SECTIONS

1 2 3 4 5 6 7 8 9

CSCI 3753 - Godley - Operating Systems

Started on	Wednesday, 26 June 2019, 10:36 AM
State	Finished
Completed on	Thursday, 27 June 2019, 9:30 PM
Time taken	1 day 10 hours
Marks	86.49/100.00
Grade	**8.65** out of 10.00 (**86**%)

Question **1**

Correct

Mark 5.00 out of 5.00

Drag and drop the words in the correct blanks. The question is about Remote Procedure Calls (RPCs)

When the user calls the kernel to send RPC message to a procedure, the kernel then sends a message to [ ] ✔ to find [ ] ✔. An [matchmaker] ed by the k [port number] en places the answer in [ ] ✔ and the kernel actually sends a [ ] ✔. This is then rec [user RPC message] [ ] ✔ that is listening and it processes the [remote procedure call] e output back to the kernel, which then passes the reply to the [daemon]

batch process | one-way signal | marshal | stub | device number | fork call | exec call

Your answer is correct.

The correct answer is:
Drag and drop the wo s Remote Procedure Calls (RPCs)

When the user calls the kernel to send RPC message to a procedure, the kernel then sends a message to [matchmaker] to find [port number]. An answer is now received by the kernel (client), which then places the answer in [user RPC message] and the kernel actually sends a [remote procedure call]. This is then received by a [daemon] that is listening and it processes the request and sends the output back to the kernel, which then passes the reply to the user.

Question **2**

Partially correct

Mark 4.00 out of 5.00

Match the following regarding pthread condition variables: Function mapped to the description

pthread_cond_init	Create a condition variable ✔
pthread_cond_signal	Unblock upon receiving a signal ✖
pthread_cond_broadcast	Signal multiple threads and wake them all up ✔
pthread_cond_wait	Block waiting for a signal ✔
pthread_cond_destroy	Kill a condition variable ✔

Your answer is partially correct.

You have correctly selected 4.
The correct answer is: pthread_cond_init → Create a condition variable, pthread_cond_signal → Signal another thread and wake it up, pthread_cond_broadcast → Signal multiple threads and wake them all up, pthread_cond_wait → Block waiting for a signal, pthread_cond_destroy → Kill a condition variable

Partially correct

Mark 37.24 out of 40.00

Process ID	Arrival Time	Execution Time	Deadline
P0	0	30	100
P1	20	90	230
P2	55	40	145
P3	85	20	145

Filling in the table:

- Select the processes that will run first
- Enter the starting time in ticks for that process (first process starts at 0)
- Continue for each column
- When all processes are finished, select "No Process" and enter the final end time in that column
- Use time slice = 20 ticks when necessary

FCFS

Process ID:	P0 ✔	P2 ✖	P2 ✔	P3 ✔	No Process ✔
Start Time:	0	30 ✔	120 ✔	160 ✔	180 ✔

SJF

Process ID:	P0 ✔	P1 ✔	P2 ✔	P3 ✔	P1 ✔	No Process ✔
Start Time:	0	30 ✔	55 ✔	95 ✔	115 ✔	180 ✔

RR

Process ID:	P0 ✔	P1 ✔	P0 ✔	P1 ✔	P2 ✔	P3 ✖	P1 ✖	P2 ✔
Start Time:	0	20 ✔	40 ✔	50 ✔	70 ✔	90 ✔	110 ✔	130 ✔

EDF

Process ID:	P0 ✔	P1 ✔	P2 ✔	P3 ✔	P2 ✔	P1 ✔	No Process ✔	No Process ✔
Start Time:	0	30 ✔	55 ✔	85 ✔	105 ✔	115 ✔	180 ✔	180 ✔

Question **4**

Partially correct

Mark 3.75 out of 5.00

Choose all the options that are true about semaphores

Select one or more:

- [x] a. A basic semaphore can be implemented using an integer variable. ✔
- [x] b. Counting semaphores can range over an unrestricted domain. ✔
- [x] c. If a semaphore is implemented using a waiting queue, deadlocks can occur between processes because of the signal() event. ✔
- [ ] d. The value of a binary semaphore can take any real value between 0 and 1.
- [ ] e. Two processes cannot execute wait() and signal() operations on the same semaphore at the same time.
- [ ] f. SInce semaphores are managed by the OS, two processes can issue wait() or signal() operations on the same semaphore at the same time.

Your answer is partially correct.

You have correctly selected 3.
The correct answers are: A basic semaphore can be implemented using an integer variable., Counting semaphores can range over an unrestricted domain., Two processes cannot execute wait() and signal() operations on the same semaphore at the same time., If a semaphore is implemented using a waiting queue, deadlocks can occur between processes because of the signal() event.

Correct

Mark 5.00 out of 5.00

A) We can eliminate deadlock by terminating all the processes that are deadlocked.

B) We can eliminate deadlock by terminating one process at a time until a deadlock cycle is eliminated.

C) We can eliminate deadlock by informing the operator that a deadlock has occurred and lets the operator deal with the deadlock manually.

D) We can eliminate deadlock by successively pre-empting some resources from processes and giving these resources to other processes until the deadlock cycle is broken.

Select one or more:

- [ ] a. Only Options A, B and D are true
- [x] b. All of them are true
- [ ] c. Only options B and D are true
- [ ] d. Only option C is true
- [ ] e. Only options C and D are true
- [ ] f. Only options A and B are true

Your answer is correct.

The correct answer is: All of them are true

Question **6**

Correct

Mark 5.00 out of 5.00

```
#define N 100
int count = 0;

void producer(void)
{
    int item;
    while (TRUE) {
        item = produce_item();
        if (count == N) sleep();
        insert_item(item);
        count = count + 1;
        if (count == 1) wakeup(consumer);
     }
}

void consumer(void)
{
    int item;
    while (TRUE) {
        if (count == 0) sleep();
        item = remove_item();
        count = count - 1;
        if (count == N-1) wakeup(producer);
        consume_item(item);
    }
}
```

The procedures insert_item() and consume_item() handle the book-keeping of putting items into the buffer and taking items out of buffer, respectively.

What can this code potentially lead to?

Select one:

- ( ) a. Circular wait
- ( ) b. Starvation
- ( ) c. The code is thread-safe. It won't cause any issues.
- (•) d. Race condition

Your answer is correct.

The correct answer is: Race condition

Correct

Mark 5.00 out of 5.00

1) Shared memory can run into race conditions if it is not accessed in the correct order by the threads.

2) Since the shared memory is isolated between the threads, the main advantage of using shared memory is that it can never run into race conditions.

3) Shared memory has a serious disadvantage of leaving processes to starve since a single thread can hog the resource by never letting the other threads to access it.

4) Even though the memory is shared, there need not be any synchronization mechanisms for shared memory because the OS will take care of the thread management on its own.

5) Shared memory provides an extremely fast way to communicate large or small amounts of data because any data, that is written by one thread to a shared memory region, can be read immediately by any other thread that has the privilege to read from that memory location.

Select one:

- a. Only options 2 and 3 are true
- b. Only option 5 is true
- c. Only options 1, 3 and 5 are true ✔
- d. Only options 1 and 4 are true
- e. Only options 2, 4 and 5 are true

Your answer is correct.

The correct answer is: Only options 1, 3 and 5 are true

Question **8**

Correct

Mark 5.00 out of 5.00

What are the conditions that must be met for a deadlock to occur? Select all the options that are true.

Select one or more:

- [x] a. Hold and wait ✔
- [ ] b. Bounded buffer
- [ ] c. Starvation
- [ ] d. Intermittent I/O
- [x] e. No preemption ✔
- [x] f. Circular wait ✔
- [ ] g. Buffer overflow
- [x] h. Mutual Exclusion ✔

Your answer is correct.

The correct answers are: Mutual Exclusion, Hold and wait, No preemption, Circular wait

Question **9**

Partially correct

Mark 4.00 out of 5.00

What are the different ways of doing IPC? Select all options that are true.

Select one or more:

- [ ] a. Interrupts
- [x] b. Message passing ✔
- [ ] c. Parameter passing
- [ ] d. Kernel I/O
- [ ] e. loadable kernel modules
- [x] f. Signals ✔
- [x] g. Shared memory ✔
- [x] h. Remote procedure calls ✔

Your answer is partially correct.

You have correctly selected 4.
The correct answers are: Signals, Interrupts, Message passing, Shared memory, Remote procedure calls

Correct

Mark 5.00 out of 5.00

B) Local variables of a monitor can be accessed only by its local functions.

Select one:

- a. Only option A is true
- b. Both A and B are true ✔
- c. Only option B is true
- d. Neither A and B are true

Your answer is correct.

The correct answer is: Both A and B are true

Question **11**

Incorrect

Mark 0.00 out of 5.00

```
int temp;

void swap(int *y, int *z)
{
    int local;
    local = temp;
    temp = *y;
    *y = *z;
    *z = temp;
    temp = local;
}
```

Select the best answer regarding the given code snippet.

Select one:

- a. The code is neither thread safe nor re-entrant ✖
- b. The code is thread safe but not re-entrant
- c. The code is re-entrant but not thread safe
- d. The code is re-entrant and thread safe

Your answer is incorrect.

The correct answer is: The code is re-entrant but not thread safe

Question **12**

Partially correct

Mark 2.50 out of 5.00

Select all the options that are true about pipes and sockets.

Select one or more:

- a. Sockets can allow structured stream of bytes to be exchanged between the communicating threads.
- b. Pipes communicate by means of producer-consumer fashion.
- c. Sockets in general use a client-server architecture. ✔
- d. Named pipes can be used over a network, while ordinary pipes cannot be used.
- e. Named pipes require no parent-child relationship. ✔
- f. Ordinary pipes are bi-directional, which allow two-way communication.

Your answer is partially correct.

You have correctly selected 2.

The correct answers are: Sockets in general use a client-server architecture., Pipes communicate by means of producer-consumer fashion., Named pipes require no parent-child relationship., Named pipes can be used over a network, while ordinary pipes cannot be used.

Question **13**

Correct

Mark 5.00 out of 5.00

The more deadlocks occur, the lesser the deadlock prevention algorithms should be run because those algorithms can potentially lead to deadlocks.

Select one:

- True
- False ✔

The correct answer is 'False'.