P1.c

1 //wap to read char by char from the file?

2 #include<stdio.h>

3 #include<fcntl.h>

4 main()

5 {

6 int fd,ret;

7 char ch;

8 fd=open("temp",O\_RDONLY);

9 if(fd<0)

10 {

11 perror("open");

12 return;

13 }

14 while(ret=read(fd,&ch,1)>0)

15 printf("%c",ch);

16 //printf("ch=%c\n",ch);

17 }

P2.c

1 #include<stdio.h>

2 #include<fcntl.h>

3 main()

4 {

5 int fd,ret;

6 char arr[20];

7 fd=open("temp",O\_RDONLY);

8 if(fd<0)

9 {

10 perror("open");

11 return;

12 }

13 ret=read(fd,arr,5);

14 if(ret<=0)

15 {

16 perror("read");

17 return;

18 }

19 arr[ret]='\0';

20 printf("%s\n",arr);

21 }

P3.c

1 /\*bzero - write zero-valued bytes

2 void bzero(void \*s, size\_t n);

3 DESCRIPTION:The bzero() function sets the first n bytes of the area starting at s to zero (bytes containing '\0').

4 \*/

5 #include<stdio.h>

6 #include<fcntl.h>

7 main()

8 {

9 int fd,ret;

10 char arr[20];

11 bzero(arr,20);

12 fd=open("temp",O\_RDONLY);

13 if(fd<0)

14 {

15 perror("open");

16 return;

17 }

18 ret=read(fd,arr,5);

19 if(ret<=0)

20 {

21 perror("read");

22 return;

23 }

24 printf("%s\n",arr);

25 }

P4.c

1 /\*memset - fill memory with a constant byte

2 void \*memset(void \*s, int c, size\_t n);

3 DESCRIPTION:The memset() function fills the first n bytes of the memory area pointed to by s with the constant byte c.

4 \*/

5 #include<stdio.h>

6 #include<fcntl.h>

7 main()

8 {

9 int fd,ret;

10 char arr[20];

11 memset(arr,0,20);

12 fd=open("temp",O\_RDONLY);

13 if(fd<0)

14 {

15 perror("open");

16 return;

17 }

18 ret=read(fd,arr,5);

19 if(ret<=0)

20 {

21 perror("read");

22 return;

23 }

24 printf("%s\n",arr);

25 }