write()

① #include <unistd.h>

② ssize_t write (int fd, const void *buf, size_t count)

- int fd → file descriptor
- const void *buf → data, we want to write
- size_t count → how much data we want to write (or how many characters)

→ write(), writes up to count bytes from the buffer starting at buf to the file referred to by the file descriptor fd.

Note:- for Screen (which is standard device), fd=1

Example

ⓐ

```
#include <~~stdio.h~~ unistd.h>
int main()
{
    write(1, "Hello", 5)
}
```

⇒ output ⇒ Hello

ⓑ

```
int main()
{
    write (1, "Hello", 2)
}
```

⇒ He

(E)

```
#include<unistd.h>
# include <stdio.h>
int main()
{
    int n;
    n= write(1,"Hello \n", 5);
    printf("value of n is %d \n", n)
}
```

output

```
Hello
value of n is 5
```

Read()

① # include <unistd.h>

② ssize_t read (int fd, void *buf, size_t Count)

$\underbrace{\text{size\_t Count}}_{\text{how much we have to read}}$

→ read(), attempts to read up to Count bytes from file descriptor fd into the buffer starting at buf.

Note:- for Key-board (standard-device), fd = 0

Ex ①

```
#include<unistd.h>
int main()
{
  Char b[30];
  read(0,b,10);
  write(1,b,10);
}
```

⇒ console screen

```
abcdefghij
abcdefghij
```

(input read from keyboard

Ex ②

```
int main()
{
    int n;
   Char b[30];
    n= read(0,b,10);
    Write(1,b,n);
}
```

⇒

```
abcdef
abcdef
```

Open()

open, openat, creat → open and possibly create a file.

① #include <sys/types.h>
# include <sys/stat.h>
# include <fcntl.h>

② int open (Const char * pathname, int flags);
→ If file exist then ~~read~~ use this

mode of file (reading mode, writing mode)

int open (Const char *pathname, int flags, mode_t mode);
→ If file not exist, then use this

③ The return value of open() is a file descriptor, a small, ~~no~~ nonnegative integer that is used in subsequent system calls (read(), write(), lseek, fcntl(), etc) to refer ~~to~~ to the open file.

④ flags ⇒ (access modes)
→ O_RDONLY → read-only
→ O_WRONLY → write-only
→ O_RDWR → read/write

Ex (1)

test.txt →

New file descriptor is 1

temp.c →

```
#include <unistd.h>
# include <fcntl.h>
# include <sys/types.h>
# include <sys/stat.h>
int main()
{   int n, fd;
    char buf[50];

    fd= open("test.txt", O_RDONLY);
    n= read(fd, buf, 10);
     write(1, buf, 10);
 }
```

Output → New file d

Ex② Let ~~tee~~ read some data ~~from~~ from file (already Created file) and write into another file (which is not exit).

test.txt

~~New file descriptor is 1~~

tample

```
# include <unistd.h>
#   include < fcntl.h>
# include < sys/types.h>
#   include <sys/stat.h>
int main()
{
        int  n, fd ;
        Char  buf [50] ;

        fd=open  ("test.txt", O_RDONLY)
        n=read (fd, buf, 10)
        o
        fd1= open("target", O_CREAT| O_WRONLY,
                                        0642)
                                         ↑
                                      permission
        write(fd1, buf, n);

}
```

Output target.txt

New file d

(append)

Let assume, if we want to add data into an existing file.

so,

```
fd1 = open("target", O_WRONLY | O_APPEND)
```