# Serial Programming/termios

## * Introduction

⇒ termios is the newer Unix API for terminal I/O.

⇒ The anatomy of a program performing serial I/O with the help of termios is as follows:

→ Open serial device with standard Unix system call open(2).

→ Configure communication parameters and other interface properties with the help of termios functions and data structures.

→ Use standard Unix system calls read(2) and write(2) for reading from & writing to the serial interface.

→ Close device with the standard Unix system call close(2) when done.

⇒ The necessary declarations and constants for termios can be found in the header file <termios.h>

↳ Some other functions and declarations can also be found in the <stdio.h>, <fcntl.h> and <unistd.h> header file.

⇒ The termios I/O API supports two different modes :-

→ **Canonical mode** { Default }

This is most useful when dealing with real terminals, or devices that provide line-by-line communication.

→ **Non-canonical mode**

In this mode, no special processing is done, and the terminal driver returns individual characters.

⇒ Configuration is done using the struct termios data structure, defined in the termios.h.

```
struct termios {
    tcflag_t c_iflag    /* input specific flag (bitmask) */
    tcflag_t c_oflag    /* Output specific flag (bitmask) */
    tcflag_t c_cflag    /* Control flag (bitmask) */
    tcflag_t c_lflag    /* local flags (bitmask) */
    cc_t     c_cc[NCCS] /* special characters */
};
```

⇒ There are more than 45 different flags that can be set with the help of the struct termios.

# *Opening / Closing a Serial Device

## Open(2)

→ Decision must be taken about should the device be opened for:

- → reading only
- → writing only
- → both reading and writting
- → blocking or non-blocking (I/O)

(recommended)

## Example

```
Const char *device = "/dev/tty0S0";
fd= open (device, O_RDWR | O_NOCTTY
                  | O_NDELAY);
  if (fd == -1) {
    Printf ("faild to Open port \n");
  }
```

fd ⇒ The returned file handle for the device.
-1 if errors occurred.

O_RDWR ⇒ Open port for reading and writing.

O_NOCTTY ⇒ The port never becomes the Controlling terminal of the process.

O_NDELAY ⇒ Use non-blocking I/O

## Close(2)

Give an open file handle fd you can close it with the following system call.

```
close(fd);
```

## * Basic configuration of a Serial Interface

⇒ After a Serial device has been opened, it is typical that it default configuration, like baud rate and line discipline needs to be overwritten with the desired parameters.

↳ This is done with a complex data structure, and the tcgetattr(3) and tcsetattr(3) function.

### Example

ICANON ⇒ Enable Canonical mode or Non-Canonical mode.

ECHO ⇒ Controls whether input is immediately re-echoed as output.

{both in local mode}