P1.c

1 //wap to find the type of file

2 //./a.out filename

3

4 #include<stdio.h>

5 #include<sys/stat.h>

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

7 {

8 struct stat v;

9 if(argc!=2)

10 {

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

12 return 0;

13 }

14 stat(argv[1],&v);

15 printf("%o\n",v.st\_mode);

16 }

P2.c

1 //wap to find the type of file

2 //./a.out filename

3 //Method1->using Bit mask

4 #include<stdio.h>

5 #include<sys/stat.h>

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

7 {

8 struct stat v;

9 if(argc!=2)

10 {

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

12 return 0;

13 }

14 stat(argv[1],&v);

15 if((v.st\_mode & (0170000))==0100000)

16 printf("Regular file..\n");

17 else if((v.st\_mode & (0170000))==0120000)

18 printf("link file..\n");

19 else if((v.st\_mode & (0170000))==0040000)

20 printf("dir file...\n");

21 else

22 printf("pipe file..\n");

23 }

P3.c

1 //wap to find the type of file

2 //./a.out filename

3 //Method2->using posix macros(portable operating systen interface for unix)

4 #include<stdio.h>

5 #include<sys/stat.h>

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

7 {

8 struct stat v;

9 if(argc!=2)

10 {

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

12 return 0;

13 }

14 stat(argv[1],&v);

15

16 if(S\_ISREG(v.st\_mode)) //macro replaces 1 or 0 based on st\_mode bits

17 printf("Regular file...\n");

18 else if(S\_ISDIR(v.st\_mode))

19 printf("Dir file...\n");

20 else if(S\_ISLNK(v.st\_mode))

21 printf("Link file...\n");

22 }

P4.c

1 //wap to display atime,mtime,ctime

2 #include<stdio.h>

3 #include<sys/stat.h>

4 #include<time.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 0;

12 }

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

14 printf("atime:%u\n",v.st\_atime);

15 printf("atime:%s\n",ctime(&v.st\_atime));

16 printf("mtime:%u\n",v.st\_mtime);

17 printf("mtime:%s\n",ctime(&v.st\_mtime));

18 printf("ctime:%u\n",v.st\_ctime);

19 printf("ctime:%s\n",ctime(&v.st\_ctime));

20 }