

ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССЫ. ЗАПУСК

Параллельное программирование

РГАТУ

2019

Грызлова Т. П.

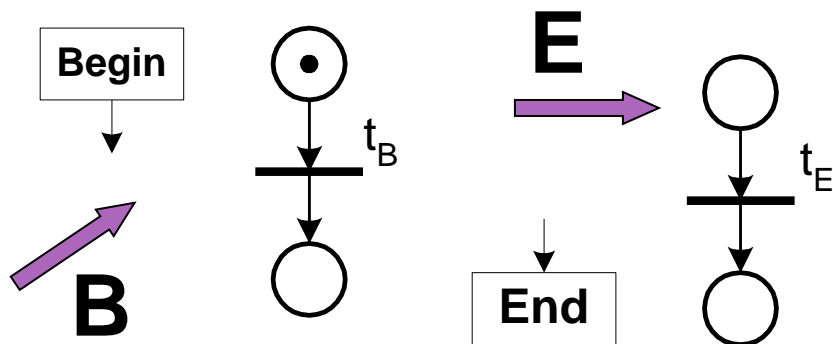
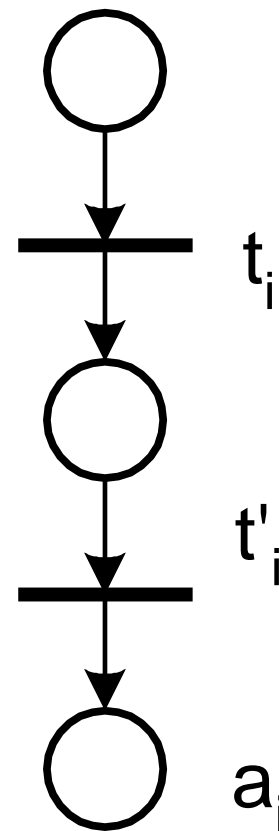
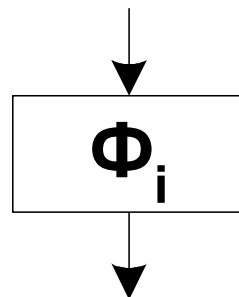
ВЕРШИНЫ ПАРАЛЛЕЛЬНЫХ ГРАФ-СХЕМ

Операторная

 **F** 

Выполнение действий над данными

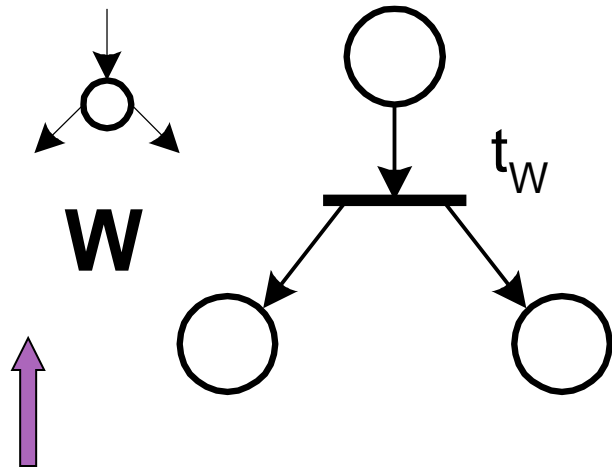
Специальные



Начало и конец алгоритма

УПРАВЛЯЮЩИЕ ВЕРШИНЫ ПГС

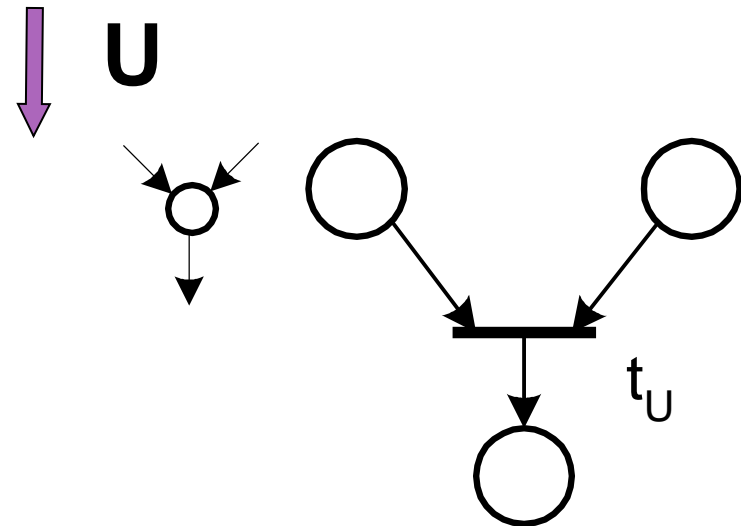
Распараллеливание



Бифуркатор.

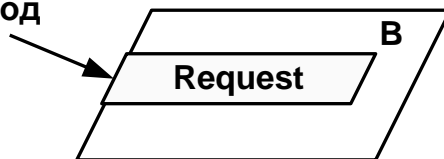
Переход к параллельному
выполнению алгоритмов

↓ **Синхронизатор**
Объединение
параллельных ветвей



ВЫЗОВ ВХОДА. АТТРИБУТЫ ЗАДАЧ И ВХОДОВ

Стандартный вход

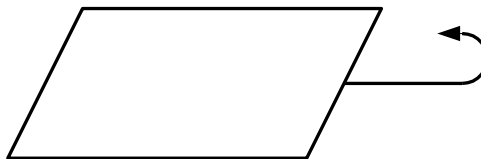


Безусловный вызов



B.Request

Условный вызов



```
select
  B.Request;
else
  Alternative Action;
end select;
```

Временной вызов



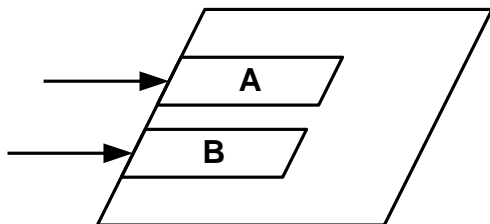
```
select
  B.Request;
or
  delay T;
  Alternative Action;
end select;
```

B'Active - True, если задача активна и False в противном случае
B'Priority - текущий приоритет задачи B;
B'Clock - накопленное время работы задачи B.

Request'Count - число необслуженных обращений ко входу, можно пользоваться только в теле задачи, которой принадлежит вход

ЗАДАЧИ

Задача с гнездами входов в нотации Бара



СПЕЦИФИКАЦИЯ

```
Task TaskName is  
  Entry A;  
  Entry B;  
End TaskName;
```

НЕ ОБЯЗАТЕЛЬНО

ТЕЛО ЗАДАЧИ (1)

Объявления
локальных
переменных

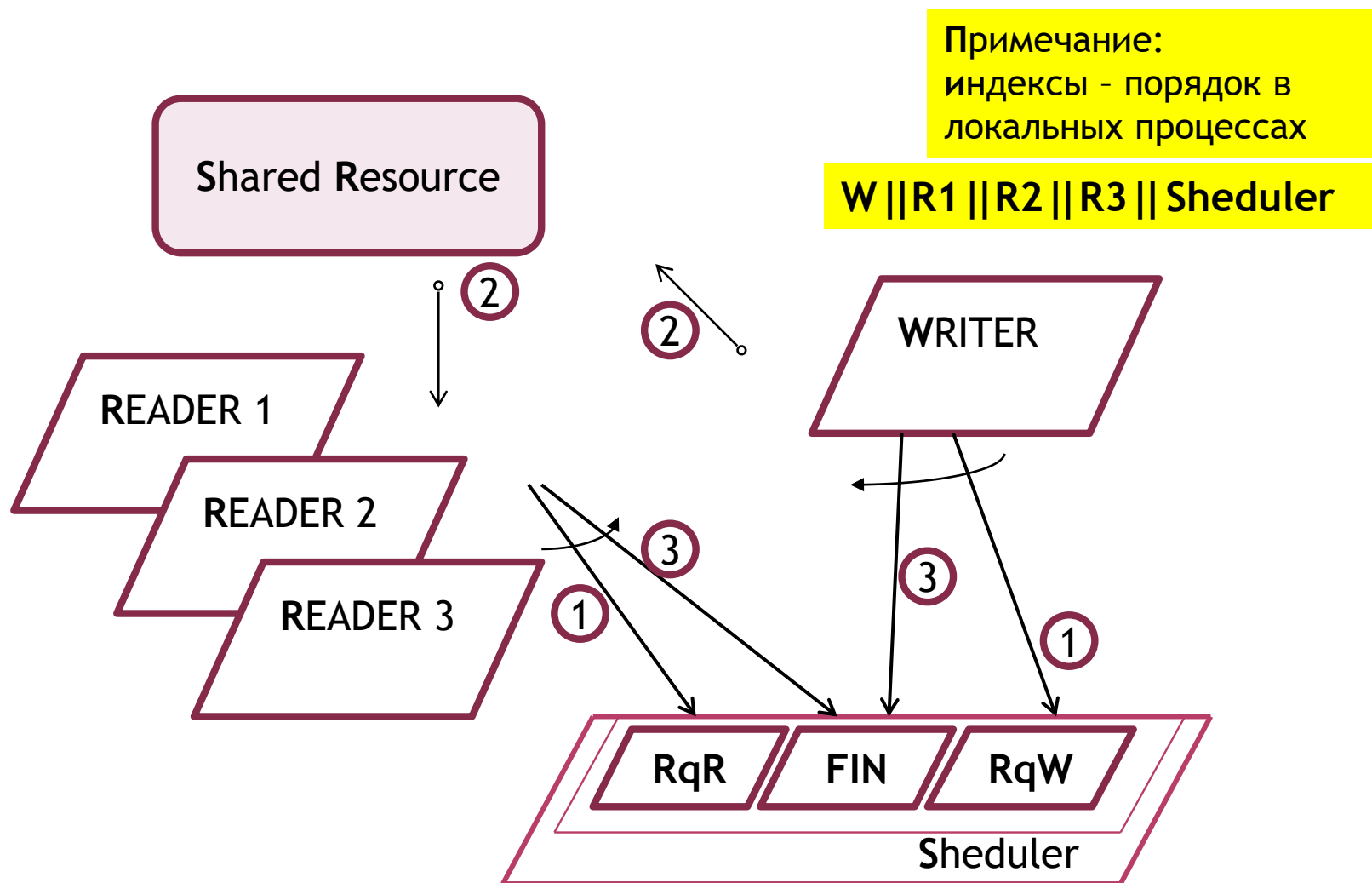
Операторы

```
Task body TaskName is  
  .....  
  Begin  
    .....  
    accept A;  
    .....  
    accept B;  
    .....  
End TaskName;
```

ТЕЛО ЗАДАЧИ (2)

```
Task body TaskName is  
  Begin  
    .....  
    accept A do ... end A;  
    .....  
    accept B do ... end B;  
    .....  
End TaskName;
```

АРХИТЕКТУРА СИСТЕМЫ ЧИТАТЕЛИ-ПИСАТЕЛИ НА ADA



ПАРАЛЛЕЛЬНО ИСПОЛНЯЕМЫЕ ЗАДАЧИ В ADA

```
with Text_IO; use text_IO;
```

```
procedure RsW_1 is
```

```
  task Sheduler is
```

```
    entry RqR;
```

```
    entry RqW;
```

```
    entry FIN;
```

```
  end Sheduler;
```

```
  task Reader;
```

```
  task Writer;
```

```
Task body Reader is
```

```
  begin
```

```
    loop
```

```
    .....
```

```
      Sheduler.RqR;
```

```
    .....
```

```
      Sheduler.FIN;
```

```
    end loop;
```

```
  end Reader;
```

```
Task body Writer is
```

```
  begin
```

```
    loop
```

```
      Sheduler.RqW;
```

```
    .....
```

```
      Sheduler.FIN;
```

```
    end loop;
```

```
  end Writer;
```

```
Task body Sheduler is
```

```
  .....
```

```
end Sheduler;
```

```
Begin
```

```
.....
```

```
End RsW_1;
```

ЗАДАЧНЫЕ ТИПЫ

```
Task type tW is  
  entry E1;  
  entry E2;  
End tW;
```

```
Writer1, Writer2 : tW;
```

```
Task type имя типа задачи is  
  entry имя входа 1(формальный параметр 11 : имя типа; формальный  
  параметр 12 : имя типа);  
  entry имя входа 2(формальный параметр 21 : имя типа; формальный  
  параметр 22 : имя типа);  
  .....  
End имя типа задачи;
```


НИТИ

НИТИ В C#

```
using System.Threading;
```

Создание нитей

```
.....  
for (int i = 1; i <= NR; i++)  
{ Thread Reader = new Thread(new ParameterizedThreadStart(ToRead));  
  Reader.Start(i);  
} //for
```

Процесс, который она будет исполнять

```
.....  
static void ToRead(object num)  
{ int iteration = 0;  
while (iteration < 15) {  
  Console.WriteLine(" {0} ", num);  
  Thread.Sleep(200); iteration++;  
} //while  
} //Reader
```