Operating Systems: Practice: Lesson 5

Sevak Amirkhanian

pthread_exit(...)

```
noreturn void
pthread_exit(void *retval);
```

The pthread_exit() function terminates the calling thread and returns a value via retval.

Semaphores

Semaphore is a another synchronization primitive which have similarities with conditional variables.

Semaphore is basically an integer counter which has 2 operations:
P() // increments the counter
V() // decrements the counter

The value of the semaphore is the number of units of the resource that are currently available

POSIX semaphores

POSIX semaphores have 4 main operations:

sem_init(...)

sem_post(..)

sem_wait(..)

sem_destroy(...)

https://man7.org/linux/man-pages/man7/sem_overview.7.html

Binary semaphores

If the semaphore count is 1, semaphore is essentially a mutex which is why in literature mutex is sometimes called *binary semaphore.*

sem_wait() -> pthread_mutex_lock()

sem_post() -> pthread_mutex_unlock()

Real-life examples of semaphores

Semaphores may be used in networking operating systems in routers for connection throttling.

Semaphores may also be used database drivers to restrict the amount of parallel connections, hence read-write operations.

Producer-Consumer Problem

The simple of description of the problem is:

We have a storage with limited storage with size N.

We have M producers who periodically produce a new item and insert into the storage if and only if there is a free space in the storage.

We have K consumers who periodically consume a new item from the storage if and only if there the storage is not empty.

Solution with semaphores

We will use 3 semaphores:

1 binary semaphore for lock

1 counting semaphore for full state

1 counting semaphore for empty state

New information about homework 4

Input argument in *run(...)* function.

Exit mechanism with *pthread_exit(...)*.

Thank you.