

## Práctica No. 2

### Introducción al intérprete 80X86 sobre la plataforma T-Juino

**Objetivo:** El alumno se familiarizará con el intérprete 80X86 (modo 16bits) sobre la plataforma T-Juino. Esto con el fin de desarrollar programas en lenguaje C para dicha plataforma.

**Material:**

- Computadora Personal (PC)
- Programa Editor de texto (ASCII), TASM y TLINK
- Tarjeta T-Juino (con intérprete 80x86)
- Manejador FTDI instalado

**Equipo:**

- Computadora Personal
- Programa emulador de terminal

**Teoría:**

- Lenguaje C, Lenguaje Ensamblador y combinación de éstos en sistemas embebidos.

#### Desarrollo:

1) Basándose en listado 1 crear in archivo texto llamado **pra2.c**

##### Listado 1

```
void putchar( char dato );
void puts( char *str );
char getch( void );
void printdec(unsigned char dato );

char msg[]="Hola UABC\r\n";

unsigned char i=0;

int main ( void ){

    while(1){

        printdec(i++);
        puts( msg );
        getch();

    }
    return 0;

}

void puts ( char *str )
{
    while( *str )
        putchar( *str++ );
}
```

```
void putchar ( char dato )
{
    asm mov dl,dato
    asm mov ah,2
    asm int 21h
}

char getch( void )
{
    char dato;

    asm mov ah,8
    asm int 21h
    asm mov dato,al

    return dato;
}

void printdec ( unsigned char dato )
{
    putchar( dato/100 + 0x30 );
    dato%=100;
    putchar( dato/10 + 0x30 );
    putchar( dato%10 + 0x30 );
}
```

2) Descargue los programas **tcc.exe**, **tasm.exe**, **tlink.exe** y **mkbintj.exe** de moodle (curso Microprocesadores y Microcontroladores en <http://uabc-lan.net>) y deposítelos en un directorio exclusivo de trabajo (por ejemplo. C:\uPuC\Prac2).

3) Compile el programa **prac2.c** mediante la línea de comando:

```
C:\uPuC\Prac2>tcc -c pra2.c
```

Esto generará el archivo **prac2.obj**

4) Encadena el archivo **obj** para generar programa **.exe** mediante la línea de comando:

```
C:\uPuC\Prac2>tlink initj prac2, prac2
```

Esto generará el archivo ejecutable **prac2.exe**

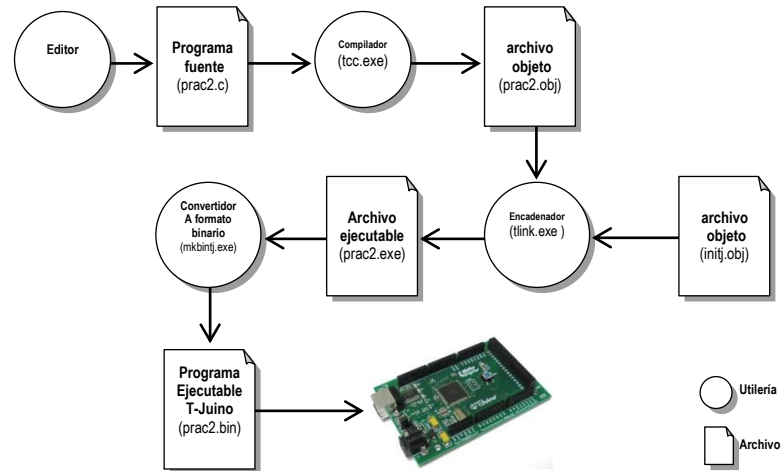


Figura 1. Proceso de generación de archivo ejecutable para T-Juino.

- 5) Convertir archivo **prac2.exe** a formato binario para la placa T-Juino haciendo uso de la herramienta **mkbintj.exe**

C:\uPuc\Prac2>**mkbintj** prac2

Esto generará el archivo **prac2.bin** el cual puede cargarse y ejecutarse en T-Juino.

- 6) Verifique el funcionamiento del programa cargándolo y ejecutándolo en T-Juino.

- **Implementar las siguientes funciones:**

- a) char **getchar**(void) : Función que retorna el carácter capturado del teclado, en ASM fuera de línea.
- b) void **putchar**(char data) : Función que manda un carácter a la pantalla, en ASM fuera de línea.
- c) void **gets**(char \*str) : Función que retorna una cadena haciendo uso de *getchar*, la cadena se retorna en el apuntador *str*.
- d) void **puts**(char \*str) : Función que imprime una cadena mediante *putchar*.
- e) void **itoa**(unsigned int number, char\* str, unsigned char base) : Función que convierte un número de 16 bits a su representación alfanumérica en la base dada, y la retorna en el apuntador *str*.
- f) unsigned int **atoi**(char \*str) : Función que convierte una cadena numérica (de base 10) y retorna su valor numérico en 16 bits.

```
typedef struct bullsAndCowsTag{
    unsigned char bulls:4;
    unsigned char cows:4;
} bullsAndCowsType;
```

- g) bullsAndCowsType **evalBullsAndCows**(char \*secrete, char \*guess) : Función que evalúa las cadenas entrantes en base a las reglas del juego "[Bulls and Cows](#)".

- Con estas funciones, portar la Práctica 1 a lenguaje C.
- Realizar los ejercicios que están relacionados con la Práctica.

### Conclusiones y Comentarios.

### Bibliografía