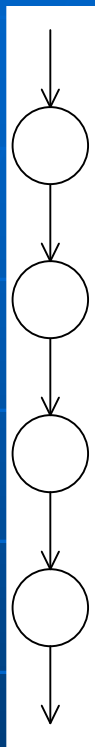
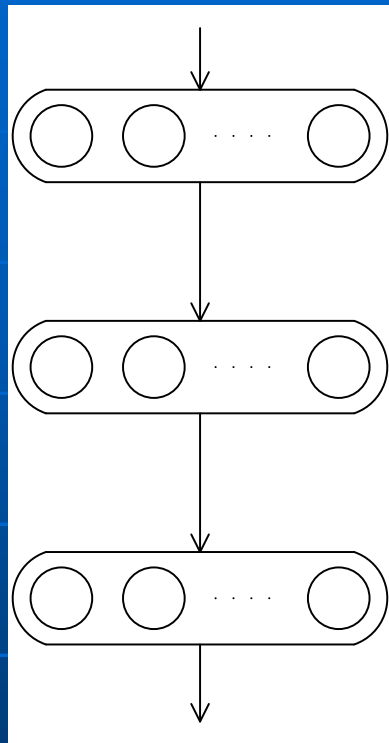


Оснoвы параллельного программирования

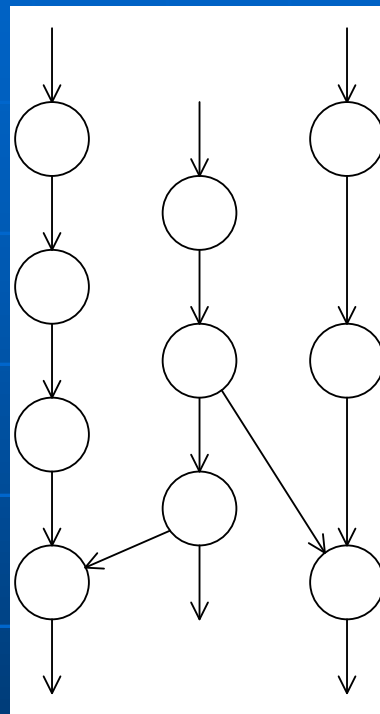
Виды алгоритмов и их реализация



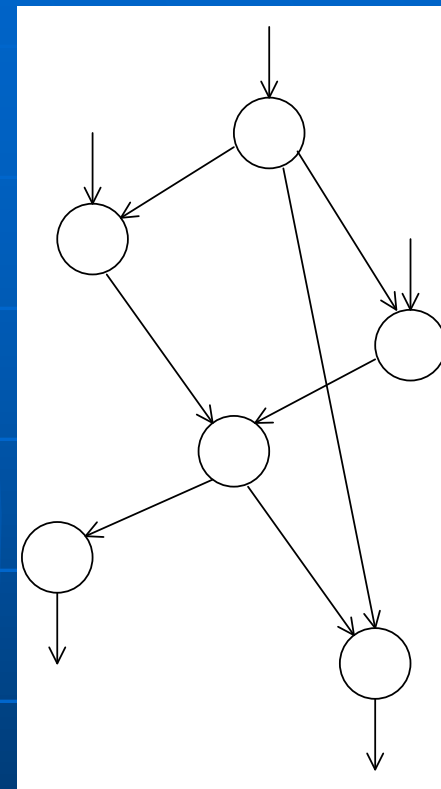
а)



б)



в)



г)

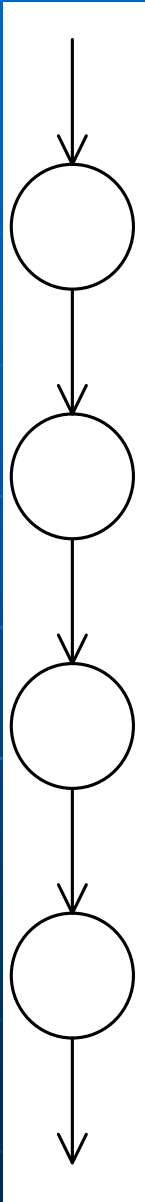
а) Последовательная алгоритмическая структура

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:

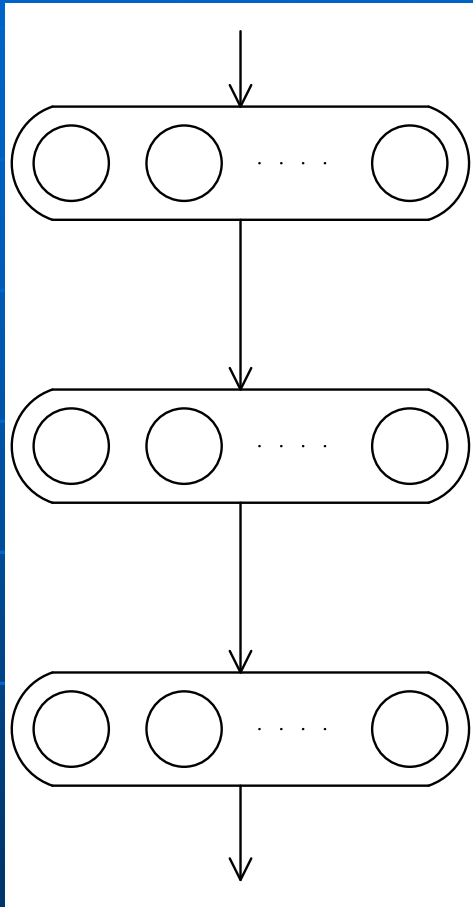
- единственная память для хранения данных и программ
- одно арифметическое устройство, исполняющее текущую команду
- одно устройство управления, осуществляющее контроль за исполнением, счетчик команд, хранящий адрес текущей команды
- простой механизм его модификации

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



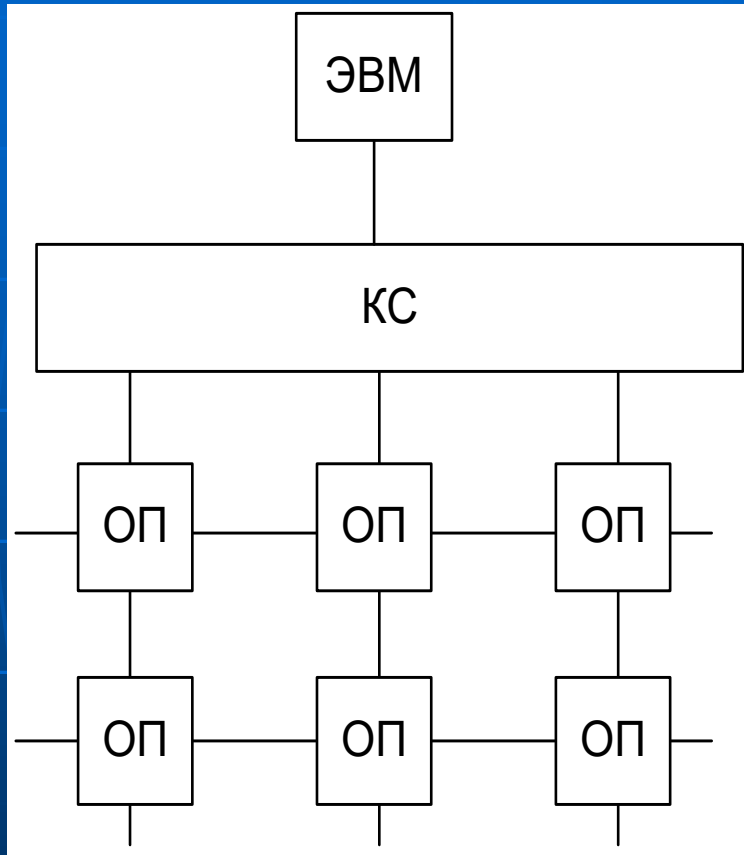
Вычислительные конструкции внутри одной группы могут быть одинаковыми или разными.

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.

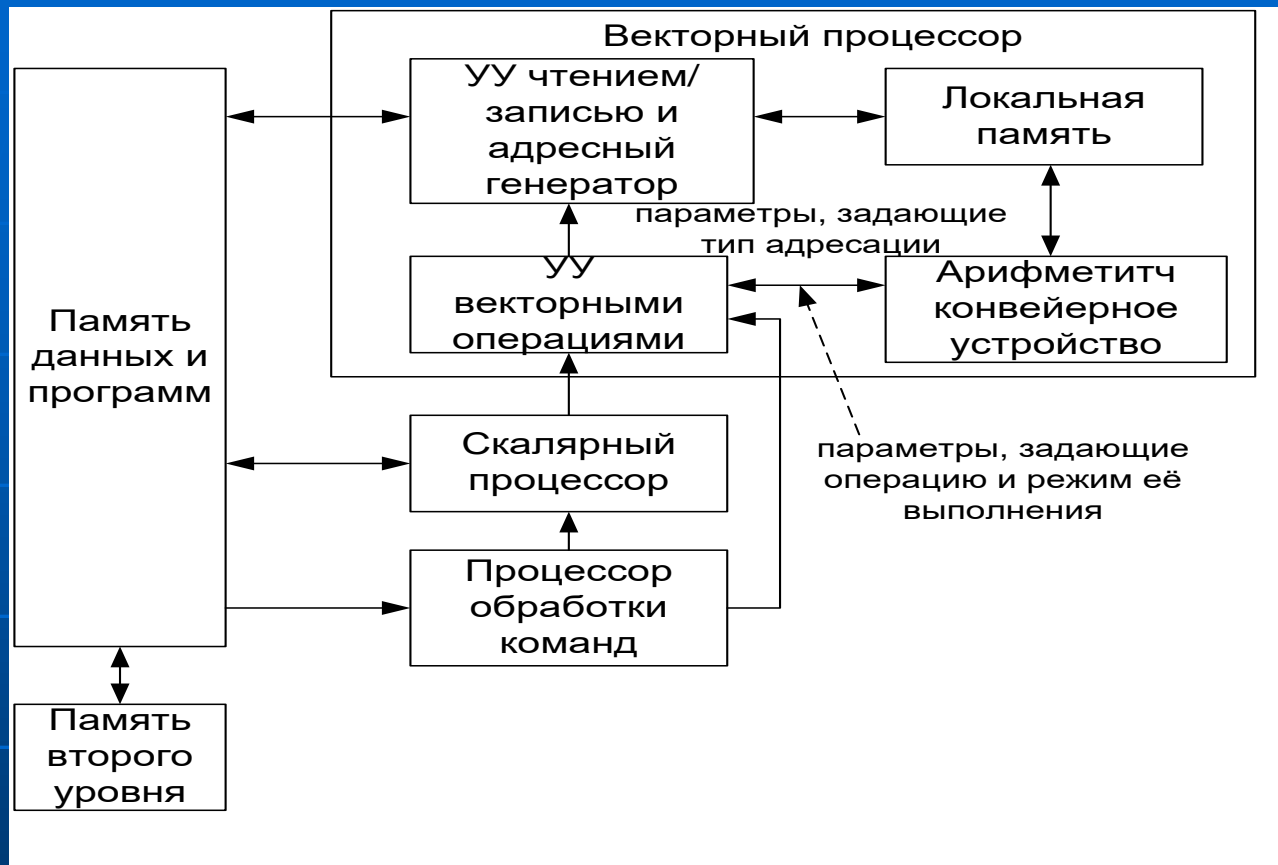
Этот подход реализуется в системах двух типов:
матричных
векторно-конвейерных.

Структура алгоритма – структура ВС



В системах с матричной архитектурой для обработки векторов используются несколько одновременно работающих процессоров

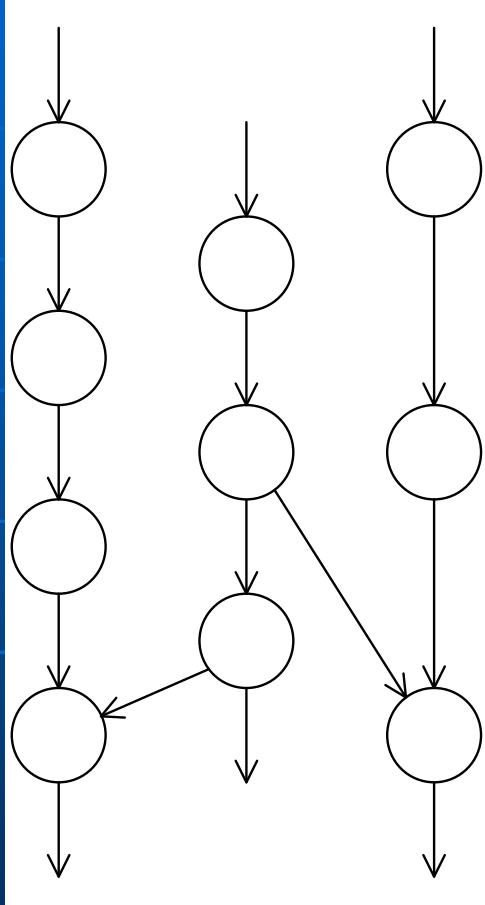
Структура алгоритма – структура ВС



В векторно-конвейерных системах вычислительные конструкции внутри групп исполняются конвейерным процессором.

Оба подхода позволяют достичь значительного ускорения по сравнению со скалярными машинами.

Структура алгоритма – структура ВС

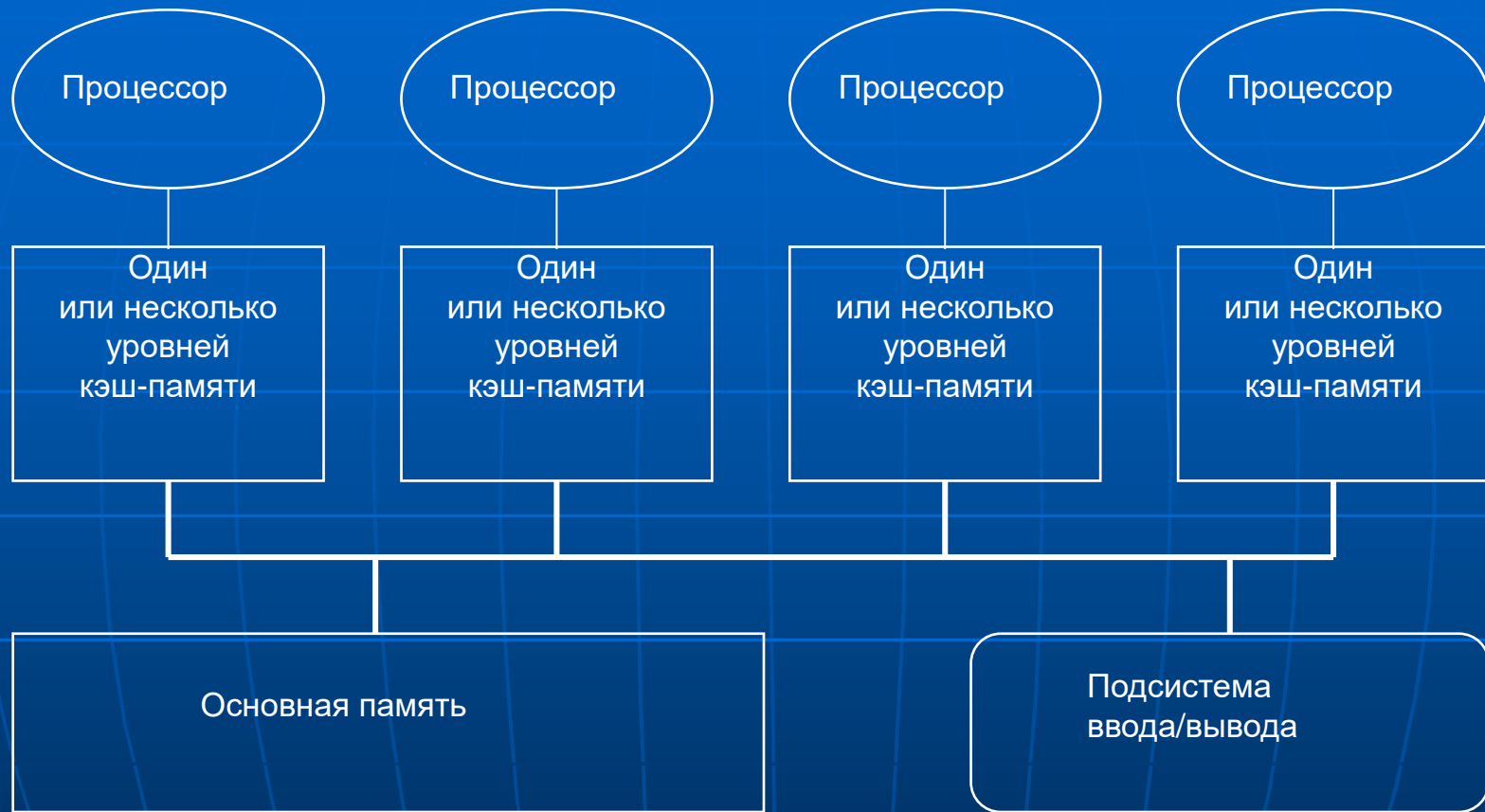


Для алгоритмов *со слабо связанными потоками* каждый процесс может исполняться на отдельном процессоре (или ЭВМ), который при необходимости осуществляет взаимодействие с другими процессорами (ЭВМ).

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

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

Структура алгоритма – структура ВС



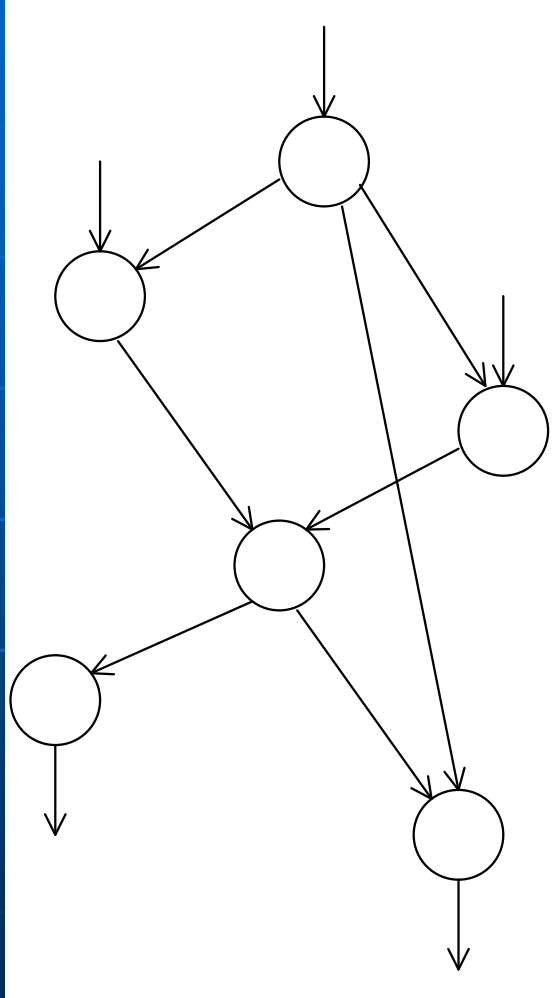
Структура вычислительной системы с общей памятью

Структура алгоритма – структура ВС



Структура вычислительной системы с распределенной памятью

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.

Эволюция языков

Первые универсальные языки

Ассемблер

- переход к символическому кодированию машинных команд;
- впервые в истории развития программирования появились два представления программы:
 - в исходных текстах и
 - в откомпилированном виде.

Фортран

Создан в 1954 году в корпорации IBM
во главе с Джоном Бэкусом (John Backus)

- первый язык программирования высокого уровня
- концепция подпрограмм
- синтаксическая структура языка достаточно сложная для машинной обработки



ЭВОЛЮЦИЯ ЯЗЫКОВ

Cobol

Создан в 1960 году.

- язык для создания коммерческих приложений
- эффективная работа с большими массивами данных

PL/1

Создан в 1964 году корпорацией IBM

- обработка исключительных ситуаций
- поддержка параллелизма
- сложная синтаксическая структура

Пример: IF ELSE = THEN THEN THEN; ELSE ELSE

ЭВОЛЮЦИЯ ЯЗЫКОВ

BASIC

Создан в 1963 году в Дартмурском колледже (Dartmouth College)

- задумывался как средство обучения
- и как первый изучаемый язык программирования

Algol

Создан в 1960 году командой во главе с Петером Науром (Peter Naur)



Эволюция языков

Дальнейшее развитие языков программирования

Pascal

Создан в 1970 году Никлаусом Виртом

- первый широко распространенный язык для структурного программирования
- внедрена строгая проверка типов
- возможность выявлять многие ошибки на этапе компиляции



Никлаус Вирт

Эволюция языков

С-подобные языки

Создан в 1972 году Керниганом и Ритчи

Создавался как язык для разработки операционной системы UNIX



Б. Керниган



Д. Ритчи

В 1986 году Бьярн Страуструп создал первую версию языка C++.

- Появились некоторые объектно-ориентированные черты
- Исправлены некоторые ошибки и неудачные решения языка

ЭВОЛЮЦИЯ ЯЗЫКОВ

Языки обработки данных

APL

Создан в 1957 году

- широкое использование математических символов
- очень мощный синтаксис

Snobol и Icon

Snobol был создан в 1962 году,

а в 1974 появился преемник язык - Icon

Оба предназначен для многофункциональной обработки строк

SETL

Создан в 1969 году.

Предназначен для описания и реализации операций над множествами

Эволюция языков

Lisp и Lisp-подобные языки

В 1958 году создан язык Lisp - язык для обработки списков

Языки Ada и Ada 95

Создан в 1983 году под эгидой Министерства Обороны США

- может выявить много ошибок на этапе компиляции
- в 1995 году был принят стандарт языка Ada 95

ЭВОЛЮЦИЯ ЯЗЫКОВ

Объектно-ориентированные языки

- **Simula** создан в 1967 году
- **Smalltalk** создан в 1972 году
- **Eiffel** создан в 1986 году

Языки параллельного программирования

- **Occam** создан в 1983 году
- **Linda** - модель параллельных вычислений, создан в 1985 году

Языки логического программирования

- **Prolog** создан в 1971 году
- **Parlog** и **Delta Prolog** созданы в 1983 году

ЭВОЛЮЦИЯ ЯЗЫКОВ

Скриптовые языки

Особенности:

- интерпретируемость
- простота синтаксиса
- легкая расширяемость

К ним относятся:

- **JavaScript;**
- **VBScript;**
- **Perl;**
- **Python;**
- **и др.**

Современный Фортран и его расширения

Fortran 90/95

Разработан в 1995. Стандарт был принят в октябре 1996

- включено 17 новых (по сравнению с Fortran 90) функций;
- частично это расширения существующих операторов

High Performance Fortran (HPF)

- Включены богатые средства для распределения данных по процессорам.
- Необходимые коммуникации и синхронизации реализуются компилятором

Другие расширения языка Fortan:

- Fortan D95; Fortan M;
- Vienna Fortran 90;
- Fx;
- Opus

Все стандарты предусматривают полную совместимость "снизу-вверх".

Современный Фортран и его расширения

Fortran-DVM

Разработан в ИПМ РАН - Институт прикладной математики им. М.В.Келдыша
(зав. отд., д.ф.-м.н. В.А. Крюковым, с.н.с., к.ф.-м.н. Н.А. Коноваловым)

DVM-модель выполнения параллельной программы охватывает параллелизм по данным и параллелизм задач.

Область применения: разработка мобильных параллельных программ вычислительного характера.

Перспективы развития языка

- ✓ Развитие средств объектно-ориентированного программирования (предполагается полный набор средств ООП).
- ✓ Средства взаимодействия с языком **Си**.
- ✓ Параметризованные производные типы.
- ✓ Новые средства ввода/вывода (асинхронный ввод/вывод и др.).
- ✓ Новые возможности, касающиеся размещаемых массивов.
- ✓ Исключительные ситуации.
- ✓ Более полная интеграция с операционной системой.

Параллельные расширения и диалекты языков C/C++

Java

Java — объектно-ориентированный язык программирования. Sun официально объявила о языке Java на торгово-промышленной выставке в мае 1995 года.

C#

C# - строго типизированный объектно-ориентированный язык, призванный обеспечить оптимальное сочетание удобства, простоты, выразительности и производительности. Был анонсирован в июне 2000 года компанией Microsoft.

Параллельные расширения и диалекты языков C/C++

Charm++

Объектно-ориентированный язык программирования, ориентированный на вычисления, управляемые потоком данных (data-driven computations).

Первая открытая реализация в 1993 году.

Другие диалекты языков C и C++:

- A++/P++; CC++;
- Cilk; High Performance C (HPC); High Performance C++;
- Maisie; Mentat Programming Language (MPL);
- mpC; MPC++;
- Parsec; pC++; sC++; uC++.

Другие параллельные языки и их расширения

НОРМА

Основные разработчики - ИПМ РАН

(Игорь Борисович Задыхайло, к.ф.-м.н. Кирилл Николаевич Ефимкин)

Предназначен для описания решения вычислительных задач сеточными методами

Adl

Был предложен в 1989 году Эдвином Педнолтом.

Функциональный язык с небольшим числом конструкций и типов данных, разработанный для написания параллельных программ.

StreamIt

Разработан специалистами из Массачусетского технологического института (MIT).

Ориентирован на программирование потоковых приложений.

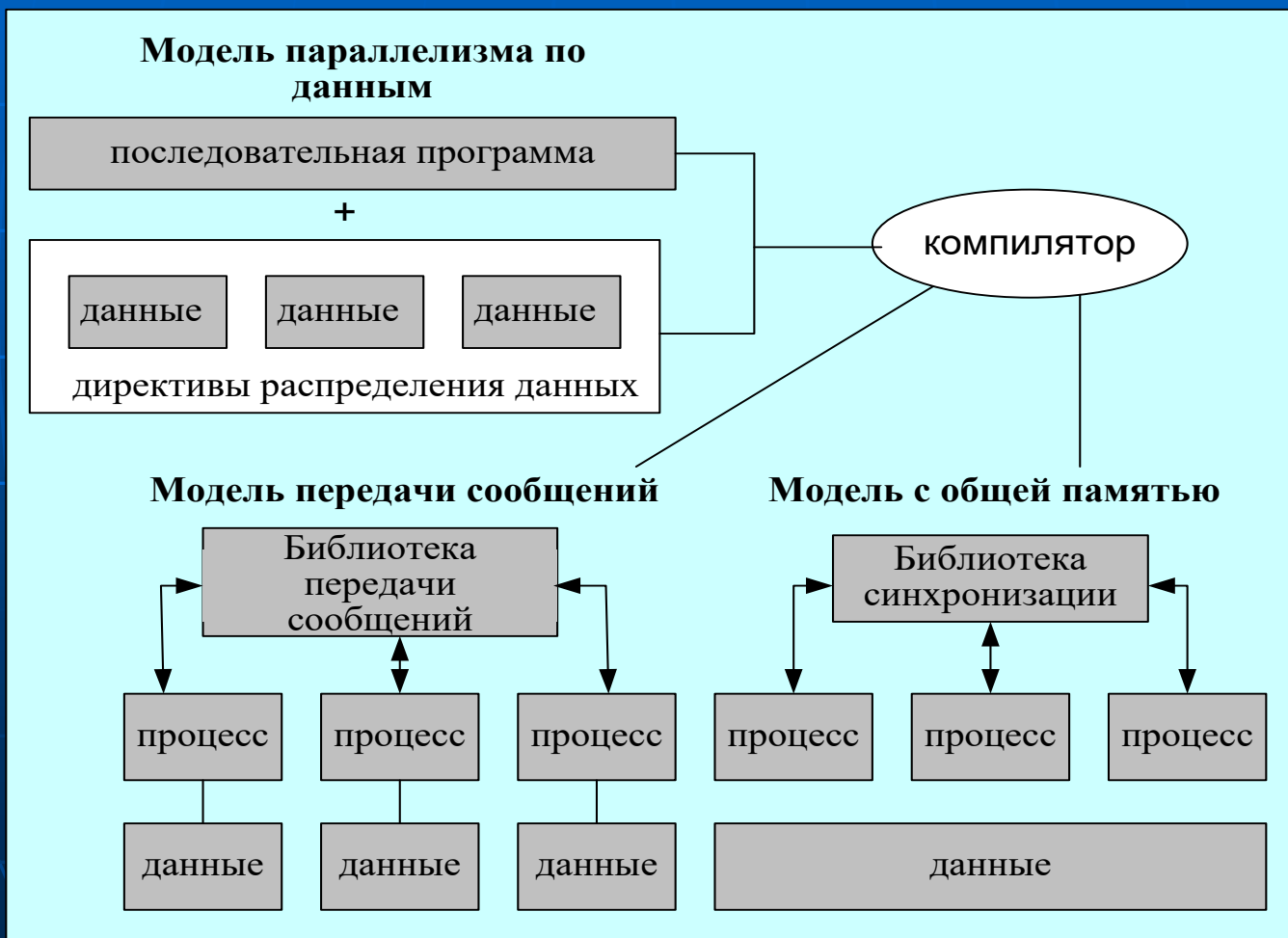
Другие параллельные языки и их расширения

OpenMP

Набор директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с разделяемой общей памятью на языках C, C++ и Fortran.

Три модели параллельного программирования

- ✓ модель с общей памятью (МОП),
- ✓ модель параллелизма по данным (МПД).
- ✓ модель передачи сообщений (МПС),



Модель параллелизма по данным. NPF.

Данные последовательной программы распределяются программистом по процессорам параллельной машины.

В современных ВС последовательная программа преобразуется компилятором в параллельную программу, выполняющуюся либо в модели передачи сообщений, либо в модели с общей памятью.

Модель параллелизма по данным имеет следующие достоинства:

- ✓ Параллелизм по данным является естественным параллелизмом вычислительных задач.
- ✓ В модели параллелизма по данным сохраняется последовательный стиль программирования. Программист не должен представлять программу в виде взаимодействующих процессов и заниматься низкоуровневым программированием передач сообщений и синхронизации.
- ✓ Распределение вычисляемых данных между процессорами – это способ повышения локализации данных.

Модель параллелизма по данным и управлению. DVM.

Эта модель, положенная в основу языков параллельного программирования Фортран-DVM и Си-DVM, объединяет достоинства модели параллелизма по данным и модели параллелизма по управлению.

Цели создания DVM-системы

- ✓ Простота разработки параллельных программ.
- ✓ Мобильность среди различных архитектур ЭВМ (последовательных и параллельных).
- ✓ Высокая эффективность выполнения программы.
- ✓ Повторное использование (композиция параллельных приложений из нескольких модулей).
- ✓ Единая модель параллелизма для языков Си и Фортран 77, как следствие этого, единая система поддержки, отладки, оценки и предсказания производительности.

Модель параллелизма по данным и управлению. DVM.

DVM-система состоит из следующих компонентов:

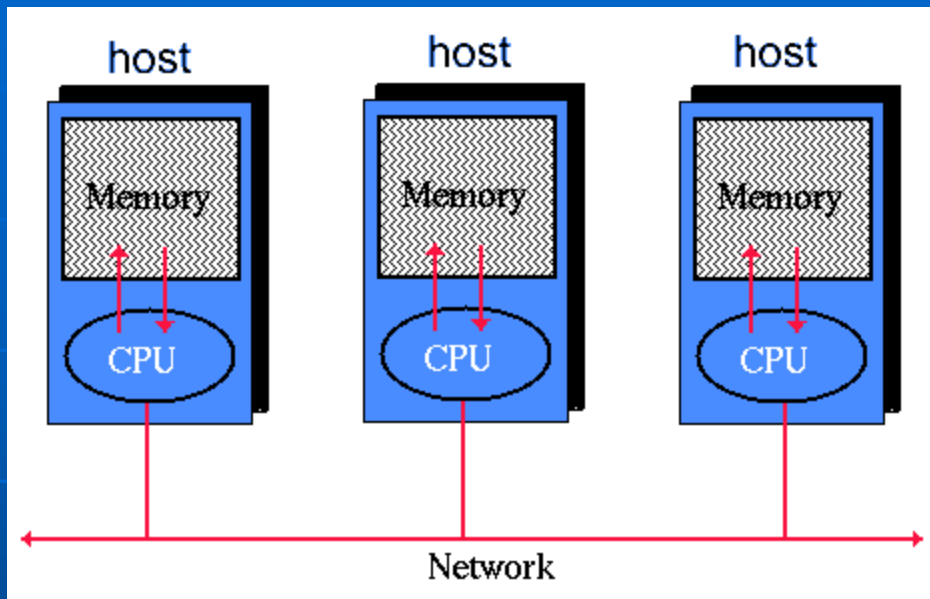
- ✓ Компилятор Fortran-DVM
- ✓ Компилятор C-DVM
- ✓ Библиотека поддержки Lib-DVM
- ✓ DVM-отладчик
- ✓ Предсказатель производительности DVM-программ
- ✓ Анализатор производительности DVM-программ

Модель параллелизма по управлению. OpenMP.

Так как нет ограничений на использование общих данных, то программист должен явно специфицировать общие данные и упорядочивать доступ к ним с помощью средств синхронизации.

В языках высокого уровня логически независимые нити (потoki) вычислений определяются на уровне функциональных задач или итераций цикла.

Модель передачи сообщений. MPI.



Обмен сообщениями через коммуникационную сеть

Что содержит сообщение

- ✓ идентификатор процесса-отправителя (rang)
- ✓ адрес, по которому размещены пересылаемые данные процесса отправителя
- ✓ тип пересылаемых данных
- ✓ количество данных
- ✓ идентификатор процесса-получателя
- ✓ адрес, по которому должны быть размещены данные процессом получателем

Модель передачи сообщений. MPI.

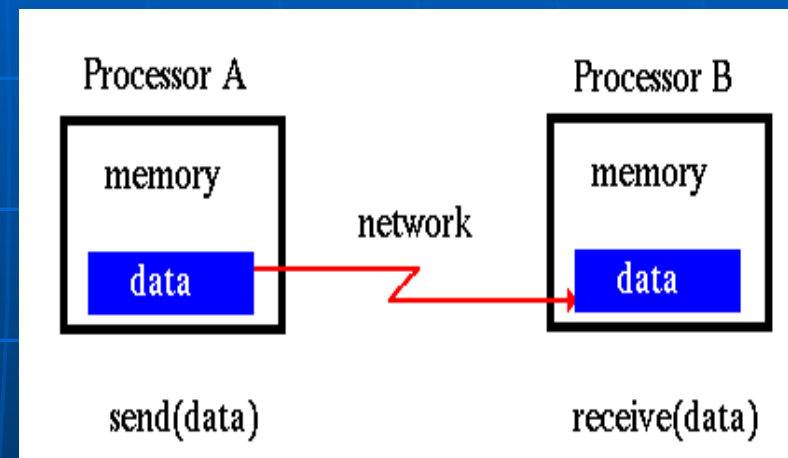
Формы обмена сообщениями

Двухточечный обмен (point-to-point)

- ✓ Синхронный обмен
- ✓ Асинхронный обмен
- ✓ Блокирующие прием/передача
- ✓ Неблокирующие прием/передача

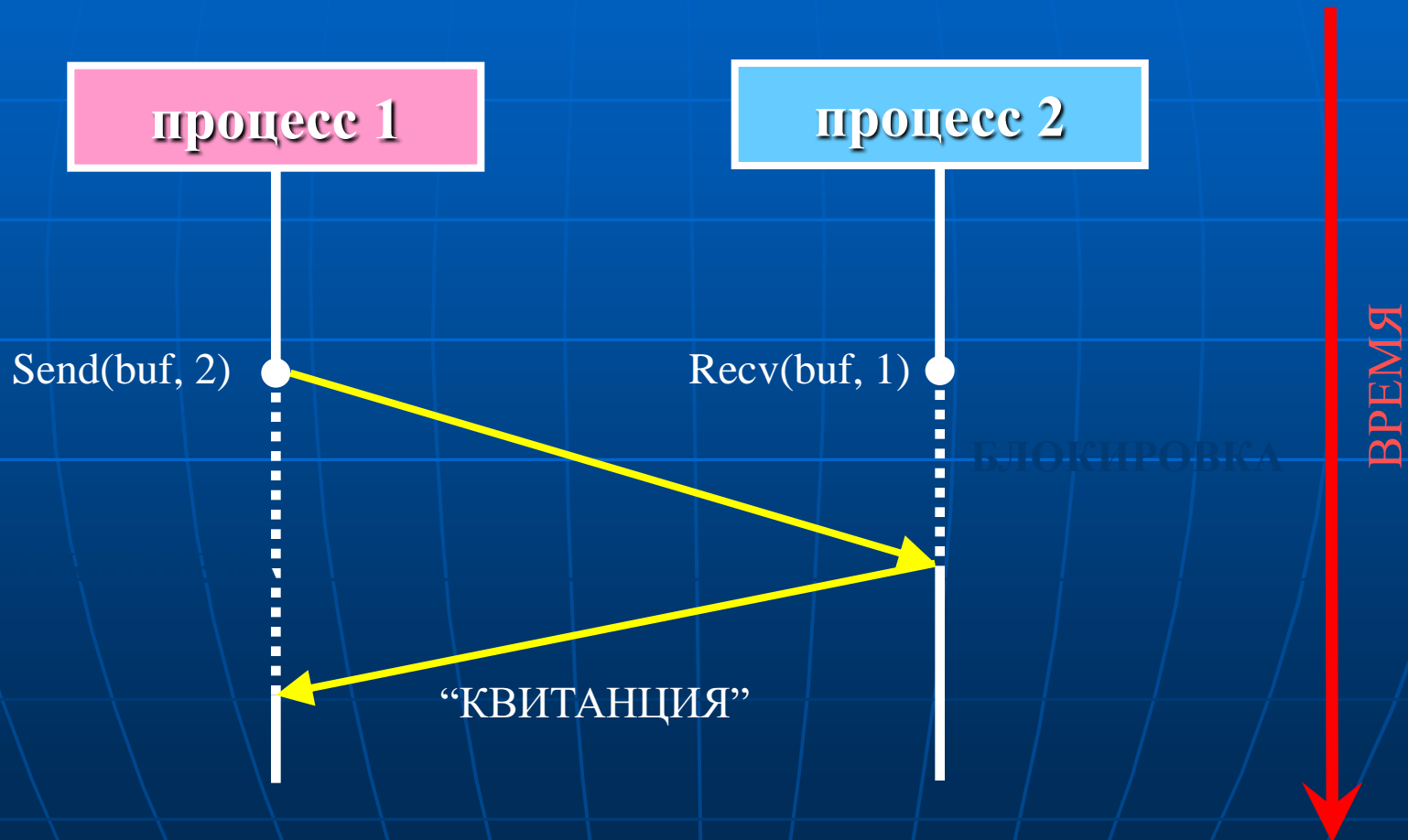
Коллективный обмен

- ✓ Широковещательная передача
- ✓ Обмен с барьером
- ✓ Операции приведения



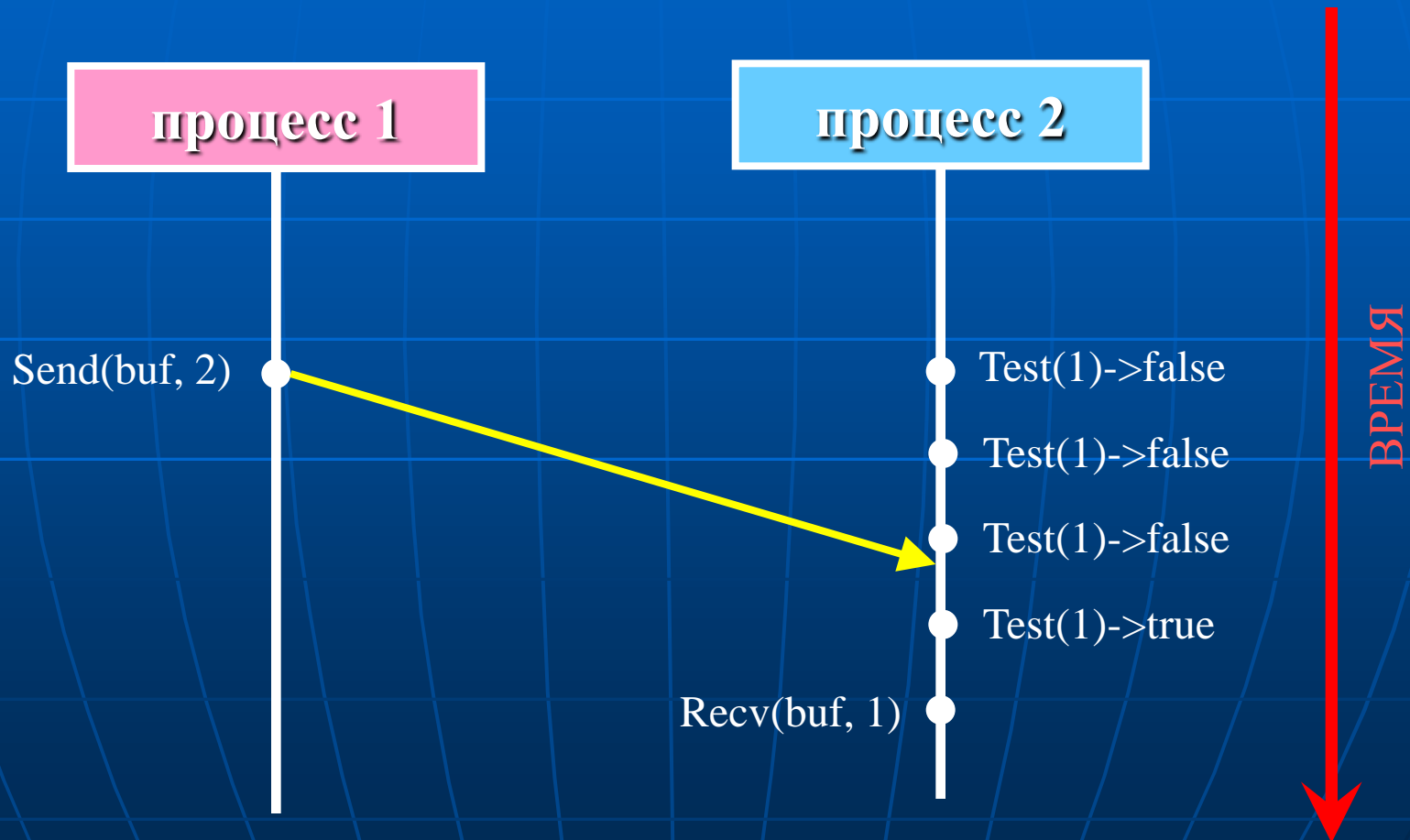
Блокирующий обмен сообщениями

Управление вызывающему процессу возвращается только после того, как данные приняты или переданы (или скопированы во временный буфер).



Неблокирующий обмен сообщениями

Управление возвращается немедленно (т.е. процесс блокируется до завершения операции), и фактическая приемопередача происходит в фоне.



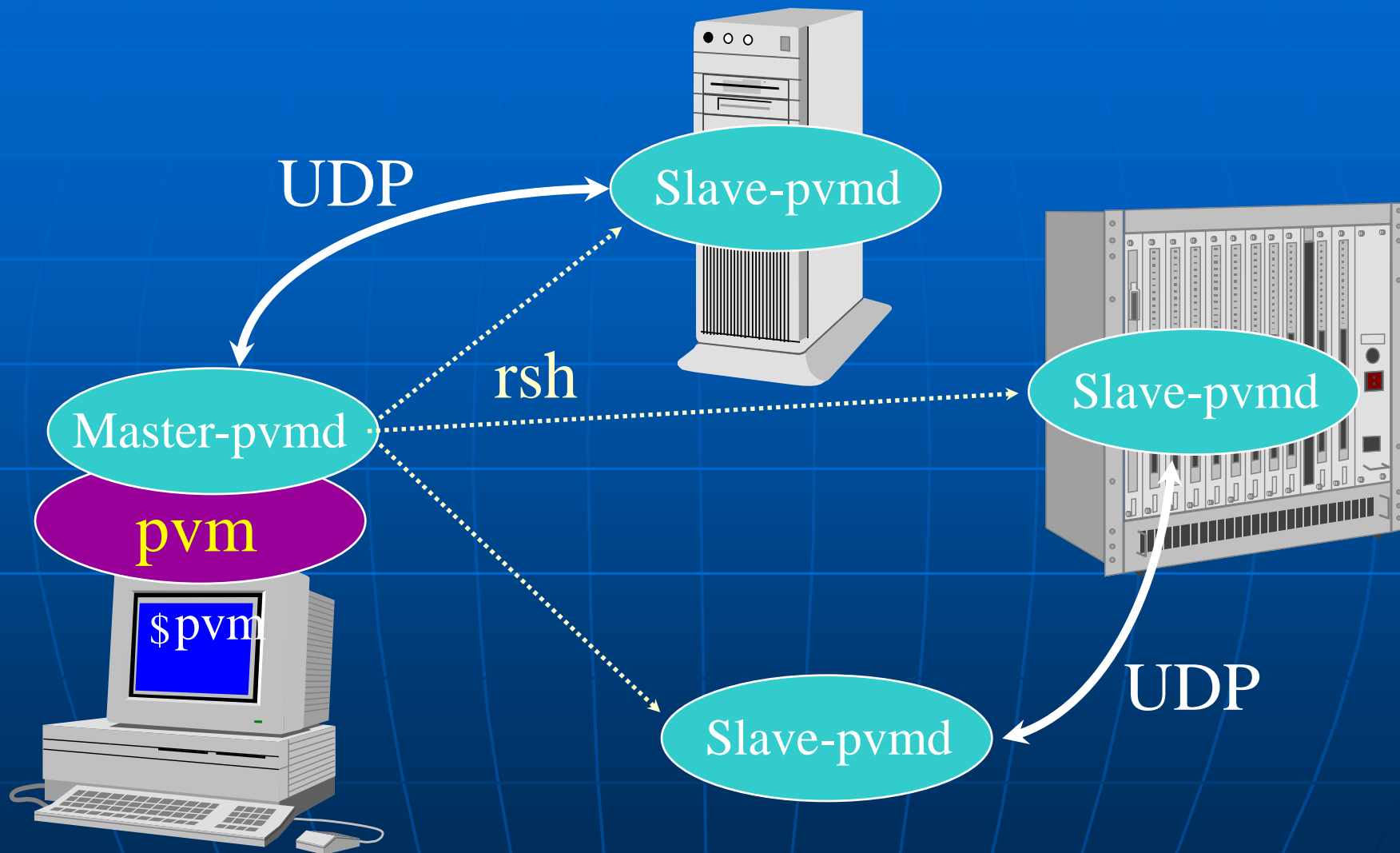
PVM

PVM (Параллельная Виртуальная Машина) - это пакет программ, который позволяет использовать связанный в локальную сеть набор разнородных компьютеров, работающих под операционной системой Unix, как один большой параллельный компьютер.

Структурно PVM состоит из двух обязательных частей:

- ✓ библиотек программирования с заголовочными файлами (изначально поддерживались Фортран и Си), которыми проблемный программист пользуется для написания приложения,
- ✓ программы-демона, называемой PVMD ("демон" - термин из жаргона Юникса; это программа, предназначенная для выполнения автоматических, без диалога с пользователем, сервисных действий, например, обслуживания запросов от других задач. Если вспомнить DOS - это полный аналог резидентной программы).

Порядок загрузки в вычислительной сети



Контроль за функционированием



Выводы

Параллельные вычисления – область науки, которая затрагивает:

- разработку вычислительных методов,
- изучение структурных свойств алгоритма,
- создание новых языков программирования,
- вопросы, связанные с самим процессом **конструирования вычислительной техники**.

Эффективность решения задачи зависит от производительности компьютеров, размеров и структуры их памяти, пропускной способности каналов связи.

Но в не меньшей степени она зависит и от уровня развития языков программирования, компиляторов, операционных систем, численных методов и многих сопутствующих математических исследований.