



Universidad Autónoma de  
Baja California

Facultad de Ciencias  
Químicas e Ingeniería

Organización de las Computadoras y Lenguaje Ensamblador

# **Programas en el Lenguaje Ensamblador del Procesador 8088 (strDec2num)**

**Integrantes:** Saul Ivan Flores – 01217102

**Fecha:** Abril 30, 2018

## Teoría

**itoa:** La función itoa() convierte un entero a una cadena de caracteres. La cadena de caracteres es colocado en un buffer pasado, que tiene estar lo suficientemente grande para almacenar la salida. Las bases aceptadas pueden ser Octal, Decimal, o Hexadecimal. Cuando la base es decimal esta función produce el mismo resultado que :

```
(void) sprintf(buffer, "%d", n);
```

Cuando la base es octal, se modifica el entero 'n' a una constante octal sin signo y en el caso de ser Hexadecimal el entero se convierte en un hexadecimal constante.

**atoi:** La función atoi() convierte una cadena de caracteres a un tipo entero, como parámetros recibe una constante char con los valores de un número integral. La función regresa el número convertido a entero en el caso de que no se genera una conversión válida se regresara el valor cero.

## Desarrollo

Tomando el método atoi como base implementamos un método que replica la funcionalidad de atoi construida en ensamblador y llamada **strDec2num**. Declarando tres variables cadena(Contiene la cadena de caracteres que será convertida), tamaño(Valor decimal que contiene el tamaño de la cadena), y base(Valor decimal se utiliza para determinar la base en el que se despliega el valor). Se asigna la variable tamaño a SI, utilizando DI como contador de la cantidad de iteraciones que se ejecutan en el programa y CX se le asigna un valor cuatro utilizado para corrimientos hacia la izquierda.

Durante la primera iteración se asigna el contenido de BX a AX, se resta el valor 30h a AX se compara SI a cero para determinar el fin de la cadena. En el caso que la cadena sea mayor a un carácter, entra a un loop donde se incrementa DI, se recorre AX cuatro bits y se almacena en dx, se limpia AX asignándole un cero y se continúa a almacenar en AX el contenido en BX+DI, se le resta un treinta hexadecimal, y se suma al valor almacenado en DX y la respuesta se de la operación es almacenada en AX. Se compara DI con SI mientras no sean iguales el loop se mantiene hasta terminar.

Al terminar de convertir la cadena de caracteres el valor se encuentra almacenado en ax y se manda a llamar el método **printNumBase** para desplegar el contenido en AX con la base deseada.

## Conclusión

El estar traduciendo métodos implementados en el lenguaje c a ensamblador x86 constantemente refuerza los temas tocados en clase al igual que permite ensamblar programas con más velocidad tomando en cuenta que la práctica vuelve a uno más competente. El estar utilizando métodos previamente utilizados como **printNumBase** para complementar el ejercicio impulsa el diseño de mejores algoritmos y buenas practicas de programacion para evitar problemas en soluciones futuras.