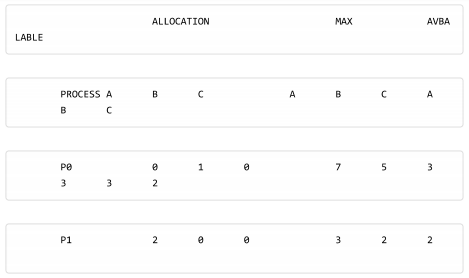
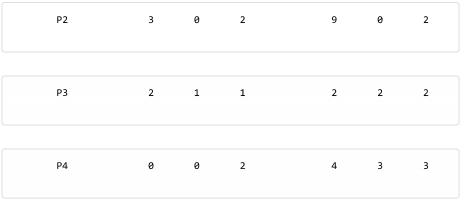
1. Which are the CORRECT **conditions** of **critical section?**

Select one:

* mutual exclusion, progressive, bounded waiting
* **mutual exclusion, bounded waiting, progress mutual**
* exclusion, protection, bounded waiting mutual
* exclusion, protection, bounded using

2. Given the following system information:





Select one:

* FINISH=(F, F, T, T, F) WORK=(7, 4, 3)
* **FINISH=(F, T, F, T, F) WORK=(7, 4, 3)**
* FINISH=(F, T, T, F, F) WORK=(7, 4, 3)
* FINISH=(F, F, T, F, F) WORK=(7, 4, 3)

**3.** Given the code for **bounded-buffer problem:**

Write process P:

do {

wait(empty);

wait(mutex);

Write (item);

signal(mutex);

signal(full);

} while (TRUE);

Read process Q:

do {

wait(full);

wait(mutex);

Read(item);

signal(mutex);

signal(empty);

} while (TRUE);

What will be the problem if the initialized value of the full variable is 1?

Select one:

* **the reader can read an invalid value**
* the writer process can not run
* no problem at all
* the reader process can not run

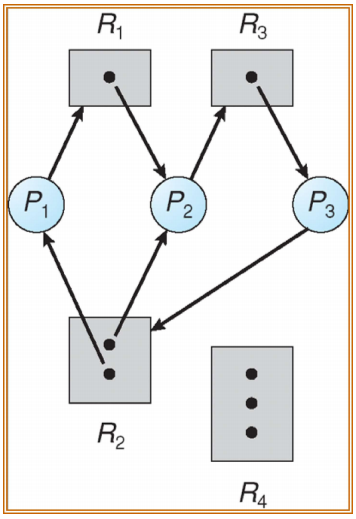
4. Which is **NOT** an **implementation** of **critical section**?

* Monitor
* Semaphore
* Peterson's solution
* **Condition**

**5.**  Which is **not Interprocess Communication**?

* A process sends data to another one via a pipe (in a UNIX like platform)
* A process shares a semaphore with another process.
* A web browser views a webpage from a web server.
* **A process reads data from a file.**

**? 6.** Given the following resource allocation graph, which is correct?

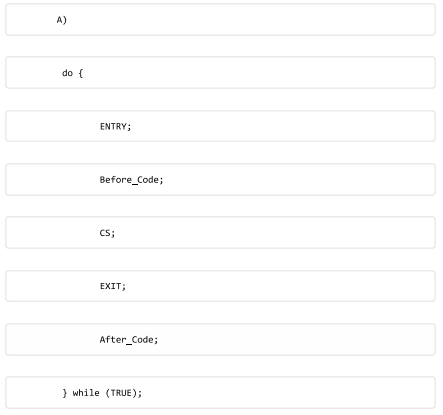


* There is no cycle in the graph.
* There is one cycle in the graph: P2-R3-P3-R2-P2
* **There are two cycle in the graph: P1-R1-P2-R2-P3-R3-P1 and P2-R3-P3-R2-P2**
* There is one cycle in the graph: P1-R1-P2-R2-P3-R3-P1

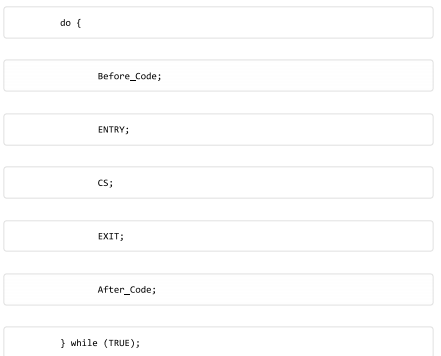
7. Which is **INCORRECT** about **Inter-process Communication (IPC)**?

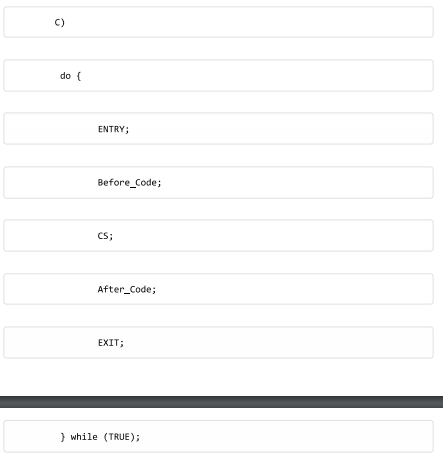
* **IPC can only be used among processes in the same system**
* In uni-programming operating system there may be NO need of local IPC
* Examples of IPC mechanism in Linux are message queue, semaphore, shared memory, …
* The IPC mechanism in different operating systems may be different

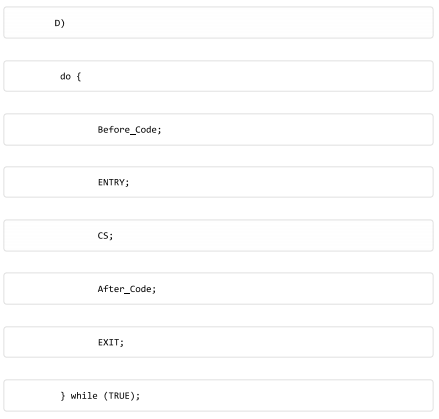
8. Which is the correct **protocol** to run a **critical section named CS**?











#### **?9**. Given the following resource allocation graph, which is correct about **resource request algorithm**?

* The goal of the algorithm is to avoid a process requesting a resource which results in a cycle.
* **The goal of the algorithm is to avoid a process claiming a resource which results in a cycle.**
* The goal of the algorithm is to avoid changing a claim edge into an assignment edge which results in a cycle.
* The goal of the algorithm is to avoid changing a request edge into an assignment edge which results in a cycle.

#### ? 10. Given the following system information:

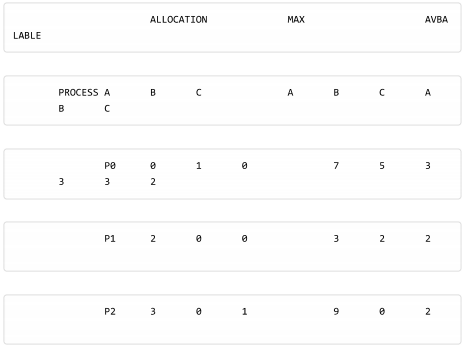


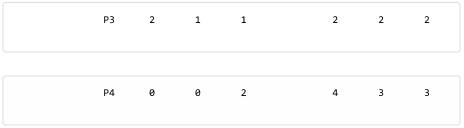


Which is the **correct** value of **FINISH** and **WORK** vectors during the **running of Banker's algorithm?**

* **FINISH=(F, F, F) WORK =(3)**
* FINISH=(F, F, T) WORK =(3)
* FINISH=(F, F, F) WORK =(4)
* FINISH=(T, F, F) WORK =(3)

#### ? 11. Given the following system information, and process PO requests (0, 2, 0) more resources:

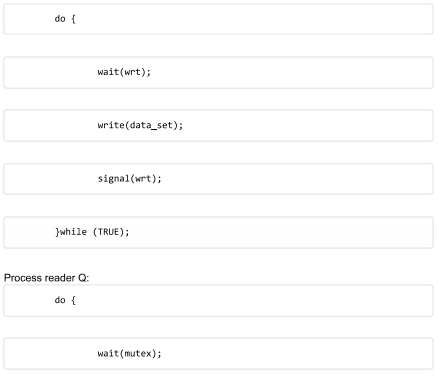




Which is the correct value of FINISH and WORK vectors during the running of Banker's algorithm which is called in the Resource-Request algorithm (to avoid deadlock)?

* FINISH=(F, F, T, F, F), WORK=(5, 3, 3)
* **FINISH=(F, F, F, T, F), WORK=(5, 2, 3)**
* FINISH=(F, F, F, F, T), WORK=(5, 3, 2)
* FINISH=(F, T, F, F, F), WORK=(5, 1, 2)

12:



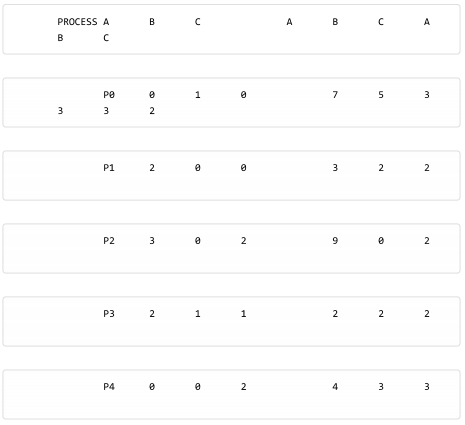


Which is the **purpose** of **mutex variable**?

* To safely access the data\_set
* To safely access the wrt variable
* **To safely access the readcount variable**
* We may remove this variable without affecting the program

#### 13. Given the following system information, and process P1 requests (1, 0, 2) more resources:

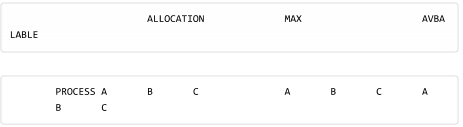


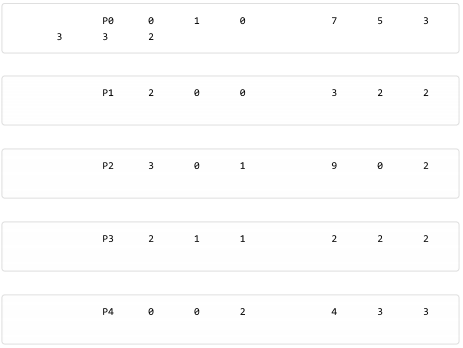


Which is the correct value of FINISH and WORK vectors during the running **of Banker's algorithm** which is called in the **Resource-Request algorithm (to avoid deadlock)**?

* FINISH=(F, F, F, T, T), WORK=(10, 5, 3)
* **FINISH=(T, T, T, T, F), WORK=(10, 5, 5)**
* FINISH=(F, T, F, T, F), WORK=(10, 4, 2)
* FINISH=(F, T, F, T, F), WORK=(10, 5, 3)

#### 14. Given the following system information, and process PO requests (0, 2, 0) more resources:

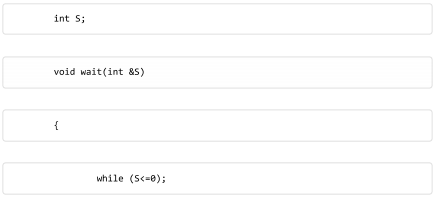


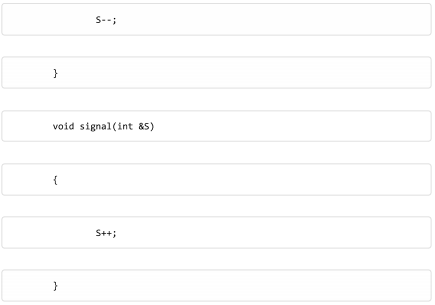


Which is the **correct** value of FINISH and WORK **vectors during the running of Banker's algorithm** which is called in the **Resource-Request algorithm (to avoid deadlock)?**

* **FINISH=(T, T, T, T, T), WORK=(10, 5, 6)**
* FINISH=(T, T, F, T, T), WORK=(10, 5, 7)
* FINISH=(T, T, T, T, T), WORK=(10, 5, 7)
* FINISH=(F, T, T, F, T), WORK=(10, 5, 6)

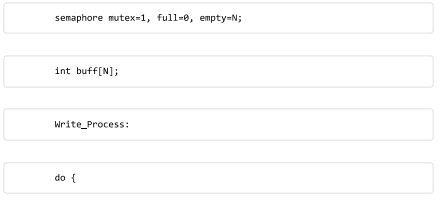
15. Which is **incorrect** about the following **code**?





* The while loop in the wait() procedure will cause the calling process to wait if the resource is in use.
* This is an implementation of semaphore.
* When the resource is in use, and another process call wait(), the CPU is waste to run the while loop.
* **The while loop can be removed.**

**16.** Given the two bellow processes sharing three semaphores full, empty, mutex, and a buffer buff having initial N empty slots:

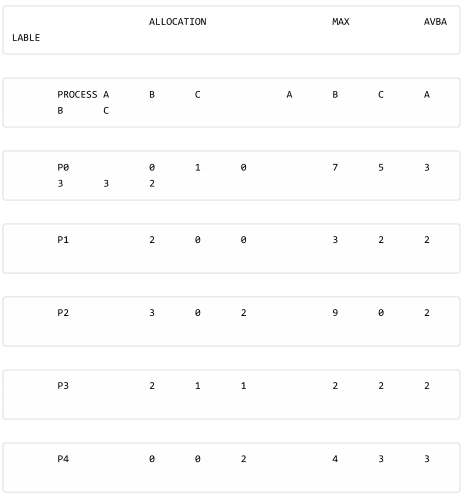




Which is **correct** about the **semaphore mutex?**

* mutex is used to force Read Process to wait until Write Process write data into buff.
* **mutex is initially set to 1. wait(mutex); and signal(mutex); operators follow a protocol to execute a critical section, i.e. to allow at most 1 process to access the buff.**
* Both semaphore mutex and full are used to ensure at most 1 process can access the buff at a time.
* mutex is used to force Write Process to write at most N data items into buff.

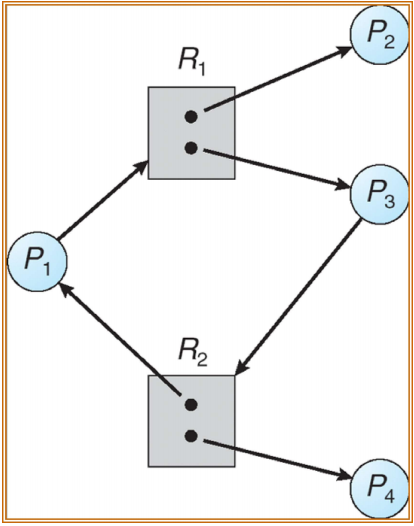
#### ? 17. Given the following system information:



Which is the correct value of FINISH and WORK vectors during the running of **Banker's algorithm?**

* FINISH=(F, T, T, F, F) WORK=(7, 5, 3)
* FINISH=(F, F, T, F, T) WORK=(7, 5, 3)
* **FINISH=(T, T, F, T, F) WORK=(7, 5, 3)**
* FINISH=(F, F, T, T, T) WORK=(7, 5, 3)

#### ? 18. Given the following resource allocation graph, provide the name of the edge from R1 to P2 (Assignment, Claim,or Request)?



#### 19 Given the following system information:

ALLOCATION REQUEST AVBALABLE

PROCESS A B C A B C A B C

PO 0 1 0 0 0 0 0 0 0

P1 2 0 0 2 0 2

P2 3 0 3 0 0 0

P3 2 1 1 1 0 0

P4 0 0 2 0 0 2

Which is correct value of FINISH and WORK vectors during the running of the deadlock detection algorithm?

Select one:

FINISH=(F, T, T, T, F), WORK=(5, 1, 2)

FINISH=(F, F, T, T, T), WORK=(5, 1, 2)

FINISH=(F, F, T, T, T), WORK=(5, 2, 3)

**FINISH=(T, T, T, F, F), WORK=(5, 1, 3)**

20. Which is **not Interprocess Communication**?

* A process writes data to a named pipe for another process to read (in a UNIX like platform)
* **A process reads data from a file.**
* A process shares a semaphore with another process.
* A process connects to a web server via socket.

22. cho [Bounded-buffer]

Which is the best explanation why semaphore full is set to 0?

Select one:

* **To force Read\_Process to wait until Write\_Process writes data into buff.**
* To force at most 1 process can access the buff at a time.

23. Given two code snippets:

|  |  |
| --- | --- |
| wait(mutex);  CS1;  wait(mutex); | wait(mutex);  CS2;  signal(mutex); |

Which is the problem when they run concurrently?

Select one:

* Two process will be blocked forever if snippet 1 run and execute the second wait(mutex) instruction first
* Two process will be blocked forever if snippet 2 run and execute the wait(mutex) instruction first
* Snippet 2 never be blocked forever
* No problem at all

24. Which is CORRECT about the goal of mutual exclusion condition of critical section?

Select one:

It makes the implementation of operating system simpler

It supports the priority of processes

It tries to utilize the shared resource more effectively

It ensures the correct use of the shared resource

25. Given the following code, with the globally shared variable int hit. Which is the critical section when multiple instances of the code are running in the system?

1. while (true) {

2. int val;

3. waitForNewRequest();

4. if(found){

5. hit+=1;

6. val=hit;

7. }

8. Respond(val);

9. }

Select one:

* The critical section should be the lines marked: 2, 3, 4, 5, 6, 7, 8
* The critical section should be the lines marked: 2, 3, 4, 5, 6, 7
* The whole code is the critical section
* The critical section should be the lines marked: 4, 5, 6, 7

27. Which is the BEST definition of deadlock?

Select one:

* Deadlock is a process waiting for a certain resource
* Deadlock is a set of processes, in which each process held some resources and waits for other resources that are held by another process. There is a circular wait among those processes.
* Deadlock is a set of processes waiting for a shared resource
* Deadlock is a set of processes, in which each process held some resources and waits for other resources that are held by another process

28. Given the following information of the system.

Pro Allocation Max Available

A A A A

P0 5 10 3

P1 2 4

P2 2 9

If P2 requests 1 more resource A, the resource request algorithm will produce:

Select one:

* P2 P1 P0 is a safe sequence, the request can be granted immediately
* P0 P1 P2 is a safe sequence, the request can be granted immediately
* P1 P0 P2 is a safe sequence, the request can be granted immediately
* The process is forced to wait

30. Given the following code, with the globally shared variable int buf, and count() is a function. Which is correct when multiple instances of the code are running in the system?

*while (true) {*

*//Some code*

*int val=buf;*

*val += count();//A time consuming function*

*buf=val*

*//Some other code*

*}*

Select one:

* The globally shared variable buf will always get the correct value.
* The globally shared variable buf will always get an incorrect value when there are more than 2 instances of the code running.
* The globally shared variable buf will always get an incorrect value when only 2 instances of the code running.
* The globally shared variable buf can get an incorrect value when there are more than 1 instance of the code running.

33.

PROCESS ALLOCATION MAX AVBALABLE

A B C A B C A B C

P0 0 1 0 7 5 3 3 3 2

P1 2 0 0 3 2 2

P2 3 0 1 9 0 2

P3 2 1 1 2 2 2

P4 0 0 2 4 3 3

Which is the correct value of FINISH and WORK vectors during the running of Banker's algorithm which is called in the Resource-Request algorithm (to avoid deadlock)?

Select one:

FINISH=(F, F, F, F, T), WORK=(4, 3, 2)

FINISH=(F, F, F, F, T), WORK=(4, 3, 2)

FINISH=(F, T, F, F, F), WORK=(4, 3, 2)

FINISH=(F, F, F, T, F), WORK=(4, 3, 2)

34. There are two processes below running concurrently:

Process A:

{

...

Lock\_file(F1);

...

Open\_file(F2);

...

Unlock(F1);

}

Process B:

{

...

Lock\_file(F2);

...

Open\_file(F1);

...

Unlock(F1);

}

Suppose the Lock\_file() system call will force the file to be used by only one process. In other words, later call to open\_file() will cause the calling process to wait. When does the deadlock occur?

A locks F1 and F2 and B waits for F2

A locks F1 and waits for F2

A locks F1 and waits for F2 and B locks F2 and waits for F1

A locks F2 and waits for F1 and B waits for F1

36. Which is incorrect about the following code?

[wait and signal]

Select one:

The while loop in the wait() procedure will cause the calling process to wait if the resource is in use.

When the resource is in use, and another process call wait(), the CPU is waste to run the while loop.

This is an implementation of semaphore.

The while loop can be removed.

Given the following code, with the globally shared variable int hit. Which is incorrect

when multiple instances of the code are running in the system?

1. while (true) {

2. int val;

3. waitForNewRequest();

4. if(found){

5. hit+=1;

6. val=hit;

7. }

8. Respond(val);

9. }

Select one:

The critical section should be the lines marked: 4, 5, 6, 7, 8

The critical section should be the lines marked: 4, 5, 6, 7

There is a race condition in the code.

There is a critical section in the code.

38. Given the following system information, and process P0 requests (0, 2, 0) more resources:

PROCESS ALLOCATION MAX AVBALABLE

P0 0 1 0 7 5 3 3 3 2

P1 2 0 0 3 2 2

P2 3 0 1 9 0 2

P3 2 1 1 2 2 2

P4 0 0 2 4 3 3

Which is the correct value of FINISH and WORK vectors during the running of Banker's algorithm which is called in the Resource-Request algorithm (to avoid deadlock)?

Select one:

FINISH=(F, T, F, T, F), WORK=(7, 2, 3)

FINISH=(F, F, F, F, T), WORK=(5, 3, 2)

FINISH=(F, T, F, T, F), WORK=(5, 2, 3)

FINISH=(F, F, T, F, T), WORK=(7, 3, 3)

41. Given the two bellow processes sharing three semaphores full, empty, mutex, and a buffer buff having initial N empty slots:

[Bounded-buffer]

Which is correct about the method to satisfy the constraint: the buff can be accessed (read or write) at most by 1 process at a time?

Select one:

* The semaphore full to ensure this constraint.
* Both semaphore mutex and full are used to ensure this constraint.
* The semaphore mutex is used to ensure this constraint.
* The semaphore empty is used to ensure this constraint.

42. Given the two bellow processes sharing three semaphores full, empty, mutex, and a buffer buff having initial N empty slots:

[Bounded-buffer]

Which is correct about the semaphore mutex?

* mutex is used to force Write\_Process to write at most N data items into buff.
* mutex is used to force Read\_Process to wait until Write\_Process write data into buff.
* Both semaphore mutex and full are used to ensure at most 1 process can access the buff at a time.
* mutex is initially set to 1. wait(mutex); and signal(mutex); operators follow a protocol to execute a critical section, i.e. to allow at most 1 process to access the buff.

43 . Given the following information of the system.

Pro Allocation Max Available

P0 0 0 1 0 0 1 1 5 2

P1 1 0 0 1 7 5

P2 1 3 5 2 3 5

P3 0 6 3 0 6 5

P4 0 0 1 0 6 5

If P1 requests resource A B C (0 4 2), the resource request algorithm will produce:

PO P2 P4 P3 P1 is a safe sequence, the request can be granted immediately

The system is not in the safe state, the request is not granted

PO P1 P2 P3 P4 is a safe sequence, the request can be granted immediately

PO P2 P1 P3 P4 is a safe sequence, the request can be granted immediately

45. Given the following system information, and process PO requests (0, 2, 0) more resources:

PROCESS AVBALABLE ALLOCATION MAX

PO 0 3 0 7 5 3 2 1 0

P1 3 0 2 3 2 2

P2 3 0 2 9 0 2

P3 2 1 1 2 2 2

P4 0 0 2 4 3 3

Which is the final value of FINISH and WORK vectors of Banker's algorithm which is called in the Resource-Request algorithm (to avoid deadlock)?

FINISH=(F, F, F, T, F), WORK=(2, 1, 0)

FINISH=(F, F, F, F, F), WORK=(2, 1, 0)

FINISH=(F, T, F, F, F), WORK=(2, 1, 0)

FINISH=(F, F, F, F, F), WORK=(2, 3, 0)

47. Which is incorrect about Semaphore?

Semaphore includes an integer and two atomic operators.

We can manipulate the semaphore's integer directly.

signal (or V) operator is corresponding to EXIT in the protocol of a critical section

wait (or P) operator is corresponding to ENTRY in the protocol of a critical section

52. Given the following system information, and process P2 requests 1 more tape:

PROCESS ALLOCATION MAX AVAILABLE

PO 5 10 3

P1 2 4

P2 2 9

Which is correct?

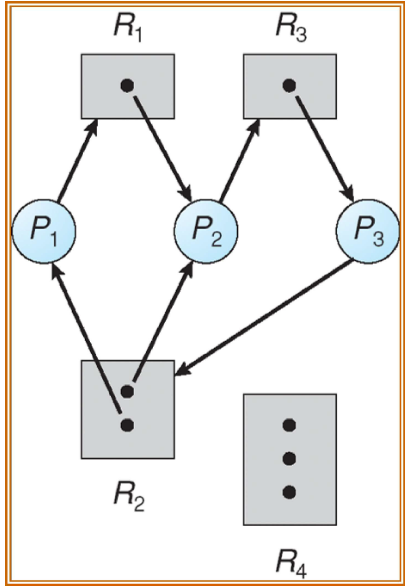
The banker's algorithm results in FINISH=(F, T, F), that means it is not safe to grant resource for P2.

The banker's algorithm results in FINISH=(T, F, F), that means it is not safe to grant resource for P2.

The banker's algorithm results in FINISH=(T, T, T), that means it is safe to grant resource for P2.

The banker's algorithm results in FINISH=(F, T, T), that means it is not safe to grant resource for P2.

53. Given the following resource allocation graph, which is correct?



There is no cycle in the graph, there is no deadlock.

There is no deadlock.

There are cycles in the graph, thus there is a deadlock.

There is a multiple instance resource type in the graph, thus there is no deadlock in the system.

54.

PROCESS ALLOCATION MAX AVAILABLE

PO 5 10 3

P1 2 4

P2 2 9

Which is the correct value of FINISH and WORK vectors during the running of Banker's algorithm?

FINISH=(F, F, F) WORK =(3)

FINISH=(T, F, F) WORK =(3)

FINISH=(F, F, T) WORK =(3)

FINISH=(F, F, F) WORK =(4)

56. Which is INCORRECT about Resource Allocation Graph (RAG)?

An edge from a resource vertex to a process vertex represents an instance of the resources is allocated to the process

A rectangle represents a process

A circle represents a process

A request edge is from a process vertex to a resource vertex

60. Given two code snippets:

...

wait(mutex);

//Critical section

signal(mutex);

snippet 2

…

signal(mutex);

//Critical section

wait(mutex);

…

Which is the problem of the above code snippets when they run concurrently?

Select one:

No problem at all

No process can enter its critical section

The mutual exclusion condition may be violated

No process can exit its critical section

-------------------------Slide------------------------

*1. Mutual Exclusion*

*o If a process is in its critical section, then no other*

*processes can be in their critical sections*

*2. Progress*

*o If no process is in its critical section*

*o other processes waiting to enter their critical section,*

*o then the selection of the process to enter the critical*

*section cannot be postponed indefinitely*

*3. Bounded Waiting*

*o No process has to wait indefinitely to enter its critical*

*section*

Which is the purpose of the first condition implementation of critical section CS (**Mutual Exclusion )** ?

* It supports the priority of process
* **? It ensures** the correct use of the shared resource
* It tries to utilize the shared resource effectively
* It makes the implementation of OS simpler

Which is the consequence of the second condition implementation of critical section CS (**Progress**) ?

**A. It reduces the waiting time of requested processes**

B. It ensures the correct use of the shared resource

C. It supports the priority of processes

D. It makes the implementation of OS simpler

Which is the consequence of the 3rd (third condition implementation of critical section CS (**Bounded Waiting)**?

A. It supports the priority of processes

B. It ensures the correct use of the shared resource

C. It utilizes the shared resource effectively

**D. It makes sure no process can never enter its critical section**

Which is the correct conditions of critical

section?

A. mutual exclusion, protection, bounded using

B. mutual exclusion, protection, bounded waiting

C. mutual exclusion, progressive, bounded

waiting

**D. mutual exclusion, bounded waiting, progress**

Which is the correct purpose the 2nd condition (second condition ) of critical section ?

or the second condition implementation of critical section CS (**Progress**) ?

**A. maximize CPU utilization**

B. maximize the shared resource utilization

C. maximize disk utilization

D. maximize RAM utilization

Which is the consequence of the 3rd condition (third condition)implementation of critical section CS (**Bounded Waiting**) ?

A. It supports the priority of processes

B. It ensures the correct use of the shared resource

**C. It ensures the relative fairness of processes to use the shared resource**

D. It utilizes the shared resource effectively

Program P i :

do {

flag[i] = TRUE;

turn = j;

while (flag[j] && turn == j) ;

**CS i ;**

flag[i] = FALSE;

REMAIN i ;

} while (1);

Which code snippet is Enter\_Section?

**A. flag[i] = TRUE;**

**turn = j;**

**while (flag[j] && turn == j) ;**

B. flag[i] = TRUE;

while (flag[j] && turn == j) ;

C. flag[i] = TRUE;

turn = j;

D. turn = j;

while (flag[j] && turn == j) ;

When counting semaphores are suitable to use?

A. When 2 processes share a single variable/resource

B. When 3 processes share a single variable/resource

C. When n processes share a single variable/resource

**D. When n processes share m variables/resources of the same type**

|  |  |
| --- | --- |
| Write process P:  do {  wait(empty);  wait(mutex);  Write (item);  signal(mutex);  signal(full);  } while (TRUE); | Read process Q:  do {  wait(full);  wait(mutex);  Read(item);  signal(mutex);  signal(empty);  } while (TRUE); |

Which is the initialized value of the **full** variable in the above algorithm?

* -1
* **0**
* 1
* NULL

What will be the problem if the initialized value of the full variable is 1?

* no problem at all
* the writer process can not run
* the reader process can not run
* the reader can read an invalid value

Why do we need readcount variable?

* We may remove this variable
* To make sure there is one reader at a time
* To make sure no readers are reading
* To make sure no readers are reading before writing

|  |  |
| --- | --- |
| Process writer P w :  do {  wait(wrt);  write(data\_set);  signal(wrt);  }while (TRUE); | Process reader P r :  do {  wait(mutex);  readcount++;  if (readcount ==1) wait(wrt);  signal(mutex);  read(data\_set);  wait(mutex);  readcount--;  if (readcount ==0) signal(wrt);  signal(mutex);  } while (TRUE); |

Which is the initialized value of the **readcount** variable in the above algorithm?

* -1
* 0
* 1
* NULL

Which is the purpose of **mutex** variable?

* To safely access the data\_set
* We may remove this variable without affecting the program
* To safely access the readcount variable
* To safely access the wrt variable

Which is the initialized value of the **mutex** variable in the above algorithm?

* -1
* 0
* 1
* NULL

Which is the purpose of wrt variable?

* To safely access the mutex variable
* To safely write the data\_set
* To safely write the readcount variable
* To safely read the data\_set

Which is the initialized value of the wrt variable in the above algorithm?

* -1
* 0
* 1
* NULL

Code of philosopher i:

do {

wait(chopstick[i]);

wait(chopstick[(i+1)%5];

Eat(i);

signal(chopstick[i]);

signal(chopstick[(i+1)%5];

Think(i);

} while (TRUE);

What value chopstick[i] is initialized?

* 1
* 2
* 0
* 5

Is there any problem with the solution?

* No problem
* Only one philosopher can eat at a time
* Only three philosophers can eat at a time
* No philosopher could eat in case each takes a chopstick and waits for the second one

Which of the following is incorrect about the solution to the above problem?

* No solution available
* Create an order of philosophers to eat
* Create an order of philosophers to think
* Allow at most 4 philosophers to request to eat at a time

|  |  |
| --- | --- |
| Snippet 1  ...  wait(mutex);  //Critical section  signal(mutex);  ... | Snippet 2  ...  signal(mutex);  //Critical section  wait(mutex);  ... |

What is the problem of the two code snippets?

* Snippet 1 has problem
* Snippet 2 has problem
* Both snippets have problem
* No problem at all

Which is the problem of the incorrect use of semaphore in the above code snippet?

* No process can enter its critical section
* No problem at all
* The mutual exclusion condition may be violated
* No process can exit its critical section

|  |  |
| --- | --- |
| Snippet 1  ...  wait(mutex);  CS 1 ;  wait(mutex);  ... | Snippet 2  ...  wait(mutex);  CS 2 ;  signal(mutex);  ... |

Which of the two code snippets has problem?

* Snippet 1 has problem
* Snippet 2 has problem
* Both snippets have problem
* No problem at all

Which is the consequence of the above problem?

* One process will be blocked
* There will be a deadlock
* No consequences if only two processes are involved
* No consequences

|  |  |
| --- | --- |
| Process P 1  ...  wait(S);  wait(Q);  ...  signal(S);  signal(Q); | Process P 2  ...  wait(Q);  wait(S);  ...  signal(Q);  signal(S); |

What is the problem of the above two processes?

* There is deadlock
  + if P 1 got S and waits for Q and
  + P 2 got Q and waits for S
* The exclusive condition is violated
* The order of semaphore calls is incorrect
* No problem at all

Which is correct about race condition?

* Happen even when there is only once process
* Happen when multiple processes use a shared resource concurrently
* Happen when multiple processes use a resource sequentially
* Happen when there are multiple processes in the system

Which is incorrect about the Peterson’s solution?

* It satisfies all the conditions of critical section
* It is easy to control even the number of processes is above 2
* It is difficult to control
* It is complicated when the number of processes is above 2

Which of the following is the most correct about critical section?

* A code snippet that operates on a global variable
* A code snippet that operates on a resource
* A code snippet that operates on a global resource
* A code snippet that operates on a shared resource

How many conditions for resolving critical section are there ?

* 1
* 2
* 3
* 4

Which is **incorrect** about the **conditions of critical section**?

* The progress condition utilizes the resource effectively
* The exclusive condition removes race condition
* The exclusive condition ensures processes to use a shared resource sequentially
* The bounded waiting condition allows a process to use a shared resource several consecutive times

Which is the purpose of the second condition of critical section?

* It reduces the waiting time of requested processes
* It ensures the correct use of the shared resource
* It makes the algorithm more complicated to implement
* It makes the algorithm less complicated to implement

Which is the purpose of the third condition of critical section?

* It supports the priority of processes
* It ensures the correct use of the shared resource
* It utilizes the shared resource effectively
* It makes sure no process is in its critical section forever

Which is **incorrect** about the semaphore?

* Semaphore is an implementation of critical section
* Semaphore does not guarantee the onditions of critical section
* A semaphore usually includes an integer variable
* Semaphore has atomic operators

How many types the semaphore are there?

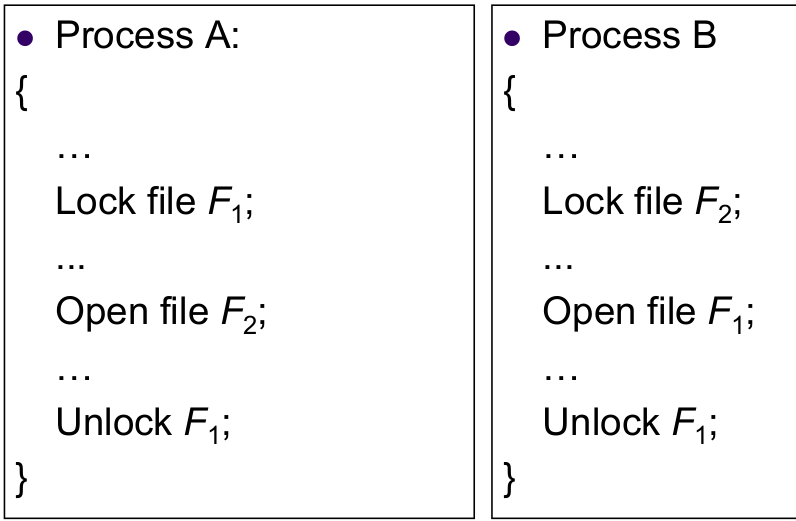
* 1
* 2
* 3
* 4

Which of the following is correct about counting semaphore?

* The value of the semaphore is 0 or 1
* The same as binary semaphore
* The value of the semaphore variable can be above 1
* The value of the semaphore variable can never be below 0

? Which of the following is the most suitable use for counting semaphore?

* Use for shared resources with a single instance
* Use for shared resources with 2 instances
* Use for shared resources with any instances
* Use for shared resources with multiple instances



When does the deadlock happen?

A. A gets F1 and waits for F2

B. A gets F2 and waits for F1 and

B waits for F1

C. A gets F1 and waits for F2 and

B gets F2 and waits for F1

D. A gets F1 and F2 and

B waits for F2

When does a deadlock happen?

A. any of the 4 conditions occur

B. any two of the 4 conditions occur

C. any 3 of the 4 conditions occur

D. all the 4 conditions occur

Which of the following is correct about the Work variable in the algorithm?

⦁ It stores the available resources when each process finishes

⦁ It is a redundant variable

⦁ It stores the state of the system

⦁ It stores possible resources for each process

Which of the following is the most correct about banker’s algorithm?

⦁ ?it detects the state of the processes

⦁ it detects the deadlock state of the system

⦁ ?it detects the safe sequence of the system

⦁ it detects the available resources

Which of the following is correct about resource-request algorithm?

⦁ it detects the unsafe state of the system

⦁ it detects the deadlock state of the system

⦁ it detects the safe sequence of the system

⦁ it detects the safe sequence of the system if the request is granted

Which of the following is correct about deadlock detection algorithm?

⦁ it only detects the unsafe state of the system

⦁ all the processes in the system are in the deadlock when it detects a deadlock

⦁ it can only detect the deadlock not the processes involved in the deadlock

⦁ it can detect deadlock as well as the involved processes

Which are the conditions for a dead lock to happen?

A. circular wait, no-preemption, hold and wait, mutual exclusion

B. circular wait, preemption, hold and wait, mutual exclusion

C. circular wait, no-preemption, hold, mutual exclusion

D. circular wait, no-preemption, hold and wait, mutual wait

Which is the most correct about deadlock prevention?

⦁ ensures the system will never enter a deadlock

⦁ ensures at least one of the four deadlock conditions will never occur

⦁ allows the system enter a deadlock and then recovers

⦁ detects the deadlock state and recovers

Which is NOT a deadlock handling method?

⦁ deadlock avoidance

⦁ deadlock prevention

⦁ deadlock prediction

⦁ deadlock ignorance

Which is the most correct about deadlock avoidance?

⦁ ensures the system will never enter a deadlock

⦁ ensures at least one of the four deadlock conditions will never occur

⦁ allows the system enter a deadlock and then recovers

⦁ ensures the circular wait condition will never occur

Which is the most correct about safe state?

⦁ the state of the system

⦁ the state of a process

⦁ the state of the system he running sequence (order) of processes that ensures the system does not enter a deadlock

⦁ the state that ensures a process can safely run

Which is correct about deadlock?

⦁ a deadlock will surely occur if the system is in unsafe state

⦁ a deadlock may occur even when the system is in safe state

⦁ there is only one method for handling deadlock

⦁ deadlock handling is available in all OSes

Which is the correct method for recovering from a deadlock?

⦁ restart the system abort each process involved in the deadlock

⦁ until the deadlock disappears

⦁ provide more resources for the system

⦁ ignore the deadlock

? Which is the criteria for selecting a process to abort when a deadlock occurs?

⦁ the number of processes in the system

⦁ the resources a process needs to complete

⦁ the available resources the system has

⦁ the available RAM