

Задача "Алгоритм Петерсона [peterson]", Ткаченко Дмитрий, команда PQ.

Решение: Приведем фрагмент кода и последовательность выполнения инструкций на процессоре, доказывающую, что при замене мест операций в doorway-секции, алгоритм Петерсона перестает гарантировать принцип взаимного исключения:

```
void lock(int t) {  
  //1 victim.store(t);  
  //2 want[t].store(true);  
  while (want[1-t].load() == victim.load()) {  
    //wait  
  }  
}
```

Здесь комментарии 1 и 2 будут означать выполнение данной строчки кода. Инструкцию процессора зададим видом n-k, где n - номер потока, который будет выполнять команду k (это 1, 2 и w - попытка захода в wait-секцию, то есть проверка условий цикла while).

Приведем же наконец порядок выполнения инструкций на процессоре:

- 1) 0 - 1 (want[0] == false, want[1] == false, victim == 0)
- 2) 1 - 1 (want[0] == false, want[1] == false, victim == 1)
- 3) 1 - 2 (want[0] == false, want[1] == true, victim == 1)
- 4) 1 - w (want[0] == false, want[1] == true, victim == 1). Попытка захода в wait-секцию, условия цикла while не выполнены (want[0] == false) и поток 1 заходит в критическую секцию.
- 5) 0 - 2 (want[0] == true, want[1] == true, victim == 1)
- 6) 0 - w (want[0] == true, want[1] == true, victim == 1). Попытка захода в wait-секцию, условия цикла while не выполнены (victim == 1) и поток 0 заходит в критическую секцию.

Принцип взаимного исключения нарушен.

Ответ: Нет, не будет.