

CS & IT ENGINEERING



Operating System

Process Synchronization

Lecture -3

By- Vishvadeep Gothi sir



Recap of Previous Lecture



Topic

Peterson's Solution

Topic

Hardware Solutions of Synchronization

Topic

Test-And-Set()

Topic

Swap()

Topic

Semaphore

Topics to be Covered



Topic

Semaphore

Topic

Spin Lock

Question 1

~~turn=0;~~ 1

```
while(true)
{
    while(turn);
    turn=1;
    //CS
    turn=0;
    RS;
}
```

lock = false

X M.E.

✓ Progress

X B.W.

✓ starvation

```
while(true)
{
```

```
    while(turn);
```

~~turn=0;~~ lock=1;

```
    //CS
```

~~turn=0;~~ lock=0;

```
    RS;
```

```
}
```

Question 2

```
lock=False;
```

```
while(true)
{
    while(lock!=False);
    CS
    lock=True;
    RS;
}
```

✓ M.E.

✗ Progress

✓ B.W.

✓ Starvation

```
while(true)
{
    while(lock!=True);
    CS
    lock=False;
    RS;
}
```

Question 3

`lock=False;` *True*

```
while(true)
{
    while(lock == False)
    {
        lock = True;
    }
    CS
    lock=False;
    RS;
}
```

X m.E.

✓ Progress

✓ B.W.

Starvation

Question 4

Boolean lock = True;

while(true)

{

while(lock)

{

CS

lock = False

}

lock = True;

RS;

}

P1 →

P2 →

X M.E.

✓ Progress

X B.W.

starvation

Starvation :-

for CPU scheduling algo

Process does not get
CPU for indefinite time

for synchronization

Process gets CPU but
not able to get critical
section.



Topic : wait() & signal()

wait(S)

```
{  
    while(S<=0);  
    S--;  
}
```

signal(S)

```
{  
    S++;  
}
```



Topic : Critical Section Solution

↓
mutual exclusion

$S = 1 \neq 1$

```
while(True)
{
    wait(S)
    ✓ C.S.
    signal(s)
}
```

#Q. If a counting semaphore S is initialized with value 2. There are 4 processes P1, P2, P3 and P4. Maximum number of processes which can be in critical section together if all process running the following code?

P1 P2

S = 2

```
while(True)
{
    wait(S)
    C.S.
    signal(s)
}
```



[NAT]



Ans = 4

#Q. If a counting semaphore S is initialized with value 2. There are 4 processes P1, P2, P3 and P4. Maximum number of processes which can run critical section if all process running the following code?

```
while(True)
{
    wait(S)
    C.S.
    signal(s)
}
```



[NAT]



#Q. Consider a counting semaphore S, initialized with value 21. What should be the value of S after executing 14 times P() and 7 time V() function on S?

$$21 - 14 + 7 = \underline{\underline{14}} \text{ Ans.}$$

#Q. Consider a semaphore S, initialized with value 37. Which of the following options gives the final value of S=12?

A ✓ Execution of 22 P() and 15 V() $37 - 22 + 15 = 30$

B ✓ Execution of 25 P() $37 - 25 = 12$

C ✓ Execution of 33 P() and 8 V() $37 - 33 + 8 = 12$

D ✓ Execution of 31 P() and 6 V() $37 - 31 + 6 = 12$

#Q. Consider a binary semaphore S, initialized with value 1. Consider 10 processes P1, P2 P10. All processes have same code as given below but, one process P10 has signal(S) in place of wait(S). If all processes to be executed only once, then maximum number of processes which can be in critical section together ?

```
process
{
    wait(S)
    C.S.
    signal(s)
}
```

Ans = 3





Topic : Solution



P1, P2,, P9

```
process
{
    wait(S)
    C.S.
    signal(s)
}
```

P10

```
process
{
    signal(S)
    C.S.
    signal(s)
}
```

$S = 1$

$P1 \Rightarrow S = 0$ in CS

$P10 \Rightarrow S = 1$ in CS

$P2 \Rightarrow S = 0$ in CS

[MCQ]



#Q. Consider a binary semaphore S, initialized with value 1. Consider 10 processes P1, P2 P10. All processes have same code as given below but, one process P10 has signal(S) in place of wait(S). If all processes to be executed only once, then maximum number of processes which can be in critical section together ?

```
while(True)
{
    wait(S)
    C.S.
    signal(s)
}
```

Ans = 10





Topic : Solution



$S = 1$

P1, P2,, P9

```
while(True)
{
    wait(S)
    C.S.
    signal(s)
}
```

P10

```
while(True)
{
    signal(S)
    C.S.
    signal(s)
}
```

$P1 \Rightarrow S = 0$ in CS

$P10 \Rightarrow S = 1$

$P2 \Rightarrow S = 0$ in CS

$P10 \Rightarrow S = 1$

$P3 \Rightarrow S = 0$ in CS

$P10 \Rightarrow S = 1$

ques 2 processes P1 and P2.

execution of P1 and P2 should like $\Rightarrow P1 \rightarrow P2 \rightarrow P1 \rightarrow P2 \dots$

write code using binary semaphore?

solⁿ

Binary semaphore $S1=1, S2=0$

P1

wait(S1)

=====
signal(S2)

P2

wait(S2)

=====
signal(S1)

ques 2 processes P1 and P2.

execution of P1 and P2 should like $\Rightarrow P2 \rightarrow P1 \rightarrow P2 \rightarrow P1 \rightarrow \dots$

write code using binary semaphore?

solⁿ

Binary semaphore $S1=0$ $S2=1$

P1

wait(S1)

=====
signal(S2)

P2

wait(S2)

=====
signal(S1)

Ques) 3 processes
print \Rightarrow cbacbacba.....

Binary $s_1 = 0$, $s_2 = 0$, $s_3 = 1$

P_1
wait(s_1)
print("a")
signal(s_3)

P_2
wait(s_2)
print("b")
signal(s_1)

P_3
wait(s_3)
print("c")
signal(s_2)

[MCQ]



#Q. Consider the following threads, T_1 , T_2 , and T_3 executing on a single processor, synchronized using three binary semaphore variables, S_1 , S_2 , and S_3 , operated upon using standard `wait()` and `signal()`. The threads can be context switched in any order and at any time. Which initialization of the semaphores would print the sequence BCABCABCA...?

[2022]

T1	T2	T3
<pre>while(true) { wait(S₃); print("C"); signal(S₂); }</pre>	<pre>while(true){ wait(S₁); print("B"); signal(S₃);} </pre>	<pre>while(true) { wait(S₂); print("A"); signal(S₁); }</pre>

$S_1 = 1$
 $S_2 = 0$
 $S_3 = 0$

A $S_1 = 1; S_2 = 1; S_3 = 1$

B $S_1 = 1; S_2 = 1; S_3 = 0$

C $S_1 = 1; S_2 = 0; S_3 = 0$

D $S_1 = 0; S_2 = 1; S_3 = 1$



2 mins Summary

Topic

Semaphore

Topic

Spin Lock

Topic

Producer Consumer Problem



Happy Learning

THANK - YOU

