

| | |
|--------------|----------------------------------|
| Started on | Thursday, 11 April 2024, 9:42 PM |
| State | Finished |
| Completed on | Thursday, 11 April 2024, 9:47 PM |
| Time taken | 4 mins 26 secs |
| Grade | 2.00 out of 2.00 (100%) |

Question 1

Correct

Mark 1.00 out of 1.00

Deadlocks: Mark the assertions which are true.

Select one or more:

- ☒ a. A **deadlock** is a state in which each member of a group is waiting for another member, including itself, to take some action. ✔ This is one of the options which is true.
- ☒ b. If a process is unable to change its state indefinitely because the resources requested by it are being used by another waiting process, then the system is said to be in a deadlock. ✔ True! (one of them)
- ☒ c. If a deadlock occurs it is necessary to have external intervention, such as terminating processes, because the deadlock will never be solved autonomously. ✔ This is one of the options which is true!

Your answer is correct.

The correct answers are: A **deadlock** is a state in which each member of a group is waiting for another member, including itself, to take some action., If a process is unable to change its state indefinitely because the resources requested by it are being used by another waiting process, then the system is said to be in a deadlock., If a deadlock occurs it is necessary to have external intervention, such as terminating processes, because the deadlock will never be solved autonomously.

Question 2

Correct

Mark 1.00 out of 1.00

In the example shown in the video, the two processes block waiting for each other (deadlock). One way to avoid the deadlock is by:

Select one or more:

- ☒ a. By using a different order for sending and receiving in process 0 and process 1: Only one of them should send first while the other should receive first. ✔ True! (one of them)
- ☒ b. Using MPI_Isend which returns immediately and does not block. ✔ True! (one of them)

Your answer is correct.

The correct answers are: Using MPI_Isend which returns immediately and does not block. , By using a different order for sending and receiving in process 0 and process 1: Only one of them should send first while the other should receive first.