## Problema

*a) Implementar la función `GenerarClave` con los siguientes requisitos:*

- *La función recibe como único parámetro el nombre de un fichero o directorio.*
- *En caso de haber recibido un directorio o de existir cualquier error de ejecución, la función devuelve -1*
- *Si el parámetro recibido es un fichero, la función deberá devolver un número entero calculado de la siguiente manera: tamaño en bytes del fichero más la suma de todos los bytes del fichero.*

*b) Se desea realizar el programa `InsertarClave` con los siguientes requisitos:*

- *El programa recibe como argumentos el nombre de un directorio y el nombre de un fichero. Se debe suponer que tanto el directorio como el fichero existen y son válidos.*
- *Calcula la clave del fichero indicado como segundo argumento, usando para su cálculo la función `GenerarClave` realizada en el anterior apartado.*
- *Una vez calculada la clave, crea en el directorio especificado como primer argumento un enlace simbólico con nombre el valor de la clave calculada, apuntando al fichero recibido como entrada.*

*Por ejemplo, en la siguiente invocación del programa:*
*InsertarClave /var/claves/ /tmp/fichDatos.txt*
*si la clave calculada es 67534, se debería crear la siguiente entrada en el directorio `/var/claves/`:*
*lrwxr-xr-x 1 luis luis 13 Jun 2 16:36 67534 -> /tmp/fichDatos.txt*

*c) Responda razonadamente. ¿Habría alguna diferencia si en vez de enlaces simbólicos el programa `InsertarClave` creara enlaces físicos?, ¿qué ventajas e inconvenientes podría suponer esta situación?*

## Solución

**a)** El programa propuesto es el siguiente.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>

int generar_clave(char *nombre)
{
      struct stat fd_info;
      int clave=0;
      char c;
      int fd;
      int s;

      /* Obtenemos la información del fichero */
      if (stat(nombre, &fd_info)<0)
      {
            perror("Error en la llamada stat");
            return -1;
      }
```

```
        /* Comprobamos si es directorio */
        if (S_ISDIR(buffer.st_mode))
        {
              clave = -1;
        }
        else
        {
              /* Obtenemos tamaño del fichero */
              clave = (int) buffer.st_size;
              fd = open(nombre,O_RDONLY);
              if (fd==-1)
              {
                    perror("Error al abrir el fichero");
                    return -1;
              }

              /* Cálculo de la suma de los bytes del fichero */

              s = read(fd, &c, 1);
              while (s>0)
              {
                    clave += (int) c;
                    s = read(fd, &c,1);
              }
        }
        close(fd);
        return clave;
}
```

**b)** El programa utilizará la función anterior para llevar a cabo las operaciones descritas.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
        int fd;
        int clave;
        char nuevoPath[200];
        char origPath[200];
        char claveStr[100];

        /* Calculamos la clave del fichero */

        clave = generar_clave(argv[2]);
        if (clave == -1)
        {
```

```
            perror("Error al generar clave");
            exit(1);
        }

        /* Ruta original */
        strcpy(origPath,"");
        if (argv[2][0]!='/')
        {
            getcwd(origPath, 200);
            strcat(origPath, "/");
        }
        /* Ahora es ruta absoluta */
        strcat(origPath, argv[2]);

        /* Nueva ruta */
        sprintf(claveStr,"/%d", clave);
        strcpy(nuevoPath,"");
        strcat(nuevoPath, argv[1]);
        strcat(nuevoPath, claveStr);

        /* Creación del enlace simbólico */
        symlink(origPath,nuevoPath);

        exit(0);
}
```

**c)** La primera diferencia es que no se pueden crear enlaces físicos entre diferentes volúmenes. La segunda diferencia vendría a la hora del borrado de los ficheros. Dependiendo de la política que nos interese podemos utilizar cada uno de los enlaces. Si usamos enlaces simbólicos podemos tener enlaces erróneos en nuestro directorio. Si usamos enlaces físicos podemos estar generando problemas con el borrado de los ficheros, ya que estamos aumentando el contador del número de enlaces en los mismos. Desde el punto de vista de la programación, sólo necesitamos cambiar la llamada symlink por link.