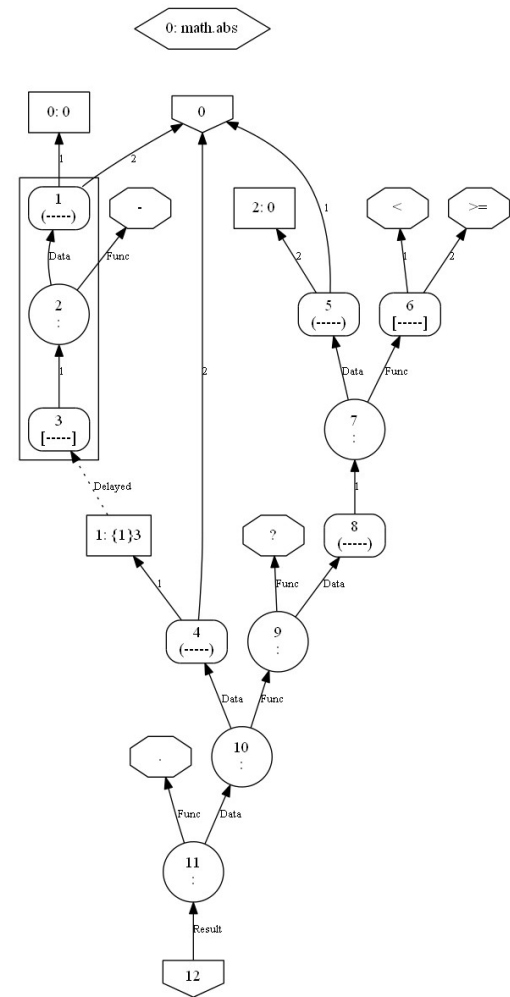
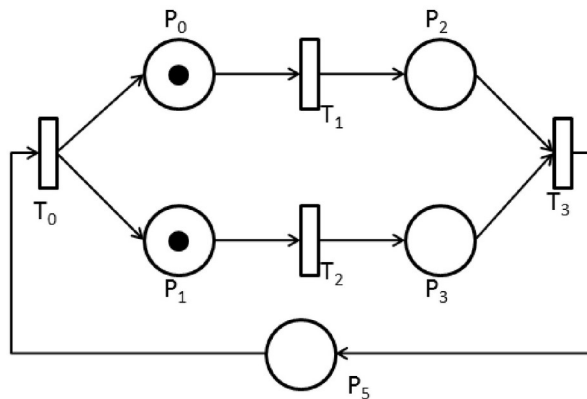
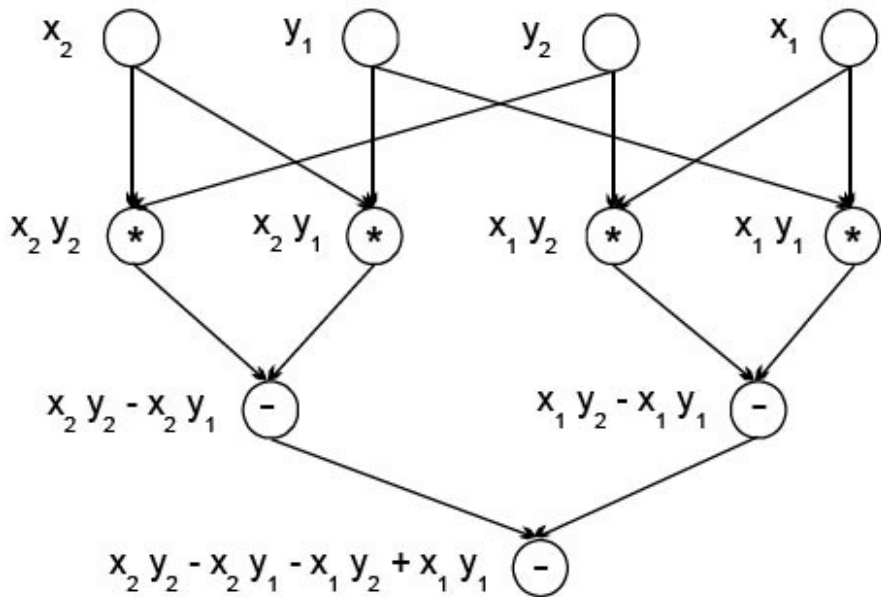


Архитектуры параллельных ВС (модельные аспекты)

(x_2, y_2)
 (x_1, y_1)
 $S = ((x_2 - x_1)(y_2 - y_1)) =$
 $= x_2 y_2 - x_2 y_1 - x_1 y_2 + x_1 y_1$



Определения

Модель вычислений - отображение различных аспектов организации вычислительных процессов. Зачастую с акцентом на частные характеристики.

Многообразие моделей вычислений определяется их направленностью на отображение не всех, а только определенных параметров, что позволяет исследовать различные аспекты организации вычислений.

Основные отношения между программными объектами

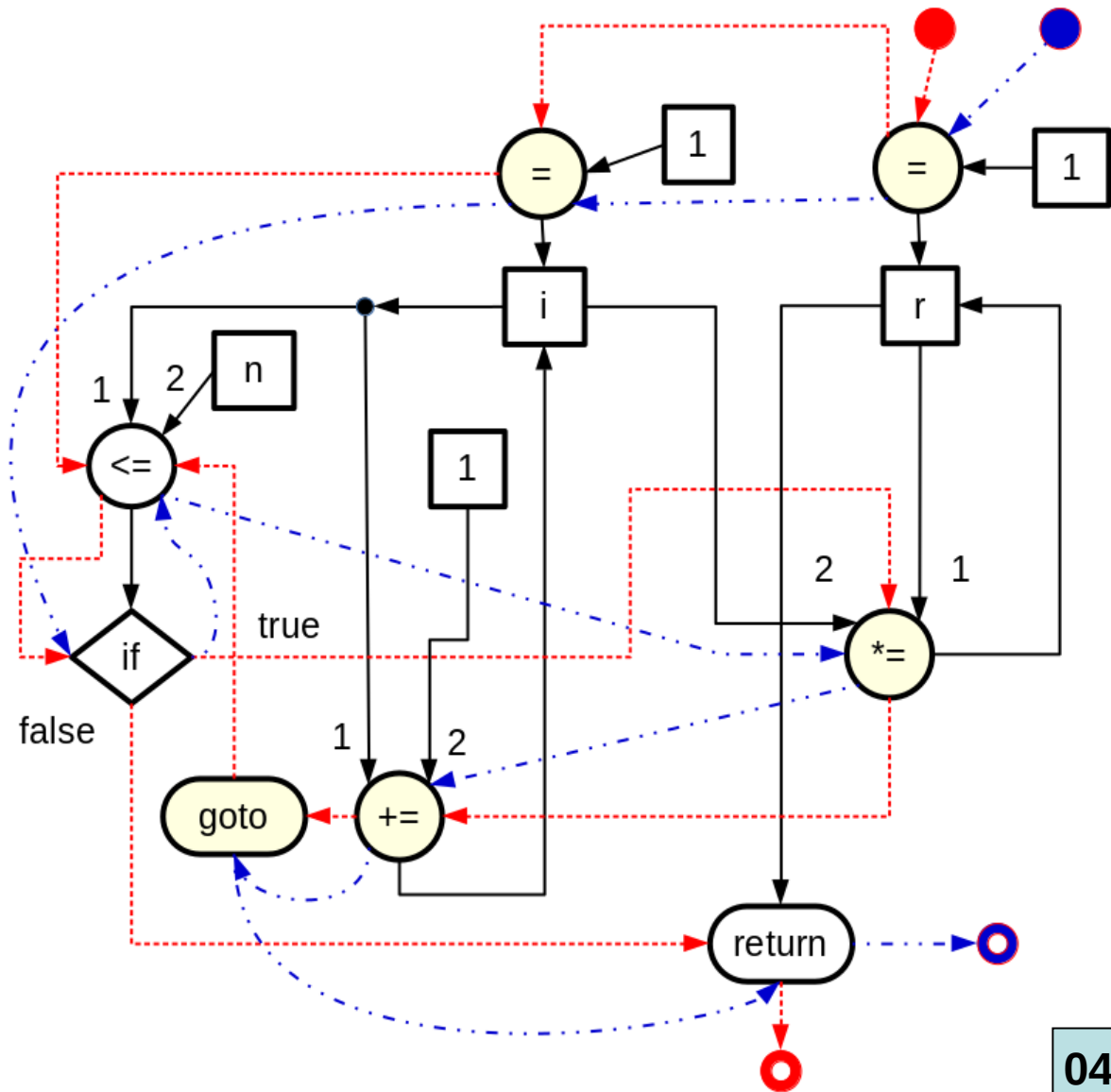
Факториал от 1 до n:

```
int fact1n(int n) {  
    int r = 1;  
    int i = 1;  
    Loop:  
    if(i <= n) {  
        r *= i;  
        i++;  
        goto loop;  
    }  
    return r;  
}
```

```

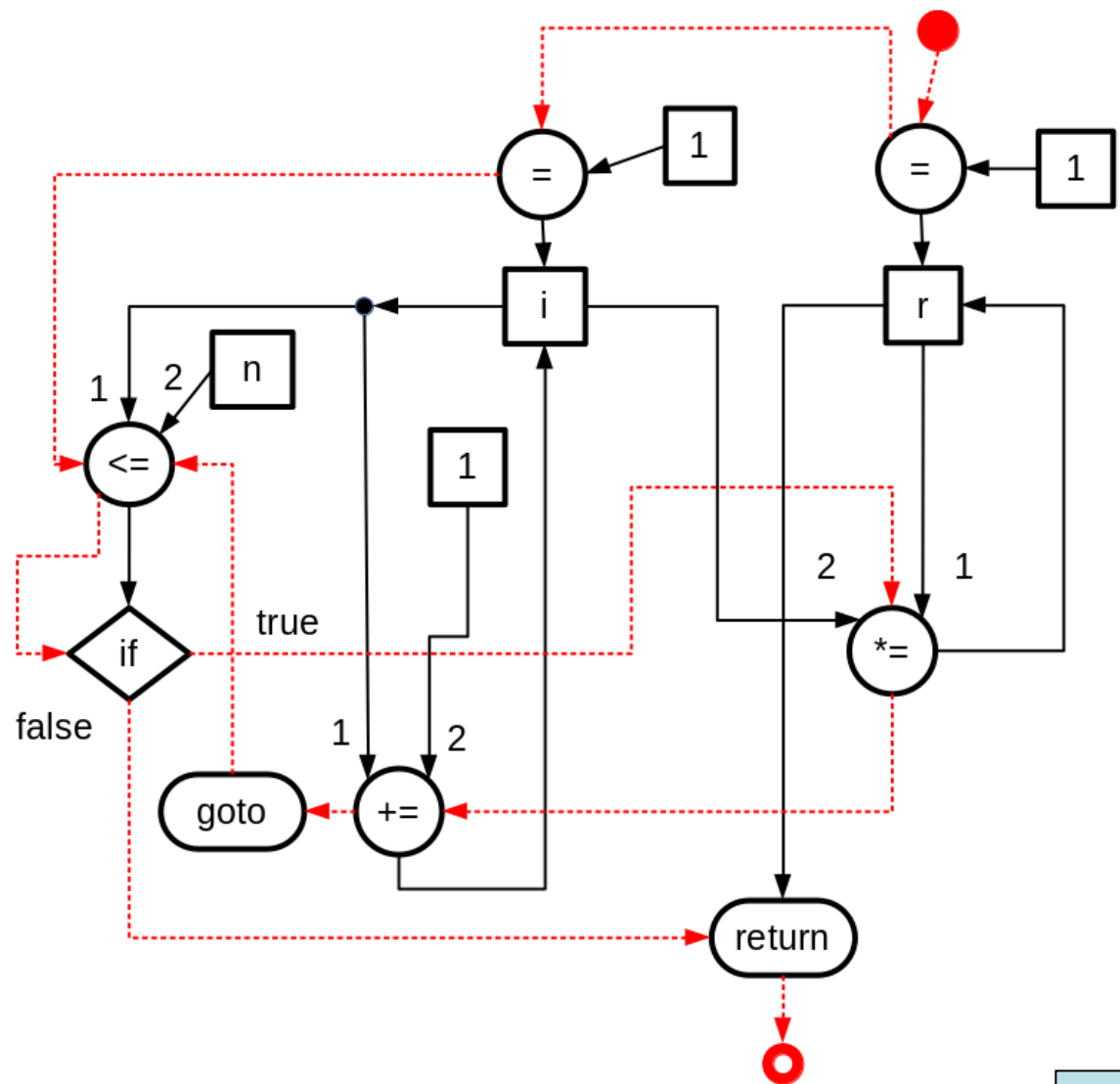
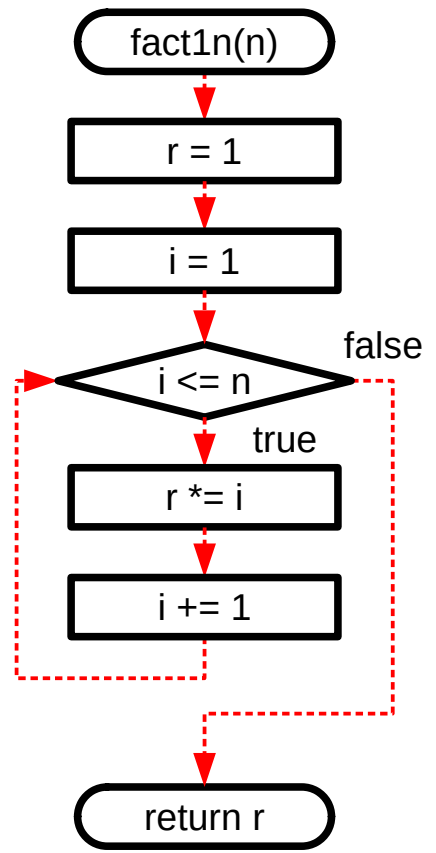
// Факториал от 1 до n
int fact1n(int n) {
    int r = 1;
    int i = 1;
Loop:
    if(i <= n) {
        r *= i;
        i++;
        goto loop;
    }
    return r;
}

```



// Факториал от 1 до n

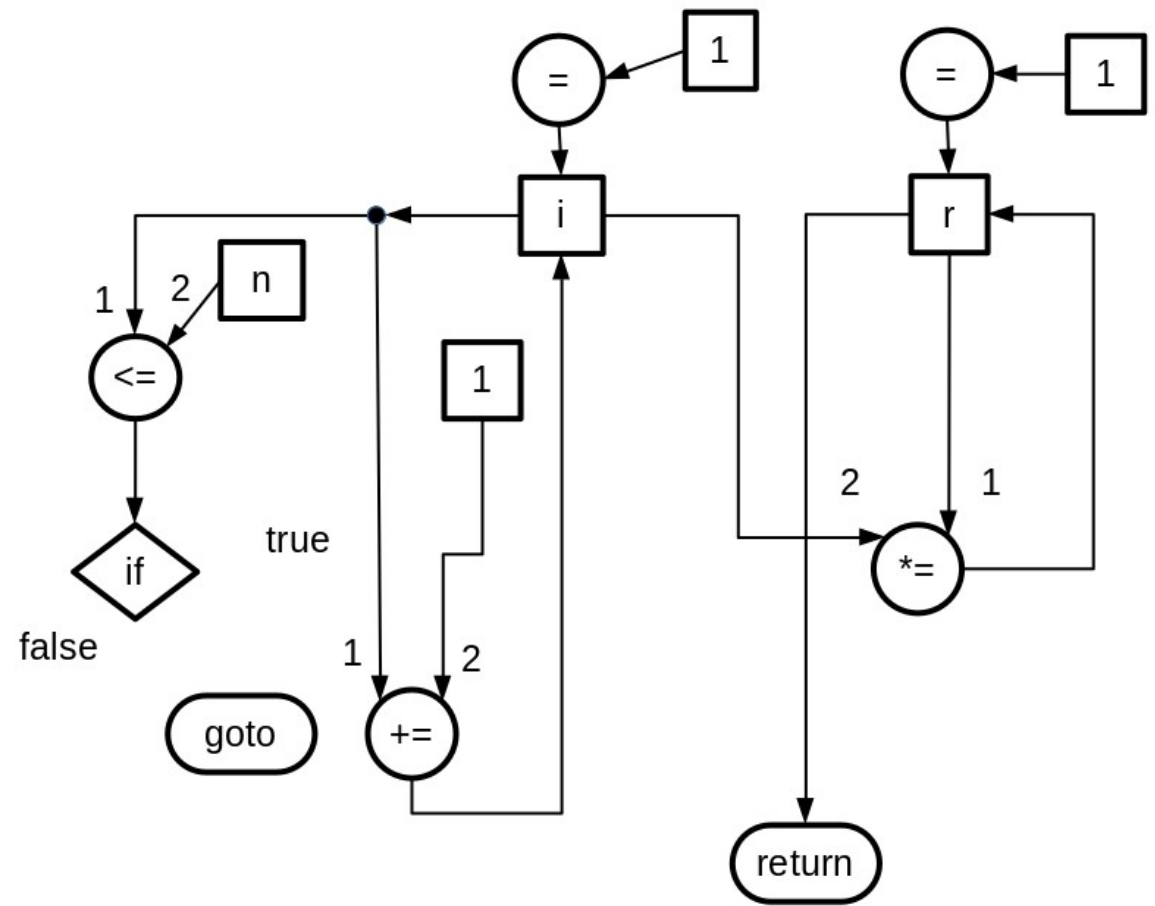
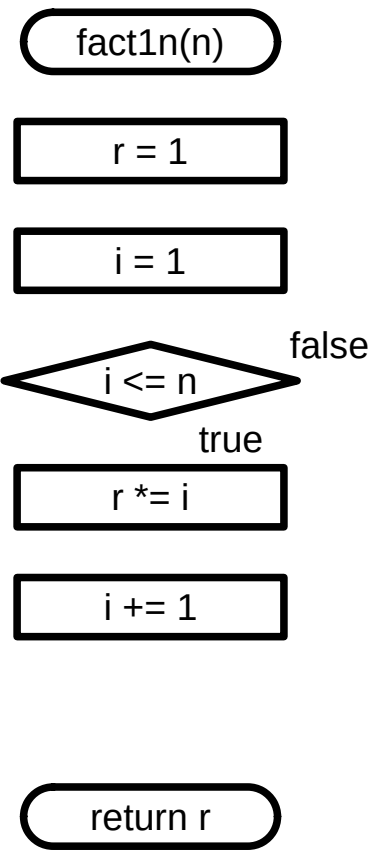
```
int fact1n(int n) {  
    int r = 1;  
    int i = 1;  
loop:  
    if(i <= n) {  
        r *= i;  
        i++;  
        goto loop;  
    }  
    return r;  
}
```



```

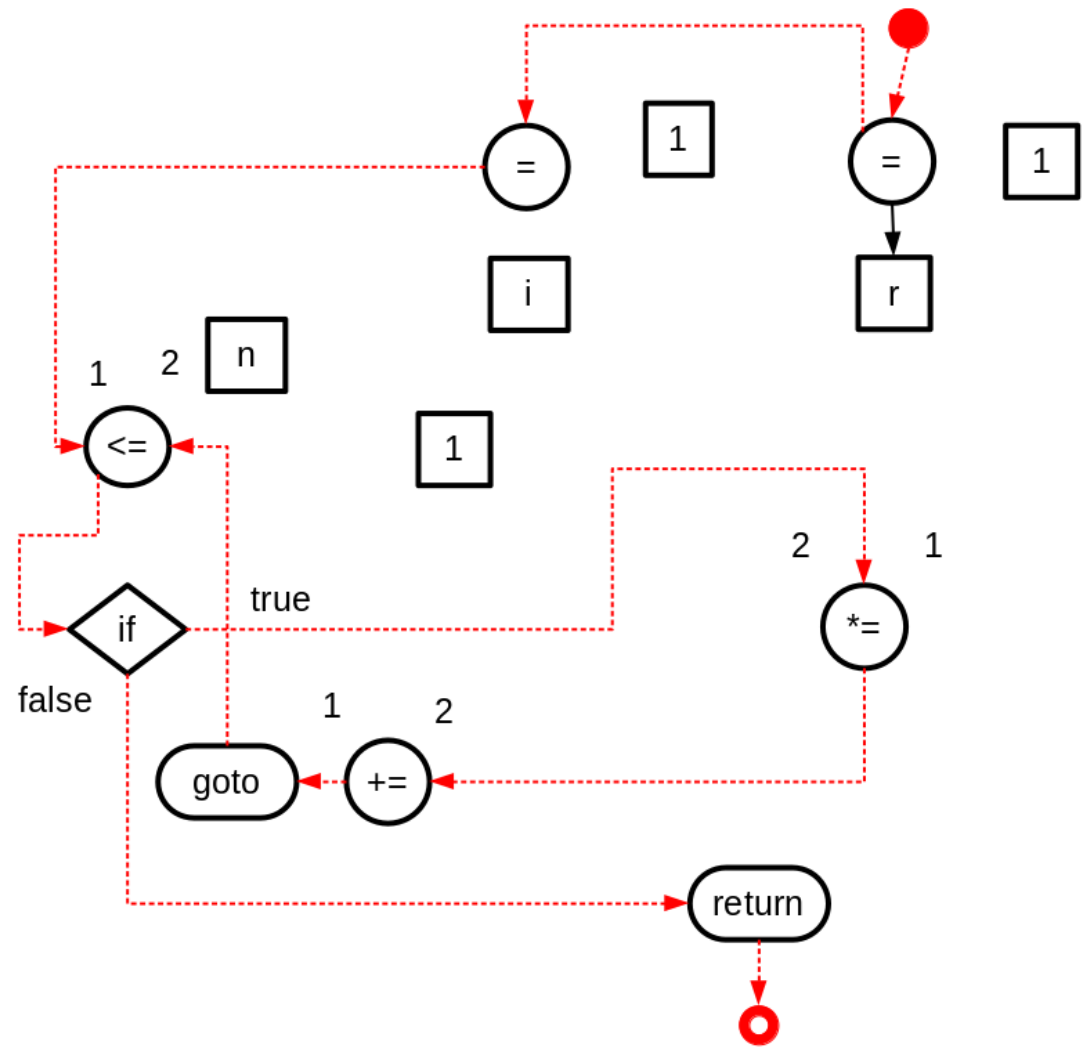
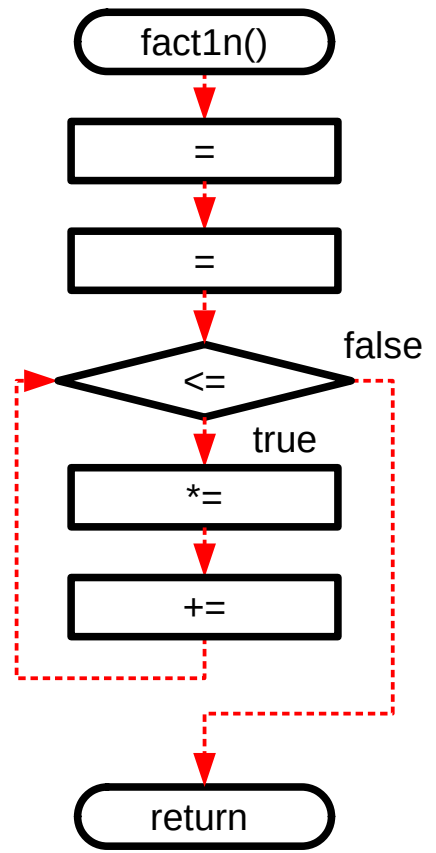
// Факториал от 1 до n
int fact1n(int n) {
    int r = 1;
    int i = 1;
loop:
    if(i <= n) {
        r *= i;
        i++;
        goto loop;
    }
    return r;
}

```



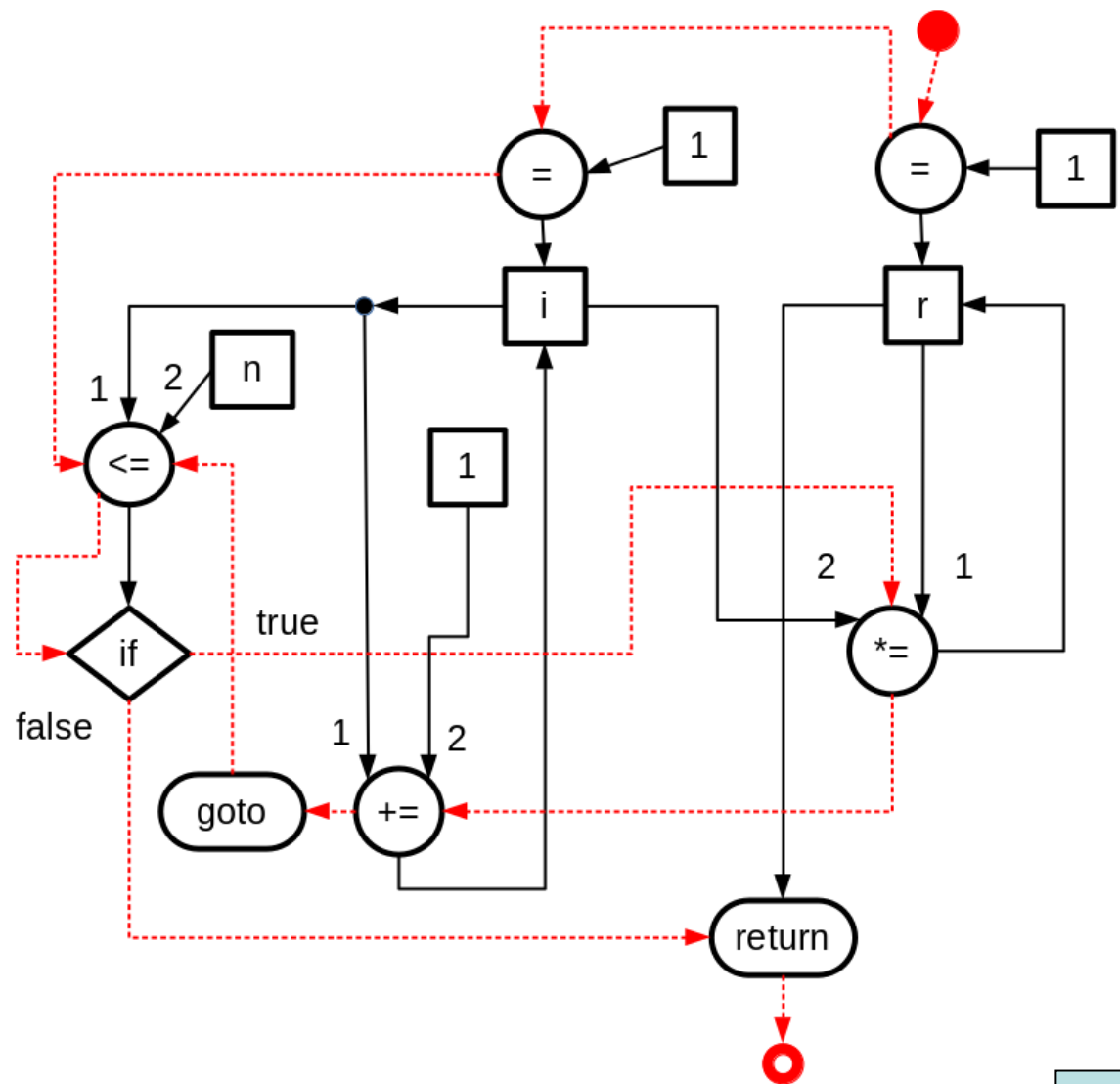
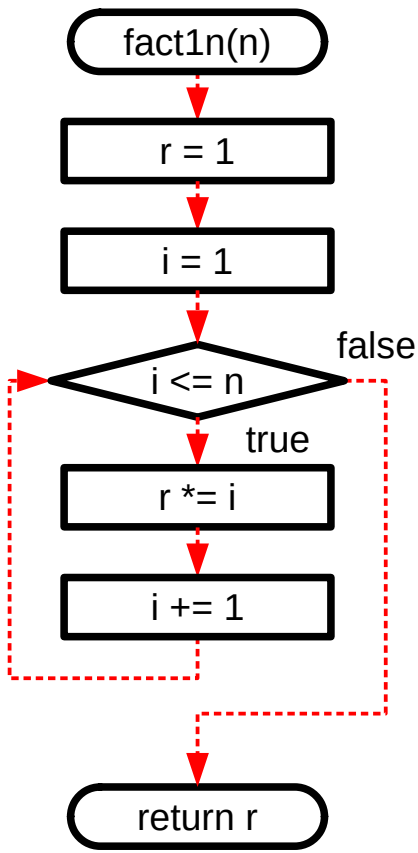
// Факториал от 1 до n

```
int fact1n(int n) {  
    int r = 1;  
    int i = 1;  
loop:  
    if(i <= n) {  
        r *= i;  
        i++;  
        goto loop;  
    }  
    return r;  
}
```



// Факториал от 1 до n

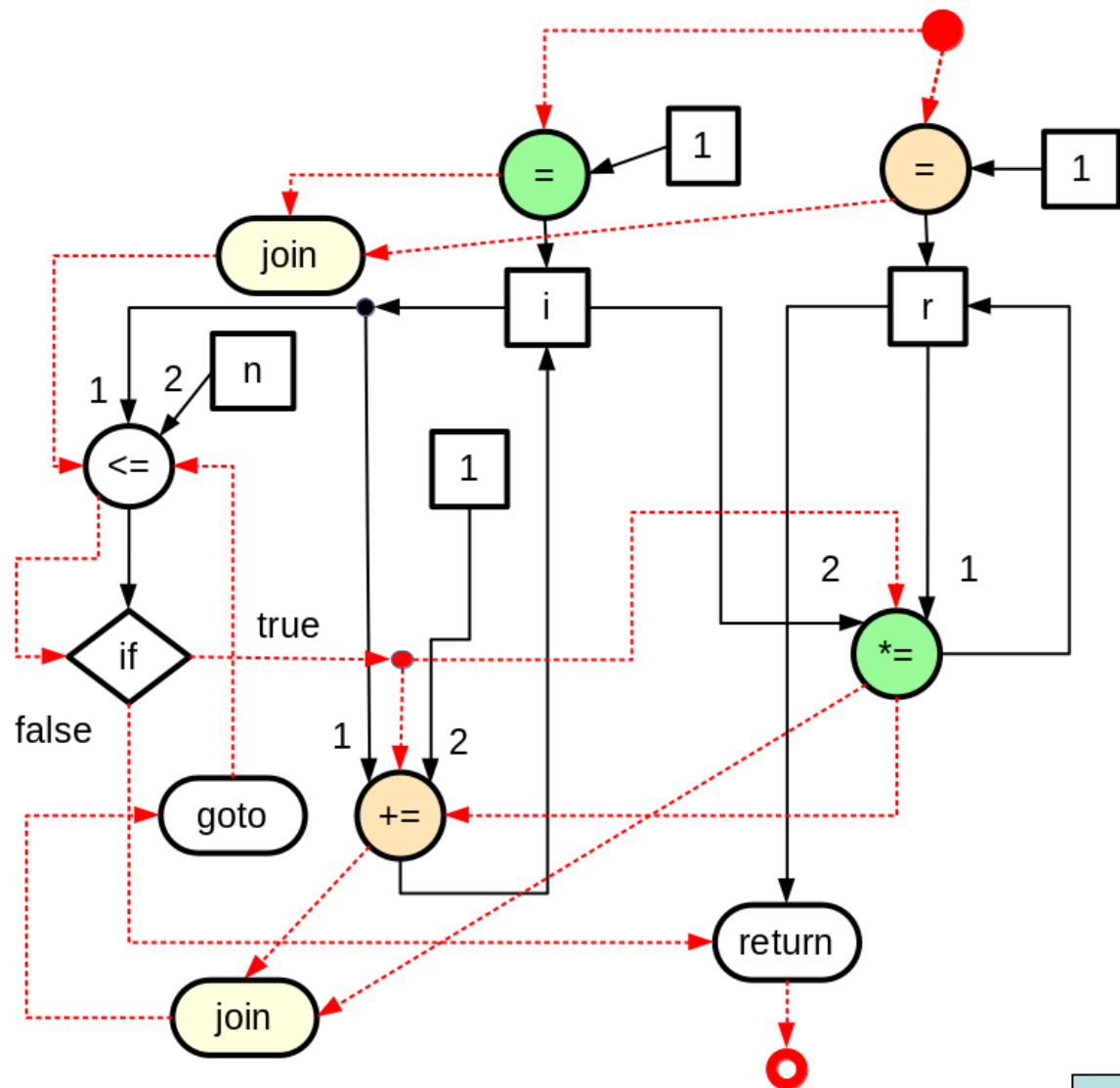
```
int fact1n(int n) {  
    int r = 1;  
    int i = 1;  
loop:  
    if(i <= n) {  
        r *= i;  
        i++;  
        goto loop;  
    }  
    return r;  
}
```



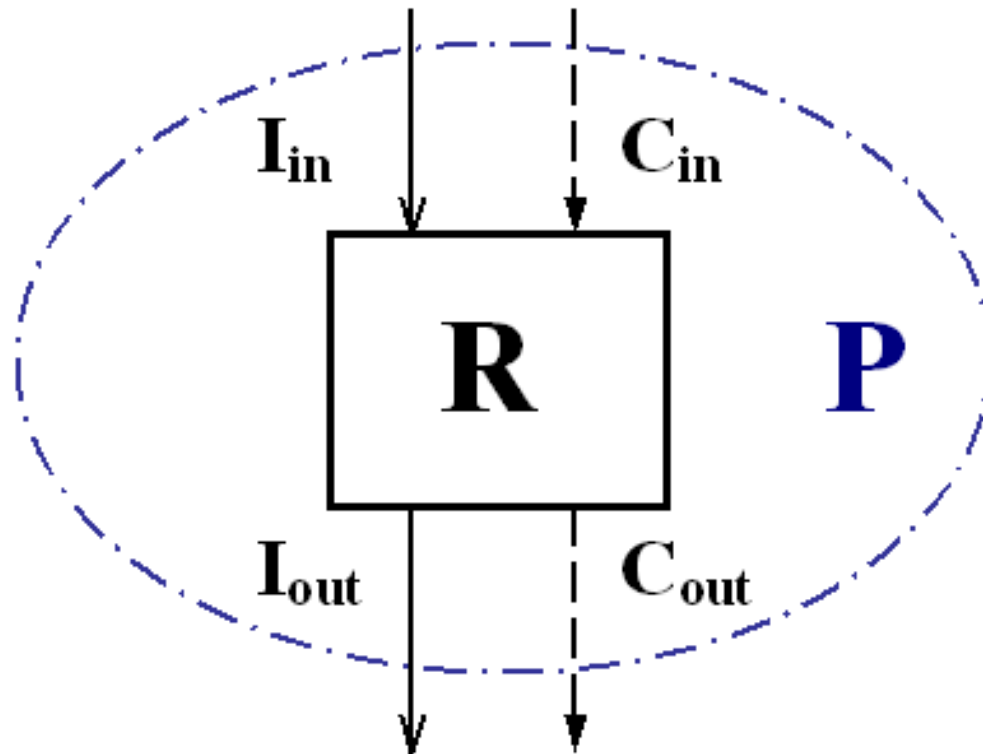

```

// Факториал от 1 до n
int fact1n(int n)
{
    fork
        int r = 1;
        int i = 1;
    join
loop:
    if(i <= n) {
        fork
            r *= i;
            i++;
        join
        goto loop;
    }
    return r;
}

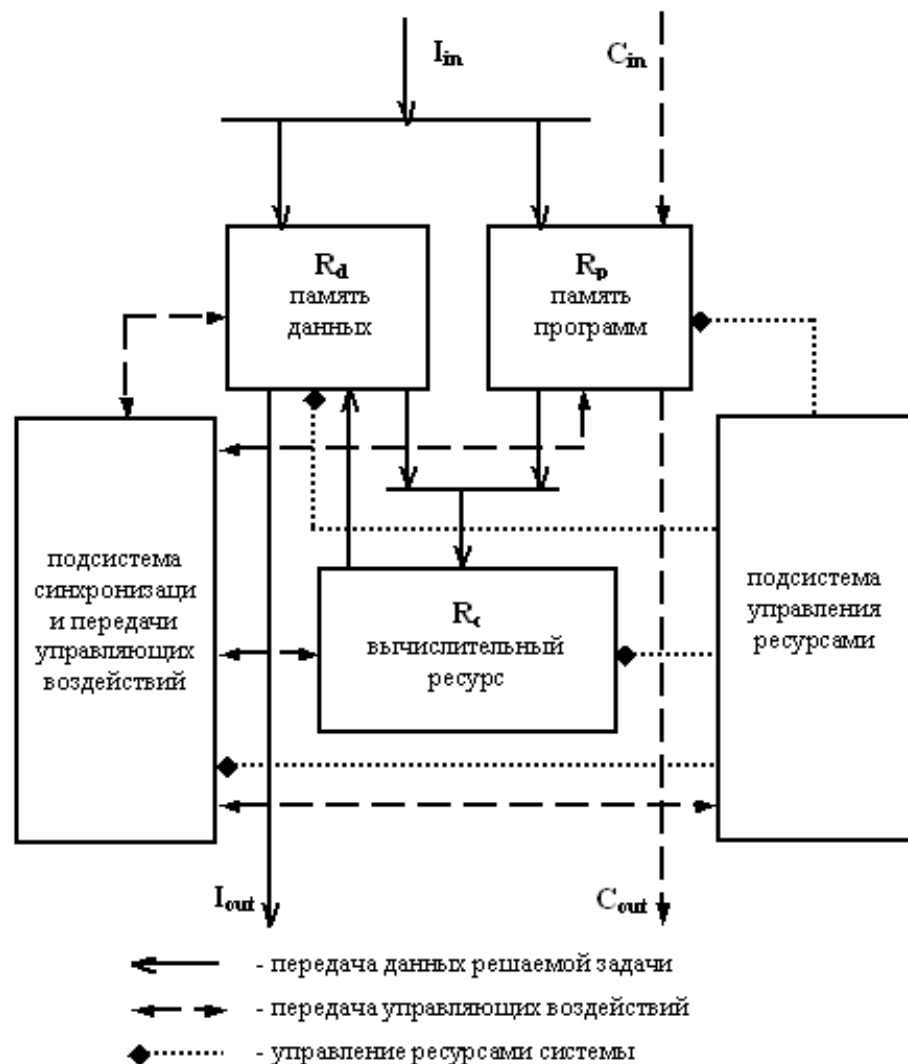
```



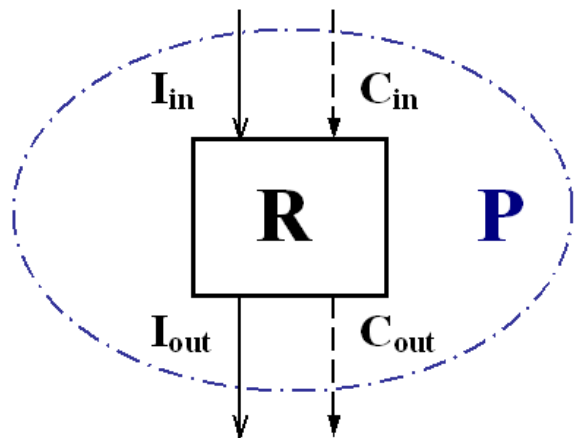
Процесс как вычисления, протекающие в ресурсе



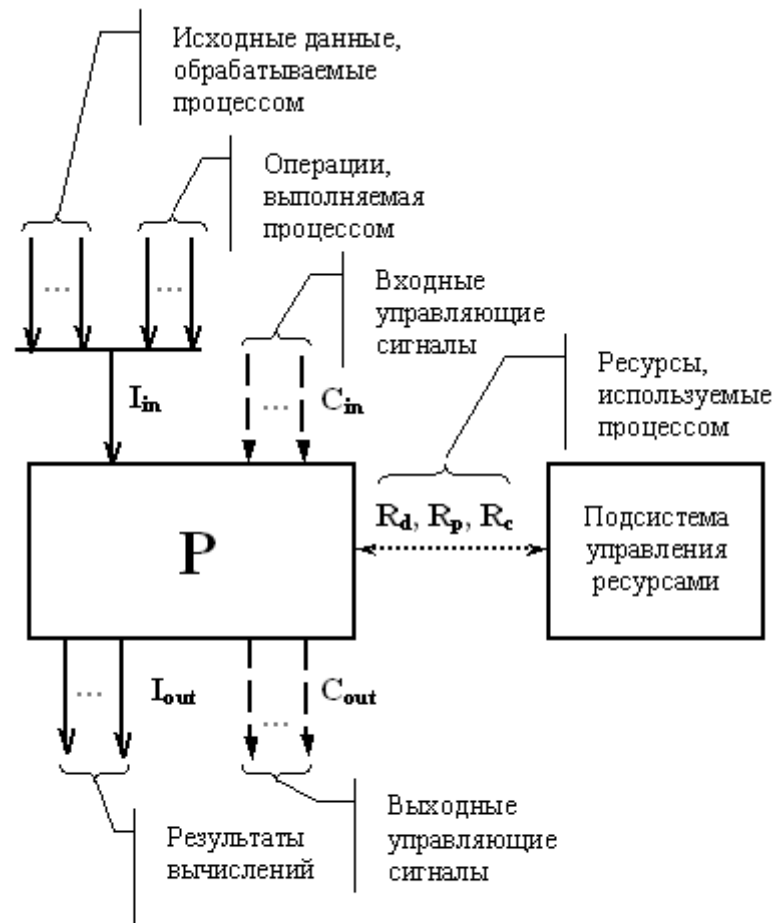
Потоки, определяющие протекание процессов в ресурсах ВС



Модель процесса

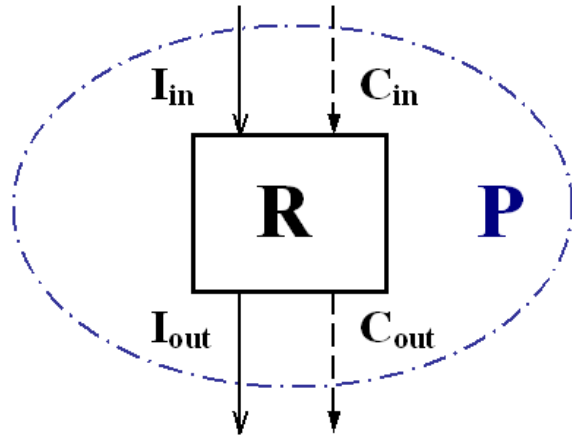


$$P = (I, R, C)$$

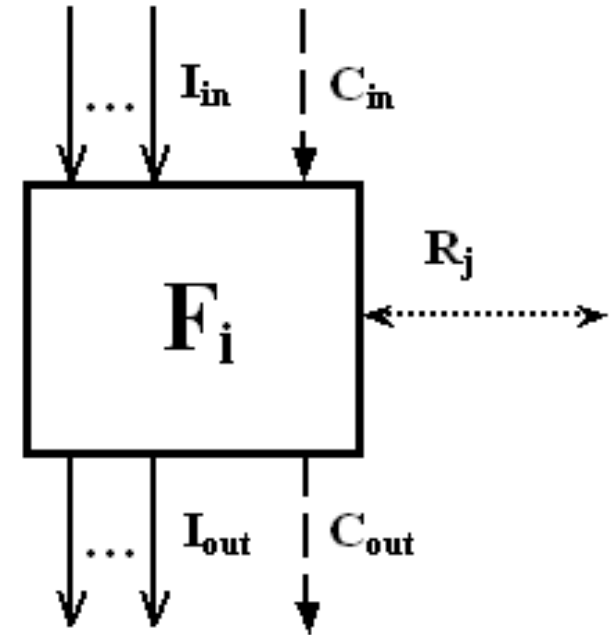



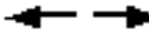

- ← - передача данных решаемой задачи
- ↔ - передача управляющих воздействий
- ←.....→ - передача информации о ресурсах

Элементарный (функциональный) процесс

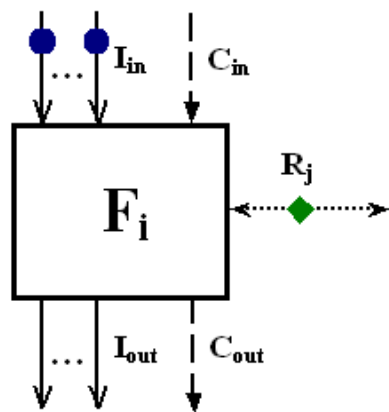


$$F_i = (I_{in}, I_{out}, R_j, C_{in}, C_{out})$$

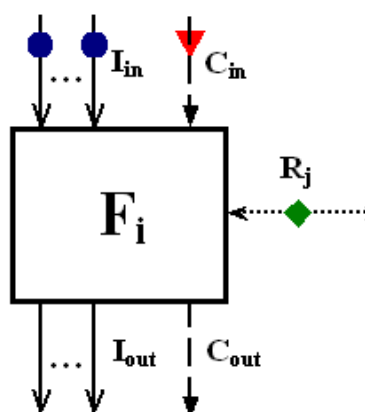


-  - передача данных решаемой задачи
-  - передача управляющих воздействий
-  - передача информации о ресурсах

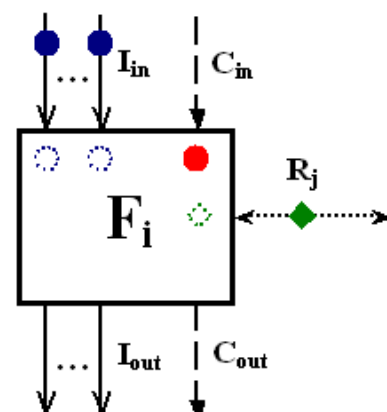
Продвижение фишек, моделирующее выполнение элементарного процесса



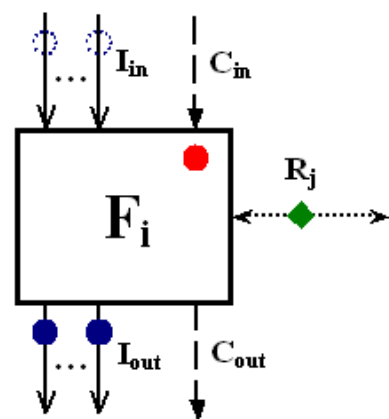
а) начальное состояние, определяющие запуск процесса



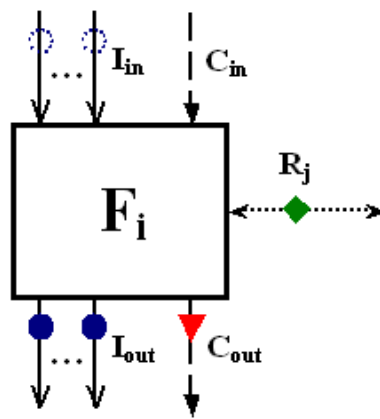
б) момент запуска процесса



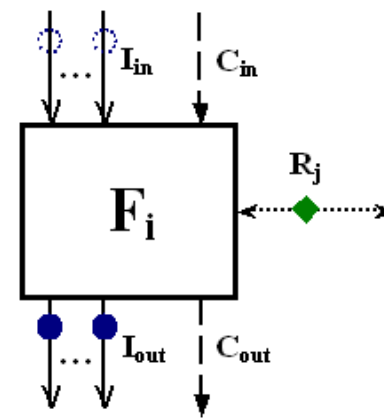
в) момент выполнения процесса



г) завершение вычислений

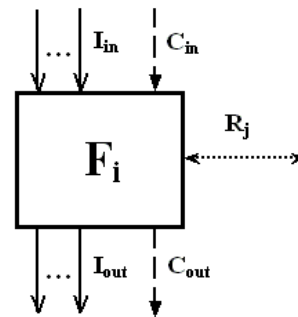


д) сигнал об окончании процесса

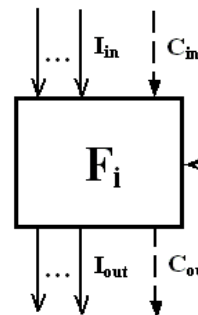


е) сформированные результаты

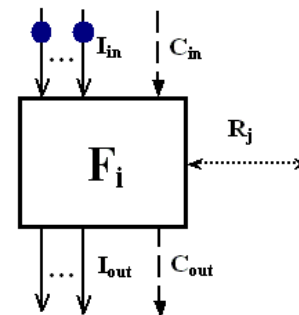
Порядок выделения ресурсов и получения исходных данных может быть произвольным



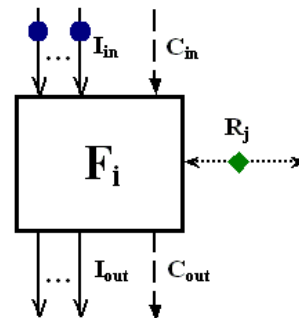
а) в начальном состоянии процесс может быть не размечен



б) в начале можно выделить ресурсы

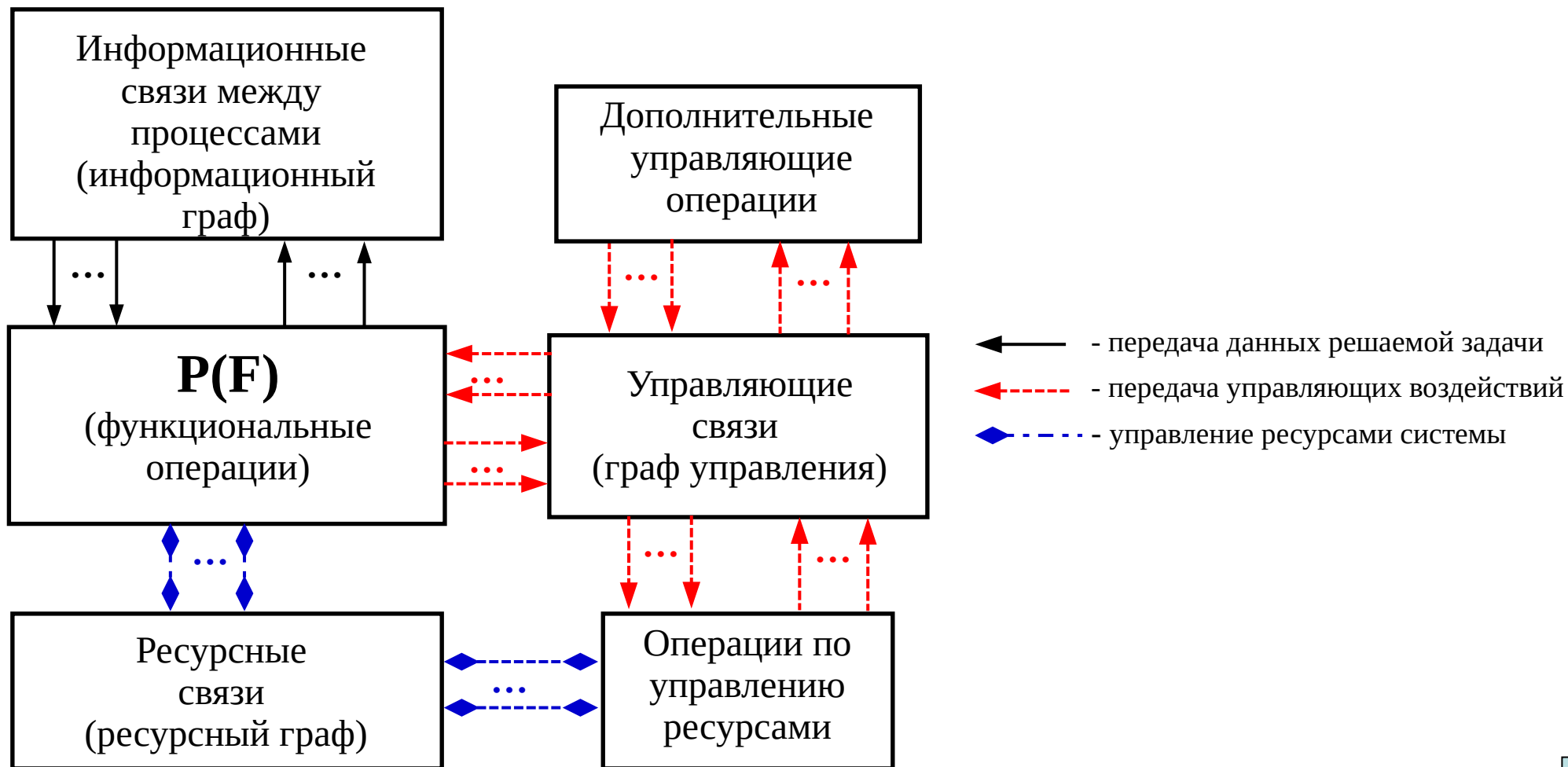


в) или в начале могут появиться аргументы



г) но необходимо их одновременное присутствие к моменту запуска процесса

ICR-сеть. Описание процессов в ресурсах ВС



ICR-сеть. Описание процессов в ресурсах ВС

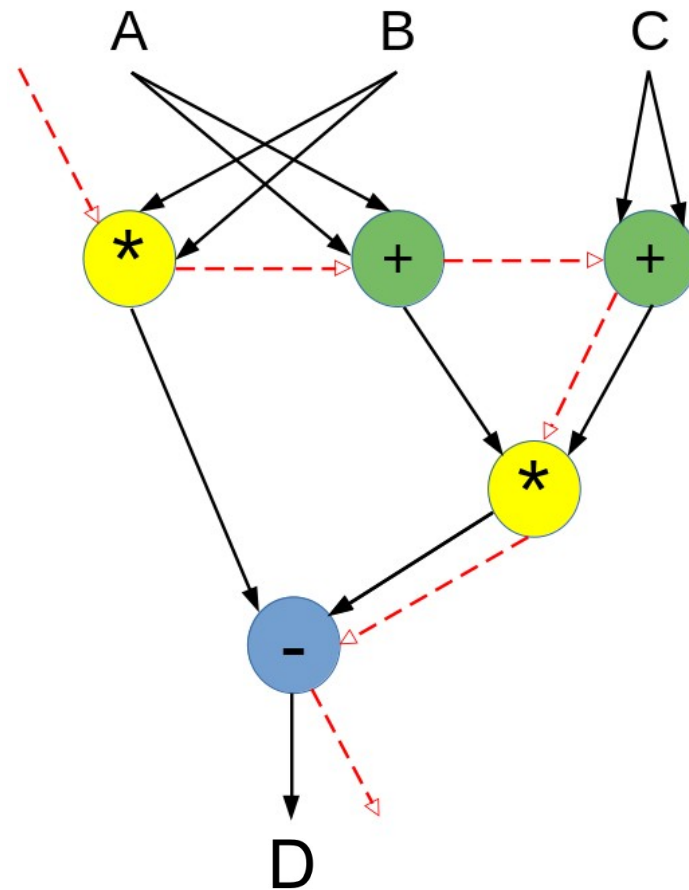
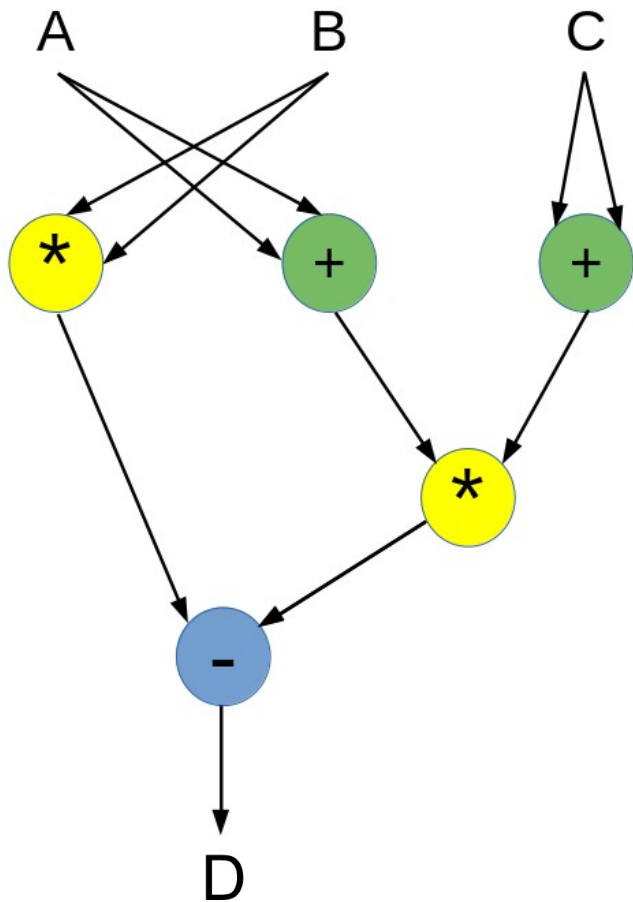
Информационный граф (ИГ) задает связи между данными, обрабатываемыми в ходе вычислений. Определяет алгоритм решаемой задачи и не изменяется при любых способах организации управления вычислениями. **Смена ИГ – смена алгоритма!**

Ресурсный граф (РГ) определяет предоставление ресурсов, без наличия которых обработка данных невозможна. Предоставление ресурсов определяется архитектурой ВС, и тем, как ресурсы отображаются в программе. **Программисту м.б. доступны команды по управлению ресурсами.**

Управляющий граф (УГ) задает передачу сигналов управления. Сигналы формируются по возникновению в сети различных событий, определяющих изменения в состоянии общего вычислительного процесса. Часть этих событий может быть связана с механизмом продвижения данных (фишек) между элементарными процессами. Сигналы управления могут порождаться и использоваться подсистемой управления ресурсами. **Для придания гибкости программисту м.б. доступны дополнительные управляющие команды (fork, join,...).**

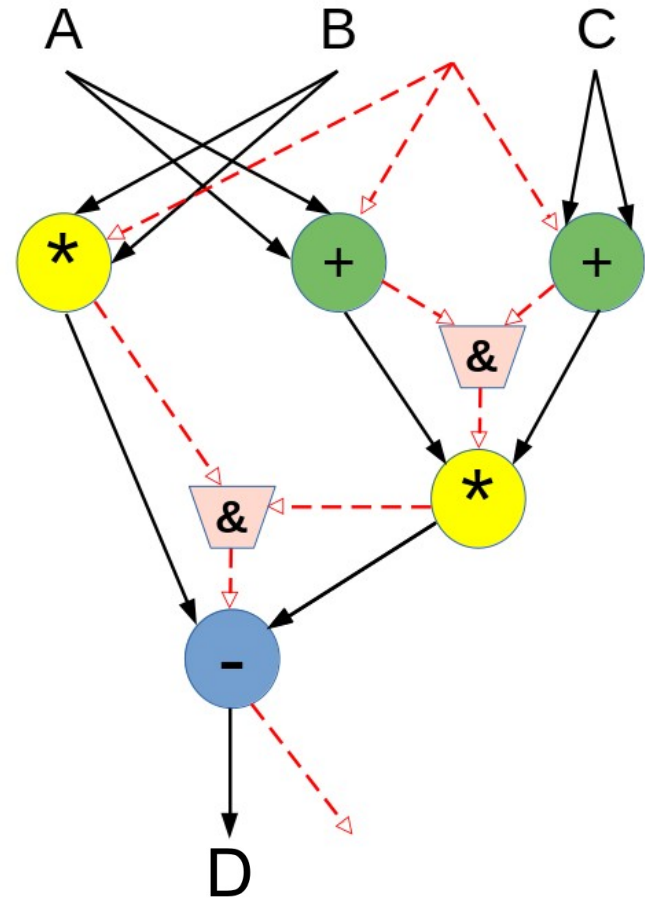
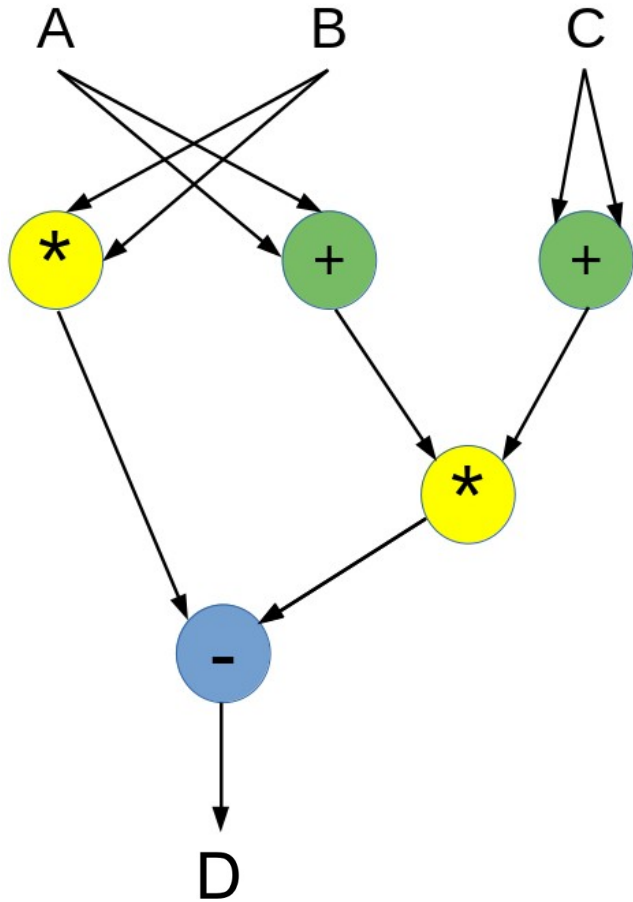
Именно управляющий граф определяет порядок выполнения операций и корректность проводимых вычислений.

$$D = B*B - (A+A)*(C+C)$$



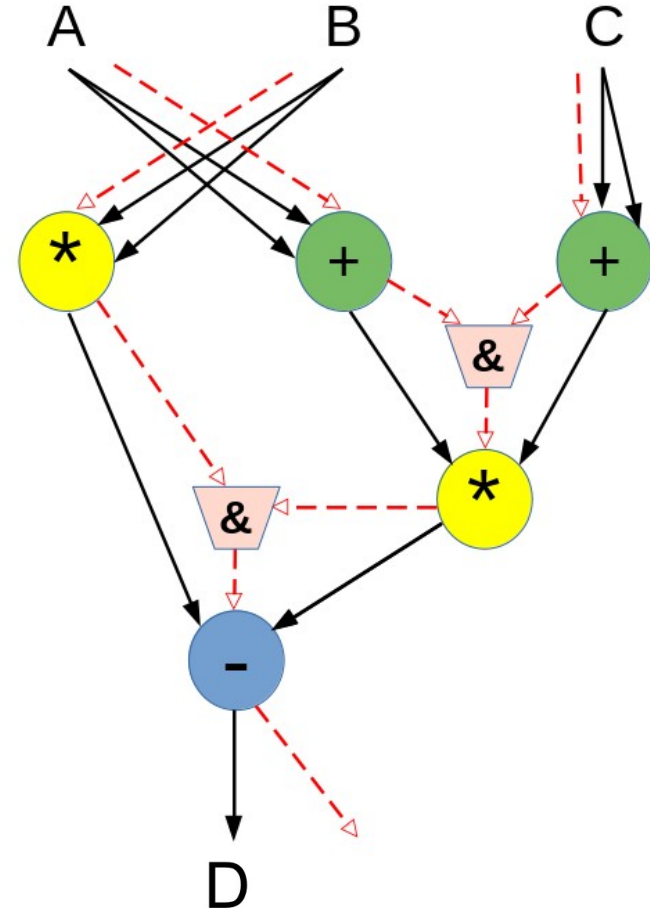
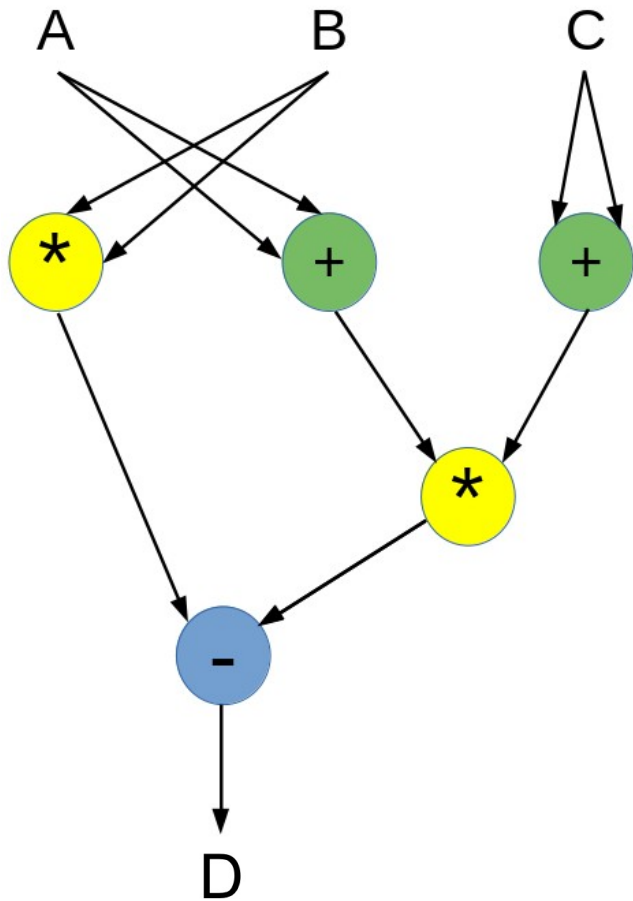
Последовательное управление вычислениями

$$D = B * B - (A + A) * (C + C)$$



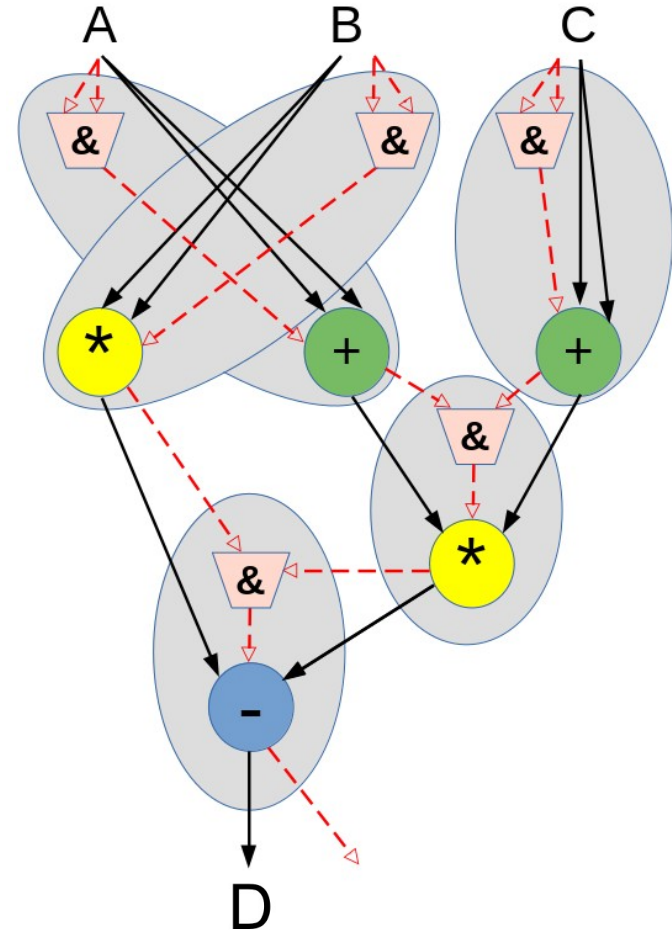
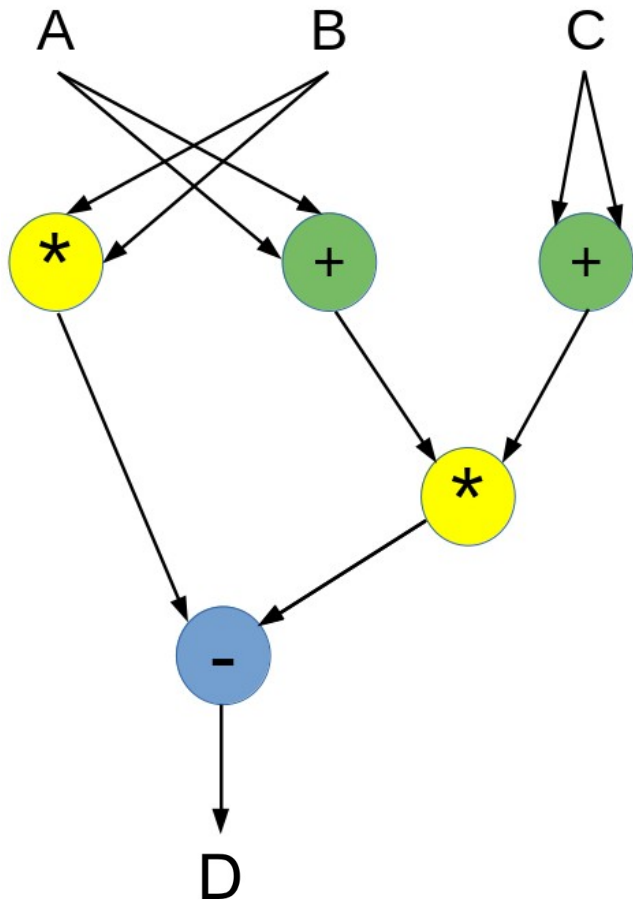
Ярусно-параллельная форма (ЯПФ)

$$D = B * B - (A + A) * (C + C)$$



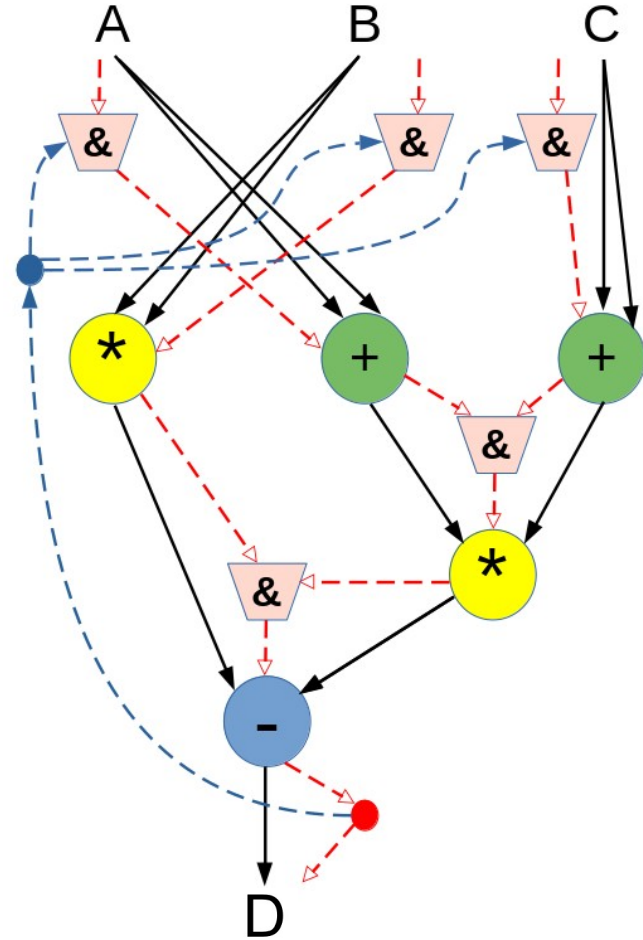
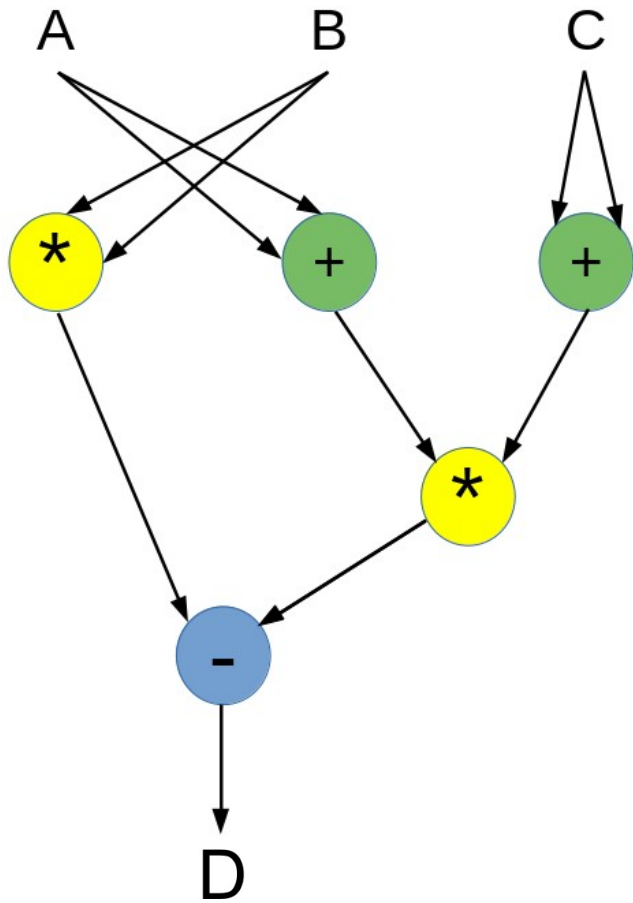
Явное управление по готовности данных

$$D = B * B - (A + A) * (C + C)$$



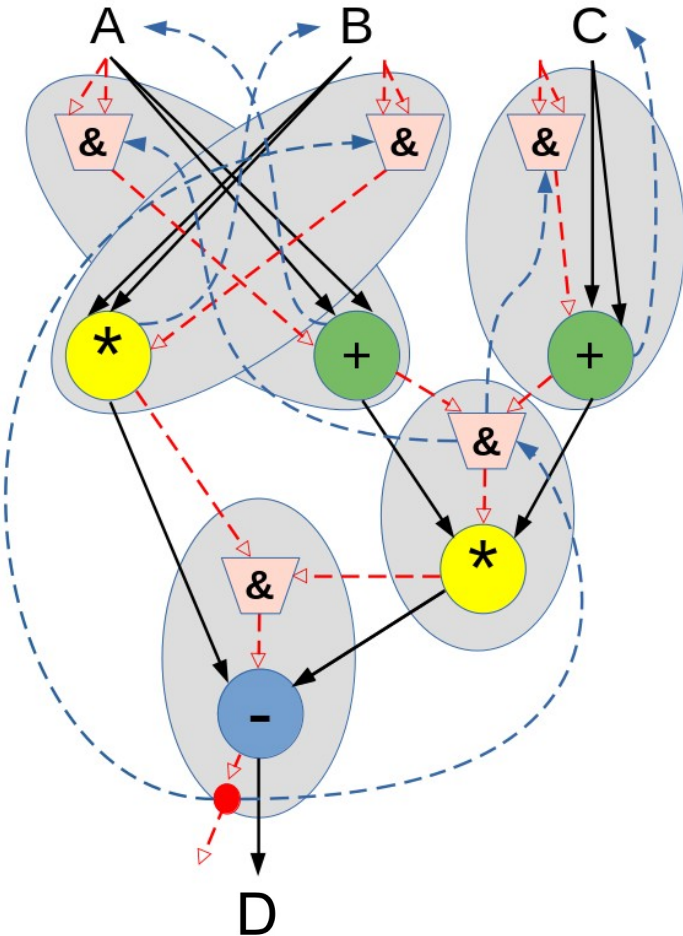
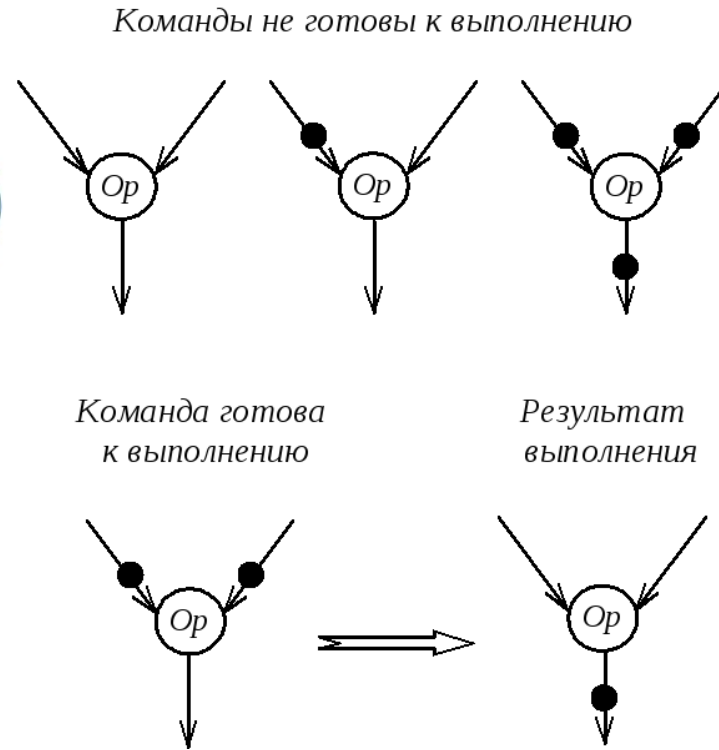
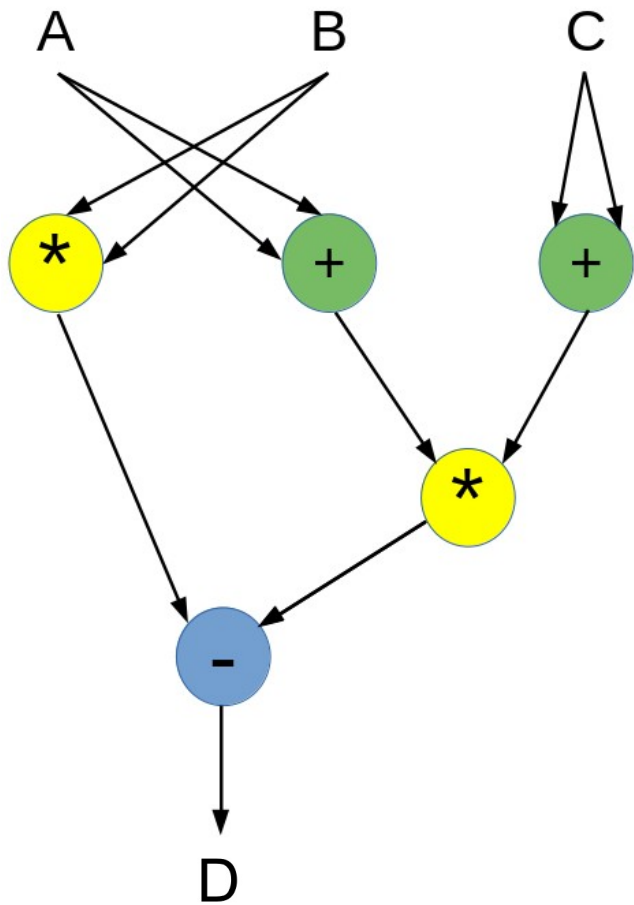
Неявное управление по готовности данных (dataflow)

$$D = B * B - (A + A) * (C + C)$$



Управление по готовности данных с явными подтверждениями

$$D = B * B - (A + A) * (C + C)$$



Управление по готовности данных с неявными подтверждениями

Стратегии управления в вычислительных системах

Условия готовности вычислений

Условие готовности данных

(Information, I-условие).

Перед запуском процесса на его информационных входах должны присутствовать все необходимые аргументы и функции, составляющие операционный пакет. Отсутствие каких-либо данных к моменту запуска процесса ведет к получению неправильного результата.

Условие выделения ресурсов

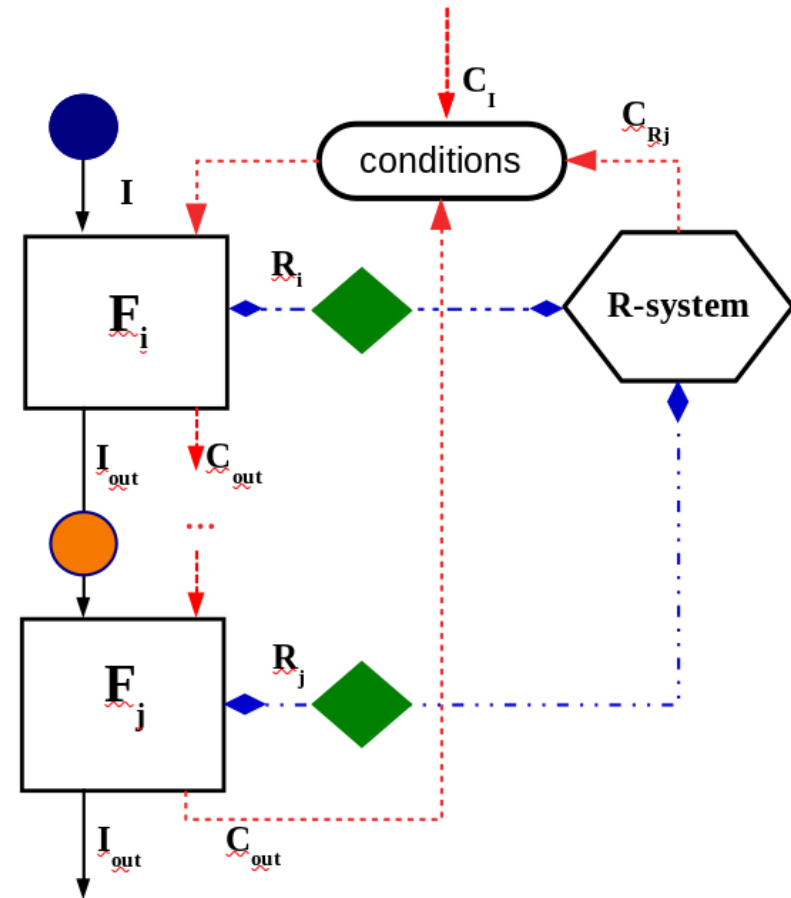
(Resource, R-условие).

Выполнение процесса протекает в соответствующих ресурсах, поэтому, перед его запуском, эти ресурсы должны быть определены и предоставлены.

Условие подтверждения

(Acknowledge, A-условие).

Ресурсы, занимаемые процессом, могут освобождаться или повторно использоваться только после того, как результаты вычислений будут использованы всеми процессами, являющихся приемниками этих результатов в качестве аргументов.



- ← - передача данных решаемой задачи
- ← - передача управляющих воздействий
- ◆ - управление ресурсами системы

Стратегии управления в вычислительных системах

Способы задания управления

Явный (субъективный) (Subjective)

В ходе написания программы разработчик сам создает логику порождения и проверки условий готовности

Неявный

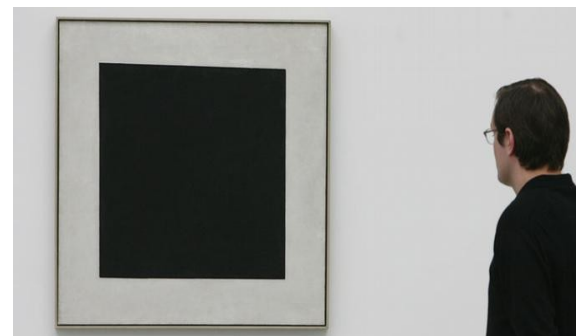
Управление тем или иным условием готовности не задается из расчета, что процессы и так будут выполняться корректно.

- Автоматический (Automatic)

Ресурсы ВС организованы таким образом, что всегда поддерживают выполнение заданного условия.

- Пустой (Empty)

Описание взаимодействия процессов может быть сделано так, что, несмотря на отсутствие в ВС механизма автоматической поддержки, условие готовности всегда истинно, поэтому его и не нужно проверять.



Стратегии управления в вычислительных системах

Множество условий готовности:

$$X = (\text{Information}, \text{Resource}, \text{Acknowledge})$$

Множество способов управления, обеспечивающих соблюдение условий готовности:

$$Y = \{\text{Subjective}, \text{Automatic}, \text{Empty}\}$$

$S_1 = \{\text{Information} \times Y\}$ – множество методов управления данными;

$S_2 = \{\text{Resource} \times Y\}$ – множество методов управления ресурсами;

$S_3 = \{\text{Acknowledge} \times Y\}$ – множество методов управления подтверждениями.

Множество стратегий управления в ВС:

$$S = \{S_1 \times S_2 \times S_3\} = 27$$

Стратегии управления в языках программирования

Множество условий готовности:

$$X = (\text{Information}, \text{Resource}, \text{Acknowledge})$$

Множество способов управления, обеспечивающих соблюдение условий готовности:

Automatic + Empty = Unevident - неявный

$$Z = \{\text{Subjective}, \text{Unevident}\}$$

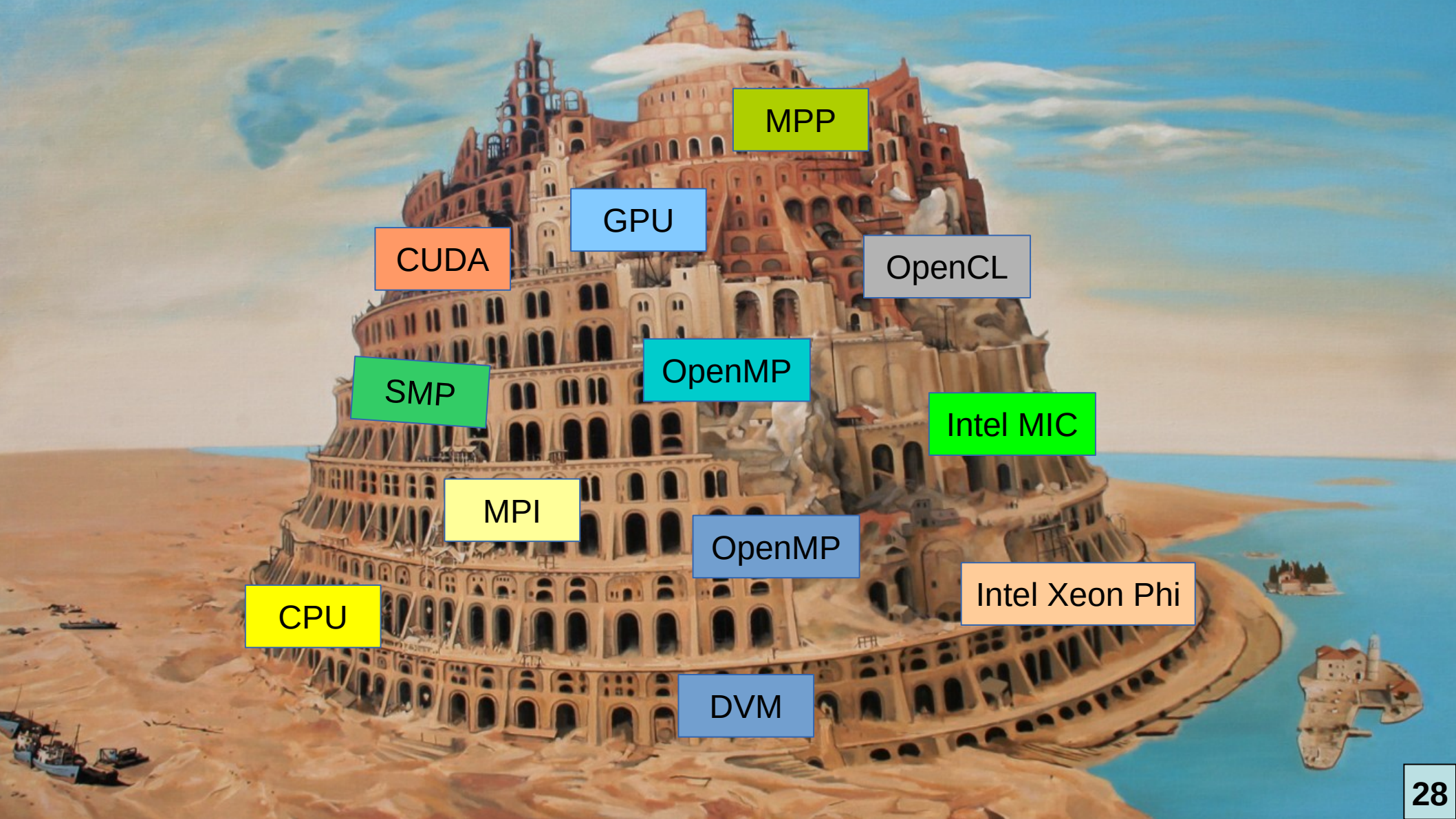
$S_1 = \{\text{Information} \times Z\}$ – множество методов управления данными;

$S_2 = \{\text{Resource} \times Z\}$ – множество методов управления ресурсами;

$S_3 = \{\text{Acknowledge} \times Z\}$ – множество методов управления подтверждениями.

Множество стратегий управления в ВС:

$$S = \{S_1 \times S_2 \times S_3\} = 8$$



MPP

GPU

CUDA

OpenCL

SMP

OpenMP

Intel MIC

MPI

OpenMP

Intel Xeon Phi

CPU

DVM

Используемые источники

1. Легалов А. И. Стратегии управления в вычислительных системах и языках программирования - 2002 г. Электронный ресурс:

<http://softcraft.ru/parallel/strat/>

2. Википедия. Модель вычислений.

https://ru.wikipedia.org/wiki/Модель_вычислений

3. Википедия. Ярусно-параллельная форма графа.

https://ru.wikipedia.org/wiki/Ярусно-параллельная_форма_графа

4. Википедия. Сеть Петри. https://ru.wikipedia.org/wiki/Сеть_Петри