// C program to illustrate

// pipe system call in C

// shared by Parent and Child

#include <stdio.h>

#include <unistd.h>

#define MSGSIZE 16

char\* msg1 = "hello, world #1";

char\* msg2 = "hello, world #2";

char\* msg3 = "hello, world #3";

int main()

{

char inbuf[MSGSIZE];

int p[2], pid, nbytes;

if (pipe(p) < 0)

exit(1);

/\* continued \*/

if ((pid = fork()) > 0) {

write(p[1], msg1, MSGSIZE);

write(p[1], msg2, MSGSIZE);

write(p[1], msg3, MSGSIZE);

// Adding this line will

// not hang the program

// close(p[1]);

wait(NULL);

}

else {

// Adding this line will

// not hang the program

// close(p[1]);

while ((nbytes = read(p[0], inbuf, MSGSIZE)) > 0)

printf("% s\n", inbuf);

if (nbytes != 0)

exit(2);

printf("Finished reading\n");

}

return 0;

}

Named pipe:

Writer:

// C program to implement one side of FIFO

// This side writes first, then reads

#include <stdio.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int fd;

// FIFO file path

char \* myfifo = "/tmp/myfifo";

// Creating the named file(FIFO)

// mkfifo(<pathname>, <permission>)

mkfifo(myfifo, 0666);

char arr1[80], arr2[80];

while (1)

{

// Open FIFO for write only

fd = open(myfifo, O\_WRONLY);

// Take an input arr2ing from user.

// 80 is maximum length

fgets(arr2, 80, stdin);

// Write the input arr2ing on FIFO

// and close it

write(fd, arr2, strlen(arr2)+1);

close(fd);

// Open FIFO for Read only

fd = open(myfifo, O\_RDONLY);

// Read from FIFO

read(fd, arr1, sizeof(arr1));

// Print the read message

printf("User2: %s\n", arr1);

close(fd);

}

return 0;

}

Reader:

// C program to implement one side of FIFO

// This side reads first, then reads

#include <stdio.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int fd1;

// FIFO file path

char \* myfifo = "/tmp/myfifo";

// Creating the named file(FIFO)

// mkfifo(<pathname>,<permission>)

mkfifo(myfifo, 0666);

char str1[80], str2[80];

while (1)

{

// First open in read only and read

fd1 = open(myfifo,O\_RDONLY);

read(fd1, str1, 80);

// Print the read string and close

printf("User1: %s\n", str1);

close(fd1);

// Now open in write mode and write

// string taken from user.

fd1 = open(myfifo,O\_WRONLY);

fgets(str2, 80, stdin);

write(fd1, str2, strlen(str2)+1);

close(fd1);

}

return 0;

}