

## Práctica 4: Mapear un fichero en Memoria MMAP

David Morales Sáez

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>DISEÑO</b>	<b>3</b>
<b>CONCLUSIÓN</b>	<b>3</b>
<b>ANEXO</b>	<b>4</b>

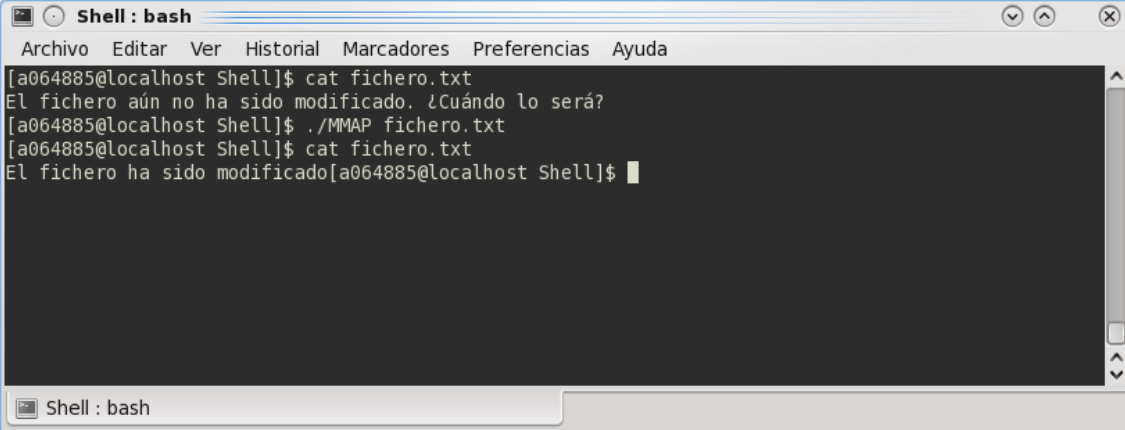
## Introducción

Para esta práctica, se ha propuesto utilizar las funciones de mapeo de ficheros en memoria. En mi caso, he decidido abrir un fichero ya creado y sustituirlo por un texto ya predefinido.

## Diseño

El código de esta práctica es bastante simple y esclarecedor. En primer lugar, abrimos el fichero. Seguidamente lo mapeamos en memoria. Después modificamos la memoria, sobrescribiendo el texto por aquel que nosotros queramos. Finalmente, cerramos el fichero y liberamos la memoria.

Para ver su funcionamiento, aquí tenemos un ejemplo algo esclarecedor:



```
Shell : bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
[a064885@localhost Shell]$ cat fichero.txt
El fichero aún no ha sido modificado. ¿Cuándo lo será?
[a064885@localhost Shell]$ ./MMAP fichero.txt
[a064885@localhost Shell]$ cat fichero.txt
El fichero ha sido modificado[a064885@localhost Shell]$
```

## Conclusión

El uso de la memoria a la hora de trabajar con ficheros puede llegar a ser más interesante que el sistema tradicional de lectura y escritura, ya que ahorramos una gestión de trabajo que puede llevar bastantes fallos.

## Anexo

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/mman.h>
#include <sys/stat.h>

void main (int argc, char *argv[])
{
    int fd, i;
    struct stat st;
    char *addr;
    int PageSize;

    if ( (PageSize = sysconf(_SC_PAGE_SIZE)) < 0)
    {
        perror("sysconf() Error=");
        exit(-1);
    }

    // Comprobamos tener dos argumentos
    if(argc != 2)
    {
        fprintf(stderr, "Usage: %s name_of_file\n", argv[0]);
        exit(1);
    }
    // Obtenemos el tamaño del fichero
    if(stat(argv[1], &st) == -1)
    {
        perror("stat");
        exit(2);
    }
    // Abrimos el fichero
    fd = open(argv[1], O_RDWR);
    if(fd == -1)
    {
        perror("open");
        exit(3);
    }
    // Proyectamos el fichero en memoria
    addr = (char *)mmap(NULL, st.st_size, PROT_WRITE,
MAP_SHARED, fd, (off_t) 0);
    if(addr == MAP_FAILED)
    {
        perror("mmap");
        close(fd);
        exit(4);
    }
    const char* texto = "El fichero ha sido modificado";
    for(i=0; i<st.st_size; i++)
    {
        addr[i] = '\0';
    }
}
```

```
    }  
    strcpy(addr, texto);  
    // Cerramos el fichero  
    close(fd);  
    // Liberamos el fichero volcado en memoria  
    if(munmap(addr, st.st_size) == -1)  
    {  
        perror("munmap");  
        close(fd);  
        exit(5);  
    }  
    exit(0);  
}
```