

## Laboratorio Sesión 06: Implementar una rutina

### Objetivo

El objetivo de esta sesión es implementar una rutina en ensamblador para realizar una actividad muy común: imprimir un número natural por pantalla.

### Enunciado de la Sesión

Diseña e Implementa la rutina en ensamblador "PrintBase" que imprime por pantalla un número natural en cualquier base entre 2 y 10. Para ello, sólo puedes utilizar la rutina "PrintChar" que permite imprimir por pantalla un carácter individual.

El programa principal para invocar la rutina "PrintBase" lo encontraréis en el fichero "Main.c".

Para implementar esta aplicación necesitarás alguna de las siguientes rutinas (las puedes encontrar en el fichero "Rutinas.c"):

```
void PrintChar(char c)
{ printf("%c", c);
}

char ConvDec(int i) //i es un número entre 0 y 9
{ return i + '0';
}

char ConvGen(int i) //i es un número entre 0 y 19
{ if (i <= 9)
    return i + '0';
    else
    return i + 'A' - 10;
}

char ConvTab(int i) //i es un número entre 0 y 19
{ char T[20] = "0123456789ABCDEFGHIJ";
  return T[i];
}
```

### Trabajo previo a la sesión

- 1) Escribe los comandos exactos necesarios para compilar y ejecutar esta práctica.
- 2) Traduce a ensamblador del IA32 la rutina "ConvTab".
- 3) Traduce a ensamblador del IA32 la siguiente sentencia:

```
PrintChar(x);
```

sabiendo que x es un caracter almacenado en el registro %ah (en %ah, ino es un error!).

- 4) Explica qué hace y cómo lo hace cada una de las funciones del fichero "Rutinas.c".
- 5) ¿Qué rutina es más eficiente, ConvGen o ConvTab? Razona la respuesta.

### Resultados de la sesión

Antes de acabar la sesión tienes que haber hecho, por lo menos, lo siguiente:

- Diseñar, implementar, y conseguir que funcione la rutina para imprimir por pantalla un número en cualquier base entera entre 2 y 9.
- Modificar la rutina para que pueda imprimir numeros en bases comprendidas entre 2 y 20. Para ello, los dígitos mayores que 9 se representarán con letras mayusculas consecutivas a partir de la A, como hacemos en base 16.
- Si hay tiempo, modificar la rutina para que compruebe que la base está entre 2 y 20 e imprima un '\*' en caso contrario.

## Notas

- No implementes la rutina de forma recursiva. Busca un algoritmo iterativo.
- La instrucción de división de naturales en IA32 tiene la siguiente sintaxis:

```
divl op    #op ha de ser un registro o una posición de memoria.
```

El resultado de esta instrucción es el siguiente:

```
%eax <- (%edx:%eax) / op    # cociente de la división entera  
%edx <- (%edx:%eax) % op    # resto de la división entera
```

Esta instrucción utiliza como operandos fuente los registros `%edx` y `%eax` concatenados como si fueran un solo registro de 64 bits.

- Para que una secuencia de caracteres aparezca por pantalla hay que imprimir un salto de línea `'\n'` al final de la secuencia.
- Como ayuda, dispones de un ejecutable llamado TEST que acepta 2 parámetros (en decimal): el número y la base, e imprime el resultado por pantalla.  
`./TEST numero base`
- Los ficheros para esta sesión los puedes encontrar en la página web de la asignatura: `"Programas.Sesion06.tar.gz"`.