|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_Информатика, искусственный интеллект и системы управления\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_Программное обеспечение ЭВМ и информационные технологии\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по Лабораторной работе №1**

**по курсу**

**«Математические основы верификации ПО»**

**Тема**

**«Знакомство с языком Promela»**

Студент \_\_\_ИУ7-и42М\_\_\_\_ Абди О.У.\_\_\_\_

(Группа) (И.О.Фамилия)

Преподаватель \_\_\_\_Кузнецова О.В.\_\_\_\_

(И.О.Фамилия)

*2024г.*

**Задание**

Для небольшого фрагмента программы необходимо описать модель этой программы на Promela и изучить её (SPIN).

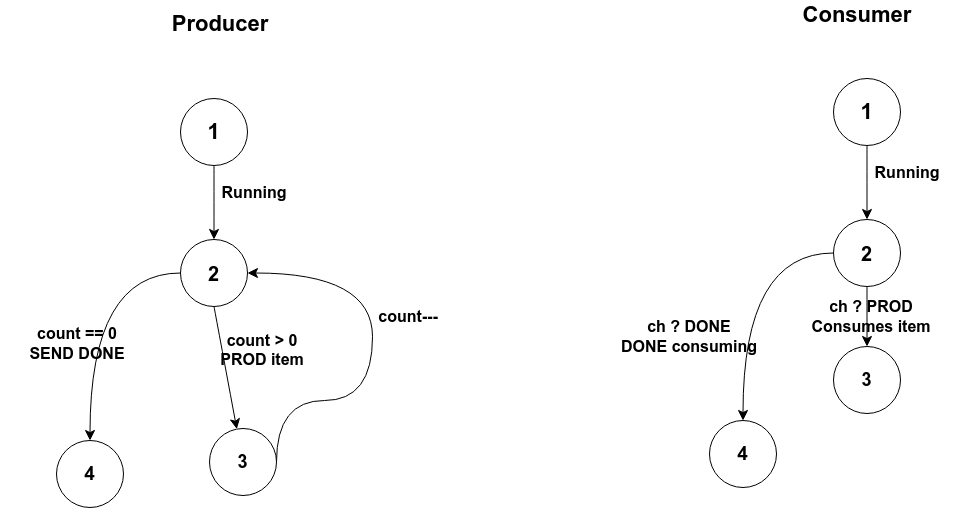
**Описание модели**

Модель описывает систему производителя-потребителя, где процесс Производителя производит элементы и отправляет их процессу Потребителя через канал c. Процесс Производителя запускается с заданным параметром count, производя и отправляя элементы до тех пор, пока count не достигнет 0, затем он отправляет сообщение DONE и завершается. Процесс Потребителя потребляет элементы и завершается при получении сообщения DONE. Модель обеспечивает эффективную коммуникацию с использованием буферизованного канала и конечного параметра count.

**Множество состояний и текстовое объяснение:** Модель имеет следующие состояния:

* **Производитель:** работает, производство | отправка, производство завершено.
* **Потребитель:** работает, потребление, потребление завершено.

Процесс Производителя начинает в состоянии "работает", создает элементы и переходит в состояние "производство". Когда счетчик достигает 0, он переходит в состояние "производство завершено". Процесс Потребителя начинает в состоянии "работает", потребляет элементы и переходит в состояние "потребление". Когда он получает сообщение DONE, он переходит в состояние "потребление завершено". Диаграммы состояний производителя и потребителя показаны на рисунке 1.

Рисунок 1 - Диаграмма переходов состояний Производителя и Потребителя.

**Фрагмент кода**

Фрагменты 1 и 2 предоставляют описания моделей производителя и потребителя соответственно

Листинг 1 — Код модели производителя.

|  |
| --- |
| proctype Producer(chan c; int count)  {  byte item;  printf("PRODUCER: running, pid=%d\n", \_pid);  do  :: count > 0 ->  item = count;  printf("PRODUCER: producing item %d\n", item);  c ! PROD, item;  count = count - 1;  :: count == 0 ->  printf("PRODUCER: done producing\n");  c ! DONE;  break;  od  } |

Листинг 2 — Код модели потребителя.

|  |
| --- |
| proctype Consumer(chan c)  {  byte item;  printf("CONSUMER: running, pid=%d\n", \_pid);  do  :: c ? PROD, item ->  printf("CONSUMER: consuming item %d\n", item);  :: c ? DONE ->  printf("CONSUMER: done consuming\n");  break;  od  } |