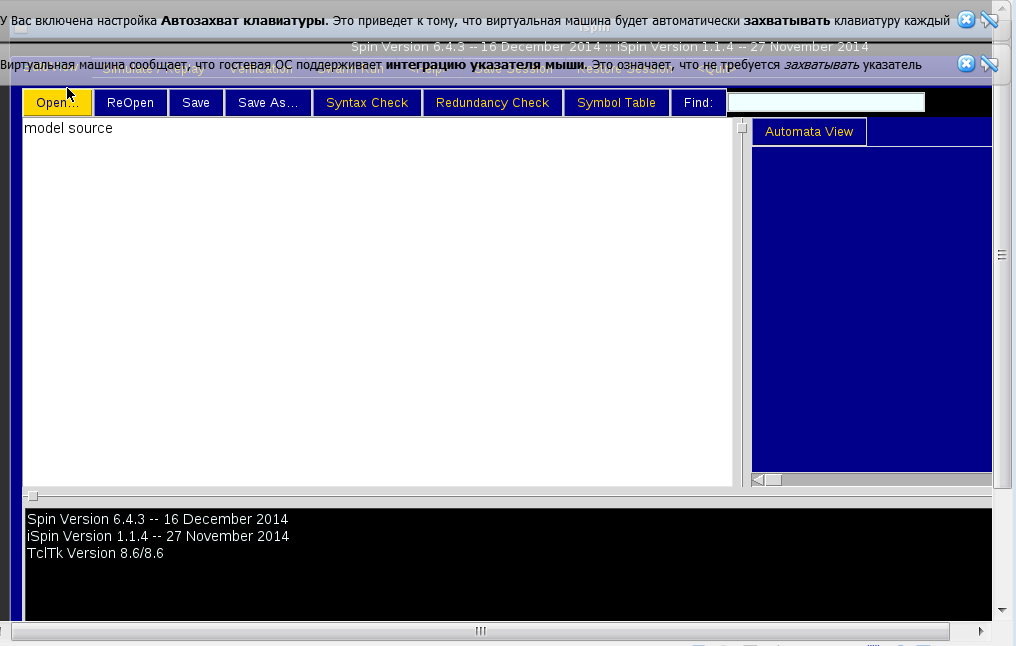
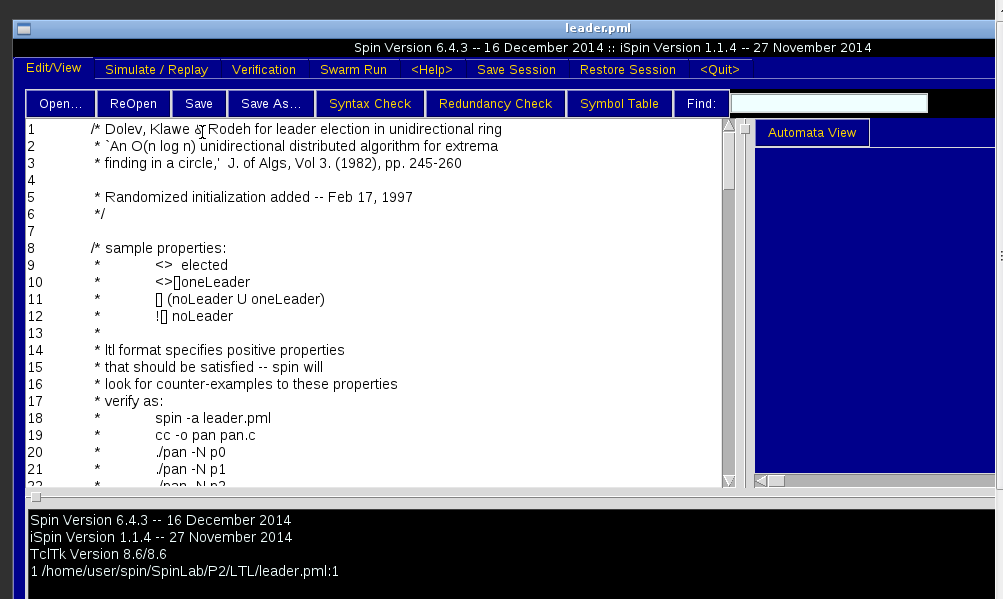
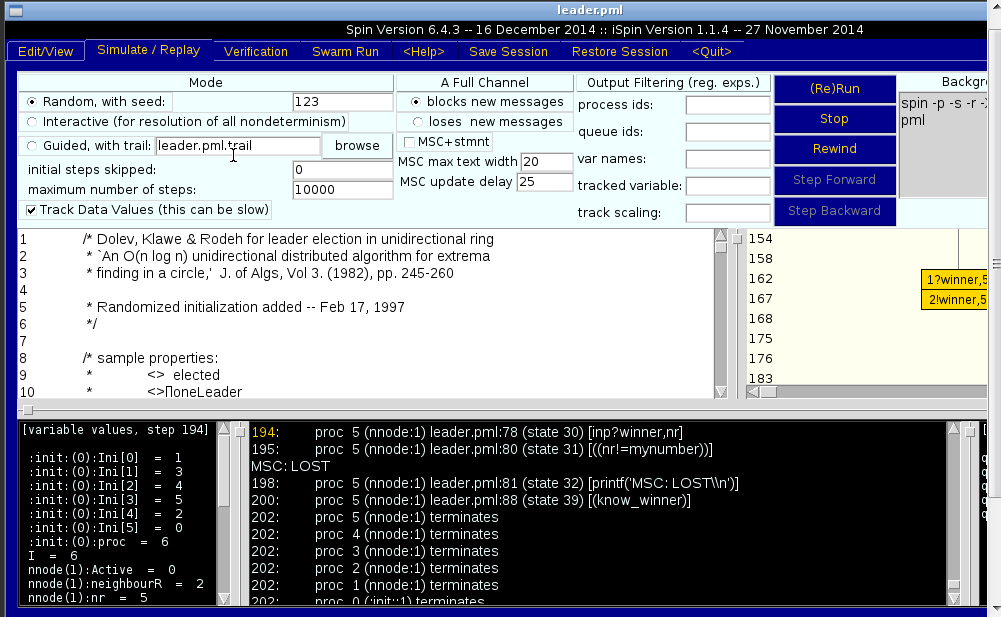
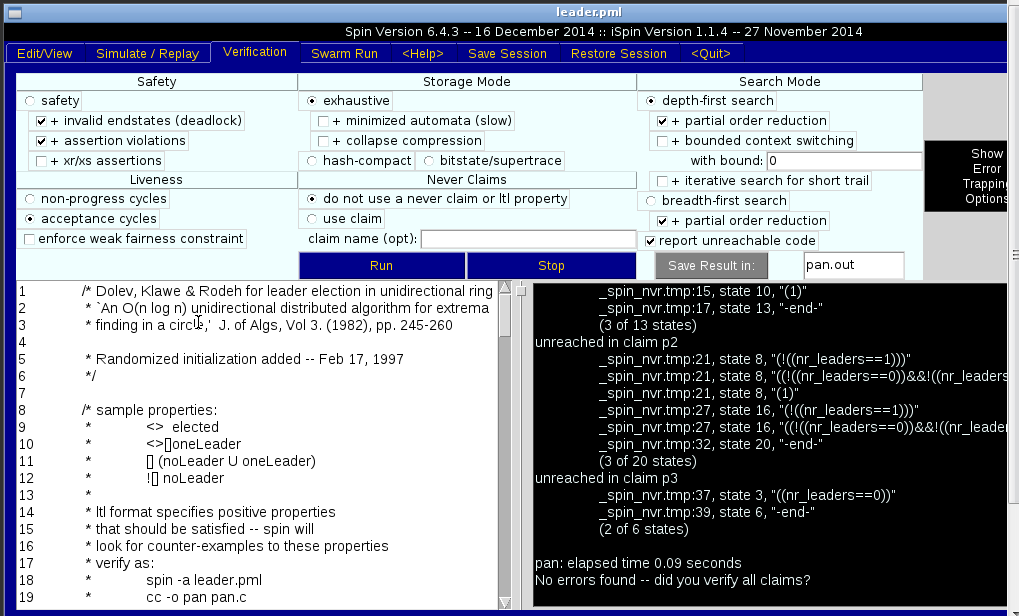
После запуска команды ispin.

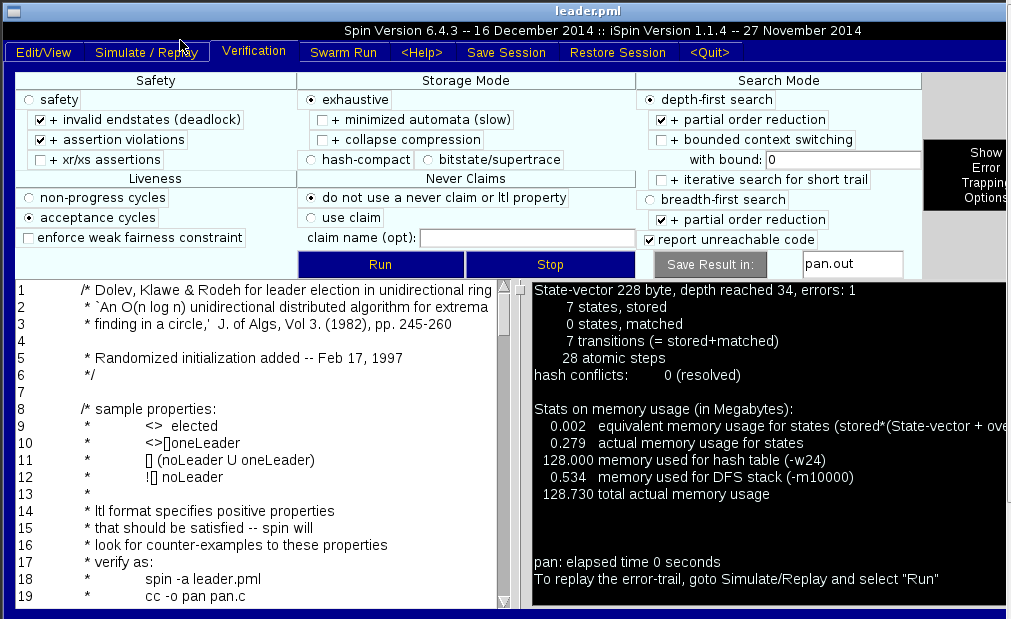


После открытия leader.pml  


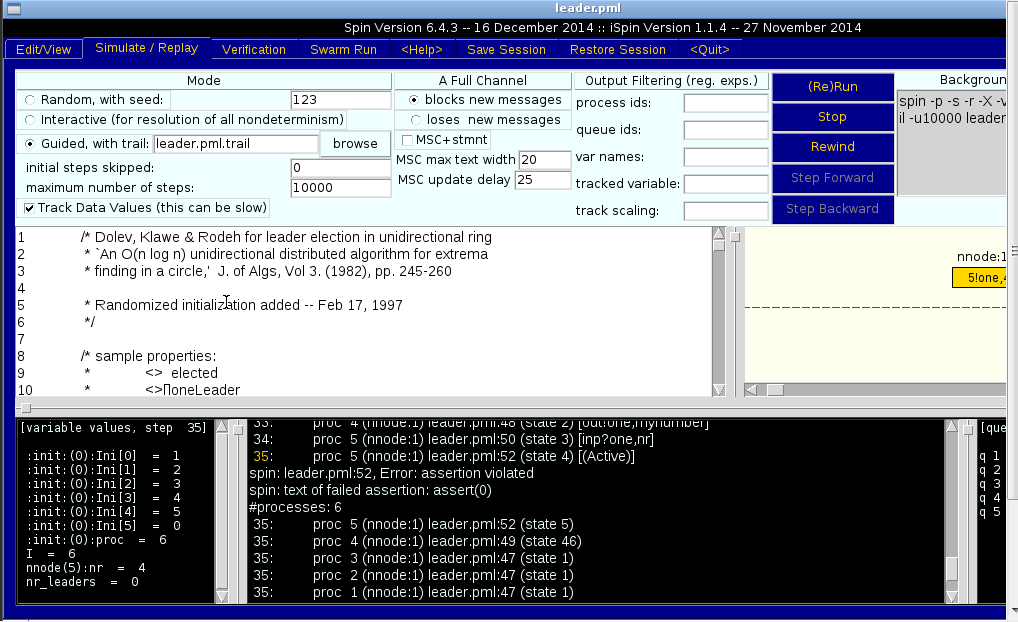
После запуска симуляции:  


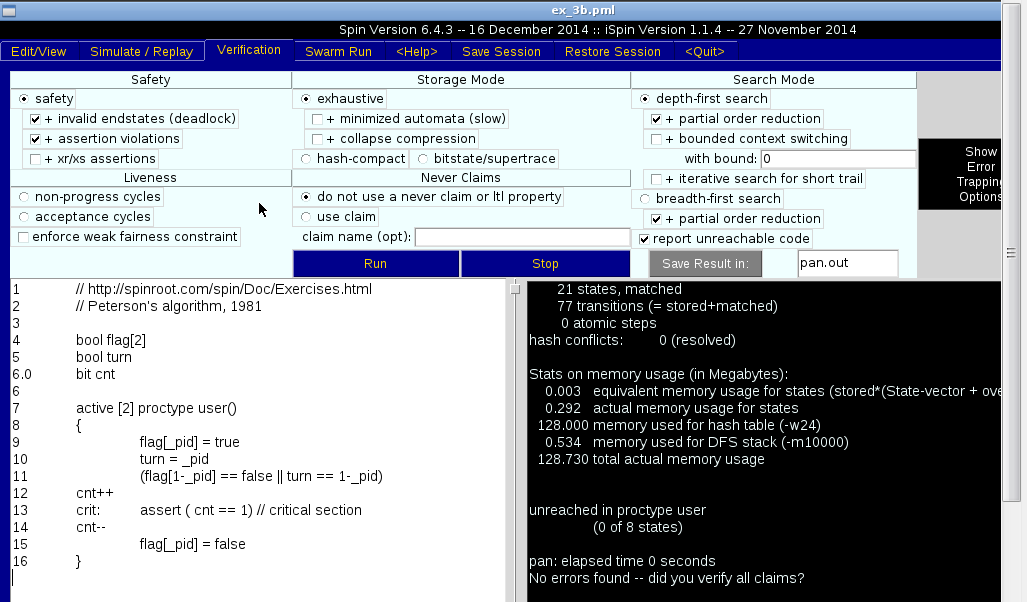
После запуска процесса верификации.  


После внесения ошибки в текст, результат верификации:



Ручной режим при наличии ошибки:



Решение задачи обеспечения взаимного исключения прочесов в критической области, предложенное Петерсоном, результат верификации.  


bool flag[2] // индикаторы заинтересованности процессов

bool turn // флаг очереди исполнения

*bit count // переменная-счётчик количества процессов в критической секции.*

active [2] proctype user() // запуск сразу 2 процессов

{

flag[\_pid] = true // заинтересованность в текущем процессе

turn = \_pid // на очереди исполнения текущий поток

(flag[1-\_pid] == false || turn == 1-\_pid) // цикл ожидания, если какой-то процесс уже занял критическую секцию (если заинтересованности в противоположном процессе нет ИЛИ исполняются текущий поток и противоположный процесс)

count++ // увеличиваем количество процессов

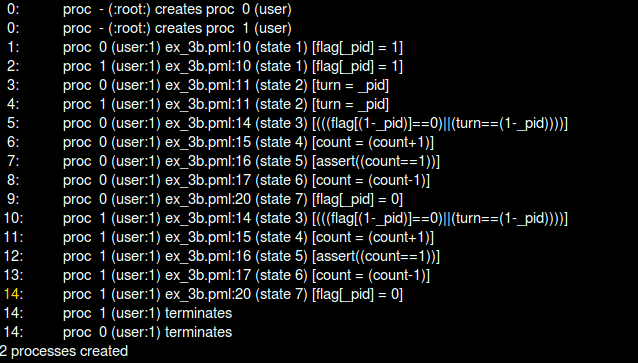
crit: assert(count==1) // сравниваем количество процессов в критической секции

count-- // при выходе из критической секции уменьшаем количество процессов

flag[\_pid] = false // сбрасываем флаг заинтересованности

}

Лог сообщений симуляции



Первый создает 0 процесс, а затем 1 процесс. Устанавливаем флаги заинтересованности для этих процессов. Затем указываем что оба процесса будут исполняемыми. Т.к. оба процесса исполняемыми, то первый из процессов занимает критическую секцию, а второй процесс ожидает выполнения первого. После прохождения критической секции, сбрасываем флаг заинтересованности с первого процесса.

Лог верификации показывает, что ошибок при выполнении программы не найдено и происходит взаимное исключение процессов.

Дополнительные вопросы. Переменная cnt может иметь тип bit, т.к. для взаимного исключения процессов необходимо, чтобы в критической секции находился 1 процесс, а bit принимает значение 0..1. В случае, если значение bit превысит 1 (количество процессов в критической секции будет больше 1), система сгенерирует ошибку. Также переменная cnt может иметь тип int, но это приведет к неоправданному использованию памяти. Переменная cnt должна быть глобальной, чтобы можно было отследить количество процессов, которые будут находиться в критической секции.