



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 и системы управления»

КАФЕДРА _____ «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №2 по курсу «Математические основы верификации ПО»

Тема Моделирование гонки процессов

Студент Якуба Д. В.

Группа ИУ7-43М

Оценка (баллы) _____

Преподаватели Кузнецова О.В.

Москва — 2024 г.

Оглавление

Введение	3
1 Выполнение	4
1.1 Описание модели взаимодействия процессов	4
1.2 Описание модели с мьютексом	7
ЗАКЛЮЧЕНИЕ	10

ВВЕДЕНИЕ

Цель работы – на примере гонки процессов научиться описывать взаимодействие двух процессов, работающих с одними данными.

Для достижения поставленной цели потребуется:

- описать модель взаимодействия процессов;
- продемонстрировать логи SPIN, в которых видна гонка;
- описать модель с мьютексом;
- привести результат корректного взаимодействия процессов.

1. Выполнение

1.1 Описание модели взаимодействия процессов

Описание модели представлено в листинге 1.

Листинг 1: Описание модели взаимодействия процессов.

```
1  short var1 = 3;
2  short var2 = 2;
3  short res1 = 0;
4
5  byte busy = 0;
6
7  proctype Thread1() {
8      (busy == 0) -> busy++;
9      assert(busy > 1);
10
11      var1++;
12      var2++;
13
14      busy--;
15  }
16
17  proctype Thread2() {
18      (busy == 0) -> busy++;
19      assert(busy > 1);
20
21      var1++;
22      res1 = var1 + var2;
23
24      busy--;
25  }
26
27  init {
28      run Thread1();
29      run Thread2();
30  }
```

На рисунках 1.1-1.3 продемонстрированы логи SPIN, в которых видна гонка.

```

> ../spin lab.pml
spin: lab.pml:19, Error: assertion violated
spin: text of failed assertion: assert((busy>1))
#processes: 3
        var1 = 3
        var2 = 2
        res1 = 0
        busy = 1
5:   proc  2 (Thread2:1) lab.pml:19 (state 3)
5:   proc  1 (Thread1:1) lab.pml:8 (state 1)
5:   proc  0 (:init::1) lab.pml:30 (state 3) <valid end state>
3 processes created

```

Рис. 1.1: Логи SPIN, демонстрирующие гонку.

```

> ./a.out
pan:1: assertion violated (busy>1) (at depth 4)
pan: wrote lab.pml.trail

(Spin Version 6.5.2 -- 15 February 2024)
Warning: Search not completed
        + Partial Order Reduction

Full statespace search for:
        never claim           - (none specified)
        assertion violations   +
        acceptance  cycles    - (not selected)
        invalid end states    +

State-vector 36 byte, depth reached 4, errors: 1
        5 states, stored
        0 states, matched
        5 transitions (= stored+matched)
        0 atomic steps
hash conflicts:          0 (resolved)

Stats on memory usage (in Megabytes):
        0.000      equivalent memory usage for states (stored*(State-vector + overhead))
        0.291      actual memory usage for states
       128.000     memory used for hash table (-w24)
        0.534      memory used for DFS stack (-m10000)
       128.730     total actual memory usage

pan: elapsed time 0 seconds

```

Рис. 1.2: Логи SPIN, демонстрирующие гонку.

```

> ../spin -p -t lab.pml
Starting Thread1 with pid 1
  1:   proc  0 (:init::1) lab.pml:28 (state 1) [(run Thread1())]
Starting Thread2 with pid 2
  2:   proc  0 (:init::1) lab.pml:29 (state 2) [(run Thread2())]
  3:   proc  2 (Thread2:1) lab.pml:18 (state 1)      [((busy==0))]
  4:   proc  2 (Thread2:1) lab.pml:18 (state 2)      [busy = (busy+1)]
spin: lab.pml:19, Error: assertion violated
spin: text of failed assertion: assert((busy>1))
  5:   proc  2 (Thread2:1) lab.pml:19 (state 3)      [assert((busy>1))]
spin: trail ends after 5 steps
#processes: 3
      var1 = 3
      var2 = 2
      res1 = 0
      busy = 1
  5:   proc  2 (Thread2:1) lab.pml:21 (state 4)
  5:   proc  1 (Thread1:1) lab.pml:8 (state 1)
  5:   proc  0 (:init::1) lab.pml:30 (state 3) <valid end state>
3 processes created

```

Рис. 1.3: Логи SPIN, демонстрирующие гонку.

Если не использовать ассерты, можно также отметить большую достигнутую глубину и количество результирующих состояний, на рисунке 1.4 представлены такие логи.

```

) ./a.out

(Spin Version 6.5.2 -- 15 February 2024)
+ Partial Order Reduction

Full statespace search for:
    never claim          - (none specified)
    assertion violations  +
    acceptance cycles    - (not selected)
    invalid end states   +

State-vector 36 byte, depth reached 15, errors: 0
    46 states, stored
    19 states, matched
    65 transitions (= stored+matched)
    0 atomic steps
hash conflicts:          0 (resolved)

Stats on memory usage (in Megabytes):
    0.003      equivalent memory usage for states (stored*(State-vector + overhead))
    0.291      actual memory usage for states
   128.000     memory used for hash table (-w24)
    0.534      memory used for DFS stack (-m10000)
   128.730     total actual memory usage

unreached in proctype Thread1
    (0 of 6 states)
unreached in proctype Thread2
    (0 of 6 states)
unreached in init
    (0 of 3 states)

pan: elapsed time 0 seconds

```

Рис. 1.4: Логи SPIN, демонстрирующие гонку без ассертов.

1.2 Описание модели с мьютексом

Описание модели представлено в листинге 2.

Листинг 2: Описание модели с мьютексом.

```

1    short var1 = 3;
2    short var2 = 2;
3    short res1 = 0;
4
5    byte mutex = 0;
6
7    proctype Thread1() {
8        atomic {
9            if
10               :: mutex < 1 ->
11               mutex++;

```

```

12             var1++;
13             var2++;
14             mutex--;
15         fi
16     }
17 }
18
19 proctype Thread2() {
20     atomic {
21         if
22             :: mutex < 1 ->
23             mutex++;
24             res1 = var1 + var2;
25             mutex--;
26         fi
27     }
28 }
29
30 init {
31     atomic {
32         run Thread1();
33         run Thread2();
34     }
35 }

```

На рисунке 1.5 представлен результат корректного взаимодействия процессов.


```

) ./a.out

(Spin Version 6.5.2 -- 15 February 2024)
+ Partial Order Reduction

Full statespace search for:
    never claim           - (none specified)
    assertion violations  +
    acceptance  cycles   - (not selected)
    invalid end states    +

State-vector 36 byte, depth reached 7, errors: 0
    9 states, stored
    1 states, matched
    10 transitions (= stored+matched)
    1 atomic steps
hash conflicts:          0 (resolved)

Stats on memory usage (in Megabytes):
    0.001    equivalent memory usage for states (stored*(State-vector + overhead))
    0.290    actual memory usage for states
   128.000   memory used for hash table (-w24)
    0.534    memory used for DFS stack (-m10000)
   128.730   total actual memory usage

unreached in proctype Thread1
    (0 of 9 states)
unreached in proctype Thread2
    (0 of 8 states)
unreached in init
    (0 of 4 states)

pan: elapsed time 0 seconds

```

Рис. 1.5: Результат корректного взаимодействия процессов.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы были получены навыки описания взаимодействия двух процессов, работающих с одними данными на ЯП Promela.

Были решены следующие задачи:

- описана модель взаимодействия процессов;
- продемонстрированы логи SPIN, в которых видна гонка;
- описана модель с мьютексом;
- приведен результат корректного взаимодействия процессов.