```c
#include <sys/types.h>
#include <sys/sem.h>
#include <sys/stat.h>
#include "semun.h"                      /* Definition of semun union */
#include "tlpi_hdr.h"

int main(int argc, char *argv[]){
        int semid, key, perms;
        struct sembuf sops[2];
        key = 12345;
        perms = S_IRUSR | S_IWUSR;
        semid = semget(key, 1, IPC_CREAT | IPC_EXCL | perms);

        if (semid != -1) {                  /* Successfully created the semaphore
*/
                union semun arg;
                /* XXXX */
                arg.val = 0;                    /* So initialize it */
                if (semctl(semid, 0, SETVAL, arg) == -1)
                        errExit("semctl");
        }else {                             /* We didn't create semaphore set */
                if (errno != EEXIST) {          /* Unexpected error from semget()
*/
                        errExit("semget 1");
                }else{                          /* Someone else already created it */
                        semid = semget(key, 1, perms);      /* So just get ID */
                        if (semid == -1)
                                errExit("semget 2");
                }
        }
    /* Now perform some operation on the semaphore */
        sops[0].sem_op = 1;             /* Add 1 */
        sops[0].sem_num = 0;            /* ... to semaphore 0 */
        sops[0].sem_flg = 0;
        if (semop(semid, sops, 1) == -1)
                errExit("semop");

        exit(EXIT_SUCCESS);
}
```



```c
#include <sys/types.h>
#include <sys/sem.h>
#include <sys/stat.h>
#include "semun.h"                      /* Definition of semun union */
#include "tlpi_hdr.h"

int main(int argc, char *argv[]){
        int semid, key, perms;
        struct sembuf sops[2];

        if (argc != 2 || strcmp(argv[1], "--help") == 0)
                usageErr("%s sem-op\n", argv[0]);

        key = 12345;
        perms = S_IRUSR | S_IWUSR;
        semid = semget(key, 1, IPC_CREAT | IPC_EXCL | perms);

        if (semid != -1) {                  /* Successfully created the semaphore */
                union semun arg;
                struct sembuf sop;

                sleep(5);
                printf("%ld: created semaphore\n", (long) getpid());
                arg.val = 0;                    /* So initialize it to 0 */
                if (semctl(semid, 0, SETVAL, arg) == -1)
                        errExit("semctl 1");

                printf("%ld: initialized semaphore\n", (long) getpid());

                /* Perform a "no-op" semaphore operation - changes sem_otime so
other processes can see we've initialized the set. */
                sop.sem_num = 0;                /* Operate on semaphore 0 */
                sop.sem_op = 0;                 /* Wait for value to equal 0 */
                sop.sem_flg = 0;
                if (semop(semid, &sop, 1) == -1)
                        errExit("semop");

                printf("%ld: completed dummy semop()\n", (long) getpid());
        }else{                              /* We didn't create the semaphore set */
                if(errno != EEXIST){            /* Unexpected error from semget() */
                        errExit("semget 1");
                }else{                          /* Someone else already created it */
                        const int MAX_TRIES = 10;
                        int j;
                        union semun arg;
                        struct semid_ds ds;
                        semid = semget(key, 1, perms);      /* So just get ID */
                        if (semid == -1)
                                errExit("semget 2");

                        printf("%ld: got semaphore key\n", (long) getpid());
                        /* Wait until another process has called semop() */
                        arg.buf = &ds;
                        for (j = 0; j < MAX_TRIES; j++){
                                printf("Try %d\n", j);
                                if (semctl(semid, 0, IPC_STAT, arg) == -1)
                                        errExit("semctl 2");

                        /* Semop() performed? If not, wait and retry else quit
loop */
                                if (ds.sem_otime != 0)
                                        break;
                                sleep(1);
                        }

                if(ds.sem_otime == 0)           /* Loop ran to completion! */
```

```
                        fatal("Existing semaphore not initialized");
                }
        }
    /* Now perform some operation on the semaphore */
    sops[0].sem_num = 0;                        /* Operate on semaphore 0... */
    sops[0].sem_op = getInt(argv[1], 0, "sem-op");
    sops[0].sem_flg = 0;
    if (semop(semid, sops, 1) == -1)
        errExit("semop");

        exit(EXIT_SUCCESS);
}
```