

```

/* 信号量通用文件：初始化、删除、PV 操作 */
/* sem_com.c */

/* 信号量初始化（赋值）函数*/
int sem_init(int sem_id, int sem_num, int init_value)
{
    union semun{
        int val;
        struct semid_ds *buf;
        unsigned short *array;
    };
    union semun sem_union;
    sem_union.val = init_value; /* init_value 为初始值 */
    /* 为信号量集中的第 sem_num 个信号量设置初值 */
    if (semctl(sem_id, sem_num, SETVAL, sem_union) == -1)
    {
        perror("Initialize semaphore");
        return -1;
    }
    return 0;
}

/* 删除信号量集的函数 */
int sem_del(int sem_id)
{
    if (semctl(sem_id, 0, IPC_RMID, NULL) == -1)
    {
        perror("Delete semaphore");
        return -1;
    }
}

/* P 操作函数 */
int sem_p(int sem_id, int sem_num)
{
    struct sembuf sem_b;
    sem_b.sem_num = sem_num; /* 对第 sem_num 个信号量进行操作*/
    sem_b.sem_op = -1; /* 表示 P 操作 */
    sem_b.sem_flg = 0;
    //sem_b.sem_flg = SEM_UNDO; /* 系统自动释放将会在系统中残留的信号量*/
    if (semop(sem_id, &sem_b, 1) == -1)

```

```
{
perror("P operation");
return -1;
}
return 0;
}

/* V 操作函数*/
int sem_v(int sem_id, int sem_num)
{
    struct sembuf sem_b;
    sem_b.sem_num = sem_num; /* 为第 sem_num 个信号量进行操作*/
    sem_b.sem_op = 1; /* 表示 V 操作 */
    sem_b.sem_flg = 0;
    //sem_b.sem_flg = SEM_UNDO; /* 系统自动释放将会在系统中残留的信号量*/
    if (semop(sem_id, &sem_b, 1) == -1)
    {
        perror("V operation");
        return -1;
    }
    return 0;
}
```