



EXPERIMENT - 7

Operating Systems Lab

AIM

Write a program to implement reader/writer problem using semaphore.

Syeda Reeha Quasar

14114802719

6C7

EXPERIMENT – 7

Aim:

Write a program to implement reader/writer problems using semaphore.

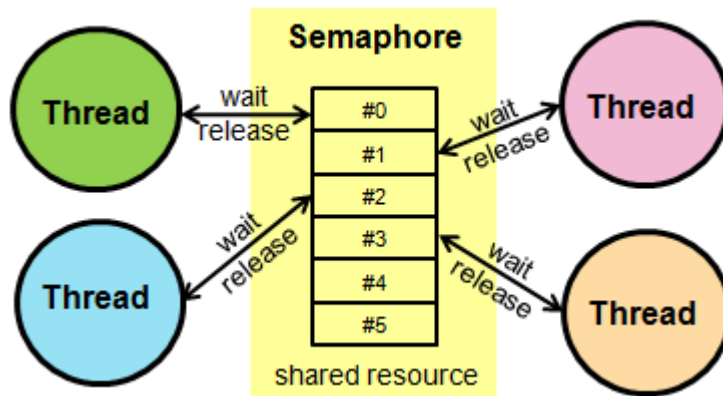
Theory:

The readers-writers problem is a classical problem of process synchronization; it relates to a data set such as a file that is shared between more than one process at a time.

Among these various processes, some are Readers - which can only read the data set they do not perform any updates, some are Writers – who can both read and write in the data sets.

The readers-writers problem is used for managing synchronization among various reader and writer processes so that there are no problems with the data sets, i.e., no inconsistency is generated.

A semaphore is simply an integer variable that is shared between threads. This variable is used to solve the critical section problem and to achieve process synchronization in the multiprocessing environment.



Source Code:

```
function sem_wait(){
    while [ $1 -le 0 ]
    do
        $1=$(( $1 - 1 ))
    done
}

function signal(){
    result=$(( $1 + 1 ))
}

x=1 y=1 z=1
wsem=1 rsem=1 readcount=0 writecount=0 var=5

function reader(){
    echo "-----"
    echo "reader " $1 " is reading"
```

```

sem_wait $z
sem_wait $rsem
sem_wait $x
readcount=$((readcount+1))
if [ $readcount -eq 1 ]
then
    sem_wait $wsem
fi
signal $x
signal $rsem
signal $z
echo "Updated value = " $var
sem_wait $x
readcount=$((readcount-1))
if [ $readcount -eq 0 ]
then
    signal $wsem
fi
signal $x
}

function writer(){
    echo
    echo "writer " $1 " is writing"
    sem_wait $y
    writecount=$((writecount+1))
    if [ $writecount -eq 1 ]
    then
        sem_wait $rsem
    fi
    signal $y
    sem_wait $wsem
    var=$((var+5))
    signal $wsem
    sem_wait $y
    writecount=$((writecount-1))
    if [ $writecount -eq 0 ]
    then
        signal $rsem
    fi
    signal $y
}

```

```

reader 0
writer 0
reader 1
reader 2
reader 3
writer 3
reader 4

```

Output:

```
reeha@Reeha:/mnt/e/sem 6/Operating Systems$ ./semaphore.sh
-----
reader 0 is reading
Updated value = 5

writer 0 is writing
-----
reader 1 is reading
Updated value = 10
-----
reader 2 is reading
Updated value = 10
-----
reader 3 is reading
Updated value = 10

writer 3 is writing
-----
reader 4 is reading
Updated value = 15
```