

Assignment 5

Thread Synchronization & Mutual Exclusion using Mutex

→ Aim:

Application to demonstrate Reader Writer problem with reader priority.

→ Objective :

Implement C program to demonstrate Reader - Writer problem with readers having priority using counting semaphores and mutex.

→ Theory :

A] Reader-Writer Problem:

There is a data area shared among a number of processes. The data area could be a file, a block of main memory, or even a bank of processor registers. There are a number of processes that only read the data area and a number that only write to the data area.

Conditions that must be followed :

1. Any number of readers may simultaneously read the file.
2. Only one writer at a time may write to the file.
3. If a writer is writing to the file, no reader may read it.

Readers are processes that are not to be excluded from one another. Writers are processes that need to be excluded from other processes. Readers do not also write to the data area nor do the writers read the data while writing.

1. We can allow any of the processes to read or write the data. Hence, we can declare any portion of a process to access the data area to be a critical section & impose the general mutual exclusion solution.
2. We can also impose the processes to read the data one at a time but this would cause intolerable delays. At the same time, writers need to be restricted while reading is taking place.

Solution using Semaphore :

```
int readcount; *  
semaphore r = 1, wsem = 1;
```

```
void reader() {
```

```
    while (true) {
```

```
        SemWait (r);  
        readcount++;  
        if (readcount == 1)  
            semWait (wsem);  
        semSignal (r);  
        Read the value.  
        SemWait (r);  
        readcount--;  
        if (readcount == 0)  
            semSignal (wsem);  
        semSignal (r)
```

```
}
```

```
void writer() {
```

```
    while (true) {
```

```
        semWait (wsem);
```

```
        Write value
```

```
} }
```

```
        semSignal (wsem);
```

```
void main () {
```

```
    readcount = 0;  
    parbegin (reader, writer);  
}
```

→ Conclusion :

Topics Covered :

Different types of solution for Reader-Writer problem.