

Caso 1: Manejo de Concurrency

Infraestructura Computacional

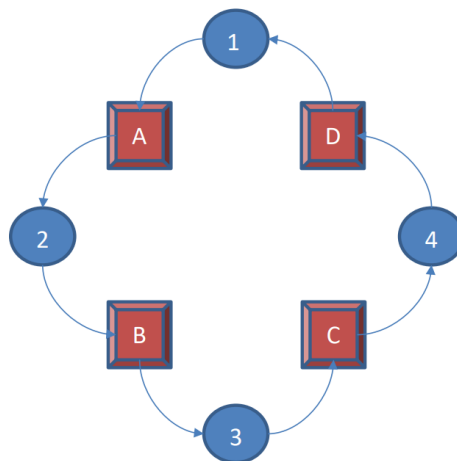
2022-1

Contenido

1. Introducción	1
2. Diagrama de solución	2
3. Funcionamiento global del Sistema.....	3
3.1 Transmisión de mensajes	4
3.2 Perspectiva de un buffer	4
3.3 Sincronización entre threads	5
3.4 Caso especial: Proceso 1.....	5
3.5 Caso especial: Ultimo mensaje (FIN)	5
4. Pruebas de funcionamiento	5

1. Introducción

En el presente reporte se documenta el desarrollo e implementación de la simulación de un sistema de mensajes mediante el uso de Threads en un programa en Java.



El problema a solucionar consiste en encontrar una manera que en un sistema con 4 buzones y 4 procesos (threads) que conectan entre si a cada uno de los buzones en un patrón circular (ver figura 1). Cuando se inicia la ejecución el proceso 1 debe mandar el número fijo de mensajes, el cual fue dado por parámetro, al buzón que le sigue a ese thread (en este caso el A). Ya cuando un mensaje está dentro de un buzón este es procesado, y se le añade al final de este un mensaje que indica que ya paso por este buzón. Después, otro thread le pide al buzón sacar el mensaje y decide procesarlo, y al final del mensaje se le añade un mensaje que indica que paso por ese thread y se manda a dormir en thread un tiempo determinado. Eventualmente ese thread/proceso va a pasar el mensaje modificado al siguiente buzón, y el proceso continuara hasta que el mensaje vuelva a llegar al proceso 1. Este proceso se realiza para el número dado de mensajes que se desea dar, y una vez todos los mensajes hayan vuelto a llegar a el proceso 1 se decide mandar un último mensaje llamado fin que es el responsable de matar a todos los thread y terminar la ejecución del sistema de mensajería.

2. Diagrama de solución

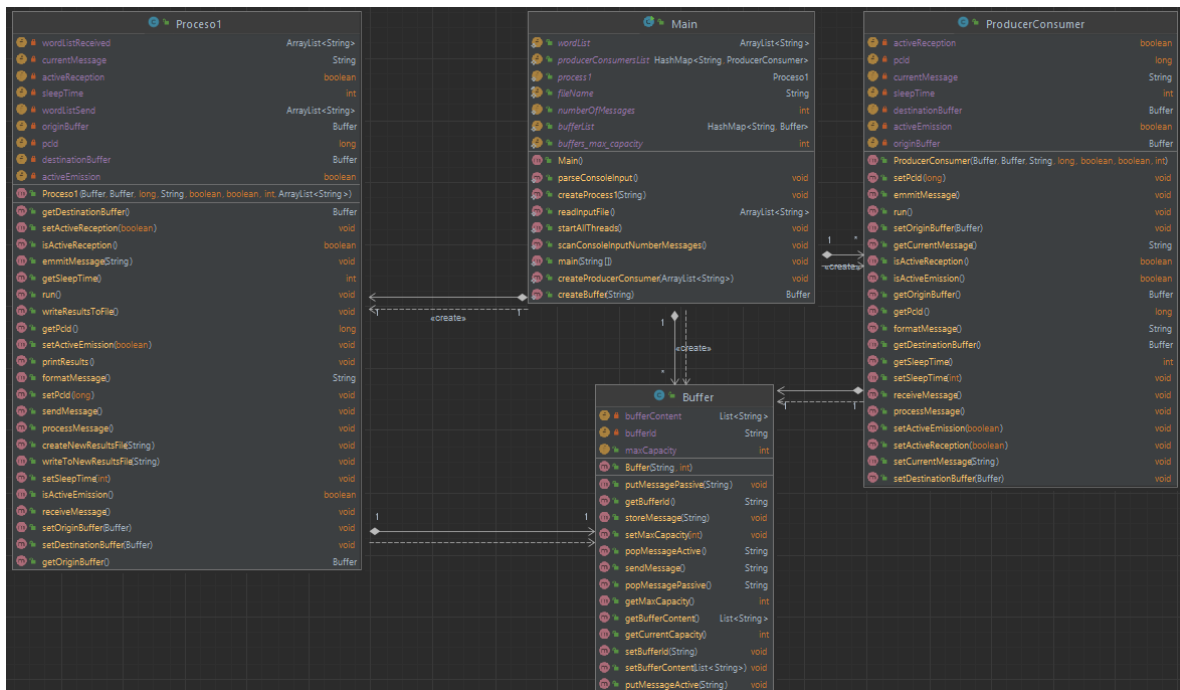
Para la creación del sistema de mensajería se implementaron 3 clases: Main, ProducerConsumer y Buffer.

En la Main es la responsable de leer la configuración de los buzones (buffers) y los procesos (producerConsumer) e inicializarlos, eventualmente esta misma es responsable de inicializar los threads y autorizar que el proceso 1 sea quien manda el número de mensajes dados por parámetro.

La clase de ProducerConsumer es una mezcla de las clases base de Producer y Consumer del ejemplo dado en clase del problema de Productores y Consumidores. Esta clase tiene la responsabilidad de sacar el mensaje del buzón de origen, procesarlo y eventualmente mandarlo al siguiente buzón.

La clase de Process1 es una especialización de la clase de ProducerConsumer (extiende producer consumer) para tratar con el caso específico de procesos 1, el cual tiene la particularidad que es la clase que manda todos los mensajes iniciales y es responsable de recibir al final los mensajes una vez le dan la vuelta a todos los buffers. Se necesita esta especialización dado que se necesita incluir parámetros nuevos como messagesToSend y una estructura del run que difiere de los otros procesos. La estructura del run en este caso consiste en primero mandar el numero de mensajes especificados al buffer A, y después prepararse para recibir el numero especificado de mensajes del buffer D, una vez se completa este ciclo este también tiene que mandar el mensaje de fin para finalizar a todos los otros procesos (del 2 al 4).

La última clase del modelo es buffer, es en esta clase que se implementa la mayoría de la sincronización del programa, ya que es aquí que se implementa las funcionalidades de intentar insertar un mensaje (putMessage) y de intentar sacar un mensaje (popMessage) los cuales los threads (ProducerConsumer) que conectan con los buzones ejecutaran para pasar los mensajes. Adicionalmente, buffer tiene métodos específicos para implementar comunicación activa y pasiva, por eso hay dos versiones de putMessage y popMessage, cada una con el tipo de comunicación que utiliza.



3. Funcionamiento global del Sistema

Inicialmente pensamos solo en crear una clase general para todos los Threads(procesos) en donde cada Thread compartía los mismos atributos y los mismos métodos, sin embargo en el momento de la ejecución el orden de instrucciones cambiaba, ya que el Thread 1 empieza generando o enviando mensajes al buffer A, mas no recibiendo es por eso que creamos una clase “Proceso1 (Thread1)”, en donde por parámetro le entra una lista de palabras previamente digitalizadas por sistema, y este mismo lo que hace es enviarlas al buffer “A” dependiendo su estado previamente seleccionado (Activo/Pasivo) y también dependiendo el almacenamiento del buffer, después de enviar las palabras por ultimo se le envía al Buffer un mensaje diciendo “Fin” .

De la misma manera el buffer A y, al igual que los demás Buffers, almacena los mensajes dependiendo su capacidad previamente instaurada, para que después el Thread siguiente (2) de manera pasiva o activa le solicite un mensaje al buffer. Cabe agregar que los Buffer no mandan mensajes y no retiran mensajes a menos que el Thread se lo solicite. Cuando se solicita un mensaje al Buffer y este no tiene nada adentro de el usamos “.wait()” o “.yield”. Cuando usamos “.wait()” lo que haremos será esperar a que algún mensaje le llegue al Buffer, o bien que se le pueda mandar a este algún mensaje, esto causara que se consuma RAM debido a que no realizara otras operaciones hasta que pueda tener un mensaje. Igualmente con el “yield” se realizara una espera activa, lo que significa que una vez se llegue a esta instrucción estará constantemente preguntando si ya hay mensajes debido a que se mandara a una cola de espera y este al ser único

Thread se repetirá hasta que consuma un mensaje. Este mismo proceso entre el Thread y el buffer se hace secuencialmente hasta que llegue al Buffer D.

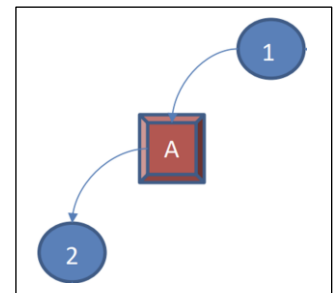
Al llegar los mensajes al Buffer D el Thread 1 de manera pasiva o activa estará solicitando los mensajes, y este una vez que tiene los mensajes los imprime por consola para después ser verificados, cabe agregar que cada vez que un mensaje pasa por un Thread este lo marca con un símbolo distintivo el cual es el número del Thread y si llega y se manda de manera pasiva o activa.

3.1 Transmisión de mensajes

- Recibir mensaje: esta función es única para todos los Threads, sin embargo dependiendo como se instaura el Thread, este podrá mensajes de manera activa al Buffer usando la función “.yield()” la cual una vez implementada estará de manera constantemente solicitando mensajes al Buffer, o de manera pasiva haciendo una de “.wait()” en el cual el Thread esperara consumiendo RAM mientras llega un mensaje en el Buffer
- Procesar mensaje: Una vez el Thread tiene el mensaje se tiene que identificar que el mensaje que se recibió no es “Fin”, una vez hecho esto se le tiene que agregar al mensaje el número del Thread en el que esta actualmente y como ese mensaje llega ahí bien sea de manera activa o de manera pasiva.
- Emitir mensaje: finalmente el Thread enviara el mensaje ya marcado al buffer, sin embargo este envío dependerá del almacenamiento del buffer y de igual forma si su envío es de forma activa o pasiva.

3.2 Perspectiva de un buffer

Un Buffer consta de 4 métodos los cuales son PutMessageActive, PutMessagePassive, PopMessageActive, PopMessagePasive, en el cual dependiendo la composición del Thread se usaran los métodos adecuados.



Cabe agregar que los buffer solo funcionaran si los Threads los utilizan, sin embargo cada buffer contiene las funciones auxiliares, que le permiten el desarrollo y el funcionamiento al mismo, tales como “getCurrentCapacity()” la cual nos dice la capacidad actual del buffer y también storeMessage(String message) en la cual su función es guardar en el buffer el mensaje que se da por parámetro. De igual forma en cada instrucción usamos condicionales como “While” ya que gracias a este podremos tener a la instrucción en espera hasta se cumpla o hasta que pase lo contrario.

3.3 Sincronización entre threads

Toda la sincronización de los Threads se realiza en la clase main

3.4 Caso especial: Proceso 1

Como se menciona anterior mente El proceso 1 o Thread 1 contiene los mismos métodos que la clase “Producerconsumer” sin embargo su proceso de ejecución es diferente, ya que este al recibir una lista por entrada tendrá que enviar los mensajes al Buffer A y no como los demás que simplemente solicitan y después envían.

3.5 Caso especial: Ultimo mensaje (FIN)

Este mensaje que representa el final de la lista de mensajes, es enviado por la clase “Proceso1” la cual le indica a los Thread que ese mensaje no se marca, y que de igual forma no vendrán mas mensajes después del mismo. Este mensaje se manda después de enviados todos los mensajes por parte de la lista.

4. Pruebas de funcionamiento

En nuestro programa tenemos implementada una funcionalidad para ver el modo de debugging, donde se puede ver el recorrido de los mensajes a través de los diferentes threads y buffers. Esto se habilita con el parámetro de debugMode en main, por default en el Proyecto que mandamos este está habilitado para hacer más visible el funcionamiento del programa y sus diferentes threads en acción.

Para entrar la configuración de los buffers y procesos esta se tiene que meter en el formato especificado en el enunciado en el archive de inputFile.txt que está en el Proyecto. Una vez se ejecuta un caso el resultado de este se guarda en la carpeta de output files con toda la información relevante de la configuración, mensajes mandados, mensajes recibidos.

A continuación, presentaremos algunos de los casos y como procedió su ejecución. Como se puede ver en los siguientes ejemplos se puede ver la sincronización en acción. Si se desean ver más ejemplos se pueden ver en output Files.

EJEMPLO 1:

Enter the number of messages to be sent by process 1:

4

Input word 0:

bonjour

Input word 1:

hello

Input word 2 :

hola

Input word 3 :

ni hao

All threads are started....

Waiting for all messages to arrive back at process 1....

Message emitted at process 1: bonjour

Message emitted at process 1: hello

Message emitted at process 1: hola

Message received from process 2: bonjour

Message emitted at process 1: ni hao

Message emitted at process 1: FIN

Message processed at process 2 : bonjour 2AA

Message emitted at process 2 : bonjour 2AA

Message received from process 2: hello

Message received from process 3: bonjour 2AA

Message processed at process 2 : hello 2AA

Message emitted at process 2 : hello 2AA

Message received from process 2: hola

Message processed at process 3 : bonjour 2AA 3PA

Message emitted at process 3 : bonjour 2AA 3PA
Message received from process 4: bonjour 2AA 3PA
Message received from process 3: hello 2AA
Message processed at process 2 : hola 2AA
Message emitted at process 2 : hola 2AA
Message received from process 2: ni hao
Message processed at process 3 : hello 2AA 3PA
Message emitted at process 3 : hello 2AA 3PA
Message received from process 3: hola 2AA
Message processed at process 2 : ni hao 2AA
Message emitted at process 2 : ni hao 2AA
Message received from process 2: FIN
Message processed at process 2 : FIN
Message processed at process 4 : bonjour 2AA 3PA 4PP
Message emitted at process 4 : bonjour 2AA 3PA 4PP
Message received from process 1: bonjour 2AA 3PA 4PP
Message received from process 4: hello 2AA 3PA
Message processed at process 1: bonjour 2AA 3PA 4PP 1AP
Message processed at process 3 : hola 2AA 3PA
Message emitted at process 3 : hola 2AA 3PA
Message received from process 3: ni hao 2AA
Message emitted at process 2 : FIN
Message processed at process 4 : hello 2AA 3PA 4PP
Message received from process 1: hello 2AA 3PA 4PP
Message emitted at process 4 : hello 2AA 3PA 4PP
Message received from process 4: hola 2AA 3PA
Message processed at process 1: hello 2AA 3PA 4PP 1AP
Message processed at process 3 : ni hao 2AA 3PA
Message emitted at process 3 : ni hao 2AA 3PA

Message received from process 3: FIN
Message processed at process 3 : FIN
Message emitted at process 3 : FIN
Message processed at process 4 : hola 2AA 3PA 4PP
Message emitted at process 4 : hola 2AA 3PA 4PP
Message received from process 1: hola 2AA 3PA 4PP
Message received from process 4: ni hao 2AA 3PA
Message processed at process 1: hola 2AA 3PA 4PP 1AP
Message processed at process 4 : ni hao 2AA 3PA 4PP
Message received from process 1: ni hao 2AA 3PA 4PP
Message emitted at process 4 : ni hao 2AA 3PA 4PP
Message received from process 4: FIN
Message processed at process 4 : FIN
Message emitted at process 4 : FIN
Message processed at process 1: ni hao 2AA 3PA 4PP 1AP
Message received from process 1: FIN
Message processed at process 1: FIN

EXECUTION RESULTS

These are the messages received , in order:

Message 1 content: bonjour 2AA 3PA 4PP 1AP

Message 2 content: hello 2AA 3PA 4PP 1AP

Message 3 content: hola 2AA 3PA 4PP 1AP

Message 4 content: ni hao 2AA 3PA 4PP 1AP

You can see the results in the following file in the outputfiles folder: results-1645500844683.txt

These are the parameters for the buffers and producer consumers

These are the messages that were sent

Process finished with exit code 0

EJEMPLO 2:

THE CONFIGURATION FOR THE PROCESSES AND THE BUFFERS MUST BE PUT IN THE inputFile.txt
WITH THE SPECIFIED PATTERN GIVEN IN THE CASE SPECIFICATIONS

The input configuration has been completed/n

Enter the number of messages to be sent by process 1:

20

Input word 0 :

did

Input word 1 :

you

Input word 2 :

ever

Input word 3 :

hear

Input word 4 :

the

Input word 5 :

tragedy

Input word 6 :

of

Input word 7 :

darth

Input word 8 :

plagueis

Input word 9 :

the

Input word 10 :

wise

Input word 11 :

?

Input word 12 :

i

Input word 13 :

thought

Input word 14 :

not

Input word 15 :

a

Input word 16 :

story

Input word 17 :

the

Input word 18 :

jedi

Input word 19 :

would

All threads are started....

Waiting for all messages to arrive back at process 1....

Message emitted at process 1: did

Message emitted at process 1: you

Message emitted at process 1: ever

Message emitted at process 1: hear

Message received from process 2: did

Message emitted at process 1: the

Message emitted at process 1: tragedy

Message emitted at process 1: of
Message emitted at process 1: darth
Message emitted at process 1: plagueis
Message emitted at process 1: the
Message emitted at process 1: wise
Message processed at process 2 : did 2AA
Message emitted at process 2 : did 2AA
Message emitted at process 1: ?
Message received from process 2: you
Message received from process 3: did 2AA
Message processed at process 2 : you 2AA
Message emitted at process 2 : you 2AA
Message received from process 2: ever
Message emitted at process 1: i
Message processed at process 3 : did 2AA 3PA
Message emitted at process 3 : did 2AA 3PA
Message received from process 3: you 2AA
Message received from process 4: did 2AA 3PA
Message processed at process 2 : ever 2AA
Message emitted at process 2 : ever 2AA
Message received from process 2: hear
Message emitted at process 1: thought
Message processed at process 3 : you 2AA 3PA
Message emitted at process 3 : you 2AA 3PA
Message received from process 3: ever 2AA
Message processed at process 2 : hear 2AA
Message emitted at process 2 : hear 2AA
Message received from process 2: the
Message emitted at process 1: not

Message processed at process 4 : did 2AA 3PA 4PP

Message emitted at process 4 : did 2AA 3PA 4PP

Message received from process 4: you 2AA 3PA

Message processed at process 2 : the 2AA

Message processed at process 3 : ever 2AA 3PA

Message emitted at process 3 : ever 2AA 3PA

Message received from process 3: hear 2AA

Message emitted at process 2 : the 2AA

Message received from process 2: tragedy

Message emitted at process 1: a

Message processed at process 2 : tragedy 2AA

Message processed at process 4 : you 2AA 3PA 4PP

t Message processed at process 3 : hear 2AA 3PA

Message emitted at process 3 : hear 2AA 3PA

Message received from process 3: the 2AA

Message emitted at process 2 : tragedy 2AA

Message emitted at process 1: story

Message received from process 2: of

Message processed at process 2 : of 2AA

Message processed at process 3 : the 2AA 3PA

Message emitted at process 3 : the 2AA 3PA

Message received from process 3: tragedy 2AA

Message emitted at process 2 : of 2AA

Message received from process 2: darth

Message emitted at process 1: the

Message processed at process 2 : darth 2AA

Message processed at process 3 : tragedy 2AA 3PA

Message emitted at process 3 : tragedy 2AA 3PA

Message received from process 3: of 2AA

Message emitted at process 2 : darth 2AA

Message emitted at process 1: jedi

Message received from process 2: plagueis

Message processed at process 2 : plagueis 2AA

Message processed at process 3 : of 2AA 3PA

Message emitted at process 3 : of 2AA 3PA

Message received from process 3: darth 2AA

Message emitted at process 2 : plagueis 2AA

Message received from process 2: the

Message emitted at process 1: would

Message processed at process 2 : the 2AA

Message processed at process 3 : darth 2AA 3PA

Message emitted at process 3 : darth 2AA 3PA

Message received from process 3: plagueis 2AA

Message emitted at process 2 : the 2AA

Message received from process 2: wise

Message emitted at process 1: FIN

Message received from process 1: did 2AA 3PA 4PP

Message emitted at process 4 : you 2AA 3PA 4PP

Message received from process 4: ever 2AA 3PA

Message processed at process 1: did 2AA 3PA 4PP 1AP

Message received from process 1: you 2AA 3PA 4PP

Message processed at process 2 : wise 2AA

Message processed at process 1: you 2AA 3PA 4PP 1AP

Message processed at process 3 : plagueis 2AA 3PA

Message emitted at process 3 : plagueis 2AA 3PA

Message received from process 3: the 2AA

Message emitted at process 2 : wise 2AA

Message received from process 2: ?

Message processed at process 4 : ever 2AA 3PA 4PP

Message emitted at process 4 : ever 2AA 3PA 4PP

Message received from process 1: ever 2AA 3PA 4PP

Message received from process 4: hear 2AA 3PA

Message processed at process 1: ever 2AA 3PA 4PP 1AP

Message processed at process 2 : ? 2AA

Message processed at process 3 : the 2AA 3PA

Message emitted at process 3 : the 2AA 3PA

Message received from process 3: wise 2AA

Message emitted at process 2 : ? 2AA

Message received from process 2: i

Message processed at process 4 : hear 2AA 3PA 4PP

Message emitted at process 4 : hear 2AA 3PA 4PP

Message received from process 1: hear 2AA 3PA 4PP

Message received from process 4: the 2AA 3PA

Message processed at process 2 : i 2AA

Message processed at process 1: hear 2AA 3PA 4PP 1AP

Message processed at process 3 : wise 2AA 3PA

Message emitted at process 3 : wise 2AA 3PA

Message received from process 3: ? 2AA

Message emitted at process 2 : i 2AA

Message received from process 2: thought

Message processed at process 2 : thought 2AA

Message processed at process 4 : the 2AA 3PA 4PP

Message received from process 1: the 2AA 3PA 4PP

Message emitted at process 4 : the 2AA 3PA 4PP

Message received from process 4: tragedy 2AA 3PA

Message processed at process 3 : ? 2AA 3PA

Message emitted at process 3 : ? 2AA 3PA

Message received from process 3: i 2AA

Message emitted at process 2 : thought 2AA

Message received from process 2: not

Message processed at process 1: the 2AA 3PA 4PP 1AP

Message processed at process 2 : not 2AA

Message processed at process 3 : i 2AA 3PA

Message emitted at process 3 : i 2AA 3PA

Message received from process 3: thought 2AA

Message emitted at process 2 : not 2AA

Message received from process 2: a

Message processed at process 4 : tragedy 2AA 3PA 4PP

Message received from process 1: tragedy 2AA 3PA 4PP

Message emitted at process 4 : tragedy 2AA 3PA 4PP

Message received from process 4: of 2AA 3PA

Message processed at process 2 : a 2AA

Message processed at process 1: tragedy 2AA 3PA 4PP 1AP

Message processed at process 3 : thought 2AA 3PA

Message emitted at process 3 : thought 2AA 3PA

Message received from process 3: not 2AA

Message emitted at process 2 : a 2AA

Message received from process 2: story

Message processed at process 4 : of 2AA 3PA 4PP

Message received from process 1: of 2AA 3PA 4PP

Message emitted at process 4 : of 2AA 3PA 4PP

Message received from process 4: darth 2AA 3PA

Message processed at process 2 : story 2AA

Message processed at process 1: of 2AA 3PA 4PP 1AP

Message processed at process 3 : not 2AA 3PA

Message emitted at process 3 : not 2AA 3PA

Message received from process 3: a 2AA

Message emitted at process 2 : story 2AA

Message received from process 2: the

Message processed at process 2 : the 2AA

Message processed at process 4 : darth 2AA 3PA 4PP

Message received from process 1: darth 2AA 3PA 4PP

Message emitted at process 4 : darth 2AA 3PA 4PP

Message received from process 4: plagueis 2AA 3PA

Message processed at process 3 : a 2AA 3PA

Message emitted at process 3 : a 2AA 3PA

Message received from process 3: story 2AA

Message emitted at process 2 : the 2AA

Message received from process 2: jedi

Message processed at process 1: darth 2AA 3PA 4PP 1AP

Message processed at process 2 : jedi 2AA

Message processed at process 3 : story 2AA 3PA

Message emitted at process 3 : story 2AA 3PA

Message received from process 3: the 2AA

Message emitted at process 2 : jedi 2AA

Message received from process 2: would

Message processed at process 4 : plagueis 2AA 3PA 4PP

Message received from process 1: plagueis 2AA 3PA 4PP

Message emitted at process 4 : plagueis 2AA 3PA 4PP

Message received from process 4: the 2AA 3PA

Message processed at process 1: plagueis 2AA 3PA 4PP 1AP

Message processed at process 2 : would 2AA

Message processed at process 3 : the 2AA 3PA

Message emitted at process 3 : the 2AA 3PA

Message received from process 3: jedi 2AA

Message emitted at process 2 : would 2AA

Message received from process 2: FIN

Message processed at process 2 : FIN

Message processed at process 4 : the 2AA 3PA 4PP

Message received from process 1: the 2AA 3PA 4PP

Message emitted at process 4 : the 2AA 3PA 4PP

Message received from process 4: wise 2AA 3PA

Message processed at process 1: the 2AA 3PA 4PP 1AP

Message processed at process 3 : jedi 2AA 3PA

Message emitted at process 3 : jedi 2AA 3PA

Message received from process 3: would 2AA

Message emitted at process 2 : FIN

Message processed at process 4 : wise 2AA 3PA 4PP

Message received from process 1: wise 2AA 3PA 4PP

Message emitted at process 4 : wise 2AA 3PA 4PP

Message received from process 4: ? 2AA 3PA

Message processed at process 3 : would 2AA 3PA

Message emitted at process 3 : would 2AA 3PA

Message received from process 3: FIN

Message processed at process 3 : FIN

Message emitted at process 3 : FIN

Message processed at process 1: wise 2AA 3PA 4PP 1AP

Message processed at process 4 : ? 2AA 3PA 4PP

Message received from process 1: ? 2AA 3PA 4PP

Message emitted at process 4 : ? 2AA 3PA 4PP

Message received from process 4: i 2AA 3PA

Message processed at process 1: ? 2AA 3PA 4PP 1AP

Message processed at process 4 : i 2AA 3PA 4PP

Message received from process 1: i 2AA 3PA 4PP

Message emitted at process 4 : i 2AA 3PA 4PP

Message received from process 4: thought 2AA 3PA

Message processed at process 1: i 2AA 3PA 4PP 1AP

Message processed at process 4 : thought 2AA 3PA 4PP

Message emitted at process 4 : thought 2AA 3PA 4PP

Message received from process 1: thought 2AA 3PA 4PP

Message received from process 4: not 2AA 3PA

Message processed at process 1: thought 2AA 3PA 4PP 1AP

Message processed at process 4 : not 2AA 3PA 4PP

Message received from process 1: not 2AA 3PA 4PP

Message emitted at process 4 : not 2AA 3PA 4PP

Message received from process 4: a 2AA 3PA

Message processed at process 1: not 2AA 3PA 4PP 1AP

Message processed at process 4 : a 2AA 3PA 4PP

Message received from process 1: a 2AA 3PA 4PP

Message emitted at process 4 : a 2AA 3PA 4PP

Message received from process 4: story 2AA 3PA

Message processed at process 1: a 2AA 3PA 4PP 1AP

Message processed at process 4 : story 2AA 3PA 4PP

Message emitted at process 4 : story 2AA 3PA 4PP

Message received from process 1: story 2AA 3PA 4PP

Message received from process 4: the 2AA 3PA

Message processed at process 1: story 2AA 3PA 4PP 1AP

Message processed at process 4 : the 2AA 3PA 4PP

Message received from process 1: the 2AA 3PA 4PP

Message emitted at process 4 : the 2AA 3PA 4PP

Message received from process 4: jedi 2AA 3PA

Message processed at process 1: the 2AA 3PA 4PP 1AP

Message processed at process 4 : jedi 2AA 3PA 4PP

Message received from process 1: jedi 2AA 3PA 4PP
Message emitted at process 4 : jedi 2AA 3PA 4PP
Message received from process 4: would 2AA 3PA
Message processed at process 1: jedi 2AA 3PA 4PP 1AP
Message processed at process 4 : would 2AA 3PA 4PP
Message emitted at process 4 : would 2AA 3PA 4PP
Message received from process 1: would 2AA 3PA 4PP
Message received from process 4: FIN
Message processed at process 4 : FIN
Message emitted at process 4 : FIN
Message processed at process 1: would 2AA 3PA 4PP 1AP
Message received from process 1: FIN
Message processed at process 1: FIN

EXECUTION RESULTS

These are the messages received , in order:

Message 1 content: did 2AA 3PA 4PP 1AP
Message 2 content: you 2AA 3PA 4PP 1AP
Message 3 content: ever 2AA 3PA 4PP 1AP
Message 4 content: hear 2AA 3PA 4PP 1AP
Message 5 content: the 2AA 3PA 4PP 1AP
Message 6 content: tragedy 2AA 3PA 4PP 1AP
Message 7 content: of 2AA 3PA 4PP 1AP
Message 8 content: darth 2AA 3PA 4PP 1AP
Message 9 content: plagueis 2AA 3PA 4PP 1AP
Message 10 content: the 2AA 3PA 4PP 1AP
Message 11 content: wise 2AA 3PA 4PP 1AP
Message 12 content: ? 2AA 3PA 4PP 1AP

Message 13 content: i 2AA 3PA 4PP 1AP

Message 14 content: thought 2AA 3PA 4PP 1AP

Message 15 content: not 2AA 3PA 4PP 1AP

Message 16 content: a 2AA 3PA 4PP 1AP

Message 17 content: story 2AA 3PA 4PP 1AP

Message 18 content: the 2AA 3PA 4PP 1AP

Message 19 content: jedi 2AA 3PA 4PP 1AP

Message 20 content: would 2AA 3PA 4PP 1AP

You can see the results in the following file in the outputfiles folder: results-1645501230874.txt

These are the parameters for the buffers and producer consumers

These are the messages that were sent

Process finished with exit code 0