

# Лабораторная работа

## Реализация взаимного исключения

### 1. Теоретическая часть.

Изучить алгоритмы *строгое чередование* и *алгоритм Петерсона*, а также особенности их работы, используя материалы соответствующих лекций.

#### Листинг алгоритма *Строгое чередование*

Процесс №0 входит в критическую область.

```
while (TRUE) {  
  
    while (turn!=0) ;  
    critical_region();  
    turn=1;  
    noncritical_region();  
  
}
```

Процесс №1 входит в критическую область

```
while (TRUE) {  
  
    while (turn!=1) ;  
    critical_region();  
    turn=0;  
    noncritical_region();  
  
}
```

#### Листинг Алгоритма Петерсона

```
#define FALSE 0  
#define TRUE 1  
#define N 2  
  
int turn;  
int interested[N];  
  
void enter_region(int process){  
  
    int other;  
    other=1-process;  
    interested[process]=TRUE;  
    turn=process;  
    while(turn==process && interested[other]==TRUE)
```

```
}
```

```
void leave_region(int process) {  
interested[process]=FALSE  
}
```

## 2. Практическая часть

Разработать программу с оконным интерфейсом, моделирующую работу алгоритмов «Строгое чередование» и «Алгоритм Петерсона» для двух работающих процессов.

При этом:

1. Адаптировать рассматриваемые алгоритмы с учётом активного ожидания.
2. Предусмотреть индикацию входа в критическую область и выхода из неё.
3. Изначально оба процесса находятся вне критических областей.
4. Изначально переменной `turn` присвоить значение 0.

В качестве модели работающих процессов выбрать два из ниже перечисленных в соответствии со своим вариантом:

1. Вводится вещественное число, возвращается квадрат этого числа.
2. Вводится целое число, возвращается факториал этого числа.
3. Вводится символ, возвращается ASCII-код этого символа.
4. Вводится символ, возвращается символ, ASCII-код которого на две единицы больше введённого.
5. Вводится целое число  $n$  от 1 до 3, подаётся  $n$  звуковых сигналов.
6. Вводится строка символов, если встречаются цифры, они заменяются знаком 'х'.

Вариант	Процесс №1	Процесс №2
1	1	2
2	1	3
3	1	4
4	1	5
5	1	6
6	2	3
7	2	4
9	2	5
10	2	6
11	3	4
12	3	5
13	3	6
14	4	5
15	5	6

## Контрольные вопросы

1. Понятие процесса и потока.
2. Диаграмма состояний процесса.
3. Состояние состязания.
4. Критическая область.
5. Взаимное исключение
6. Условия реализации взаимного исключения.
7. Способы реализации взаимного исключения. Достоинства и недостатки.
8. Запрещение прерываний.
9. Переменные блокировки.
10. Строгое чередование.
11. Достоинства и недостатки строгого чередования.
12. Алгоритм Петерсона.
13. Достоинства и недостатки алгоритма Петерсона.
14. Реализация взаимного исключения с помощью команды **TSL**.
15. Понятие семафора и мьютекса.