P1.c

1 /\*alarm - set an alarm clock for delivery of a signal

2 unsigned int alarm(unsigned int seconds);\*/

3 #include<stdio.h>

4 #include<signal.h>

5 main()

6 {

7 printf("hi....\n");

8 alarm(10);

9 printf("hello...\n");

10 printf("process under execution...\n");

11 while(1);

12 }

P2.c

1 /\*alarm - set an alarm clock for delivery of a signal

2 unsigned int alarm(unsigned int seconds);\*/

3 #include<stdio.h>

4 #include<signal.h>

5 main()

6 {

7 printf("hi....\n");

8 alarm(10);

9 alarm(5);

10 alarm(1);

11 printf("hello...\n");

12 printf("process under execution...\n");

13 while(1);

14 }

P3.c

1 #include<stdio.h>

2 #include<signal.h>

3 void user\_isr(int n)

4 {

5 printf("n=%d\n",n);

6 printf("in isr..\n");

7 printf("alarm signal scheduled for this process again after 2sec\n");

8 alarm(2);

9 }

10 main()

11 {

12 printf("hi....\n");

13 signal(SIGALRM,user\_isr);

14 printf("alarm signal scheduled for this process after 10sec\n");

15 alarm(10);

16 printf("hello...\n");

17 printf("process under execution...\n");

18 while(1);

19 }

P4.c

1 #include<stdio.h>

2 #include<signal.h>

3 int count=0;

4 void user\_isr(int n)

5 {

6 count++;

7 printf("n=%d\n",n);

8 printf("in isr..\n");

9 printf("alarm signal scheduled for this process again after 2sec\n");

10 if(count==5)

11 signal(SIGALRM,SIG\_DFL);

12 alarm(2);

13 }

14 main()

15 {

16 printf("hi....\n");

17 signal(SIGALRM,user\_isr);

18 printf("alarm signal scheduled for this process after 10sec\n");

19 alarm(10);

20 printf("hello...\n");

21 printf("process under execution...\n");

22 while(1);

23 }

P5.c

1 /\*RETURN VALUE:

2 alarm() returns the number of seconds remaining until any previously scheduled alarm was due to be delivered, or zero if there was no previously scheduled alarm.\*/

3 #include<stdio.h>

4 main()

5 {

6 int ret;

7 ret=alarm(10);

8 printf("1)ret:%d\n",ret); //0

9 ret=alarm(5);

10 printf("2)ret:%d\n",ret); //10

11 while(1);

12 }

P6.c

1 /\*RETURN VALUE:

2 alarm() returns the number of seconds remaining until any previously scheduled alarm was due to be delivered, or zero if there was no previously scheduled alarm.\*/

3 #include<stdio.h>

4 main()

5 {

6 int ret;

7 ret=alarm(20);

8 printf("1)ret:%d\n",ret); //0

9 sleep(3);

10 ret=alarm(10);

11 printf("2)ret:%d\n",ret); //17

12 sleep(2);

13 ret=alarm(5);

14 printf("3)ret:%d\n",ret); //8

15 sleep(1);

16 ret=alarm(2); //4

17 printf("4)ret:%d\n",ret);

18 while(1);

19 }

P7.c

1 //wap to disable ctrl+c(signal no 2) and ctrl+\(signal no 3) in a process upto

2 //10 seconds

3 #include<stdio.h>

4 #include<signal.h>

5 void isr(int n)

6 {

7 printf("in isr...\n");

8 signal(2,SIG\_DFL);

9 signal(3,SIG\_DFL);

10 }

11 main()

12 {

13 printf("process executing...\n");

14 signal(2,SIG\_IGN);

15 signal(3,SIG\_IGN);

16 signal(SIGALRM,isr);

17 alarm(10);

18 while(1);

19 }

P8.c

1 //SIGKILL(9) and SIGSTOP(19) can not be caught or ignored

2 #include<stdio.h>

3 #include<signal.h>

4 main()

5 {

6 printf("process executing..\n");

7 printf("pid:%d\n",getpid());

8 signal(9,SIG\_IGN);

9 while(1);

10 }