STREAMS message/PIPEs/FIFO:pipe, popenand pcloseFunctions

Assignment No: 13c

Subject:- Unix Operating System

System Lab Class :- TYIT

Name	PRN
Aditi Sudhir Ghate	2020BTEIT00044

Objectives:

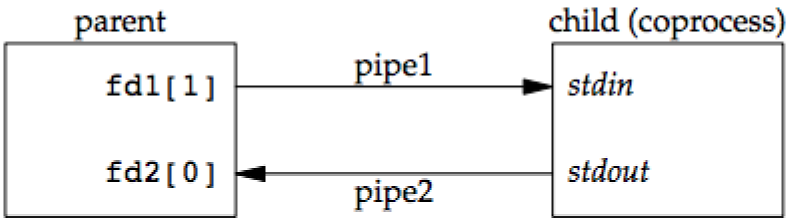
1.Simple filter to add two numbers. (B)

Theory:

A UNIX system filter is a program that reads from standard input and writes to standard output. Filters are normally connected linearly in shell pipelines. A filter becomes a coprocess when the same program generates the filter's input and reads the filter's output.

The Korn shell provides coprocesses. The Bourne shell, the Bourne-again shell, and the C shell don't provide a way to connect processes together as coprocesses. A coprocess normally runs in the background from a shell, and its standard input and standard output are connected to another program using a pipe. Although the shell syntax required to initiate a coprocess and connect its input and output to other processes is quite contorted, coprocesses are also useful from a C program.

Whereas popen gives us a one-way pipe to the standard input or from the standard output of another process, with a coprocess we have two one-way pipes to the other process: one to its standard input and one from its standard output. We want to write to its standard input, let it operate on the data, and then read from its standard output.



Example: invoking add2 as a coprocess

For example, the process creates two pipes: one is the standard input of the coprocess and the other is the standard output of the coprocess. The figure below shows this arrangement:

Figure 15.16 Driving a coprocess by writing its standard input and reading its standard output

The following program is a simple coprocess that reads two numbers from its standard input, computes their sum, and writes the sum to its standard output.

Program:

```
#include "apue.h"

int

main(void)
```

```
{
    int     n, int1, int2;
    char    line[MAXLINE];

    while ((n = read(STDIN_FILENO, line, MAXLINE)) > 0) {
        line[n] = 0;        /* null terminate */
        if (sscanf(line, "%d%d", &int1, &int2) == 2) {
            sprintf(line, "%d\n", int1 + int2);
            n = strlen(line);
            if (write(STDOUT_FILENO, line, n) != n)
                err_sys("write error");
        } else {
            if (write(STDOUT_FILENO, "invalid args\n", 13) != 13)
                err_sys("write error");
        }
    }
    exit(0);
}
```

Input:

```
10 12
```

Output:

```
22
```

Conclusion:

In this way two numbers are added.

References:

1.[https://notes.shichao.io/apue/ch15/