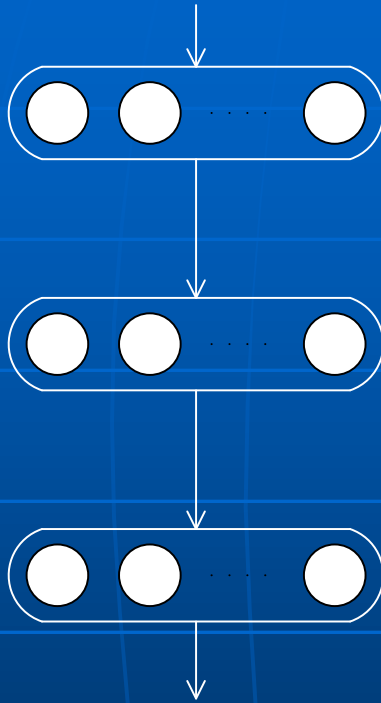


# **Оснoвы параллельных вычислений**

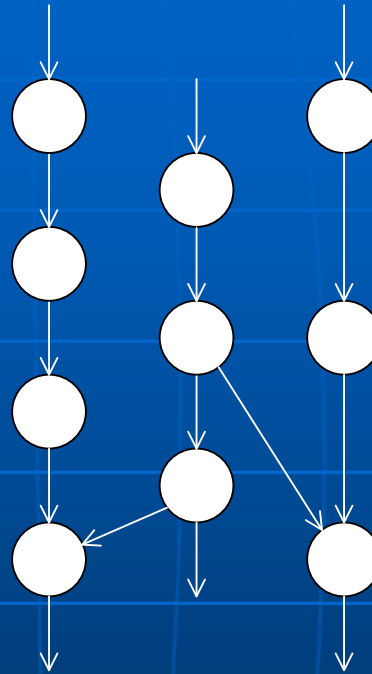
# Виды алгоритмов и их реализация



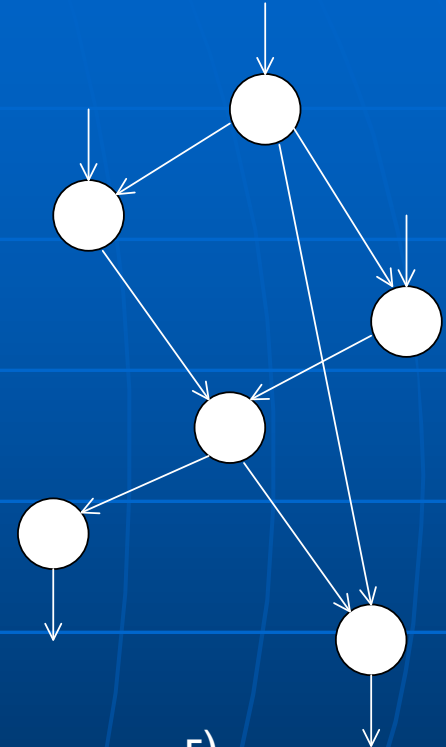
а)



б)



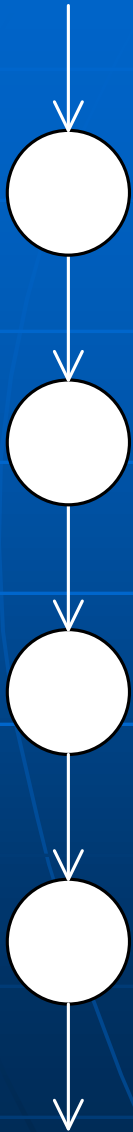
в)



г)

- а) Последовательная алгоритмическая структура
- б) Последовательно-групповая алгоритмическая структура
- в) Совокупность слабосвязанных потоков
- г) Параллельная структура общего вида

# Структура алгоритма – структура ВС



Для *последовательной алгоритмической структуры* вычислительная система проста:

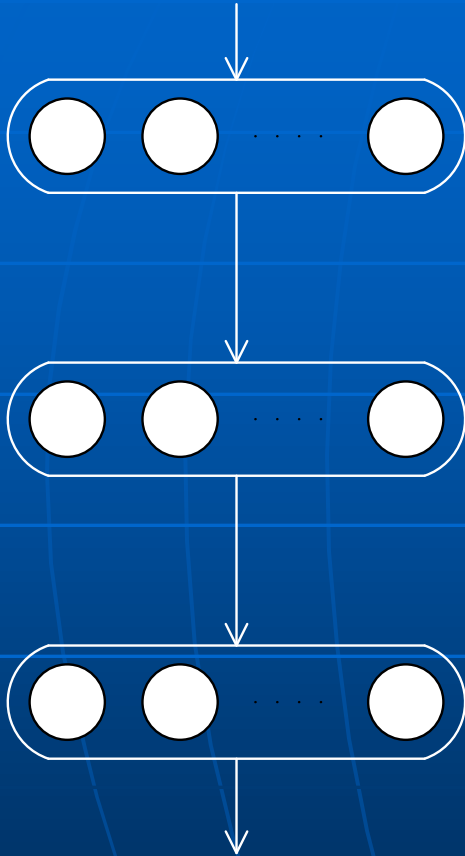
- единственная память для хранения данных и программ
- одно арифметическое устройство, исполняющее текущую команду
- одно устройство управления, осуществляющее контроль за исполнением, счетчик команд, хранящий адрес текущей команды
- простой механизм его модификации

Взаимные связи между перечисленными компонентами также просты.

Быстродействие системы ограничено ее последовательной сущностью и определяется временем исполнения каждой команды.

# Структура алгоритма – структура ВС

## *Последовательно-групповая организация*



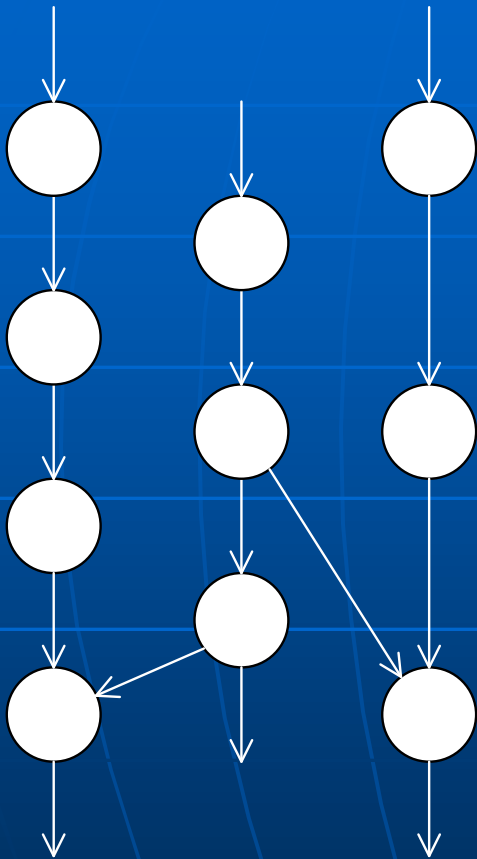
Вычислительные конструкции внутри одной группы могут быть одинаковыми или разными.

Реализация таких алгоритмов на скалярных процессорах сопровождается рядом факторов, ограничивающих максимальную производительность.

Влияние этих факторов уменьшается при введении *векторных команд* и организации системы, которая обеспечивает эффективное исполнение таких команд.

Этот подход реализуется в системах двух типов:  
**матричных**  
**векторно-конвейерных.**

# Структура алгоритма – структура ВС



Для алгоритмов **со слабосвязанными потоками** каждый процесс может выполняться на отдельном процессоре (или ЭВМ), который при необходимости осуществляет взаимодействие с другими процессорами (ЭВМ).

Такие вычислительные системы называются *многопроцессорными* или *многомашинными*.

Ключевым элементом в них является механизм синхронизации и взаимосвязи между процессами.

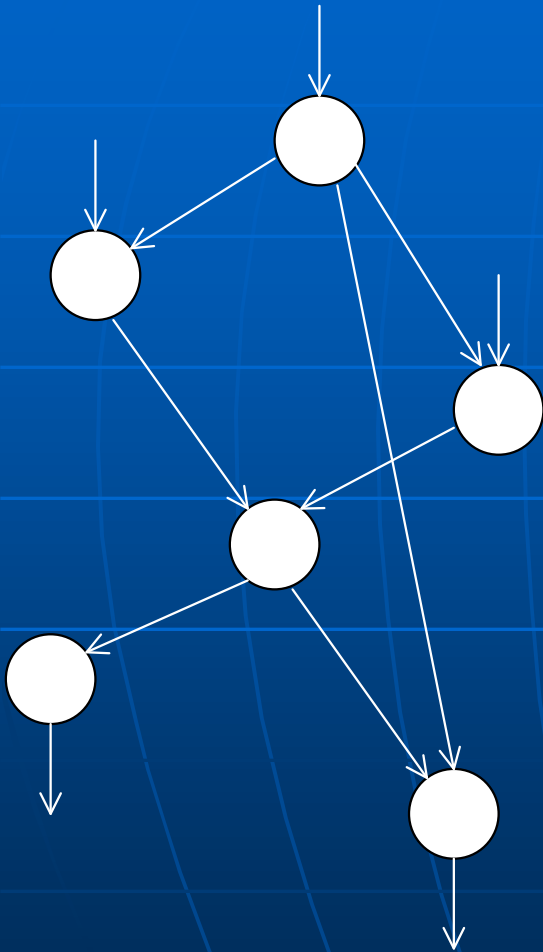
# Структура алгоритма – структура ВС

Для *параллельных структур общего вида* практически невозможно явно задать последовательность выполнения команд в объектном коде.

Из-за того, что в условие инициализации операции входит завершение предшествующих операций, управление порядком исполнения команд становится более сложным.

Для осуществления такого управления предложено два механизма:

- С помощью потоков признаков.
- С помощью потоков данных.



**См. «Численные методы  
параллельных вычислений»**