Para ver el funcionamiento de número (32 bit para int o float) o texto numérico (secuencia de caracteres==dígitos) al trabajar con ficheros.

Empecemos por escribir la palabra “**Ion.**” Ahora leerlo como valor numérico, ¿Qué muestra? ¿778989385?

Hacer lo mismo con la palabra “**.noI**” ¿Qué muestra? ¿1232039470?

Se hace el comentario de arriba para entender el tratamiento de los bits según se lea como texto o número binario.

Poner la palabra “**Ion.**” En binario y calcular su valor numérico, lo mismo con la palabra “**.noI**”. ¿Que podéis decir sobre dichos valores?. Solo deciros que Intel trabaja con los números en **litte-endian**.

|  |  |  |  |
| --- | --- | --- | --- |
| Karakterea | Ascii (dezimalean) | Ascii (Hexadezimalean) | Ascii(Binarioan) |
| I | 73 | 49 | 1001001 |
| o | 111 | 6F | 1101111 |
| n | 110 | 6E | 1101110 |
| . | 46 | 2E | 101110 |

Para ver lo comentado tenéis el programa **ion\_idatzi.c**

Como os pide de introducir un número, idea de hacer fichero de números, darle como extensión **.dat** (esto no es obligatorio, solo tema conceptual).

Luego está el **ion\_irakurri.c** este se utiliza para leer los dos ficheros que ha hecho el programa anterior.

En este programa mirar e interpretar la declaración del vector de dos enteros.

int basura[2] = {1431519554,16722};

la declaración con su inicialización si se coge como estring es o equivale a:

BASURA == {1431519554,16722} BASU, RA’\0’’\0’

El programa está hecho para que funcione con una extensión del tipo “.dat” como entrada.

Justificar los valores que se visualizan en función del fichero y del tipo de lectura que se ha realizado en cada uno de los apartados.

Justificar las 4 líneas con salida de datos (todos los valores).

Programan datuak sartzen ibili naiz eta ez dit zure balore berdina ematen, behin baino gehiagotan ahalegindu naiz binario, hexadecimal eta dezimalean sartzen, baina ez dut ulertzen programak zer egiten duen.