

int dup (int fd);

0	- std in
1	- std out
2	- stderr
3	- f1
4	- f1

Следователно двата файлови дескриптора имат следното общо:

- един и същи отворен файл;
- общ указател на текуща позиция;
- еднакъв режим на отваряне на файла.

int file = open ("f1" ...) // 3 → f1
int copy = dup (file) // 4 → f1

вързва първият свободен fd (в случая 4),
на същата позиция, където е бил и първият.
и ако тедем, то те се гвият заедно.

Искаме f1 да ни бъде stdout

1) Open()

close(1)

0	- std in
1	- std out
2	- std err
...	

→

0	- std in
1	-
2	- std err
...	

1) close(1)

2) open(file)

int file = open ("f1" ...)

0	- std in
1	- f1
2	- std err
...	

Така след като направим:

write(1, buff, 8), то ще
запишем в f1 и тои ще
ни бъде stdout

2) dup()

int file = open("f1")

0 - stdin
1 - stdout
2 - stderr
3 - f1

1) open(file)
2) close(1)
3) dup(file_fd)

close(1)

0 - stdin
1 - ~~stdout~~
2 - stderr
3 - f1

→

0 - stdin
1 -
2 - stderr
3 - f1

dup(file)

0 - stdin
1 - f1
2 - stderr
3 - f1

переназначение на стандартен вход от файл

```
if ((fd = open(infile, O_RDONLY)) == -1)
    err_sys_exit("can't open file %s\n", infile);
close(0);
dup(fd);
close(fd);
```

0 - ~~stdin~~ infile
1 - stdout
2 - stderr
3 - ~~infile~~

int dup2 (int redirect_fd , int to_this)

connect this entry to this fd

* береза fd все сдвинутого на to_this
* забирая to_this fd у my gaba entry-to на redirect_fd

```
int file_desc = open("tricky.txt", O_WRONLY | O_APPEND);  
  
// here the newfd is the file descriptor of stdout (i.e. 1)  
dup2(file_desc, 1);  
close(file_desc);  
// All the printf statements will be written in the file  
// "tricky.txt"  
printf("I will be printed in the file tricky.txt\n");
```

0 - stdin
1 - ~~stdout~~ → tricky.txt
2 - stderr
3 - ~~"tricky.txt"~~
.....

→ ізо вато бере stdout на
е tricky іто printf все
перета все файла

0 - stdin
1 - tricky
2 - stderr
3 -
.....