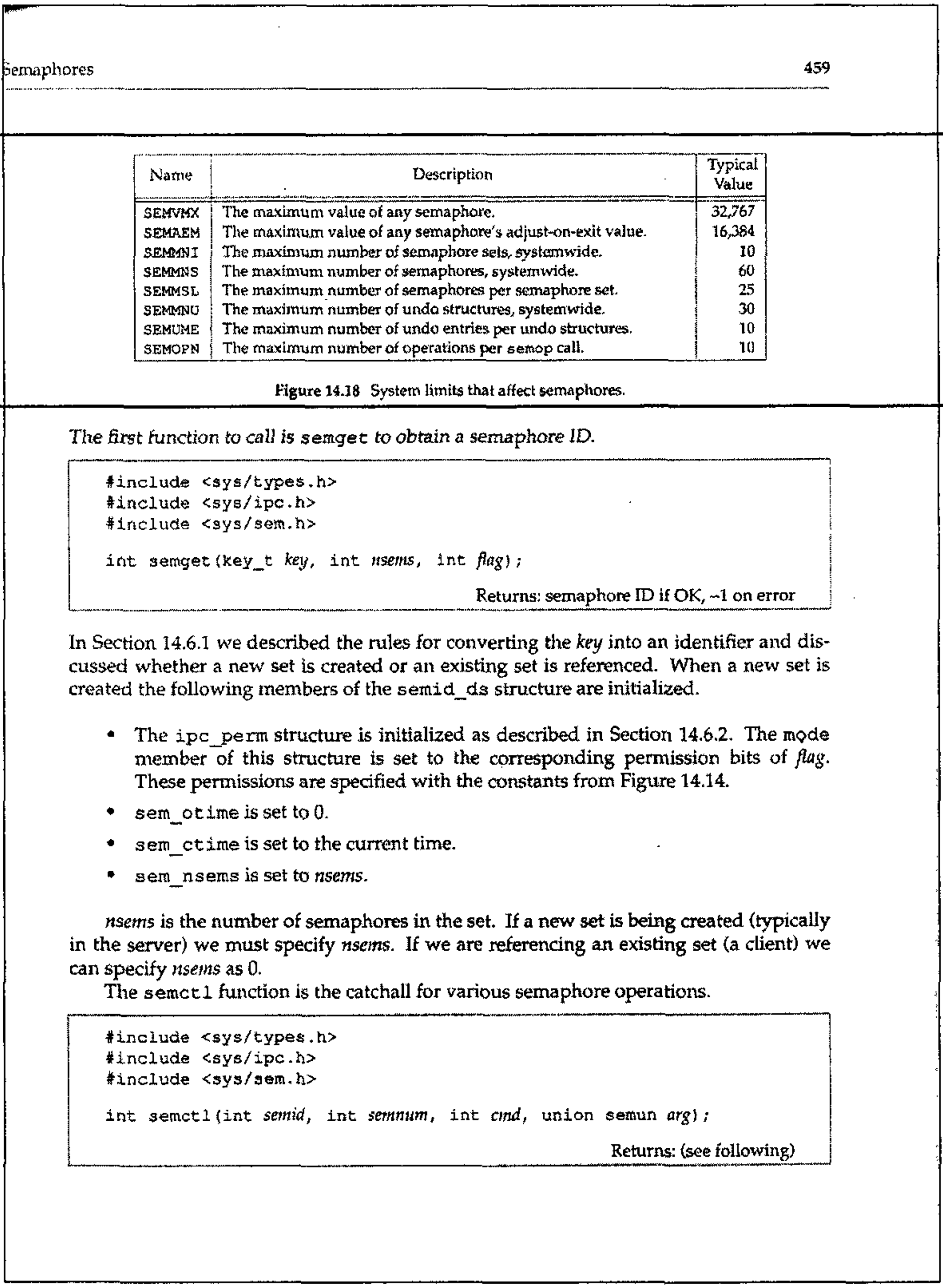

Name	Description	Typical Value
SEMVMX	The maximum value of any semaphore.	32,767
SEMAEM	The maximum value of any semaphore's adjust-on-exit value.	16,384
SEMMNI	The maximum number of semaphore sets, systemwide.	10
SEMMNS	The maximum number of semaphores, systemwide.	60
SEMMSL	The maximum number of semaphores per semaphore set.	25
SEMMNU	The maximum number of undo structures, systemwide.	30
SEMUME	The maximum number of undo entries per undo structures.	10
SEMOPN	The maximum number of operations per semop call.	10

Figure 14.18 System limits that affect semaphores.

The first function to call is `semget` *to obtain a semaphore ID.*

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

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

Returns: semaphore ID if OK, −1 on error

In Section 14.6.1 we described the rules for converting the *key* into an identifier and discussed whether a new set is created or an existing set is referenced. When a new set is created the following members of the `semid_ds` structure are initialized.

- The `ipc_perm` structure is initialized as described in Section 14.6.2. The `mode` member of this structure is set to the corresponding permission bits of *flag*. These permissions are specified with the constants from Figure 14.14.
- `sem_otime` is set to 0.
- `sem_ctime` is set to the current time.
- `sem_nsems` is set to *nsems*.

nsems is the number of semaphores in the set. If a new set is being created (typically in the server) we must specify *nsems*. If we are referencing an existing set (a client) we can specify *nsems* as 0.

The `semctl` function is the catchall for various semaphore operations.

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

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

Returns: (see following)