

## Laboratorio Sesión 05: Programando en ensamblador

### Objetivo

El objetivo de esta sesión es programar en el Laboratorio una pequeña aplicación escrita en C y en ensamblador del IA32, de forma que enlacéis código escrito en C con código escrito en ensamblador.

### Enunciado de la Práctica

Disponemos de una aplicación escrita en C que realiza la búsqueda de un elemento en un vector de structs. A partir de los campos del struct se calcula un número que debe coincidir con el elemento que se busca. La aplicación está dividida en tres ficheros: "MainBusqueda.c", "BuscarElemento.c" y "Buscar.c".

Para comprobar el funcionamiento de esta aplicación, podéis hacer lo siguiente:

```
$> gcc MainBusqueda.c BuscarElemento.c Buscar.c -o BUSCADOR
$> ./BUSCADOR 32
```

```
Encontrado elemento 32 en la posición 14 con valor z
```

```
$> ./BUSCADOR 44
```

```
NO encontrado elemento 44
```

La práctica consiste en traducir a ensamblador las rutinas "BuscarElemento" y "Buscar" que se encuentran definidas en los ficheros "BuscarElemento.c" y "Buscar.c".

Nota: Los ficheros para esta sesión los podéis encontrar en la página web de la asignatura: "Programas.Sesion05.tar.gz".

### Algo de Linux: El path

Uno de los mecanismos más útiles del intérprete de comandos son las variables de entorno. Para consultar las variables de entorno se puede utilizar el comando `printenv`. Algunas de estas variables permiten modificar el comportamiento del propio intérprete.

Una de las variables de entorno más importantes es la variable `PATH`. Se puede consultar el contenido de esa variable utilizando el siguiente comando:

```
$> printenv PATH
```

La variable `PATH` contiene una lista de directorios, separados por el símbolo ":". Cada vez que se invoca un comando (p.e. `ls`) o un programa cualquiera (p.e. `gcc`), el intérprete de comandos buscará el ejecutable en esta lista de directorios en el orden indicado. Cuando se encuentra el ejecutable, se detiene la búsqueda.

Por ejemplo, cuando ejecutáis un programa tenéis que escribir:

```
$> ./BUSCADOR
```

Es necesario indicar explícitamente en qué directorio se encuentra el ejecutable. En este caso, el directorio ".", es decir, el directorio de trabajo. Se puede modificar el `PATH` para que incluya el directorio de trabajo. El comando depende del shell que se use, por ejemplo:

Para bash shell:

```
$> export PATH=$PATH:/usr/sbin/:/usr/local/bin
```

Para tcsh or csh:

```
$> set PATH = ($PATH /usr/sbin /usr/local/bin)
```

Este comando modifica la variable `PATH`, y le añade el directorio de trabajo "." a su contenido actual. Ahora se puede ejecutar el mismo programa haciendo simplemente:

```
$> BUSCADOR
```

## Trabajo previo de la sesión

Como trabajo previo de la sesión, tenéis que realizar lo siguiente:

- Explicad qué significan y para qué sirven los parámetros "argc" y "argv" del "main".
- ¿Explicad para qué sirve la función "atoi"?
- Dibujad el struct S1.
- Dibujad el bloque de activación de las rutinas "BuscarElemento" y "Buscar".
- Dibujad el bloque de activación de la rutina "main".
- Traducid a ensamblador del IA32 la sentencia:

```
trobat = BuscarElemento(&low, &high, &mid, x, v);
```

que se encuentra dentro de la rutina "Buscar".

## Tareas a realizar en la sesión

- 1) Compilad y probad la aplicación original escrita en C.
- 2) Traducid a ensamblador del IA32 la rutina "Buscar". Utilizad como base el fichero "Busca.s". Probadla con los ficheros "MainBusqueda.c" y "BuscarElemento.c".
- 3) Traducid a ensamblador del IA32 la rutina "BuscarElemento". Utilizad como base el fichero "BuscaElem.s". Probadla con el fichero "MainBusqueda.c" y con la rutina en ensamblador del anterior apartado.