Operating Systems-2
CS 3510 Spring 2019
Theory Assignment 1

INSTRUCTOR: DR. SATHYA PERI

Report By:
Vijay Tadikamalla
CS17BTECH11040

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Answers

1. Let the consider the busy waiting implementation of semaphore. So the value of a semaphore can never go below 0.
   Initial values of semaphores empty = n,semaphores full = 0.

   a. Minimum value of semaphore full = 0 as it can never go below 0.
      Minimum value of semaphore empty = 0. This can happen when n items are produced continuously without being consumed. Max no of items which can be produced is equal to buffer size(n).

   b. Maximum value of empty = n as buffer is empty when semaphore empty is equal to n. So no more items can be consumed to increase the value of empty.
      Maximum value of full is n. This can happen when n items are produced continuously without being consumed. Max no of items which can be produced is equal to buffer size(n).

   c. Let the no of producers be p and no of consumer be c. Size of buffer is n.
      CASE 1: floor(n/2) is greater than p and c.
      Let at any point of time semaphore full is equal to floor(n/2) and semaphore empty is equal to ceil(n/2).
      Now let all the producers exit the wait(empty). So empty == ceil(n/2) - p.
      Now let all the consumers exit the wait(full). So full == floor(n/2) - c.
      So empty+full = n- p -c.
      CASE 2: floor(n/2) is less or equal to p and c.
      So now both producers and consumers exit wait(empty) and wait(full) respectively. So empty and full both become 0.

      So minimum value of (full+empty) == max(0, n-p-c).

      Maximum value of (full+empty)==n because signal() function cannot be called until a wait() function is called.

2. Set the semaphore A equal to 1

Writer

```
while(true){
  wait(A);
  wait(rw_mutex);
  signal(A);
  .
  .
  // Performing writing
  .
  .
  signal(rw_mutex);
}
```

Reader

```
while (true){
  wait(A);
  wait(mutex);
  readCount++;
  if(readCount == 1) wait(rw_mutex);
  signal(A);
  signal(mutex);
  .
  .
  //Performing reading
  .
  .
  wait(mutex);
  read count--;
  if(!readCount)signal(rw_mutex);
  signal(mutex);
}
```

3. The “compare and compare-and-swap” idiom work appropriately for implementing spinlocks because compare and swap is executed atomically. So even if two or more threads enter the conditional statement, they can not leave the while loop together i.e only one thread can enter the CS while the other threads wait in the spinlock.
4. Let the value of semaphore is 1 when two threads simultaneously called the getValue() function. Both the threads will enter the if condition and will call the wait() function. Here the problem arises as one thread will decreases the value of semaphore to 0 and the second thread will get blocked. So the purpose of getValue() function is not fulfilled as the second thread still gets blocked.