



Ejercicio

Escribir un programa, similar a la orden *cp*, que utilice para su implementación la llamada al sistema *mmap()* y una función de C que nos permite copiar memoria, como por ejemplo *memcpy()*. Para conocer el tamaño del archivo origen podemos utilizar *stat()* y para establecer el tamaño del archivo destino se puede usar *ftruncate()*.

Solución

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

int main(int argc, char *argv[]) {
    struct stat attr;
    int fd1;
    int fd2;
    int tamaño;
    char *mem1,*mem2;

    if ((fd1=open(argv[1],O_RDONLY, S_IRWXU))<0) {
        perror("\nError en open");
        exit(-1);
    }

    if ((fd2=open(argv[2],O_CREAT|O_RDWR,S_IRWXU))<0) {
        perror("\nError en open");
        exit(-1);
    }
    if (stat(argv[1],&atributos) < 0) {
        perror("\nError en lstat");
    }
    tamaño=attr.st_size;

    printf ("Ajustando el tamaño archivo %s a %d \n",argv[2],tama);
    ftruncate(fd2, tama);
```



```
mem1 = (char *)mmap(0, tamaño, PROT_READ, MAP_SHARED, fd1, 0);
if (memoria1 == NULL) {
    perror("Fallo en mapeo");
    exit(-1);
}

mem2 = (char *)mmap(0, tamaño, PROT_WRITE, MAP_SHARED, fd2, 0);
if (mem2 == NULL) {
    perror("Fallo en mapeo");
    exit(-1);
}

memcpy(mem2, mem1, tamaño);

if (munmap (mem1, tamaño) == -1) {
    perror("Error al cerrar el mapeo\n");
    exit(-1);
}
if (munmap (mem2, tamaño) == -1) {
    perror("Error al cerrar el mapeo\n");
    exit(-1);
}
return 0;
}
```