Semaphore (6A)

- Semaphore

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Young Won Lim
12/06/2012

semget()

```
int semget ( key_t key, int nsems, int semflg );

Returns semaphore set identifier (sid) on success
key – the return value of ftok()
Nsems - argument specifies the number of semaphores
semflg
IPC_CREAT - Create the semaphore set if it doesn't already exist
IPC_CREAT | IPC_EXCL - fail if semaphore set already exists.

sid = semget( mykey, numsems, IPC_CREAT | 0660 )
```

semop() - (1)

```
int semop ( int semid, struct sembuf *sops, unsigned nsops);

semid - the return value of semget()
sops - a pointer to an array of operations
       to be performed on the semaphore set
nsops -the number of operations in that array.


struct sembuf {
  ushort  sem_num;        /* semaphore index in array */
  short   sem_op;         /* semaphore operation */
  short   sem_flg;        /* operation flags */
};

sem_num   The number of the semaphore you wish to deal with
sem_op    The operation to perform (positive, negative, or zero)
sem_flg   Operational flags
```

semop() - (2)

```
int semop ( int semid, struct sembuf *sops, unsigned nsops);
```

negative sem_op
is added to the semaphore.
the calling process sleeps until the
requested amount of resources are
available in the semaphore

positive sem_op
is added to the semaphore.
returning resources back to the
application's semaphore set

zero sem_op
the calling process will sleep() until
the semaphore's value is 0.
waiting for a semaphore to reach 100%
utilization

```
struct sembuf {
  ushort  sem_num;
  short   sem_op;
  short   sem_flg;
};
```

semop() - (3)

```
int semop ( int semid, struct sembuf *sops, unsigned nsops);
```

```
struct sembuf {
  ushort  sem_num;
  short   sem_op;
  short   sem_flg;
};
```

```
struct sembuf sem_lock = { 0, -1, IPC_NOWAIT };
a value of ``-1''
will be added to semaphore number 0
in the semaphore set.

semop(sid, &sem_lock, 1);

struct sembuf sem_unlock = { 0, 1, IPC_NOWAIT };
a value of ``1''
will be added to semaphore number 0
in the semaphore set.
```

semctl() - (1)

```
int semctl ( int semid, int semnum, int cmd, union semun arg );

    /* arg for semctl system calls. */
    union semun {
        int val;                /* value for SETVAL */
        struct semid_ds *buf;   /* buffer for IPC_STAT & IPC_SET */
        ushort *array;          /* array for GETALL & SETALL */
        struct seminfo *__buf;  /* buffer for IPC_INFO */
        void *__pad;
    };
```

IPC_STAT
IPC_SET
IPC_RMID

GETPID
GETNCNT
GETZCNT

GETALL
GETVAL

SETALL
SETVAL

semctl() - (2)

int semctl (int semid, int semnum, int cmd, union semun arg);

IPC_STAT Retrieves the semid_ds structure for a set, and stores it in the address of the buf argument in the semun union.

IPC_SET Sets the value of the ipc_perm member of the semid_ds structure for a set. Takes the values from the buf argument of the semun union.

IPC_RMID Removes the set from the kernel.

GETALL Used to obtain the values of all semaphores in a set. The integer values are stored in an *array* of unsigned short integers pointed to by the array member of the union.

GETNCNT Returns the number of processes currently waiting for resources.

GETPID Returns the PID of the process which performed the last semop call.

GETVAL Returns the value of a single semaphore within the set.

GETZCNT Returns the number of processes currently waiting for 100% resource utilization.

SETALL Sets all semaphore values with a set to the matching values contained in the *array member* of the union.

SETVAL Sets the value of an individual semaphore within the set to the val member of the union.

Reference

References

[1] http://en.wikipedia.org/
[2] http://www.tldp.org/LDP/lpg/node46.html