P1.c

1 /\* lstat() is identical to stat(), except that if path is a symbolic link, then the link itself

2 is stat-ed, not the file that it refers to.\*/

3 #include<stdio.h>

4 #include<sys/stat.h>

5 main(int argc,char \*argv[])

6 {

7 struct stat v;

8 if(argc!=2)

9 {

10 printf("./a.out filename\n");

11 return;

12 }

13 if(lstat(argv[1],&v)<0)

14 {

15 perror("stat");

16 return;

17 }

18 printf("size of the file:%d\n",v.st\_size);

19 printf("inode num of the file:%d\n",v.st\_ino);

20

21 }

P2.c

1 //wap to find that given files are link files or not if they are having link then find out what type of link?

2 //$./a.out file1 file2

3 #include<stdio.h>

4 #include<sys/stat.h>

5 main(int argc,char \*argv[])

6 {

7 struct stat v1,v2;

8 if(argc!=3)

9 {

10 printf("./a.out file1 fle2\n");

11 return;

12 }

13 stat(argv[1],&v1);

14 stat(argv[2],&v2);

15 if(v1.st\_ino==v2.st\_ino) //true if files are hard link or sodt link

16 {

17 lstat(argv[1],&v1);

18 lstat(argv[2],&v2);

19 if(v1.st\_ino==v2.st\_ino)

20 printf("hard link...\n");

21 else

22 printf("soft link..\n");

23 }

24 else

25 printf("no link...\n");

26

27 }