Задача 7. Задача на понимание и обоснование свойств решения критической секции, рассмотренного на лекции.

Исследование провёл студент группы 22207 Гордеев Никита

Дата выполнения работы 25.12.2022 (Вариант 2)

Задачи

Используется циклический замок на основе машинной инструкции Test-and-Set (bool TS(bool)).

- **Обосновать выполнение свойств 1-3 критической секции**
- **Показать, что свойство 4 критической секции не гарантируется.**

Циклический замок Test-and-Set

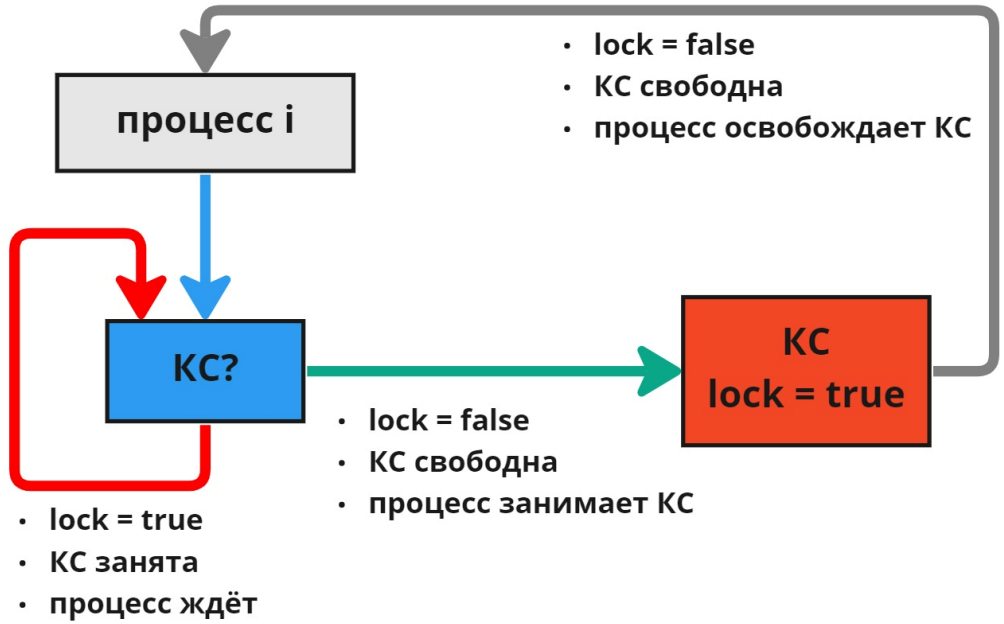
- **Инструкция проверить и установить.**
- **В соответствии с методом, каждый процесс ждёт, чтобы текущее значение замка стало ложным. Если условие выполняется, замок устанавливается в значение true, и процесс входит в критическую секцию**

Код исходной программы

```
process CS[i = 1 to n] {
    while (1) {
        while (TS(lock)); # вход
            КС;
        lock = 0; # выход

        НКС;
    }
}

bool TS(bool lock) {
    <bool init = lock; # запомнить входной аргумент
    lock = 1; # установить значение
    return init;> # вернуть исходное значение
}
```



Свойство 1

“Взаимное исключение”

Суть: **В любой момент времени только один процесс может выполнять свою критическую секцию.**

Условия выполнения: **при попытке нескольких процессов войти в критическую секцию только один из них сможет первым изменить значение lock, поэтому остальные войти в неё не смогут**

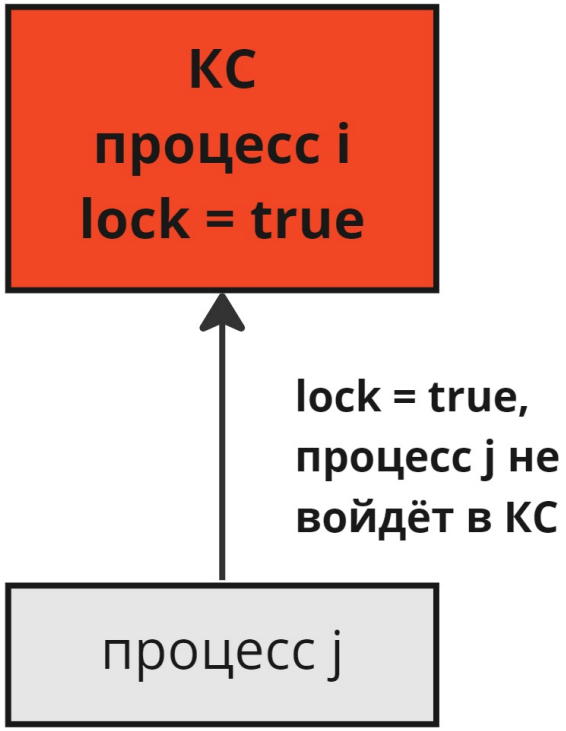
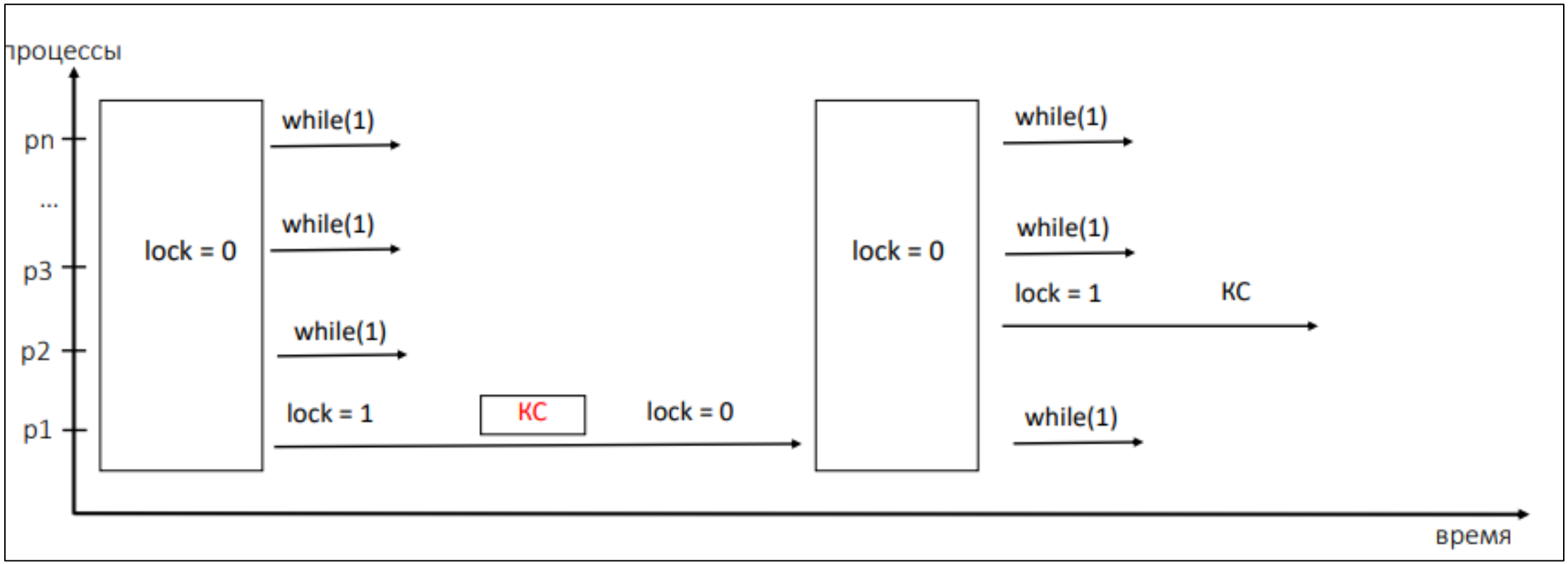


Схема свойства 1



Свойство 2

“Отсутствие взаимной блокировки” (живая блокировка)

Суть: **Если несколько процессов пытаются войти в свои критические секции, хотя бы один это осуществит**

Условие выполняется: **Если несколько процессов находятся в протоколе входа, то lock ложна, следовательно хотя бы один процесс войдёт в критическую секцию.**

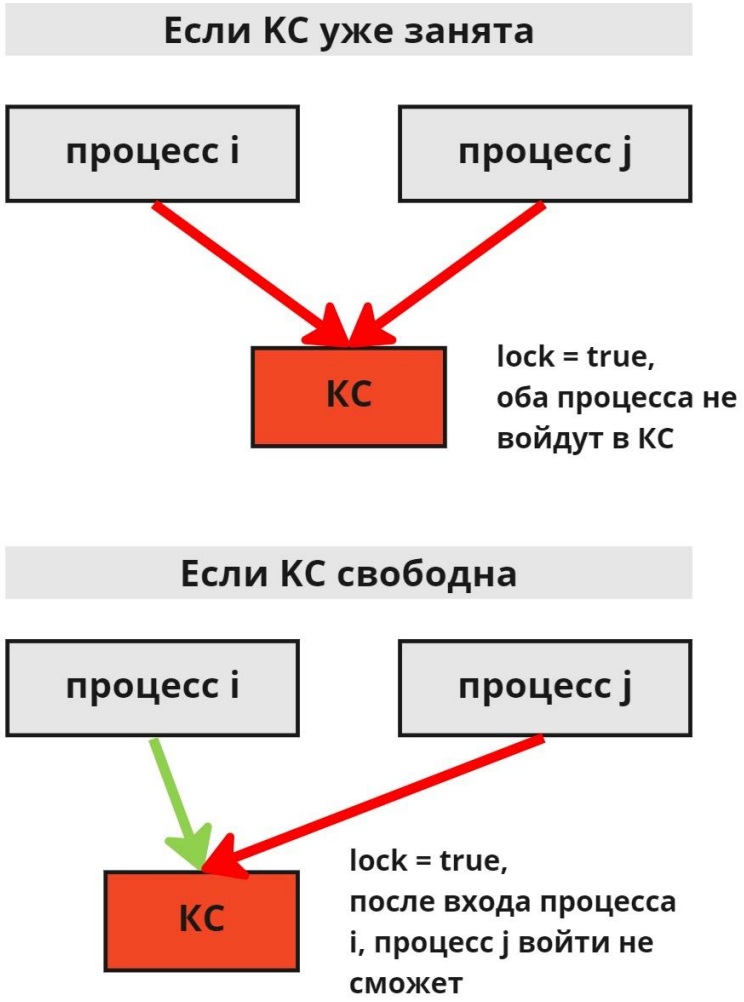
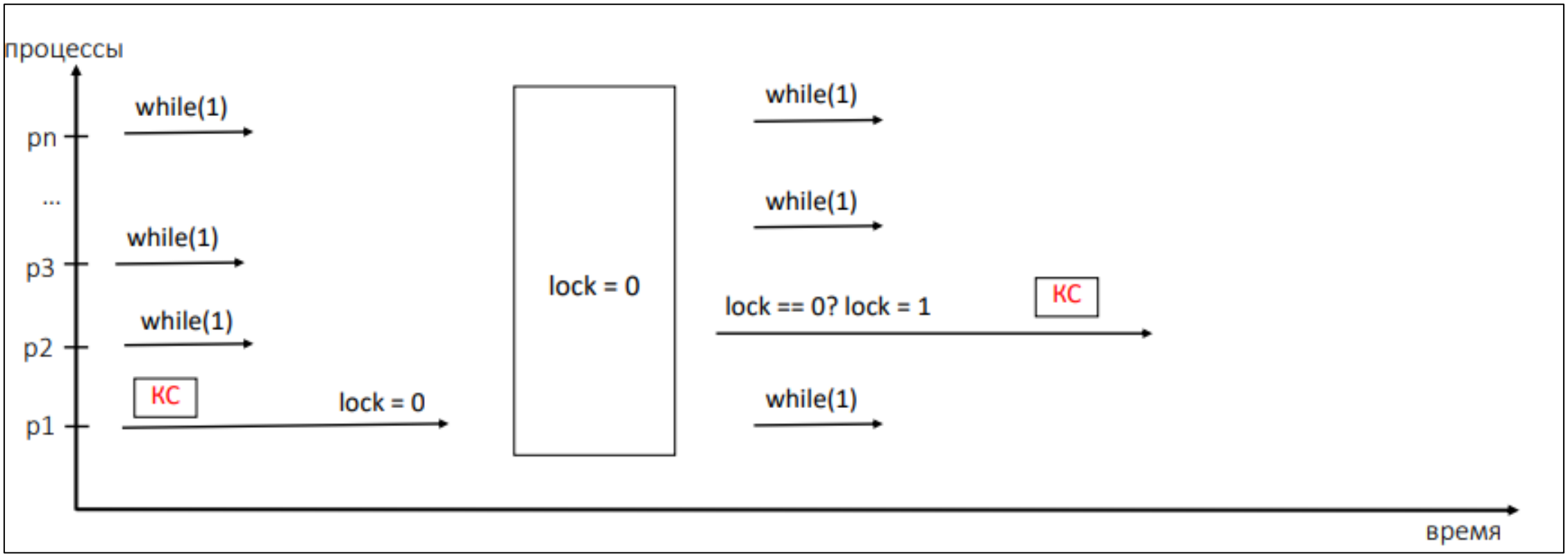


Схема свойства 2



Свойство 3

“Отсутствие излишних задержек”

Суть: **Если один процесс пытается войти в свою критическую секцию, а другие выполняют свои некритические секции или завершены, первому процессу разрешается вход в критическую секцию.**

Условие выполняется: **если остальные процессы вышли из критической секции, то lock = false, следовательно данный процесс сможет войти в критическую секцию.**

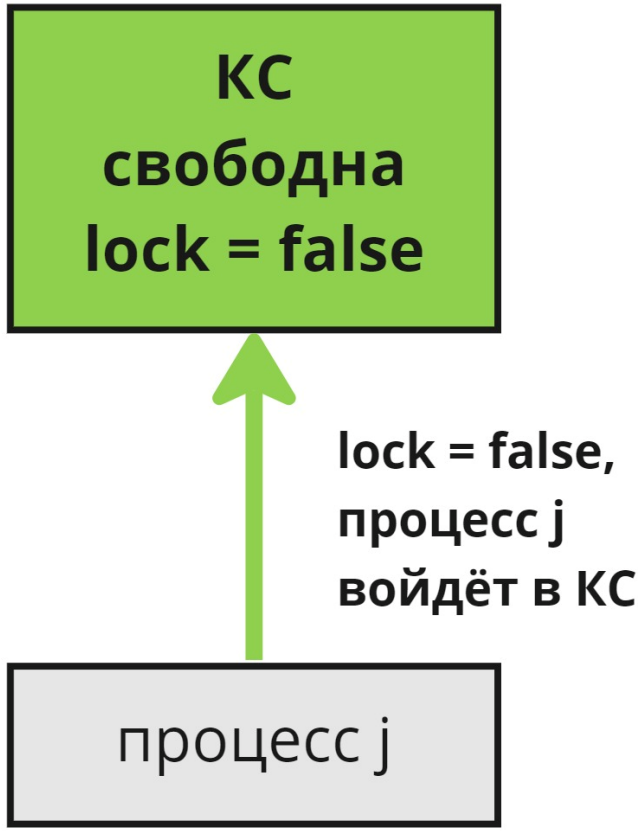
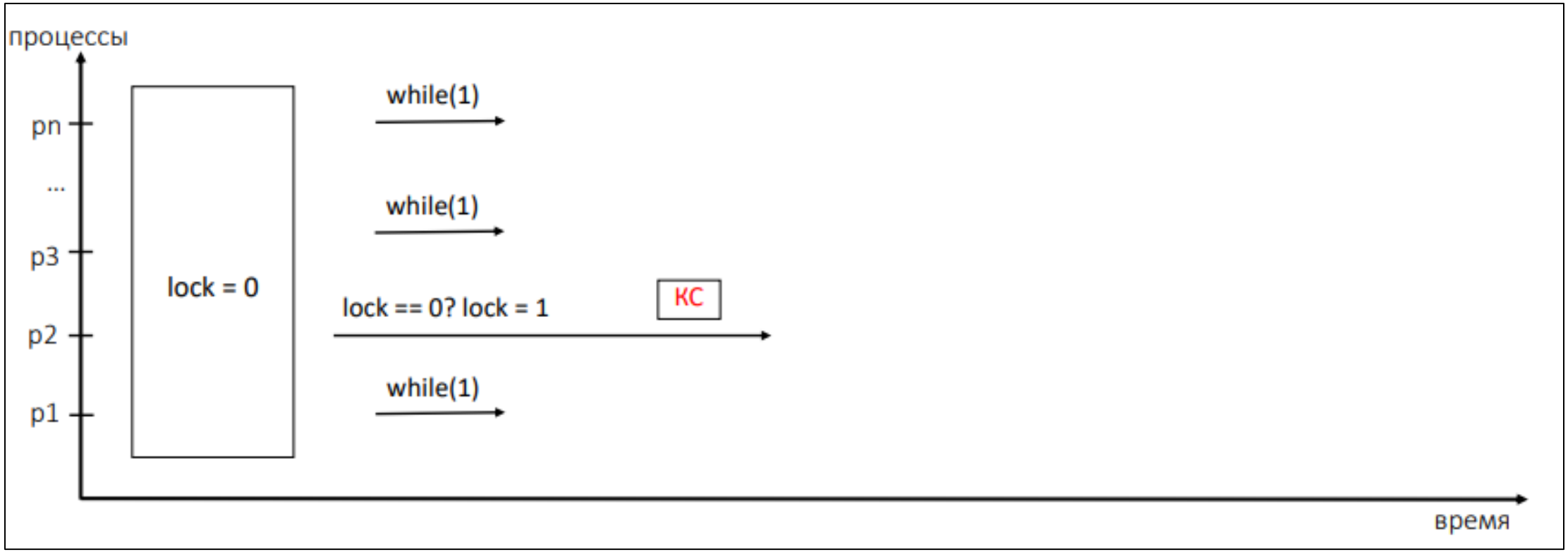


Схема свойства 3



Свойство 4

“Возможность входа”

Суть: **Процесс, который пытается войти в критическую секцию, когда-нибудь это сделает.**

Условие не гарантируется: **если некоторые процессы более приоритетны, то есть в них проверка будет выполняться быстрее чем в других, то такие процессы могут вечно блокировать менее приоритетный процесс**

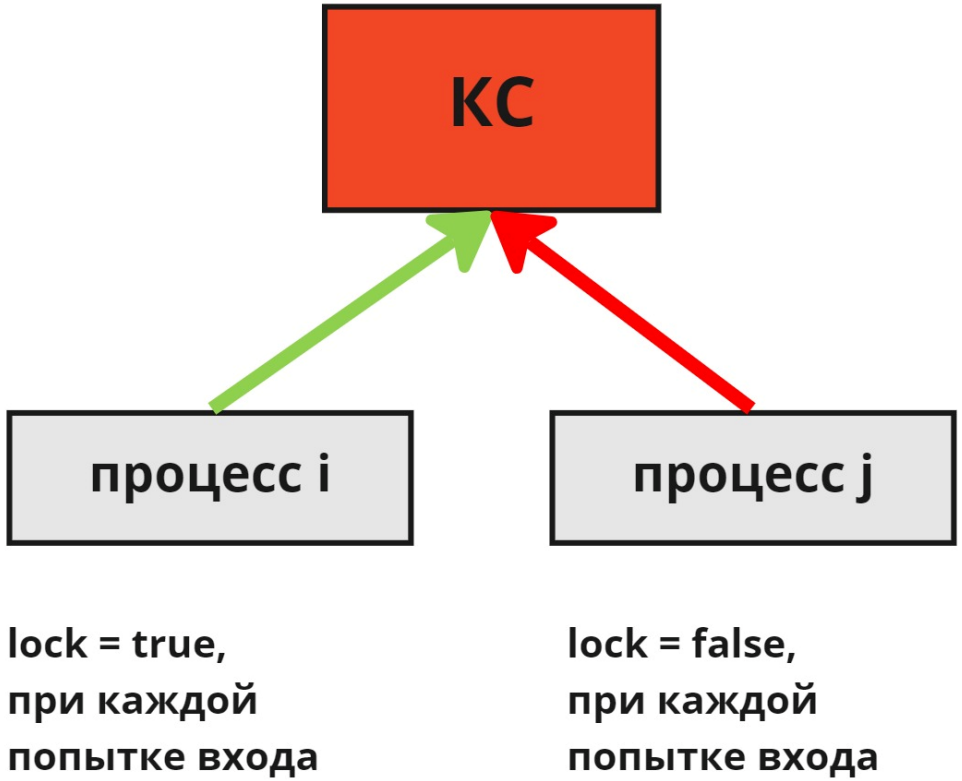
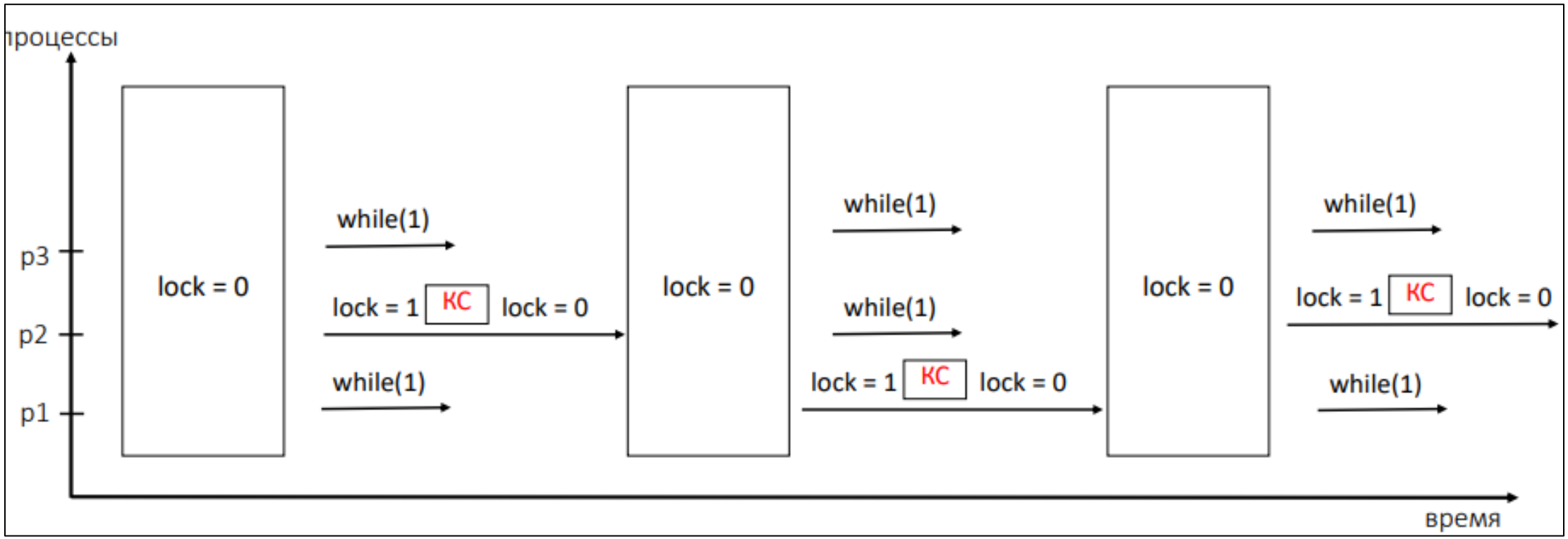


Схема свойства 4



Материалы:

- **3.1 Задача критической секции // Грегори Р. Эндрюс - Основы многопоточного, параллельного и распределенного программирования (дата обращения: 01.12.2022).**

Изменения

- **Версия 2**
  - **Изменил оформление титульного слайда**
  - **Добавил код программы**
  - **Добавил схемы выполнения свойств**