P1.c

1 /\*pipe, pipe2 - create pipe

2 int pipe(int pipefd[2]);

3 int pipe2(int pipefd[2], int flags);

4

5 DESCRIPTION:pipe() creates a pipe, a unidirectional data channel that can be used for interprocess communication. The array pipefd is used to return two file descriptors referring to the ends

6 of the pipe. pipefd[0] refers to the read end of the pipe. pipefd[1] refers to the write

7 end of the pipe. Data written to the write end of the pipe is buffered by the kernel until it is read from the read e nd of the pipe.\*/

8 #include<stdio.h>

9 main()

10 {

11 int fd[2];

12 if(pipe(fd)<0)

13 {

14 perror("pipe");

15 return;

16 }

17 printf("fd[0]=%d fd[1]=%d\n",fd[0],fd[1]);

18 }

P2.c

1 //wap to transfer the data b/w related processes using pipe

2 #include<stdio.h>

3 main()

4 {

5 int fd[2];

6 if(pipe(fd)<0)

7 {

8 perror("pipe");

9 return;

10 }

11

12 if(fork()==0)

13 {

14 char b[20];

15 printf("before read..\n");

16 read(fd[0],b,sizeof(b));

17 printf("after read\n");

18 printf("in child data:%s\n",b);

19 //exclusively child

20 }

21 else

22 {

23 //exclusively parent

24 char a[20];

25 printf("enter the data...\n");

26 scanf("%s",a);

27 write(fd[1],a,strlen(a)+1);

28 }

29 }