

Esercitazione 1

I/O non bufferizzato-1

esercizio 1

- ▶▶ Scrivere un programma in C per rispondere alle seguenti domande:
 - ▶ Supponiamo di aprire un file in modalita' R-W, con il flag O_APPEND attivato.
 - E' possibile leggere in un punto qualsiasi del file utilizzando il comando lseek?
 - E' possibile usare lseek per modificare parte del file?
 - Che succede se usiamo pwrite?

soluzione 1(1)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#define FILE_MODE (S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH)
#define FILE_FLAGS (O_RDWR | O_APPEND | O_CREAT | O_TRUNC)
int main(void)
{
    int fd, i, j;
    off_t offset;
    char *buff, c;
    char teil[20] = "1234567890abcdefghil";

    buff = calloc(25, sizeof(char));

    if((fd = open("FILE1", FILE_FLAGS, FILE_MODE)) <0)
        printf("\n an error occurred opening the file\n");

    for(j=1; j <=2; j++) /* write 40 chars in the file */
    {
        write(fd, teil, 20); /* alternating 10 letters and */
                            /* 10 numbers */
    }
}
```

soluzione 1(2)

```
lseek(fd, 0, SEEK_SET); /* Move to the beginning */
printf("\n Here is the content of the file:\n\n");
while(read(fd, &c, 1) != 0) /* and write the content of file */
    printf("%c", c);
printf("\n\n");

lseek(fd, 5, SEEK_SET); /* I'm trying to move to position 5 */
read(fd, buff, 10); /* and read 10 chars */
printf("\n The 10 chars read at offset 5:%s. Check it!\n", buff);

offset = lseek(fd, -10, SEEK_CUR); /*Move back five chars*/
write(fd, string_num, 5); /* and try to write 5 num to cover 5 chr */
printf("\n Moved to offset %d and wrote 5 nums\n", (int) offset);

offset = lseek(fd, 0, SEEK_CUR); /* Check new offset after write */
printf("\n After write the offset is %d\n", (int)offset);
printf("\n Now I try again with pwrite\n");
pwrite(fd, string_num, 5);

lseek(fd, 0, SEEK_SET); /*Move to beginning and write content */
printf("\n Here is the content of the file:\n\n");
while(read(fd, &c, 1) != 0)
    printf("%c", c);
printf("\n\n");
}
```

esercizio 2

1. Scrivere un programma in C che prende un input da tastiera e lo scrive nel file FILE1.
L'input da tastiera viene fatto usando il filedescriptor 7 (usare dup o dup2)
 2. Successivamente, copiare in un file FILE2,
 1. prima in ordine inverso la seconda meta' del contenuto di FILE1
 2. poi la prima meta' del contenuto di FILE1, quindi stampare il contenuto di FILE2 sul terminale usando solo read() e write() [no printf() !!!]
- Ex: EDCBAlihgfedcba123456789012345

soluzione 2(1)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#define FILE_MODE (S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH)
int main(){
    int fdin, fdout, n;
    char c, buf[90];

    if((fdin=open("FILE1",O_RDWR|O_CREAT|O_TRUNC,0644))<0)//CreaFILE1
        printf("FILE1 open error\n"); //in modalita' read-write

    if((fdout=open("FILE2",O_CREAT|O_RDWR|O_TRUNC,0644))<0)//ApreFILE2
        printf("FILE2 open error\n");//in cui copiare in modal. w-only

    dup2(STDIN_FILENO, 7); // duplico Standard_input su filedes 7
    n=read(7,buf,90); //leggo da standard input
    write(fdin,buf,n); //scrivo su FILE1
```

soluzione 2(2)

```
length_FILE1 = (int) lseek(fdin,0,SEEK_END);// lunghezza FILE1
half_way = length_FILE1 / 2;
n = (int) lseek(fdin,-1,SEEK_END); //vai alla fine di FILE1

while ( n >= half_way) { // leggi da FILE 1 all'indietro
    read(fdin, &c, 1); // e scrivi in FILE 2
    write(fdout, &c, 1);
    n= (int) lseek(fdin,-2,SEEK_CUR); // nota il -2
}

n = (int) lseek(fdin, 0, SEEK_SET); // vai all'inizio di FILE 1
while(n < half_way)
{ // leggi la prima meta' e
    read(fdin, &c, 1); // scrivila in FILE 2
    write(fdout, &c, 1);
    n = (int) lseek(fdin, 0, SEEK_CUR);
}
```

soluzione 2(3)

```
lseek(fdout,0,SEEK_SET);

while ( read(fdout, &c, 1) > 0)
    write(STDOUT_FILENO, &c, 1);

printf("\n");

exit(0);
}
```