*Государственное образовательное учреждение высшего профессионального образования*

*«Московский Государственный Технический Университет имени Н.Э. Баумана»*



**ОТЧЁТ**

**По** **математическим основам верификации ПО**

Лабораторная работа № 3

*Тема:* « ***Моделирование сетевого протокола*** »

Студент: Нгуен Дык Бинь

Группа: ИУ7-41М

Преподаватель: Кузнецова О.В.

***Москва, 2020***

## Задание

На языке promela необходимо описать прототип сетевого протокола, реализующего отправление и получение данных.

В лабораторной работе производим запись и чтение в/из канал/а.

## Текст программы

mtype = {msg, ack};

chan s\_r = [2] of {mtype, bit};

chan r\_s = [2] of {mtype, bit};

active proctype sender() {

bit seqno;

do

:: s\_r!msg, seqno ->

if

:: r\_s?ack, eval(seqno) -> // Считываем новое сообщение

seqno = 1 - seqno;

:: r\_s?ack, eval(1 - seqno)

fi

od

}

active proctype receiver() {

bit expect, seqno;

do

:: s\_r?msg, seqno ->

r\_s!ack, seqno;

if

:: seqno == expect; // Сохраняем сообщение

printf("Expected!\n");

expect = 1 - expect

:: else // Игнорируем сообщение

fi

od

}

## Подробное описание

* В программе 2 процесса: получатель и отправитель.
* Тип сообщений:
* Отправка сообщения через команду:
* Приём сообщения через команду:
* К каждому сообщению добавляется один бит.
* Получатель сообщает о доставке сообщения, возвращая бит отправителя.
* Если отправитель убедился в доставке сообщения, он отправляет новое, изменяя значения бита:
* Если значение бита не изменилось, получатель считает, что идёт повтор сообщения.
* В данной программе используется функция ***eval(х)****,* она отображает текущее значение ***х*** на константу (0 или 1), которая служит ограничением для принимаемых сообщений.
* Процесс передачи сообщений:

msg(0)

ack(0)

msg(1)

ack(1)

...

## Пример работы программы

* Приведен результат моделирования 20х первых шагов:

~# spin -u20 -c channel.pml

proc 0 = sender

proc 1 = receiver

q\p 0 1

1 s\_r!msg,0

1 . s\_r?msg,0

2 . r\_s!ack,0

2 r\_s?ack,0

Expected!

1 s\_r!msg,1

1 . s\_r?msg,1

2 . r\_s!ack,1

2 r\_s?ack,1

Expected!

-------------

depth-limit (-u20 steps) reached

-------------

final state:

-------------

#processes: 2

queue 1 (s\_r):

queue 2 (r\_s):

20: proc 1 (receiver:1) channel.pml:25 (state 5)

20: proc 0 (sender:1) channel.pml:15 (state 8)

2 processes created

* Граф состояний (-u20)

