

# 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; A~~

*while(true)*

{

*while(turn);*

*turn=1;*

*//CS*

*turn=0;*

*RS;*

}

*lock = false*

✗ m.e.

✓ Progress

✗ B.w.

*while(true)*

{

*while(turn);*

*turn~~lock~~=1;*

*//CS*

*turn~~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;

}

✗ m.e.

✓ Progress

✓ B.w.

Starvation

## Question 4

```
Boolean lock= True;  
while(true)  
{  
    while(lock)  
    {  
        CS  
        lock = False  
  
        p1 → }  
        p2 → lock=True;  
        RS;  
    }  
}
```

✗ M.E.

✓ Progress

✗ 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)           signal(S)  
{                {  
    while(S<=0);      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 = ~~240~~

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

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)
}
```

#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)
}
```

$$S = 1$$

$$P1 \Rightarrow S = 0 \quad \text{in CS}$$

$$P10 \Rightarrow S = 1 \quad \text{in CS}$$

$$P2 \Rightarrow S = 0 \quad \text{in CS}$$

#Q. Consider a binary semaphore S, initialized with value 1. Consider 10 processes P<sub>1</sub>, P<sub>2</sub> ... P<sub>10</sub>. All processes have same code as given below but, one process P<sub>10</sub> 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

P1, P2, ...., P9

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

P10

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

S = 1

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  $P_1$  and  $P_2$ .

execution of  $P_1$  and  $P_2$  should like  $\Rightarrow P_1 \rightarrow P_2 \rightarrow P_1 \rightarrow P_2 \dots$

write code using binary Semaphore?

---

soln Binary semaphore  $S_1 = 1, S_2 = 0$

$P_1$

wait( $S_1$ )

≡

signal( $S_2$ )

$P_2$

wait( $S_2$ )

≡

signal( $S_1$ )

ques) 2 processes  $P_1$  and  $P_2$ .

execution of  $P_1$  and  $P_2$  should like  $\Rightarrow P_2 \rightarrow P_1 \rightarrow P_2 \rightarrow P_1 \rightarrow \dots$

write code using binary Semaphore?

soln Binary semaphore  $S_1 = 0$   $S_2 = 1$

$P_1$

wait( $S_1$ )

≡

signal( $S_2$ )

$P_2$

wait( $S_2$ )

≡

signal( $S_1$ )

ques)

3 processes

print  $\Rightarrow$  cbacbaca - - -

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 BCABCABC...?

[2022]

| T1  | T2  | T3   |
|---|---|--|
| <pre>while(true) ){     wait(S<sub>3</sub>);     print("C");     signal(S<sub>2</sub>); }</pre> | <pre>while(true){     wait(S<sub>1</sub>);     print("B");     signal(S<sub>3</sub>); }</pre> | <pre>while(true) {     wait(S<sub>2</sub>);     print("A");     signal(S<sub>1</sub>); }</pre> |

$$\begin{aligned}S_1 &= 1 \\S_2 &= 0 \\S_3 &= 0\end{aligned}$$

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