

Figure 1 shows the flow in producing a product.

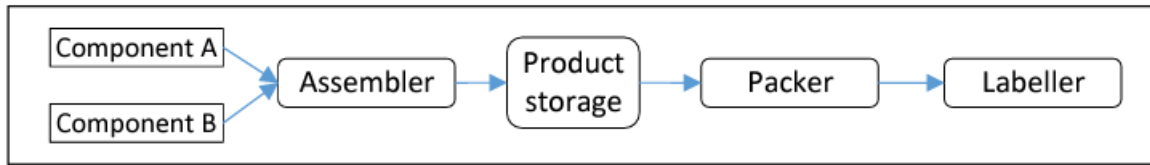


Figure 1

The product is made of two components, namely Component-A and Component-B. The Assembler will assemble both components to produce the final product and put it in the storage. Whenever there are 6 units of products in the storage, the Packer will pack all the 6 units into a box. The box will be sent to the Labeller to label the box.

Write a multithreaded program to simulate the above-mentioned production flow based on the following guidelines. You may use C/C++, Java, C#, python or other programming language that support threading and synchronisation (semaphore or condition variable).

- A thread to simulate the production of Component-A.
- A thread to simulate the production of Component-B.
- A thread to simulate the Assembler.
- A thread to simulate the Packer.
- A thread to simulate the Labeller.
- All threads should run iteratively, i.e., infinite loop.
- Assumption:
 1. Component-A is produced every second.
 2. Component-B is produced every 2 seconds.
 3. The Assembler needs 2 seconds to assemble both components to produce the final product.
 4. The Packer takes 2 seconds to pack the products into a box.
- Each thread should print a message to indicate its current operation or state. For examples:

The Component-A thread may print messages such as

Component-A: Unit-1 produced.

Component-A: Unit-2 produced.

...

The Assembler thread may print messages such as

Assembler: waiting for components

Assembler: Product-1 completed.

Assembler: waiting for components

Assembler: Product-2 completed.

...

NOTE: the output from each thread may interleave. For examples:

Component-A: Unit-1 produced.

Component-A: Unit-2 produced.

Component-B: Unit-1 produced.

Component-A: Unit-3 produced.

Assembler: waiting for components

Assembler: Product-1 completed.

Component-A: Unit-4 produced.

...