P1.c

1 /\* int dup(int oldfd);

2 int dup2(int oldfd, int newfd);

3 These system calls create a copy of the file descriptor oldfd.

4 dup() uses the lowest-numbered unused descriptor for the new descriptor.

5 dup2() makes newfd be the copy of oldfd, closing newfd first if necessary.

6 After a successful return from one of these system calls, the old and new

7 file descriptors may be used interchangeably. \*/

8 #include<stdio.h>

9 #include<fcntl.h>

10 main()

11 {

12 int fd1,fd2;

13 fd1=open("temp",O\_WRONLY|O\_CREAT|O\_TRUNC,0644);

14 if(fd1<0)

15 {

16 perror("open");

17 return;

18 }

19 printf("fd1:%d\n",fd1);

20 fd2=dup(fd1);

21 printf("fd2:%d\n",fd2);

22 write(fd1,"hello",5);

23 write(fd2,"vector",6);

24 }

P2.c

1 /\* int dup(int oldfd);

2 int dup2(int oldfd, int newfd);

3 These system calls create a copy of the file descriptor oldfd.

4 dup() uses the lowest-numbered unused descriptor for the new descriptor.

5 dup2() makes newfd be the copy of oldfd, closing newfd first if necessary.

6 After a successful return from one of these system calls, the old and new

7 file descriptors may be used interchangeably. \*/

8 #include<stdio.h>

9 #include<fcntl.h>

10 main()

11 {

12 int fd1,fd2;

13 fd1=open("temp",O\_WRONLY|O\_CREAT|O\_TRUNC,0644);

14 if(fd1<0)

15 {

16 perror("open");

17 return;

18 }

19 printf("fd1:%d\n",fd1);

20 close(1);

21 fd2=dup(fd1);

22 printf("hello\n");

23 }

P3.c

1 /\* int dup(int oldfd);

2 int dup2(int oldfd, int newfd);

3 These system calls create a copy of the file descriptor oldfd.

4 dup() uses the lowest-numbered unused descriptor for the new descriptor.

5 dup2() makes newfd be the copy of oldfd, closing newfd first if necessary.

6 After a successful return from one of these system calls, the old and new

7 file descriptors may be used interchangeably. \*/

8 #include<stdio.h>

9 #include<fcntl.h>

10 main()

11 {

12 int fd1,fd2;

13 fd1=open("temp",O\_WRONLY|O\_CREAT|O\_TRUNC,0644);

14 if(fd1<0)

15 {

16 perror("open");

17 return;

18 }

19 printf("fd1:%d\n",fd1);

20 fd2=dup2(fd1,9);

21 printf("fd2:%d\n",fd2);

22 }