

# Задача 8. Разбор алгоритма разрыва узла для $n$ процессов (на основе листинга из лекции и учебника Андрюса).

Исследование провёл студент группы 22207 Гордеев Никита

Дата выполнения работы 25.12.2022 (Вариант 2)

# Задачи

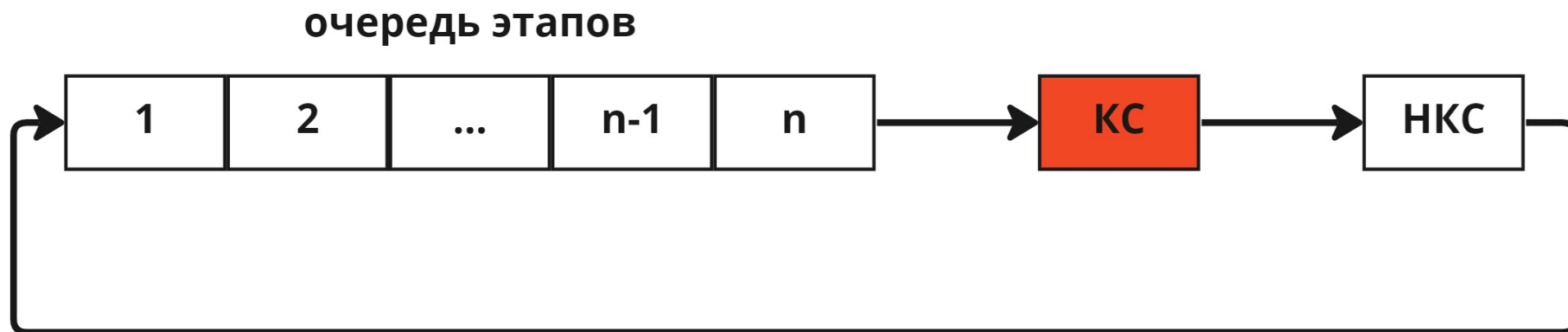
- Разбор алгоритма разрыва узла для  $n$  процессов (на основе листинга из лекции и учебника Андрюса)

## Обозначение

- $\text{int } n$  – число процессов
- $\text{int in}[1:n]$  – массив, хранящий этап ожидания для каждого процесса
- $\text{int last}[1:n]$  – массив, хранящий номер процесса на соответствующем этапе

# Описание алгоритма

- Для того, чтобы КС выполнялась только одним процессом, реализована очередь этапов ожидания. Каждый процесс должен пройти  $n$  этапов, чтобы войти в КС.

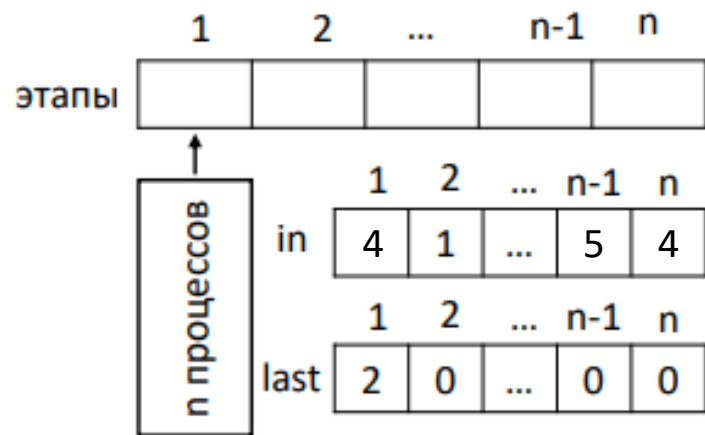


# Программа разрыва узла для n процессов

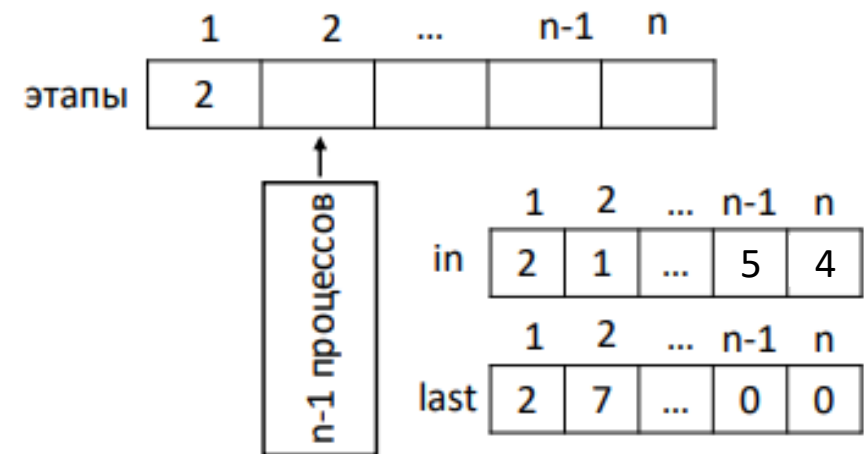
```
int in[1:n] = ([n] 0), last[1:n] = ([n] 0);  
process CS[i = 1 to n] {  
  while (true) {  
    for [j = 1 to n] {      /* протокол входа */  
      /* запомним, что процесс i находится на этапе j  
        и там является последним */  
      last[j] = i; in[i] = j;  
      for [k = 1 to n st i != k] {  
        /* ждать, если процесс k находится на этапе с большим номером  
          и процесс i был последним из прошедших на этап j */  
        while (in[k] >= in[i] and last[j] == i) skip;  
      }  
    }  
    критическая секция;  
    in[i] = 0;          /* протокол выхода */  
    некритическая секция;  
  }  
}
```

- Int n – число процессов
- in[1:n] – массив, хранящий этап ожидания для каждого процесса
- last[1:n] – массив, хранящий номер процесса на соответствующем этапе.
- in[i] - этап KC[i]
- last[j] – кто выполняет этап j
- внешний for выполняется n-1 раз
- внутренний for CS[i] проверяет остальные процессы
- не более n-1 процессов могут пройти первый этап, n-2 второй и так далее.
- Гарантирует, что пройти все n этапов и выполнять свою критическую секцию процессы могут только по одному.

# Схема работы 1



На первый этап входят все  $n$  процессов.  
Процесс с номером 2 зашел последним.  
Все остальные процессы проходят на следующий этап.



На второй этап входят  $n-1$  процессов, все, кроме 2-го.  
Процесс с номером 7 зашел последним.  
Все остальные процессы проходят на следующий этап.

# Схема работы 2



# Вывод:

- Решение для  $n$  процессов свободно от состояний активного тупика, избегает ненужных задержек и гарантирует возможность входа.
- Избегает ненужных задержек и гарантирует возможность входа
- Данный процесс задерживается, только если некоторый другой процесс находится в протоколе входа впереди данного, и из предположения, что каждый процесс в конце концов выходит из своей критической секции.

# Материалы:

- **3.3.1 Алгоритм разрыва узла // Грегори Р. Эндрюс - Основы многопоточного, параллельного и распределенного программирования (дата обращения: 11.12.2022).**



# Изменения

- **Версия 2**
  - Добавил схемы работы процессов
  - Изменил код программы