OPERATING SYSTEMS (CS F372)
SEM I 2021-2022
QUIZ I QUESTIONS & SOLUTIONS

QUESTION: Select True or False. The second argument of waitpid() system call can be NULL.
A) True
B) False
ANSWER: A

QUESTION: Select True or False. The value of the argument pid in the waitpid() system call cannot have a value less than 0.
A) True
B) False
ANSWER: B

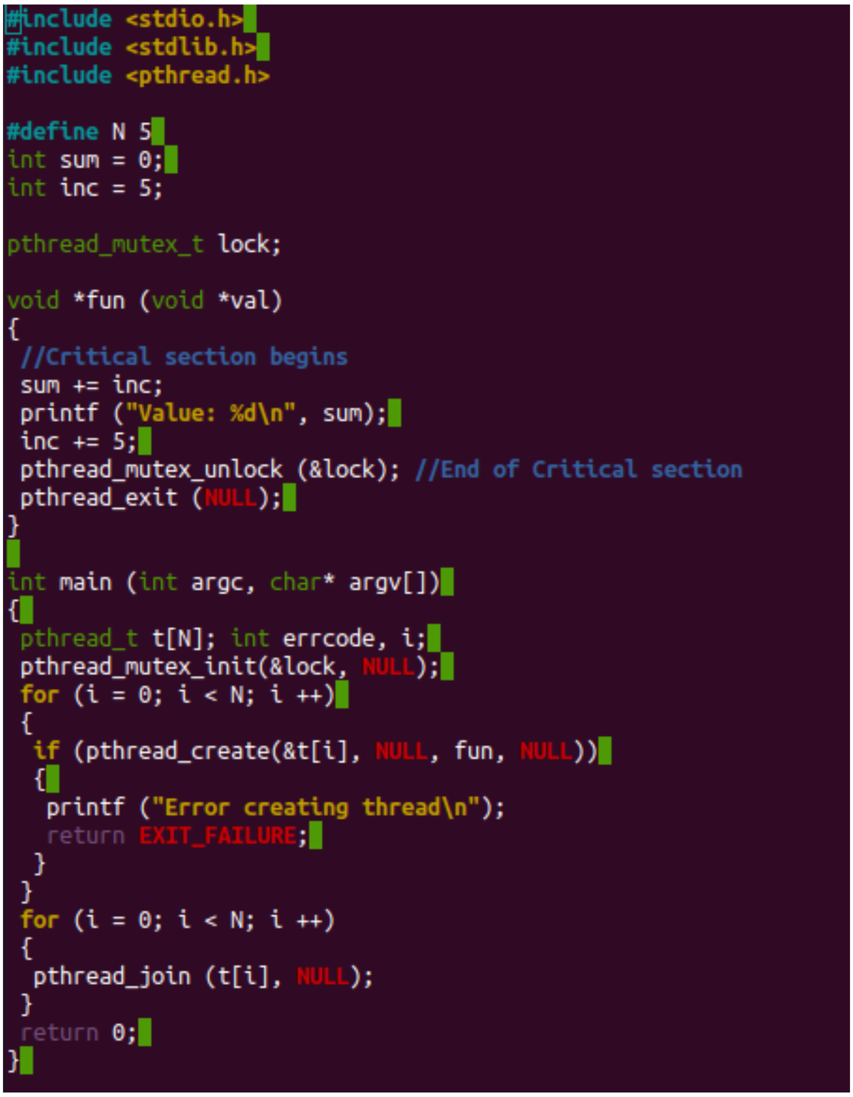
QUESTION: Select True or False. The thread specified in the second argument of pthread_join() function must be joinable.
A) True
B) False
ANSWER: A or B

QUESTION: Select True or False. A deadlock situation can never arise while the pthread_join() function is used or executed.
A) True
B) False
ANSWER: B

QUESTION: Which of the following is correct?
A) sem_wait() decrements and sem_post() increments the semaphore pointed to by the argument sem
B) sem_wait() increments and sem_post() decrements the semaphore pointed to by the argument sem
C) None of the options is correct
D) Both sem_wait() and sem_post() increment the semaphore pointed to by the argument sem
ANSWER: A

QUESTION: Which of the following is correct?
A) Both sem_wait() and sem_post() return 0 on success
B) sem_destroy() will fail if the sem argument is not a valid semaphore
C) sem_wait() decrements the semaphore pointed to by the argument sem
D) All options are correct
ANSWER: D

QUESTION: What will be printed on the screen as a result of the output of the following program?



```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define N 5
int sum = 0;
int inc = 5;

pthread_mutex_t lock;

void *fun (void *val)
{
 //Critical section begins
 sum += inc;
 printf ("Value: %d\n", sum);
 inc += 5;
 pthread_mutex_unlock (&lock); //End of Critical section
 pthread_exit (NULL);
}

int main (int argc, char* argv[])
{
 pthread_t t[N]; int errcode, i;
 pthread_mutex_init(&lock, NULL);
 for (i = 0; i < N; i ++)
 {
  if (pthread_create(&t[i], NULL, fun, NULL))
  {
   printf ("Error creating thread\n");
   return EXIT_FAILURE;
  }
 }
 for (i = 0; i < N; i ++)
 {
  pthread_join (t[i], NULL);
 }
 return 0;
}
```

A) Value: 5

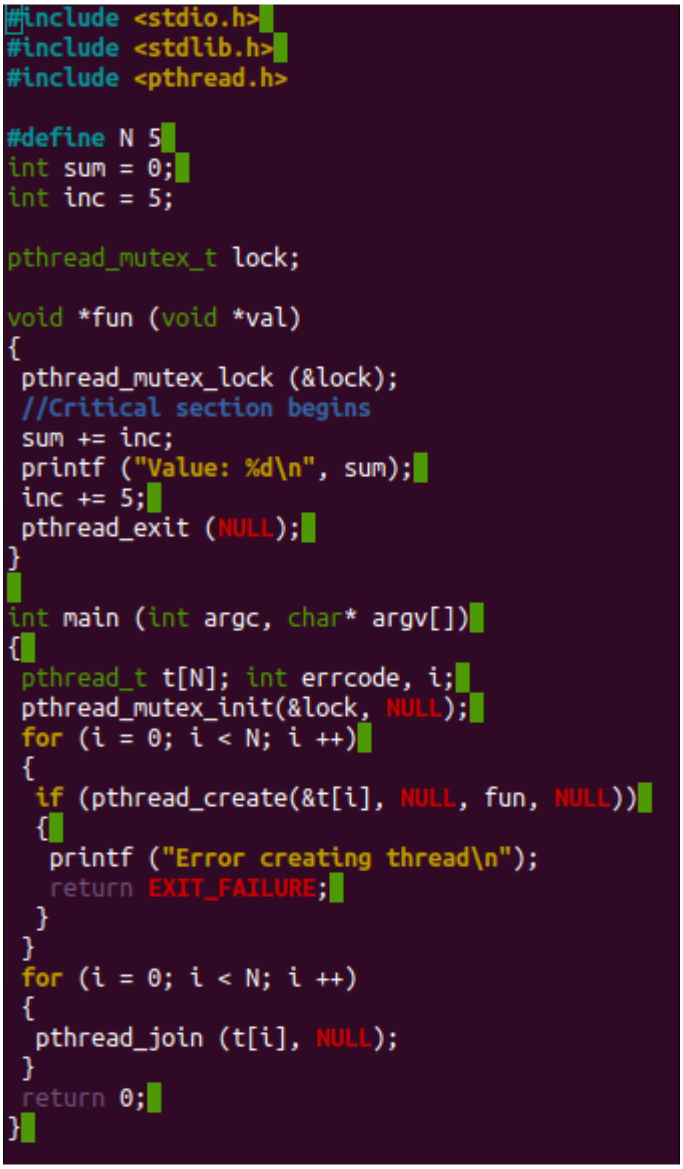
B)
Value: 5
Value: 15
Value: 30
Value: 50
Value: 75

C)
Value: 5
Value: 10
Value: 15
Value: 45
Value: 25

D) The output can be different each time when you run the code

ANSWER: D

QUESTION: What will be printed on the screen as a result of executing the following program ?



```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define N 5
int sum = 0;
int inc = 5;

pthread_mutex_t lock;

void *fun (void *val)
{
 pthread_mutex_lock (&lock);
 //Critical section begins
 sum += inc;
 printf ("Value: %d\n", sum);
 inc += 5;
 pthread_exit (NULL);
}

int main (int argc, char* argv[])
{
 pthread_t t[N]; int errcode, i;
 pthread_mutex_init(&lock, NULL);
 for (i = 0; i < N; i ++)
 {
  if (pthread_create(&t[i], NULL, fun, NULL))
  {
   printf ("Error creating thread\n");
   return EXIT_FAILURE;
  }
 }
 for (i = 0; i < N; i ++)
 {
  pthread_join (t[i], NULL);
 }
 return 0;
}
```

A) Value: 5

B)
Value: 5
Value: 15
Value: 30
Value: 50
Value: 75

C)
Value: 5
Value: 10
Value: 15
Value: 45
Value: 25

D) The output can be different each time when you run the code

ANSWER: A

QUESTION: Choose the most appropriate answer based on the thread handler function below:

```
pthread_mutex_t lock; // global
void *fun(void *val) {
   pthread_mutex_lock(&lock);
   pthread_mutex_lock(&lock);
   printf("Hello World! \n", sum); // Critical Section
   pthread_mutex_unlock(&lock);
   pthread_mutex_unlock(&lock);
   pthread_exit(NULL);
}
```

A) At most 1 thread can enter the critical section
B) At most 2 threads can enter the critical section
C) This will result in a deadlock
D) Mutual exclusion is not preserved
ANSWER: C

QUESTION: What is the most likely output of the following thread handler function? Assume 2 threads are competing to be scheduled on a uniprocessor system.

```
pthread_mutex_t lock;
void *fun(void *val) {
   pthread_mutex_lock(&lock);
   printf("First thread has entered \n");
   pthread_mutex_lock(&lock);
   printf("Second thread has entered \n");
   printf("Hello World!\n");
   pthread_mutex_unlock(&lock);
   pthread_mutex_unlock(&lock);
   pthread_exit(NULL);
}
```

A) First thread has entered
B) First thread has entered Second thread has entered Hello World!
C) Second thread has enteredFirst thread has entered Hello World!
D) Hello World!First thread has entered Second thread has entered
ANSWER: A

QUESTION: In the context of the following semaphore initialisation, choose the most appropriate option sem_init(&sem, 2, 5);
A) sem is a binary semaphore shared between different threads of the same process
B) sem is a counting semaphore shared between different threads of the same process
C) sem is a counting semaphore shared between different processes
D) sem is a binary semaphore shared between different processes
ANSWER: C

QUESTION: In the context of the following semaphore initialisation, choose the most appropriate option sem_init(&sem, 0, 5);
A) sem is a binary semaphore shared between different threads of the same process
B) sem is a counting semaphore shared between different threads of the same process
C) sem is a counting semaphore shared between different processes
D) sem is a binary semaphore shared between different processes

ANSWER: B

QUESTION: Select True or False. Ordinary pipes can be used between processes that have a parent-child relation.
A) True
B) False
ANSWER: A

QUESTION: Select True or False. Only related processes can exchange data using FIFOs.
A) False
B) True
ANSWER: A

QUESTION: Consider 5 processes whose Arrival time and CPU Burst time are shown in the following table.

Process	Arrival Time	Burst Time
P1	0	3
P2	2	3
P3	3	2
P4	5	5
P5	6	9

Assume that the processes are scheduled using the FCFS scheduling algorithm. The average waiting and average turnaround time are
A) 2.8 and 7.2
B) 2.5 and 7.0
C) 2.6 and 7.1
D) 1.9 and 6.9
ANSWER: A

QUESTION: Consider 5 processes whose Arrival time and CPU Burst time are shown in the following table.

Process	Arrival Time	Burst Time
P1	0	6
P2	2	4
P3	3	2
P4	5	5
P5	6	9

Assume that the processes are scheduled using the FCFS scheduling algorithm. The average waiting and average turnaround time are
A) 5.8 and 11.0
B) 2.5 and 17.0
C) 2.6 and 17.1
D) 1.9 and 16.9
ANSWER: A

QUESTION: Consider a TLB implemented using associative cache memory. Given hit ratio = 75%, cache memory access time = 20 ns and main memory access time is = 150 ns. What is the effective access time for accessing data corresponding to a logical address generated by the CPU?
A) 207.5 ns

B) 202.5 ns
C) 155 ns
D) 200 ns
ANSWER: A

QUESTION: Consider a TLB implemented using associative cache memory. Given hit ratio = 85%, cache memory access time = 20 ns and main memory access time is = 150 ns. What is the effective access time for accessing data corresponding to a logical address generated by the CPU?
A) 192.5 ns
B) 189.5 ns
C) 153 ns
D) 170 ns
ANSWER: A

QUESTION: The two memory access problem can be solved by the use of a special fast-lookup hardware cache called
A) associative memory
B) main memory
C) virtual memory
D) page table memory
ANSWER: A

QUESTION: Translation look-aside buffer is based on
A) cache memory
B) main memory
C) virtual memory
D) page table memory
ANSWER: A

QUESTION: Associative Memory used to solve two memory access problem is implemented using
A) cache memory
B) main memory
C) virtual memory
D) page table memory
ANSWER: A

QUESTION: The state transition that is associated with scheduler dispatch is
A) ready to running
B) running to ready
C) running to waiting
D) none of the options is correct
ANSWER: A

QUESTION: Scheduling is done so as to achieve which objective?
A) increase CPU utilization
B) decrease CPU utilization
C) keep the CPU more idle
D) none of the options is correct
ANSWER: A

QUESTION: Mutual exclusion problem occurs

A) among processes that share resources
B) between two disjoint processes that do not interact
C) among processes that do not use the same resource
D) none of the options is correct
ANSWER: A

QUESTION: Critical section is
A) a set of instructions that access common shared resource
B) a part of the operating system which is not allowed to be accessed by any process
C) the portion of the main memory which can be accessed only by one process at a time
D) none of the options is correct
ANSWER: A

QUESTION: Consider FIFO page replacement algorithm with the following reference string.1, 2, 4, 3, 2, 1, 4, 6, 2, 5The number of page faults for a 3 frame case is
A) 8
B) 5
C) 2
D) 7
ANSWER: A

QUESTION: Consider FIFO page replacement algorithm with the following reference string.1, 2, 4, 3, 1, 3, 4, 3, 2, 5The number of page faults for a 3 frame case is
A) 7
B) 5
C) 2
D) 8
ANSWER: A

QUESTION: Peterson’s solution is
A) a software based solution
B) a hardware based solution
C) a combination of hardware and software based solutions
D) none of the options is correct
ANSWER: A

QUESTION: The exit section of Peterson’s solution uses
A) only the flag array
B) only the turn variable
C) both the turn variable and flag array
D) none of the options is correct
ANSWER: A

QUESTION: Suppose you have a resource with 5 instances in a system. All the instances are shared by multiple processes and all the instances are equivalent. What will you use to solve the critical section problem in this system?
A) a single counting semaphore
B) a single binary semaphore
C) a single mutex
D) none of the options is correct
ANSWER: A

QUESTION: Which of the following statements is false for mutex lock?
A) mutex lock has an associated integer variable
B) mutex lock is also known as spinlock
C) mutex lock exhibits busy waiting
D) available() and release() operations are associated with mutex lock
ANSWER: A

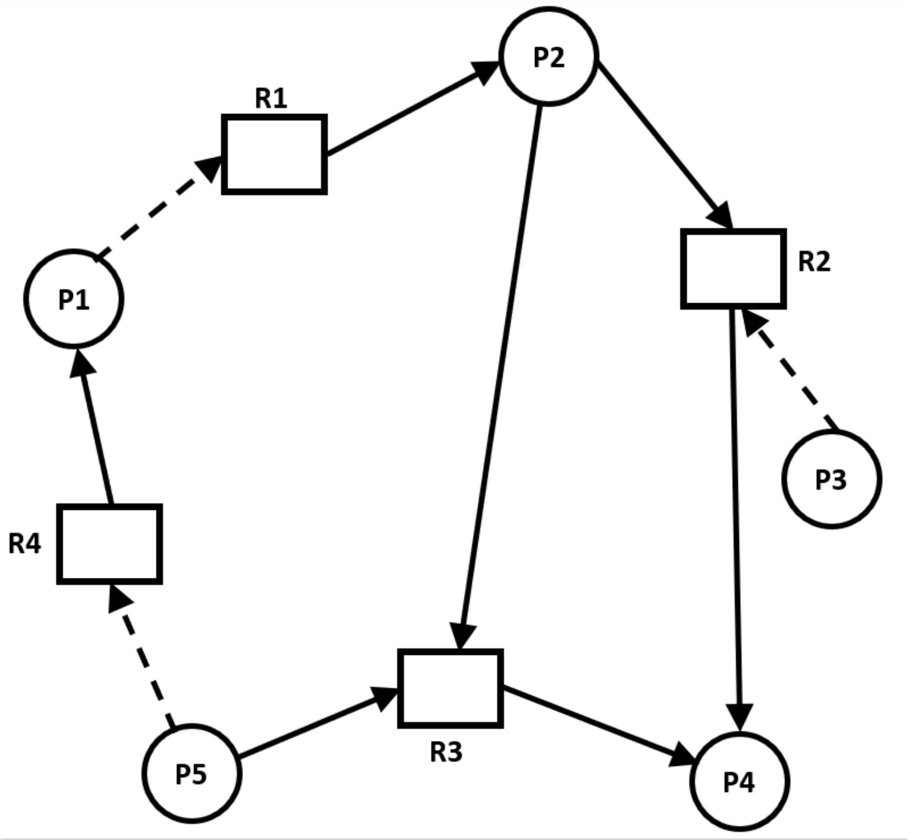
QUESTION: Consider a counting semaphore S whose value varies in the range 0 and 6. Initially, S = 6. Now, assume the following sequence of operations carried out on S.
wait(S);
wait(S);
signal(S);
wait(S);
wait(S);
signal(S);

What is the final value of S?
A) 4
B) 5
C) 6
D) 3
ANSWER: A

QUESTION: Consider a counting semaphore S whose value varies in the range 0 and 6. Initially, S = 0. Now, assume the following sequence of operations carried out on S.
signal(S);
signal(S);
wait(S);
signal(S);
wait(S);
signal(S);

What is the final value of S?
A) 2
B) 0
C) 4
D) 3
ANSWER: A

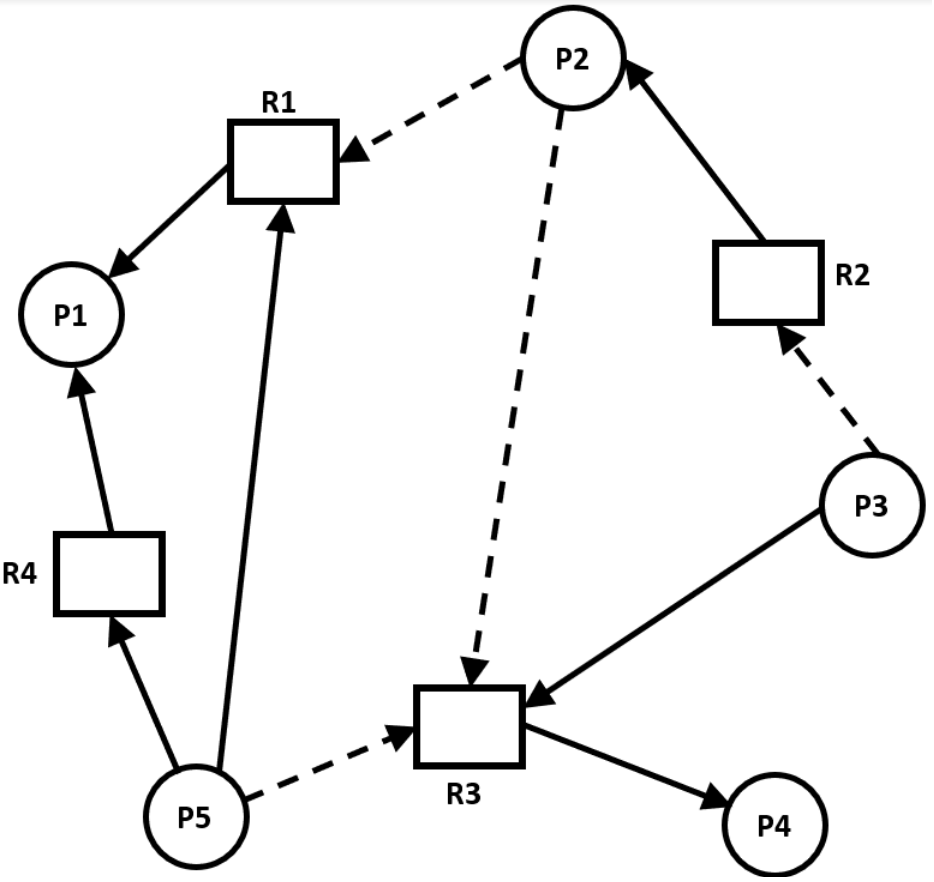
QUESTION: Consider the resource-allocation graph shown below.



This graph contains
A) 4 assignment edges, 3 request edges and 3 claim edges
B) 3 assignment edges, 4 request edges and 3 claim edges
C) 3 assignment edges, 4 request edges and 4 claim edges
D) 4 assignment edges, 4 request edges and 4 claim edges
ANSWER: A

QUESTION: Consider the resource-allocation graph shown below.

This graph contains



A) 4 assignment edges, 3 request edges and 4 claim edges
B) 3 assignment edges, 4 request edges and 3 claim edges
C) 3 assignment edges, 4 request edges and 4 claim edges
D) 4 assignment edges, 4 request edges and 4 claim edges
ANSWER: A

QUESTION: A computer system contains 2 types of resources - R1 and R2. A total of 7 instances of R1 and 10 instances of R2 are present. Currently, 4 processes, P1, P2, P3 and P4 are active in the system. The current snapshot of the system is shown in the table below.

Process	Allocation		Max	
	R1	R2	R1	R2
P1	0	2	3	9
P2	1	1	8	1
P3	1	3	4	7
P4	2	1	5	3

Determine which of the following is True.
A) a safe sequence is not possible
B) only one safe sequence is possible
C) multiple safe sequences are possible
D) none of the options is correct
ANSWER: A

QUESTION: A computer system contains 2 types of resources - R1 and R2. A total of 15 instances of R1 and 8 instances of R2 are present . Currently, 4 processes, P1, P2, P3 and P4 are active in the system. The current snapshot of the system is shown in the table below.

Process	Allocation		Max	
	R1	R2	R1	R2
P1	3	2	6	3
P2	4	1	8	5
P3	2	0	7	4
P4	1	1	6	6

Determine which of the following is True.
A) multiple safe sequences are possible
B) only one safe sequence is possible
C) a safe sequence is not possible
D) none of the options is correct
ANSWER: A

QUESTION: Consider a hypothetical system that uses an inverted page table. The CPU generates a 15-bit logical address 110100110110101. The process pid is specified using 5 bits, the page number is specified using 5 bits and the offset is also specified using 5 bits. What is the process pid for the given logical address?
A) 26
B) 13
C) 21
D) none of the options is correct
ANSWER: A

QUESTION: Consider a hypothetical system that uses an inverted page table. The CPU generates a 15-bit logical address 010111110110110. The process pid is specified using 5 bits, the page number is specified using 5 bits and the offset is also specified using 5 bits. What is the process pid for the given logical address?
A) 11
B) 29
C) 22
D) none of the options is correct
ANSWER: A

QUESTION: For a page, if the valid-invalid bit is set to i, it indicates that
A) the page is either not part of the logical address space of the process or the page is part of the logical address space of the process but is not present in memory
B) the page is memory resident
C) the page is not part of the logical address space of the process
D) the page is part of the logical address space of the process but is not present in memory
ANSWER: A

QUESTION: Dirty bit is used
A) to reduce the overhead of page replacement
B) to support locality of reference
C) as a substitute of valid-invalid bit
D) to keep track of the list of free frames
ANSWER: A