P1.c

1 //record locking using semaphores

2 #include<stdio.h>

3 #include<fcntl.h>

4 #include<sys/sem.h>

5 main()

6 {

7 char a[20]="abcdefghijklmn";

8 int fd,id,i=0,ret;

9 struct sembuf v;

10 fd=open("temp",O\_WRONLY|O\_CREAT|O\_APPEND,0644);

11 if(fd<0)

12 {

13 perror("open");

14 return;

15 }

16

17 id=semget(5,3,IPC\_CREAT|0644);

18 if(id<0)

19 {

20 perror("semget");

21 return;

22 }

23 v.sem\_num=0;//semaphore number(0/1/2)

24 v.sem\_op=0; //semaphore operation(0(or)+ve(or)-ve)

25 //v.sem\_flg=0;//not used any flag

26 v.sem\_flg=IPC\_NOWAIT;

27 printf("before semop..\n");

28 ret=semop(id,&v,1);//semop function will deside a process allowed to access resource or not

29 if(ret==-1)

30 {

31 perror("semop");

32 return;

33 }

34 printf("after semop...\n");

35 semctl(id,0,SETVAL,1);

36 printf("writing data into file..\n");

37 while(a[i]) //critical section of code

38 {

39 write(fd,a+i,1);

40 sleep(1);

41 i++;

42 }

43 printf("writing data by p1 process completed..\n");

44 semctl(id,0,SETVAL,0);

45 printf("resource released...\n");

46 }

P2.c

1 #include<stdio.h>

2 #include<fcntl.h>

3 #include<sys/sem.h>

4 main()

5 {

6 char a[20]="1234567890";

7 int fd,id,i=0,ret;

8 struct sembuf v;

9 fd=open("temp",O\_WRONLY|O\_CREAT|O\_APPEND,0644);

10 if(fd<0)

11 {

12 perror("open");

13 return;

14 }

15

16 id=semget(5,3,IPC\_CREAT|0644);

17 if(id<0)

18 {

19 perror("semget");

20 return;

21 }

22 v.sem\_num=0;//semaphore number(0/1/2)

23 v.sem\_op=0; //semaphore operation(0(or)+ve(or)-ve)

24 //v.sem\_flg=0;//not used any flag

25 v.sem\_flg=IPC\_NOWAIT;

26 printf("before semop..\n");

27 ret=semop(id,&v,1);//semop function will deside a process allowed to access resource or not

28 if(ret==-1)

29 {

30 perror("semop");

31 return;

32 }

33 printf("after semop...\n");

34 semctl(id,0,SETVAL,1);

35 printf("writing data into file..\n");

36 while(a[i])

37 {

38 write(fd,a+i,1);

39 sleep(1);

40 i++;

41 }

42 printf("writing data by p2 process completed..\n");

43 semctl(id,0,SETVAL,0);

44 printf("resource released...\n");

}