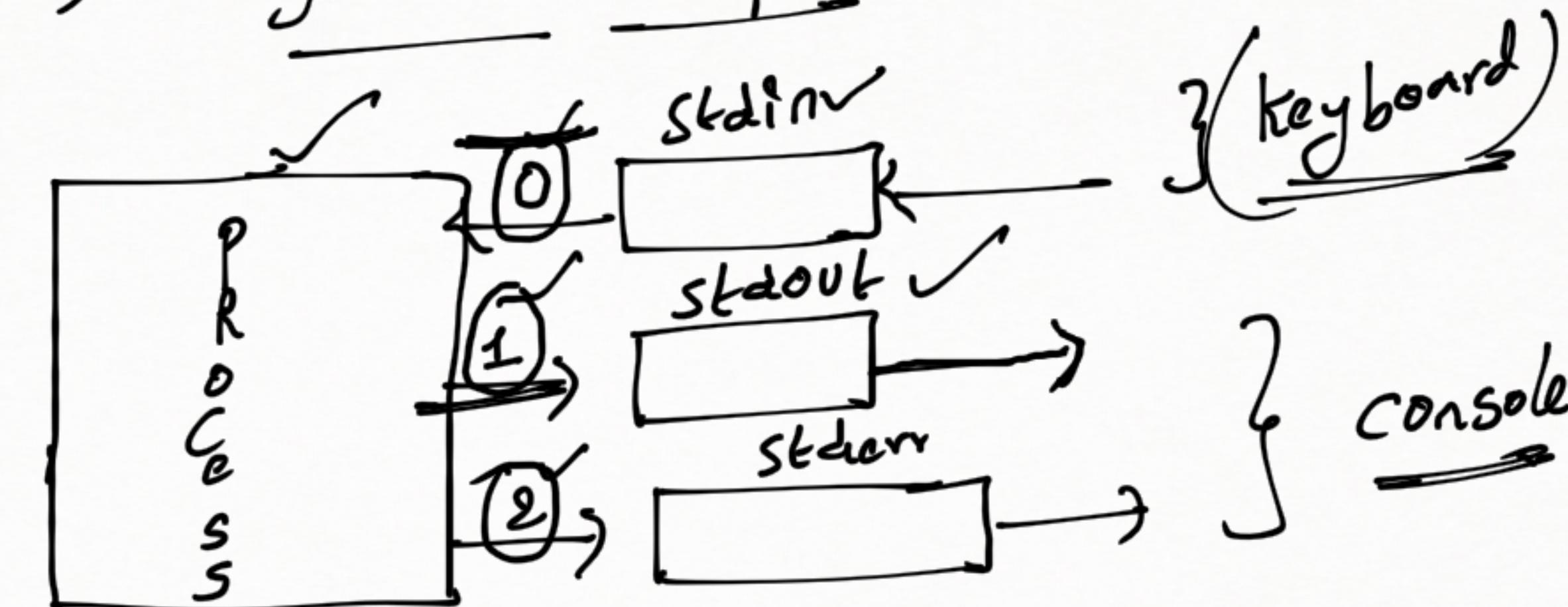


Date: 8-11-21

- When a process is created by default three buffers are attached to each process. stdin, stdout, stderr
- These buffers are represented with small non negative integers
because called as file descriptor

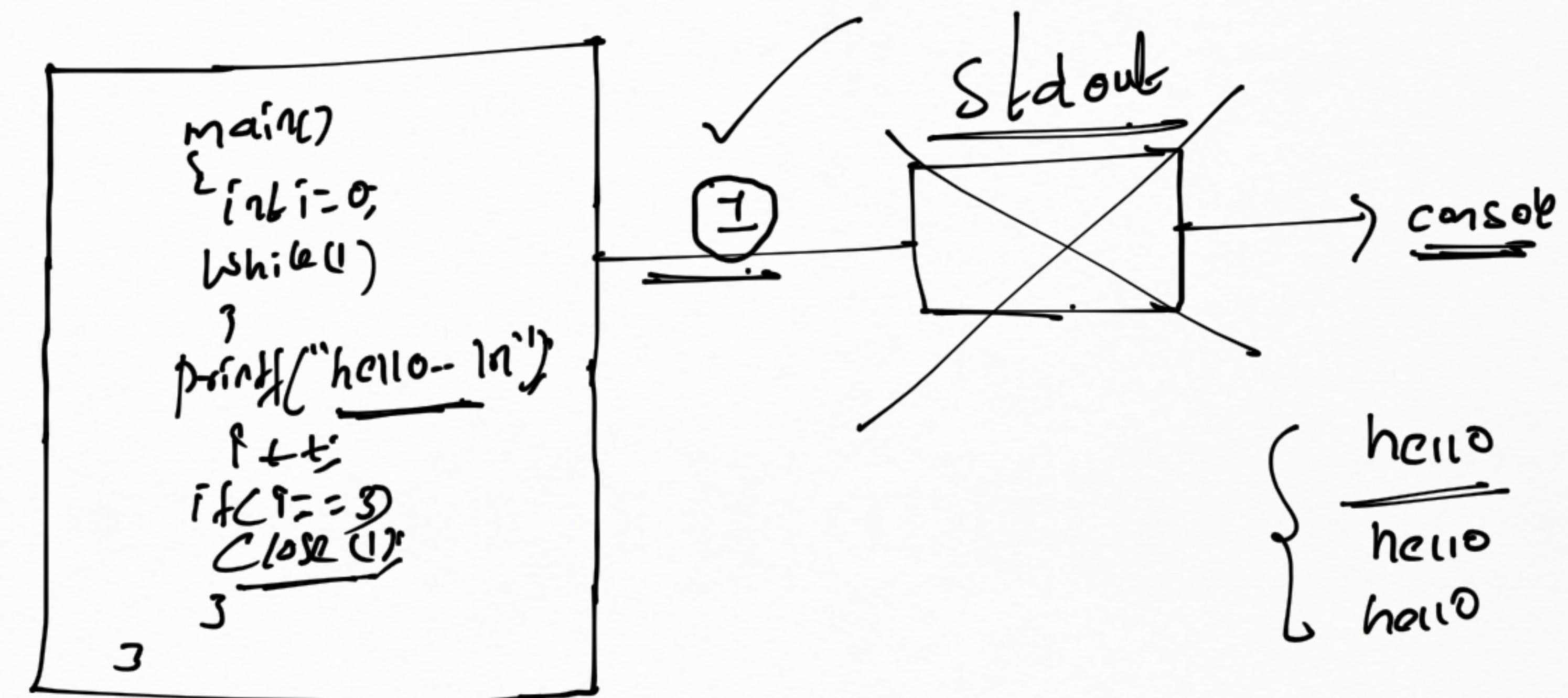
0 }
1 } \Rightarrow file descriptors
2 }



```

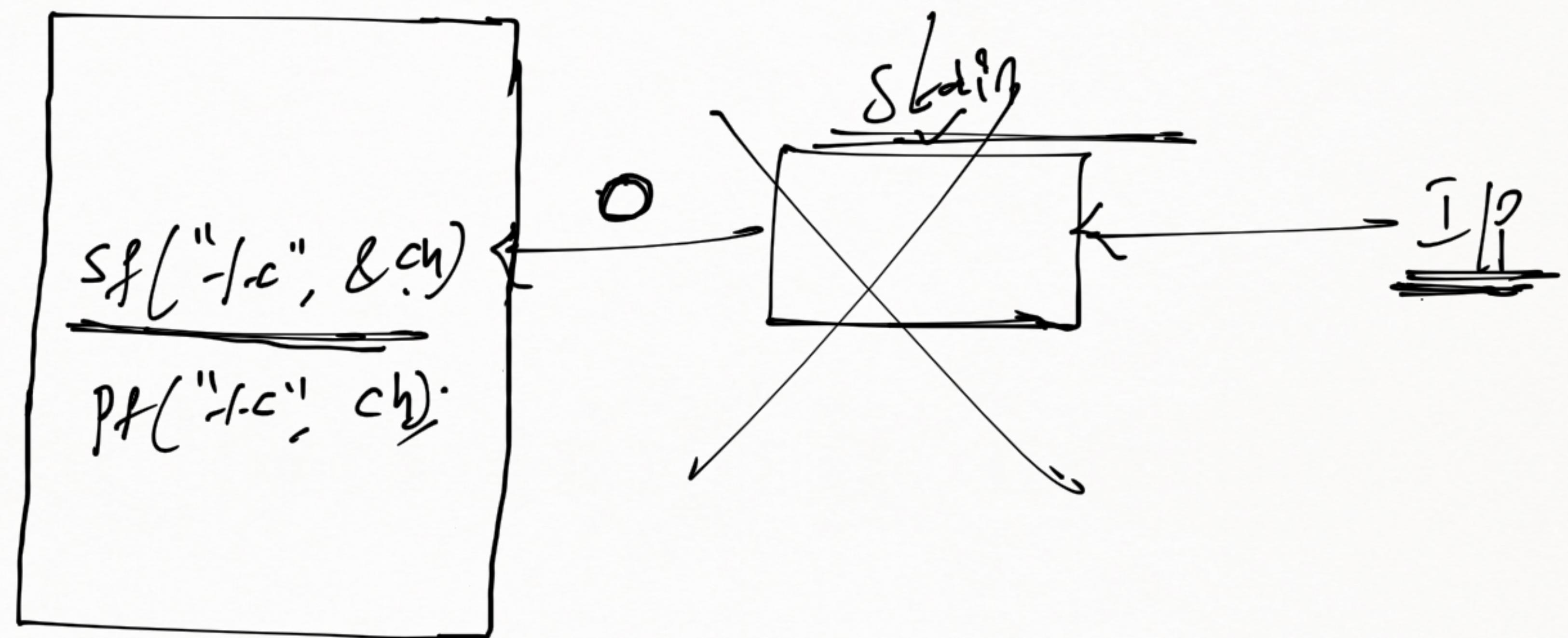
⇒ main()
{
    for i=0;
    while(1)
    {
        printf("hello--\n");
        i++;
        if(i==3)
            close(1);
    }
}

```



3

=>



→ operations on files

✓ (i) open file

✓ (ii) read / write

✓ (iii) close file

How to open a file

(Library function) → fopen ("Temp", "r")
 "r"
 "r+"
 "w"
 "w+"
 "a"
 "at"

open → system call

fopen (const char *, int flags, mode_t mode)

filename

mandatory

✓ O_RDONLY
✓ O_WRONLY

✓ O_RDWR WR

optional

O_CREAT
O_APPEND
O_TRUNC

↳ file permissions
affected to newly
created files

E₂:

open ("Temp", O_WRONLY, O_664)

↓
flag

fopen ("Temp", "W")

↓
file
create

✓ open ("Temp", O_WRONLY | O_CREAT | O_TRUNC, O_664)

↓

✓ fopen ("Temp", "W")

$\Rightarrow fopen("Temp", "r")$

\Downarrow

$open("Temp", O_RDONLY);$

$fopen("Temp", "rt");$

$open("Temp", O_RDWR);$

$\Rightarrow fopen("Temp", "w+");$

$\Rightarrow fopen("Temp", "a")$

$open("Temp", O_RDWR | O_CREAT | O_TRUNC)$

$open("Temp", O_WRONLY | O_CREAT | O_APPEND);$

$\Rightarrow fopen("Temp", "at")$

$open("Temp", O_RDWR | O_CREAT | O_APPEND);$

⇒

File create [const char *, mode_t mode]
↓
create

create is equivalent to open() with flags

equivalent to O-CREATE | O-WRONLY | O-TRUNC. ✓

✓ create ("temp", 0644) ✓

✓ open ("temp", O-CREATE | O-WRONLY | O-TRUNC, 0644)

Note: Using `cat` or `cat` `sys call` whatever the
file `cat` they are only for writing purpose
not possible to read.