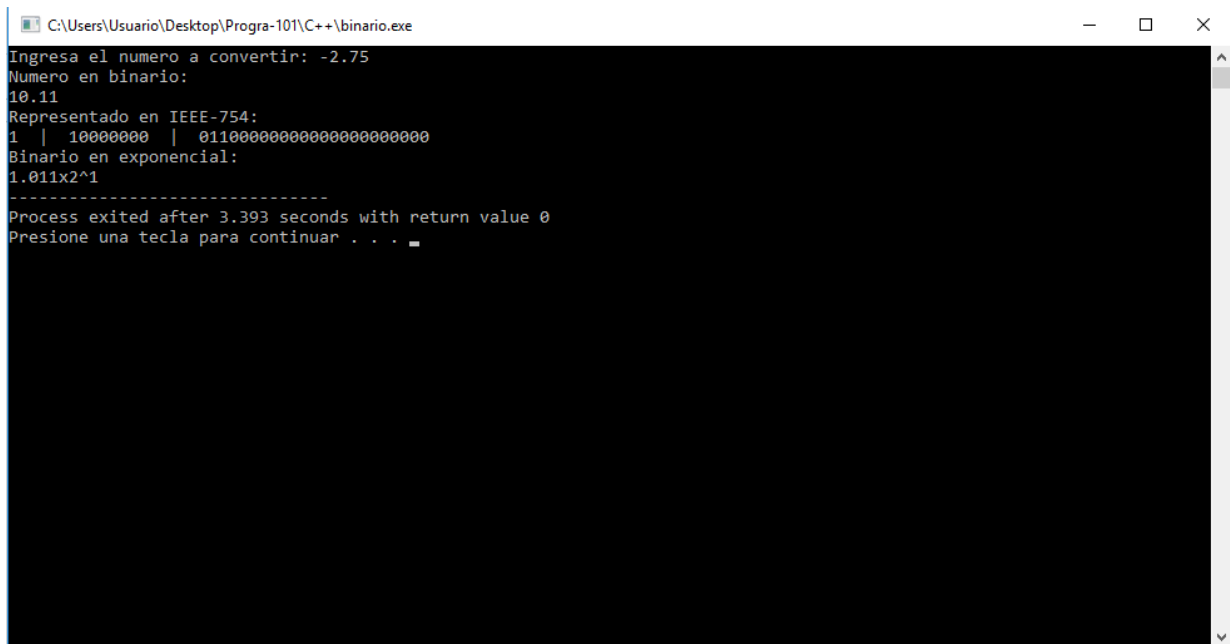
Práctica 1_ IEEE754 Punto Flotante: Estandarización.

2°P

T/M

23/02/2021

Captura del Ejecutable del Programa



```
C:\Users\Usuario\Desktop\Progra-101\C++\binario.exe
Ingresa el numero a convertir: -2.75
Numero en binario:
10.11
Representado en IEEE-754:
1 | 1000000 | 01100000000000000000
Binario en exponencial:
1.011x2^1
-----
Process exited after 3.393 seconds with return value 0
Presione una tecla para continuar . . .
```

Código del programa

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>

int main(){
    int
    binario[100],binariof[100],exponen[100],i,in,j,x=0,y,e,b=0,numint,signo,exp,mantisa[23];
    int temp,cont=0,c,sum;//variable temporal y contador
    float numf,aux;
    double dcimal;
    printf("Ingresa el numero a convertir: "); scanf("%f",&numf);
    aux=numf;
    if(numf<0) numf*=-1; //por si es negativo
    numint= numf;
    dcimal = (numf)-numint;
    i=0;

    //CONVERSION DEL SISTEMA DECIMAL A BINARIO
    if(numint==0) printf("%d",numint);
    while(dcimal!=0){ //PARTE FRACCIONARIA
        dcimal*=2;
        if(dcimal<1){
```

```

        binariof[x]= 0;
        x++;
    } else{
        binariof[x]=1;
        dcimal-=1;
        x++;
    }

}

c=x;
//PARTE ENTERA

while(numint>0){ // 5>0// 2>0// 1>0//
    binario[i]= numint%2;//bin[0] = 1 | bin[1]= 0 | bin[2]= 1
    numint/=2; //num=2 // num=1 | num=0
    ++i; //i=1 | i=2 | i=3
}
e=i;
in=i;
//impresion del numero binario

printf("Numero en binario: \n");
if(aux<0) printf("");
for(j=i-1;j>=0;--j){

    printf("%d",binario[j]);
}
printf(".");
for(y=0;y<x;y++){

    printf("%d",binariof[y]);
}
// Parte IEEE-754

//signo
if(aux>=0){
    signo=0;
} else signo=1;
//exponente

e-=1;
exp= e + 127;

while(exp>0){
    // 5>0// 2>0// 1>0//
    exponen[b]= exp%2;//bin[0] = 1 | bin[1]= 0 | bin[2]= 1
    exp/=2; //num=2 // num=1 | num=0
    ++b; //i=1 | i=2 | i=3
}

```

```

//mantisa
for(j=0;j<i-1;j++){//copia la parte entera del numero

    mantisa[j]= binario[j];

}
if(i==0){i=0;} else i--;
while(cont < i/2) //invierte el orden
{
    temp = mantisa[cont];
    mantisa[cont] = mantisa[i - 1 - cont];
    mantisa[i - 1 - cont] = temp;
    cont++;
}

sum = c+i;
for(j=i;j<sum;++j){//copia la parte fraccionaria
    mantisa[j]=binariof[j-i];
}
for(j=sum;j<23;++j){ //rellena los ceros
    mantisa[j]=0;
}
//impresion

printf("\nRepresentado en IEEE-754:\n");
printf("%d | ",signo);//signo
for(j=8-1;j>=0;--j){ //exponente

    printf("%d",exponen[j]);
}
printf(" | ");
for(j=0;j<23;++j){ //mantisa

    printf("%d",mantisa[j]);
}
//////////////////////////////////// PASAR A MODO EXPONENCIAL

printf("\nBinario en exponencial: \n"); //NUMERO BINARIO EN EXPONENCIAL
for(j=i;j>=0;--j){
    printf("%d",binario[j]);
    if(j==i) printf(".");
}
for(y=0;y<x;y++){

    printf("%d",binariof[y]);

}printf("x2^%d",i);

return 0;

```

}

Conclusión

En esta práctica pude apreciar y aprender a cómo convertir un número en sistema decimal a sistema binario y de este mismo poder estandarizarlo a IEEE-754, me pareció muy interesante que es el que se usa internacionalmente y se implementó en 1985 para de esta manera poder representar números con coma flotante de una manera estandarizada. Pude observar que consta de 3 partes: signo, exponente y mantisa. El signo va al principio del estandarizado y se representa con un '0' si es positivo o con un '1' si es negativo, además de que es de un bit; el exponente se representa con 8 bits el cual consta del exponente del número binario más la BIAS que en este caso es 127 bit porque fue de precisión simple, al final sigue la mantisa que son los 23 bits más significativos del número binario después del punto. Personalmente tuve mis dificultades para realizar el programa, en mi caso fue en el acomodo de las posiciones en los arreglos pero gracias a esta práctica puedo realizar conversiones a binario y a estándar IEEE-754 precisión simple más fácilmente.