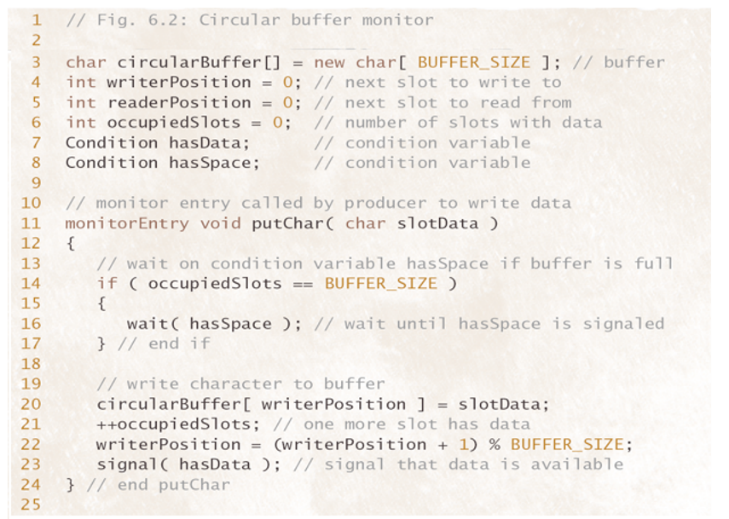
Coral Schmidt Montilla #148830

Refer to the monitor described in the following figure and answer the following questions:



1. Which procedure places data into the circular buffer?

The procedure `putChar` places data into the circular buffer.

1. Which procedure removes data from the circular buffer?

The procedure for removing data from the circular buffer is not provided in the code snippet.

1. Which queueing discipline best describes the operation of the circular buffer?

The queueing discipline that best describes the operation of the circular buffer is First-In-First-Out (FIFO), where the data that is written into the buffer first is read out first.

1. Is this true: writePosition>=readerPosition?

The statement `writerPosition >= readerPosition` is not necessarily true as it depends on the specific implementation and the current state of the circular buffer. It could be true if the buffer is full or if the reader is lagging behind the writer, but it may not always be the case.

1. Which statements perform monitor initialization?

The statement `char circularBuffer[] = new char[BUFFER\_SIZE];` initializes the circular buffer.

1. Which statement(s) can “wake up” a thread waiting on a condition variable?

The statement `signal(hasData);` can "wake up" a thread waiting on the `hasData` condition variable.

1. Which statement(s) can put a thread “to sleep”?

The statement `wait(hasSpace);` can put a thread "to sleep" while waiting for space to become available in the circular buffer.

1. Which statement(s) ensure that the buffer “wraps around”?

The statement `writerPosition = (writerPosition + 1) % BUFFER\_SIZE;` ensures that the buffer "wraps around" when the end is reached, allowing for circular behavior.

1. Which statement(s) modify a shared critical variable to indicate that another slot in the buffer is available?

The statement `++occupiedSlots;` modifies a shared critical variable to indicate that another slot in the buffer is available.