Отчет по лабораторной работе №2

по дисциплине:
«Математические основы верификации ПО»

Студент: Балякин Д.В.

Группа: ИУ7-42М

Преподаватель: Кузнецова О.В.

Москва, 2020

Описание задания:

На языке Promela описать модель взаимодействия двух процессов, разделяющих ресурсы.

Код программы

```
#define loop 30
int count = 0;

proctype inc( ) {
   int c = loop
   do
       :: (c > 0) -> count=count + 1; c--;
       :: (c == 0) -> break;
   od;
}

active proctype print( ) {
   int c = loop
   do
       :: (c > 0) ->
           if :: (count % 2 == 0) ->
               printf("MSC: count: %d\n", count);
           fi;
           c--;
           :: (c == 0) ->  break;
   od;
}

init {
   run inc();
   run inc();
   run inc();
}
```

Выполнение программы

```
      MSC: count: 6
      MSC: count: 14
      MSC: count: 18
      MSC: count: 22
      MSC: count: 28
      MSC: count: 33
      MSC: count: 44
      MSC: count: 54
      MSC: count: 61
      MSC: count: 66
      MSC: count: 68
      MSC: count: 80
      MSC: count: 82
      MSC: count: 86
      MSC: count: 88
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
      MSC: count: 90
5 processes created
```

Как видно из вывода программы, процесс вывода выводит не только четные, но и нечетные числа, что говорит о присутствии гонки процессов.

Код программы с устраненной гонкой потоков

```
#define loop 30
int count = 0;

bool mutex = true;

inline wait(sem) {
   atomic {
       sem;
       sem=false;
   }
}

inline signal(sem) {
   sem = true
}
```

```
proctype inc( ) {
   int c = loop;
   do
       :: (c > 0) ->
           wait(mutex);
           int tmp;
           tmp = count;
           count=tmp + 1;
           signal(mutex); c--;
       :: (c == 0) -> break;
   od;
}

active proctype print( ) {
   int c = loop
   do
       :: (c > 0) ->
           wait(mutex);
           if :: (count % 2 == 0) ->
               printf("MSC: count: %d\n", count);
              :: (count % 2 != 0) -> skip
           fi;
           signal(mutex);
           c--;
           :: (c == 0) ->  break;
   od;
}

init {
   atomic {
   run inc();
   run inc();
   }
}
```

Выполнение программы