

2.2. Control de procesos en Linux



Seguro que más de una vez
hemos necesitado dentro de
un programa ejecutar otro
programa que realice alguna
tarea concreta.



Linux ofrece varias funciones para realizar esto: *system()*, *Fork()* y *execl()*.



La función *system()* se encuentra en la librería estándar *stdlib.h* por lo que funciona en cualquier sistema operativo que tenga un compilador de C/C++ como por ejemplo Linux, Windows, etc.



El formato es el siguiente:

```
int system(const char *cadena)
```

La función recibe como parámetro una cadena de caracteres que indica el comando que se desea procesar.

Dicha instrucción es pasada al intérprete de comandos del ambiente en el que se esté trabajando y se ejecuta.

Devuelve el valor -1 si ocurre un error y el estado devuelto por el comando en caso contrario.



La función *execl()*

```
#include <unistd.h>
```

```
int execl(const char *fichero, const char  
*argo, ..., char *argn, (char *) NULL);
```


La función recibe el nombre del fichero que se va a ejecutar y luego los argumentos terminando con un puntero nulo.

Devuelve -1 si ocurre algún error y en la variable global *errno* se pondrá el código de error adecuado.

Para ejecutar el comando */bin/ls -/* escribimos:

```
execl("/bin/ls", "ls", "-1", (char *)NULL);
```