Semaphore operations

1. review of basic semaphore operation:
  P - if semaphore value is 0, wait; otherwise, decrement semaphore value.
  V - if any waiters, wake one; otherwise, increment semaphore value.

2. How to use semaphore in C?
(1). Include three <sys> head files

(2). Use semget(), semctl() and semop()
  - semget(), providing a key, permissions, and # of sems in a set.
  - semctl() to initialize value (using val field in semun union)
  - semop() to perform up/down operations

Example:

```
//=====================================//
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

int mysem;        //ID for a semaphore set, like file handler for a file.
#define KEY 1100  //can be any number

//get the i-th semaphore value in a semaphore set mysem.
int get_sem(int i)
{
   ushort semarr[30];
   union semun
   {
      int val;
      struct semid_ds *buf;
      ushort *array;
   } arg;

   arg.array = semarr;
   semctl(mysem, i, GETALL, arg);
   return semarr[i];
}

void show_sem(int i)
{
   int val;

   val = get_sem(i);
   printf("semaphore[%d]=%d\n", i, val);
}

//create a semaphore set with N semaphores using a KEY.
void create_sem(int N)
{
   printf("create %d semaphores\n", N);
   mysem = semget(KEY, N, 0666 | IPC_CREAT);
   if(mysem < 0)
      error_msg();
}
```

```
//initialize all semaphores value=1
void init_sem(int N)
{
   int j;
   int retval;
   union semun
   {
      int val;
      struct semid_ds *buf;
      ushort *array;
   } arg;

   arg.val = 1;
   for (j=0; j<N; j++)
   {
      retval = semctl(mysem, j, SETVAL, arg);
      if(retval < 0)  error2("Error: semctl", j);
      show_sem(j);
   }
}

//P and V operations using semop()
void PV(int i, int act)
{
   struct sembuf op;
   int retval;

   op.sem_num = i;
   op.sem_op = act;  //1=V, -1=P
   op.sem_flg = 0;   //will wait
   retval = semop(mysem, &op, 1);
   if(retval != 0)  error2("error: semop: ", act);
   //show_sem(i);
}

//P operation on semaphore i
void P(int i)
{
   PV(i, -1);
}

void V(int i)
{
   PV(i, 1);
}
```