Оглавление

[**1. Классификация Флинна вычислительных систем** 2](#_Toc185711033)

[**2. Определение ускорения параллельной программы** 2](#_Toc185711034)

[**3. Основные модели параллельного программирования** 3](#_Toc185711035)

[**4. Информационный граф параллельного алгоритма** 4](#_Toc185711036)

[**5. Законы Амдала. Законы Амдала. Каким будет максимальное ускорение параллельной программы, в которой арифметических операций выполняется только одним процессором?** 5](#_Toc185711037)

[**6. Этапы разработки параллельного алгоритма** 6](#_Toc185711038)

[**7. Основной рейтинг суперЭВМ** 7](#_Toc185711039)

[**8. Технология ОреnMP. Структура программы. Модель программирования.** 8](#_Toc185711040)

[**9. Синтаксис директив OpenMP.** 10](#_Toc185711041)

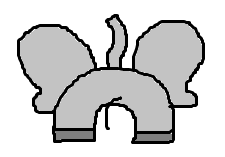
[**10. Создание параллельных потоков в OpenMP.** 10](#_Toc185711042)

[**11. Функции runtime-библиотеки. Директивы master и singlе.** 11](#_Toc185711043)

[**12. Директива for в OpenMP.** 12](#_Toc185711044)

[**13. Ошибки в многопоточных программах, состояние гонки.** 13](#_Toc185711045)

[**14. Директивы синхронизации в OpenMP .** 14](#_Toc185711046)

[**15. Параллельная редукция в OpenMP** 16](#_Toc185711047)

[**16.** **Управление видимостью переменных в OpenMP .** 16](#_Toc185711048)

[**17.** **MPI. Модель программирования.** 17](#_Toc185711049)

[**18.** **Понятие параллельной программы в MPI.** 18](#_Toc185711050)

[**19.** **Операции передачи данных в MPI.** 19](#_Toc185711051)

[**20.** **Понятие коммуникаторов.** 19](#_Toc185711052)

[**21.** **Виртуальные топологии.** 20](#_Toc185711053)

[**22.** **Базовые функции MPI.** 21](#_Toc185711054)

[**23.** **Основные операции передачи и приема в MPI.** 22](#_Toc185711055)

[**24.** **Коллективные функции обмена в MPI.** 22](#_Toc185711056)

[**25.** **Широковещательная рассылка данных в MPI.** 23](#_Toc185711057)

[**26.** **Передача данных от всех процессов одному. Операция редукции.** 24](#_Toc185711058)

[**27.** **Организация и работа с суперкомпьютерами.** 24](#_Toc185711059)

# **1. Классификация Флинна вычислительных систем**

Классификация Флинна — это подход к типологизации вычислительных систем на основе параллелизма в потоках данных и инструкций. Выделяют следующие категории:

1. **SISD** (Single Instruction, Single Data):
   * Однотипная последовательность инструкций обрабатывает один поток данных.
   * Пример: обычный процессор.
   * Характеристики:
     + Отсутствие параллелизма.
     + Используется в большинстве традиционных ПК.
2. **SIMD** (Single Instruction, Multiple Data):
   * Один поток инструкций одновременно выполняется над несколькими потоками данных.
   * Пример: графические процессоры (GPU).
   * Характеристики:
     + Высокая производительность для однотипных операций над большими массивами данных.
     + Применение: обработка изображений, научные расчеты.
3. **MISD** (Multiple Instruction, Single Data):
   * Несколько потоков инструкций обрабатывают один поток данных.
   * Характеристики:
     + Редко встречается на практике.
     + Возможное применение: системы контроля ошибок.
4. **MIMD** (Multiple Instruction, Multiple Data):
   * Несколько потоков инструкций обрабатывают разные потоки данных.
   * Пример: суперкомпьютеры, кластеры.
   * Характеристики:
     + Наиболее универсальная модель.
     + Используется в большинстве современных систем параллельных вычислений.

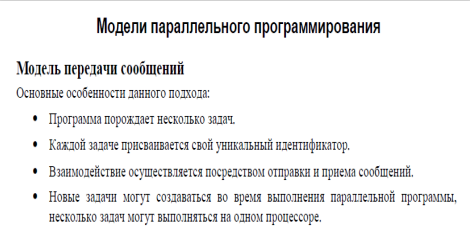
# **2. Определение ускорения параллельной программы**

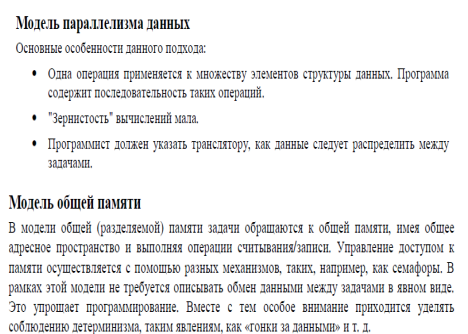
Ускорение (“speedup”) — это показатель эффективности параллельной программы, определяемый формулой:

где:

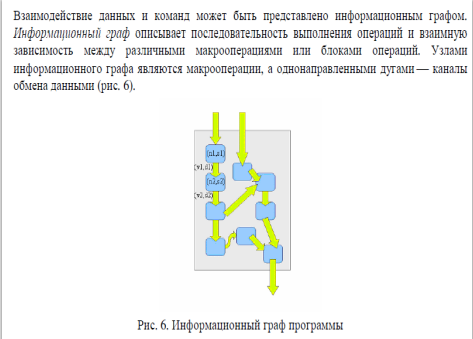
* — время выполнения программы на одном процессоре,
* — время выполнения программы на i процессорах.

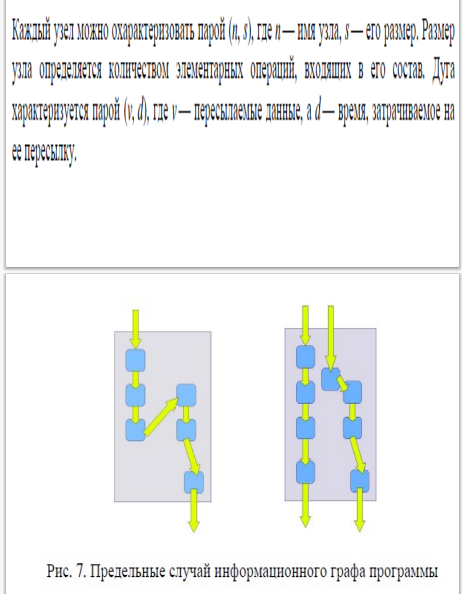
# **3. Основные модели параллельного программирования**



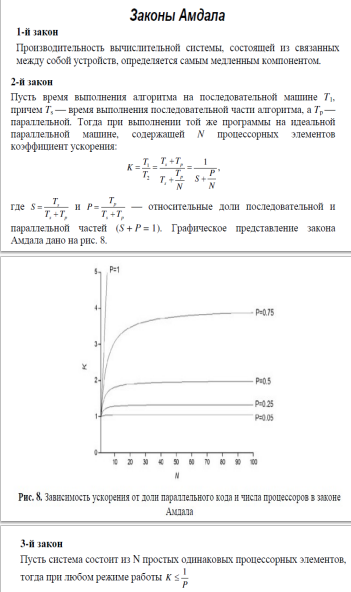


# **4. Информационный граф параллельного алгоритма**



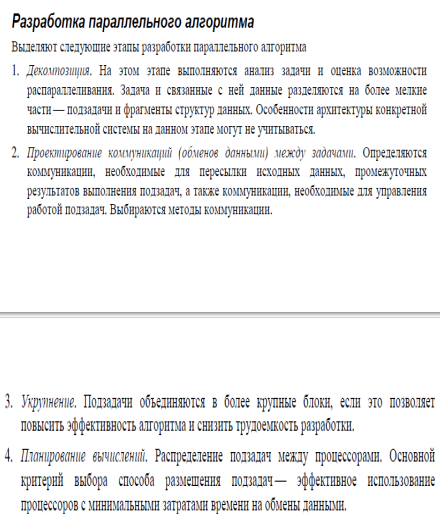
****

# **5. Законы Амдала. Законы Амдала. Каким будет максимальное ускорение параллельной программы, в которой арифметических операций выполняется только одним процессором?**



Каким будет максимальное ускорение параллельной программы, в которой арифметических операций выполняется только одним процессором?  
скорее всего 1, что будет означать что ускорения нет, потому что это то же самое что и последовательный алгоритм

# **6. Этапы разработки параллельного алгоритма**

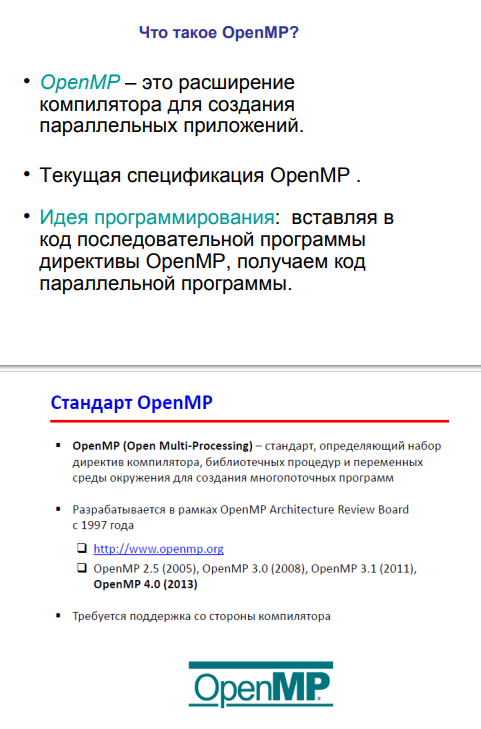


# **7. Основной рейтинг суперЭВМ**

**TOP500** — глобальный рейтинг суперкомпьютеров, обновляемый дважды в год.



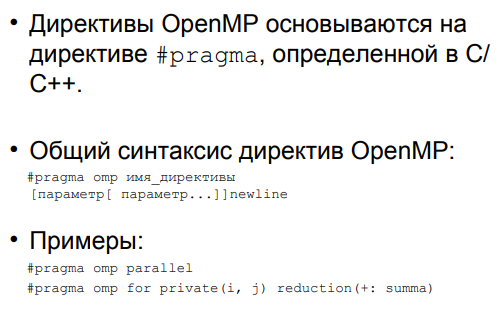
# **8. Технология ОреnMP. Структура программы. Модель программирования.**

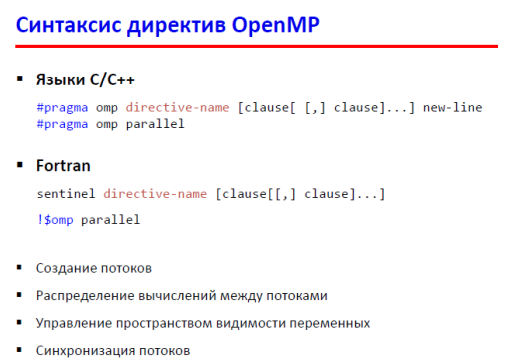




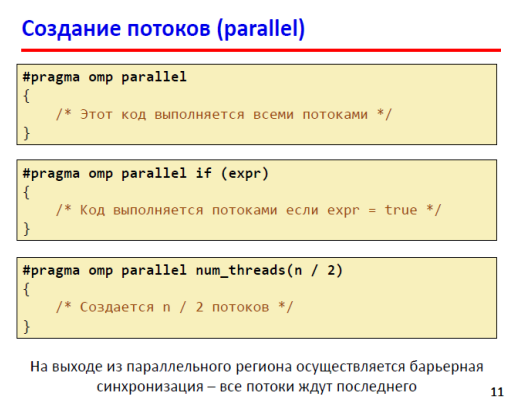
****

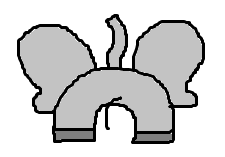
# **9. Синтаксис директив OpenMP.**



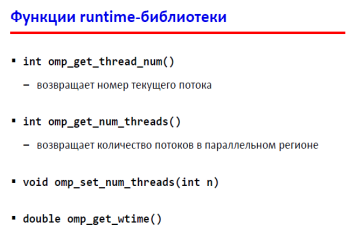
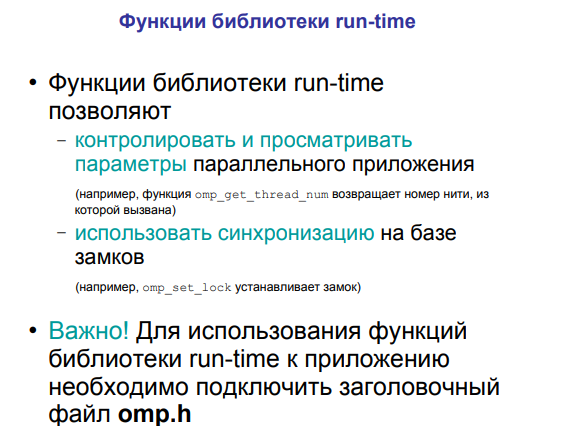


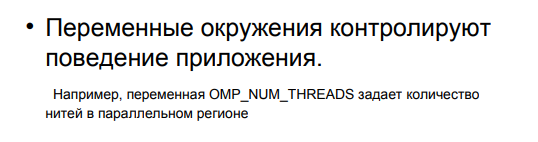
# **10. Создание параллельных потоков в OpenMP.**

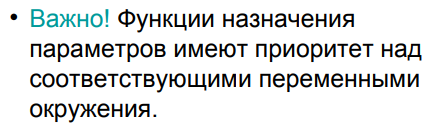


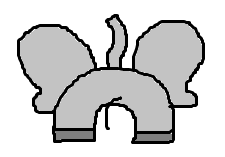
Директива OpenMP **#pragma omp parallel if (expr)** используется для условного создания параллельной области. Если логическое выражение **expr** оценивается как true, тогда создаётся параллельная область, и код внутри неё выполняется несколькими потоками. Если **expr** оценивается как false, параллельная область не создаётся, и весь код выполняется только главным потоком.(Чатик написал) 

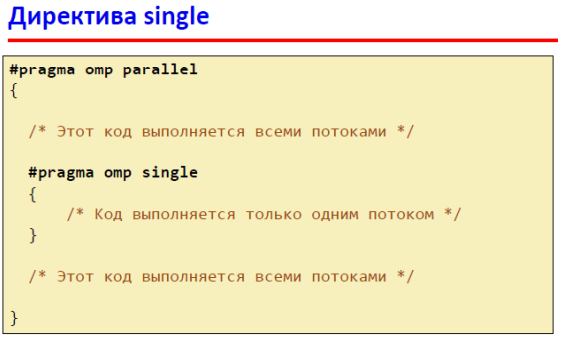
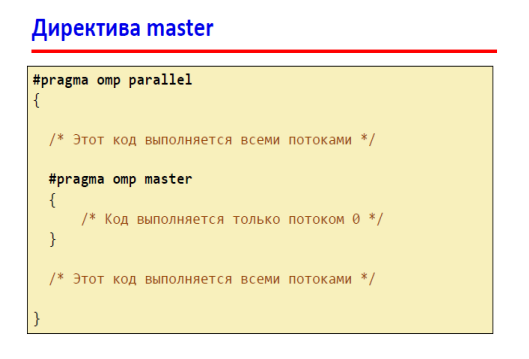
# **11. Функции runtime-библиотеки. Директивы master и singlе.**







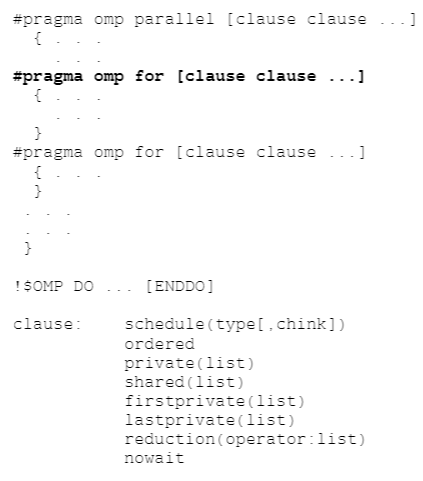


**Короче Master это когда кусок кода выполняется именно процессом с номером 0, это кажется главный процесс(поток)**

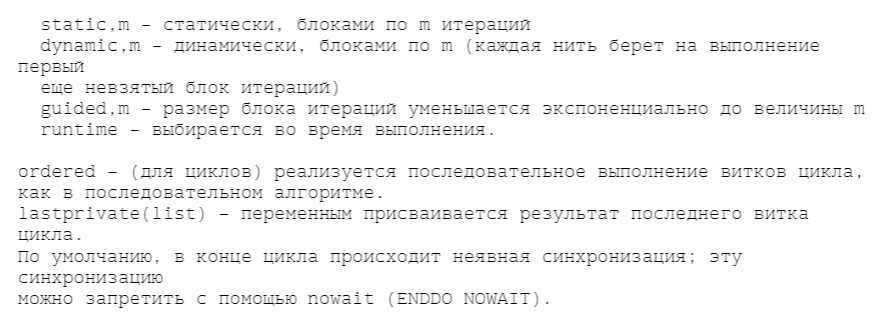
**А Single выполняется видимо любым**

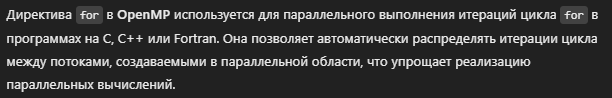


# **12. Директива for в OpenMP.**

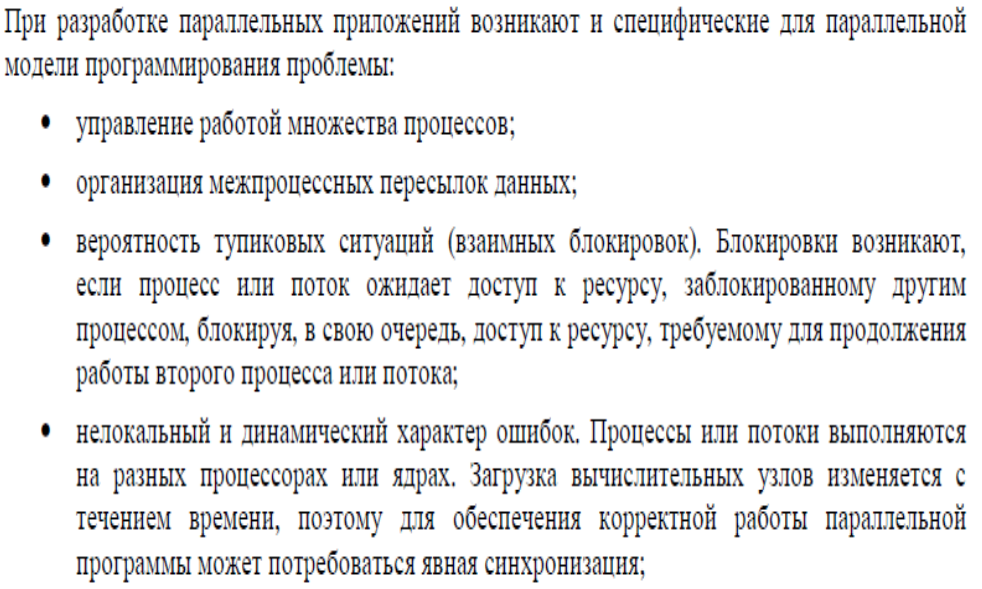
 

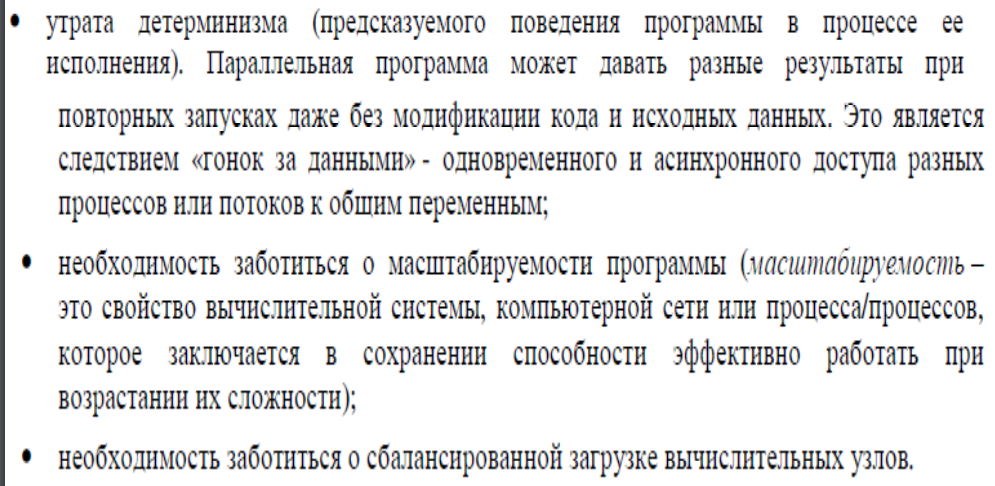


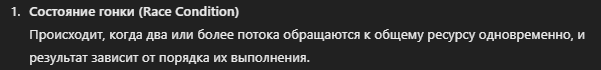


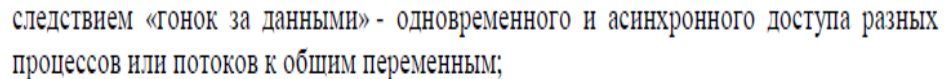


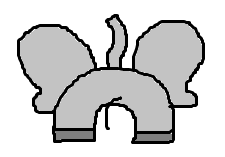
# **13. Ошибки в многопоточных программах, состояние гонки.**



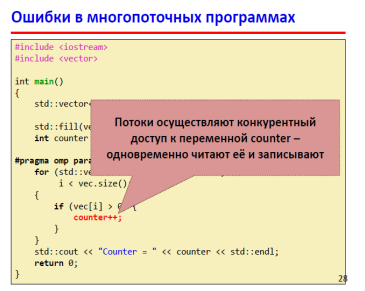




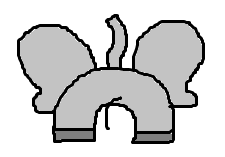
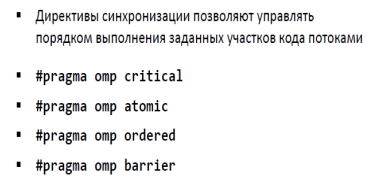


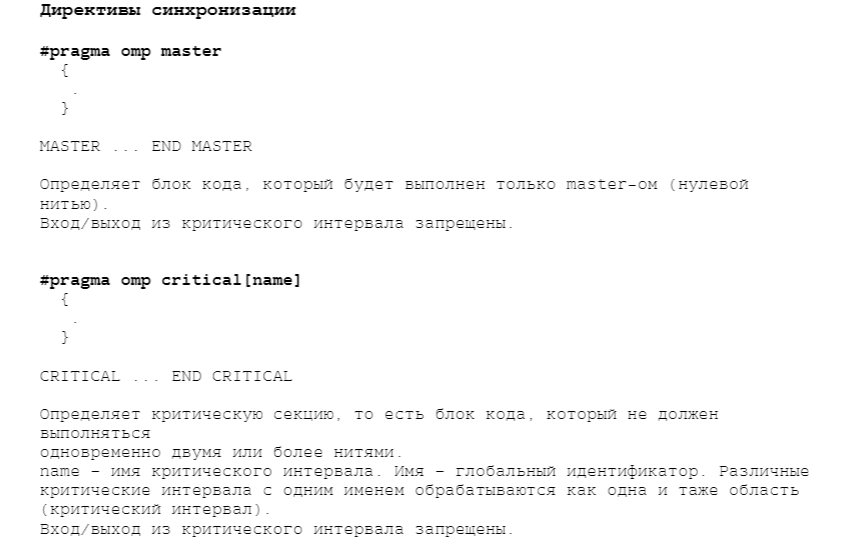


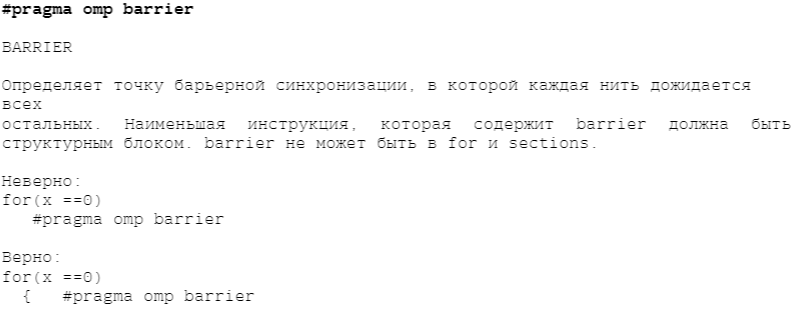


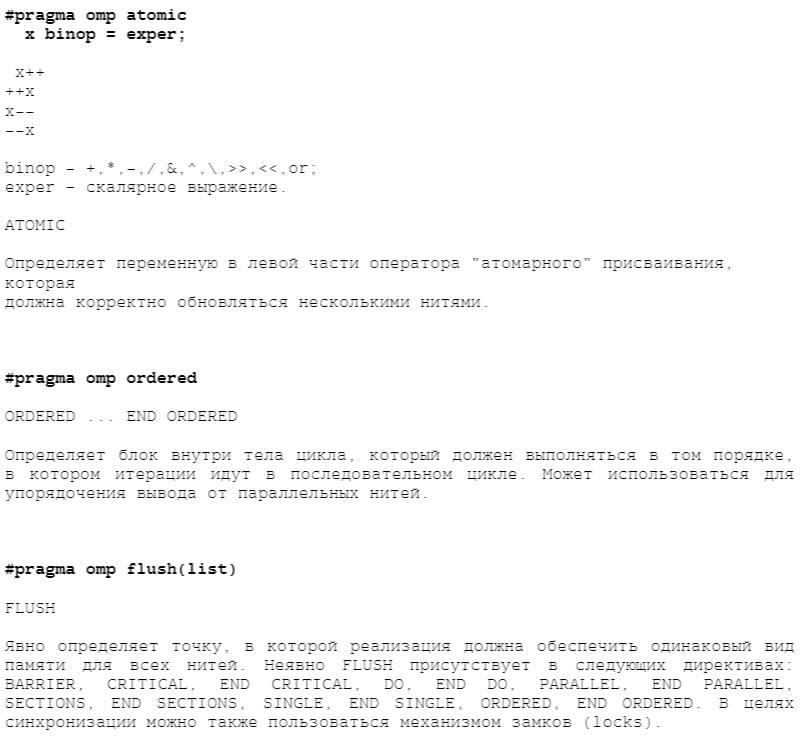


# **14. Директивы синхронизации в OpenMP .**

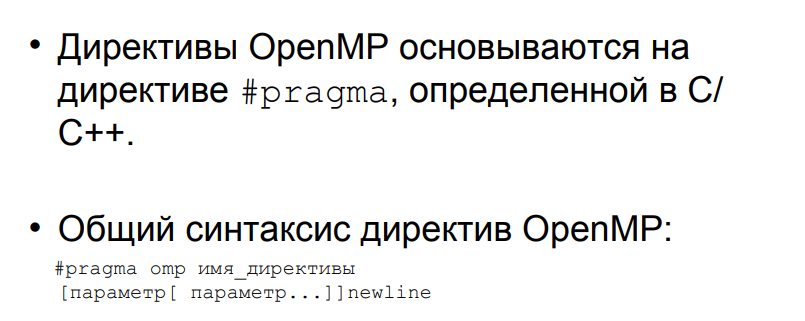


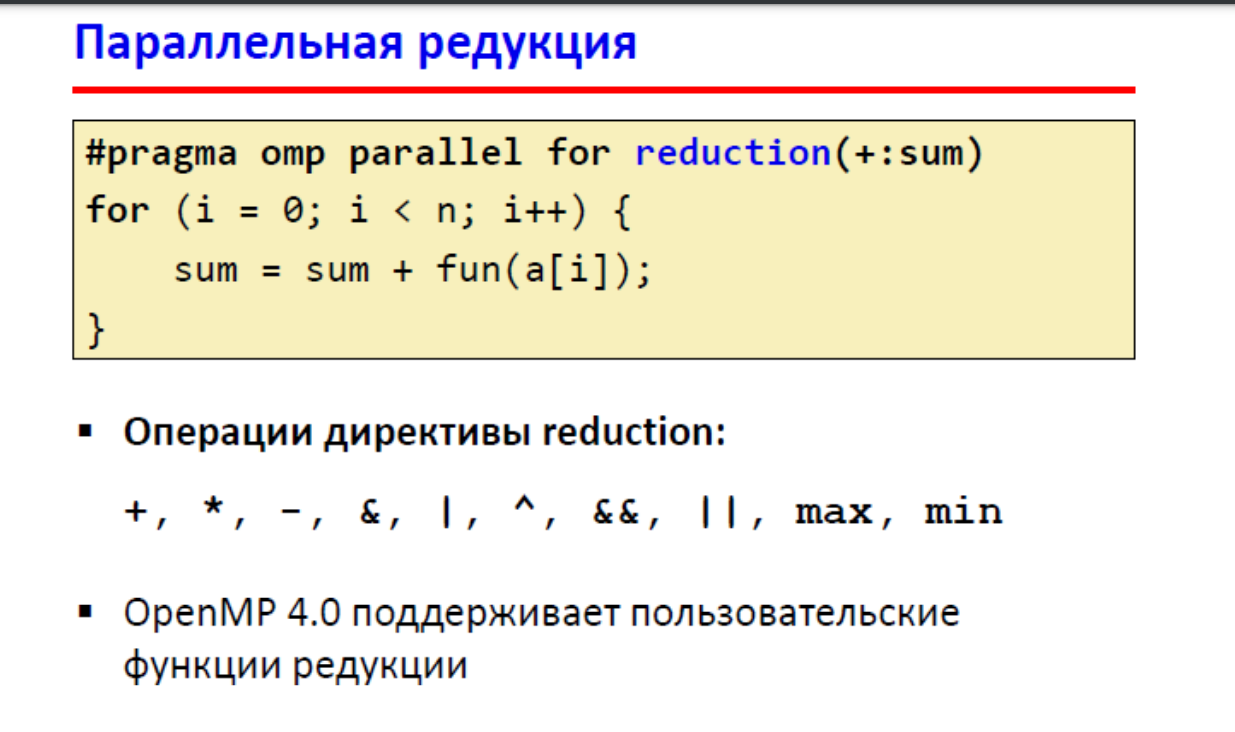




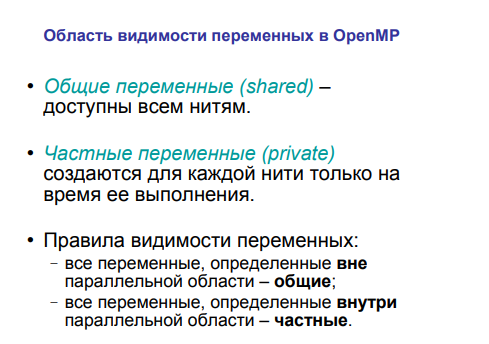


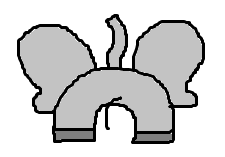
# **15. Параллельная редукция в OpenMP**

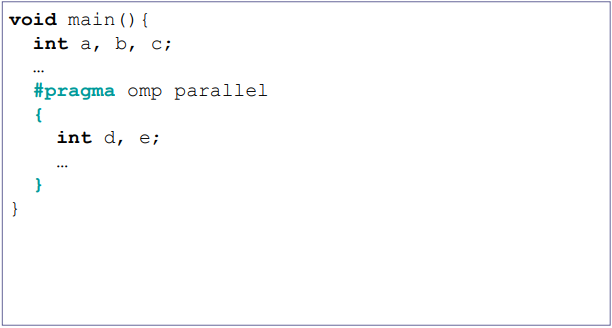


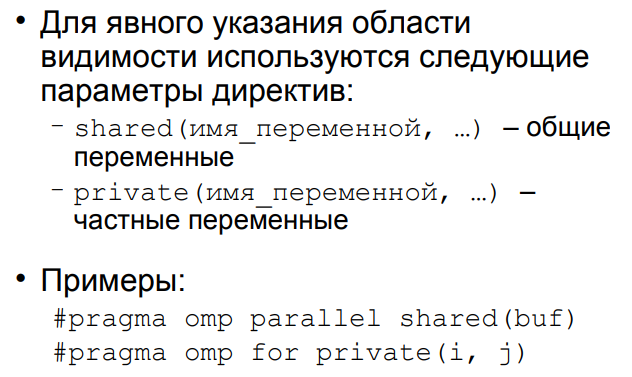


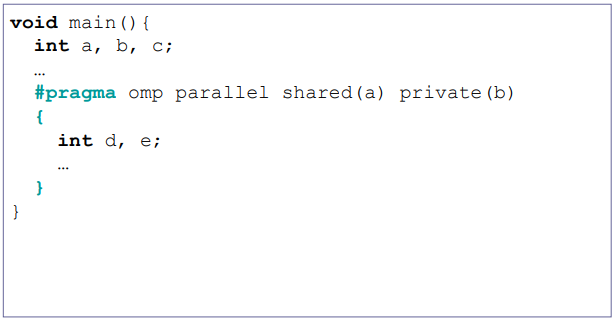
1. **Управление видимостью переменных в OpenMP .**

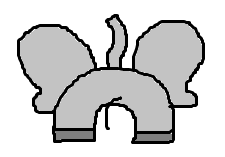




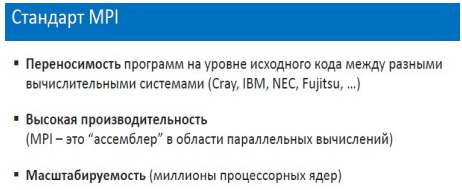
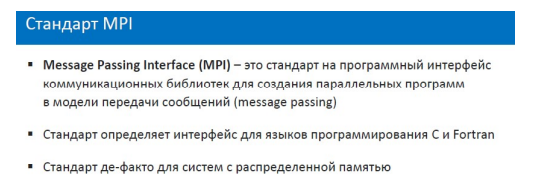


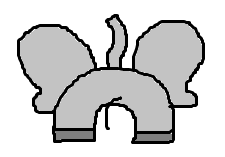


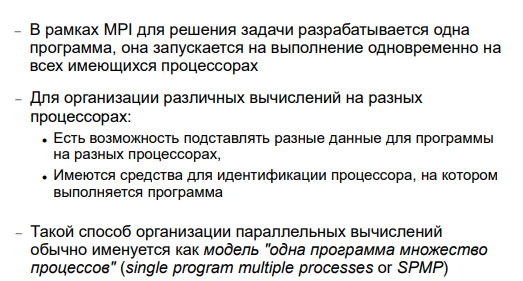




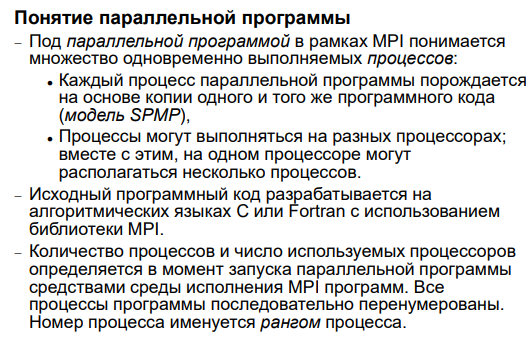
1. **MPI. Модель программирования.**



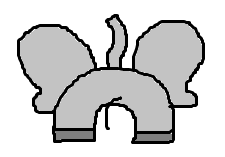
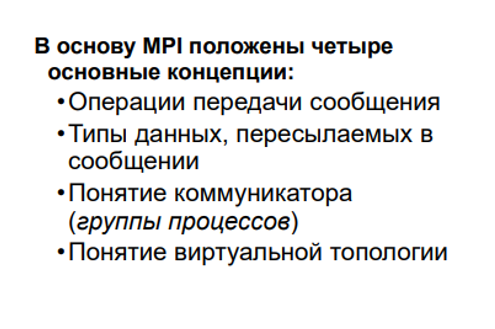




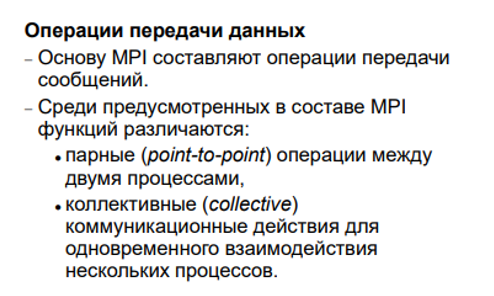
1. **Понятие параллельной программы в MPI.**



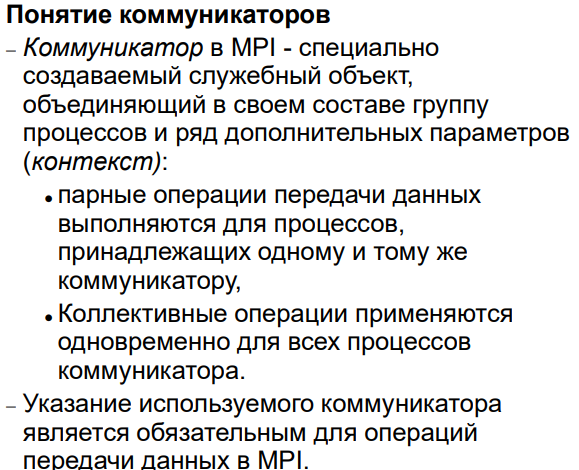
**Перед 19 вопросом, Какие то концепции MPI можно типа сказать что вот такие есть а дальше рассказывать 19 20 или 21 вопрос. Про типы данных отдельно вроде не говорится но вообще это видимо MPI\_Datatype там в функциях приема и передачи такое есть**

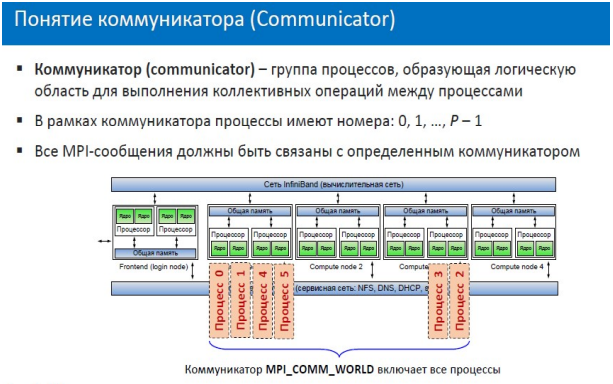
**Вот это**

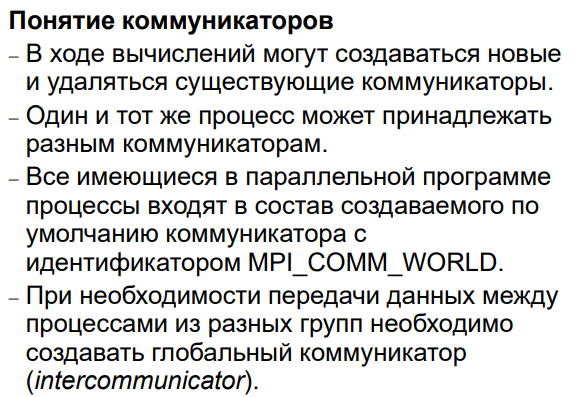
1. **Операции передачи данных в MPI.**



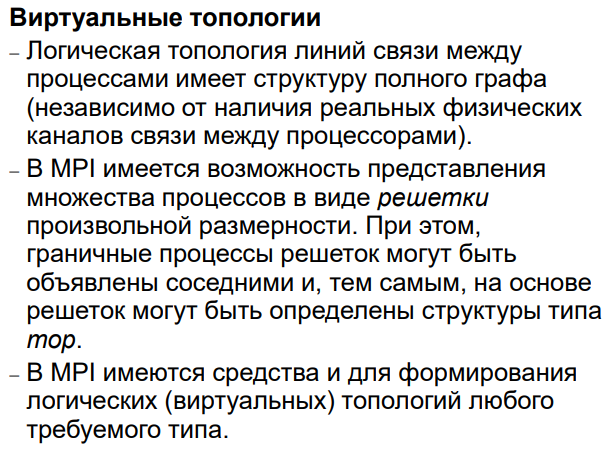
1. **Понятие коммуникаторов.**

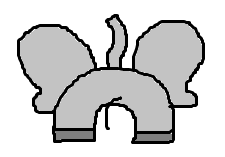




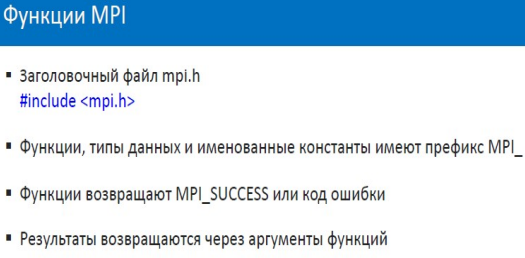


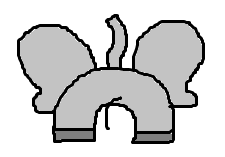
1. **Виртуальные топологии.**

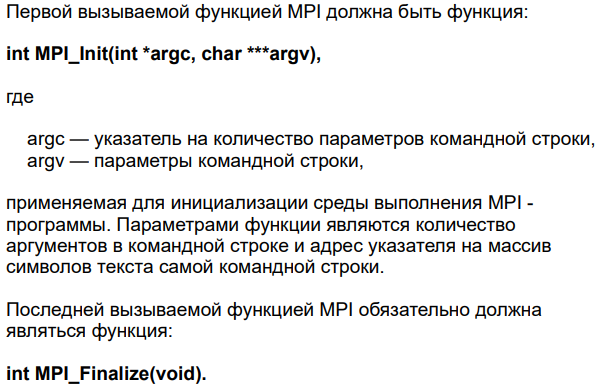


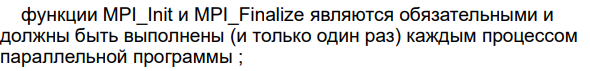


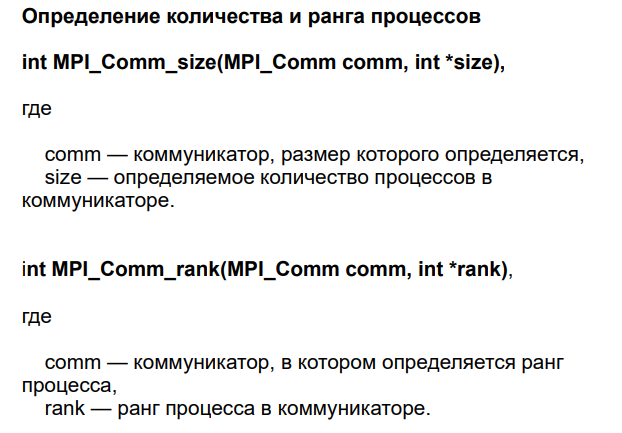
1. **Базовые функции MPI.**

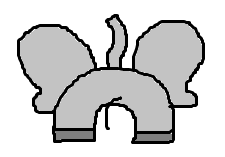




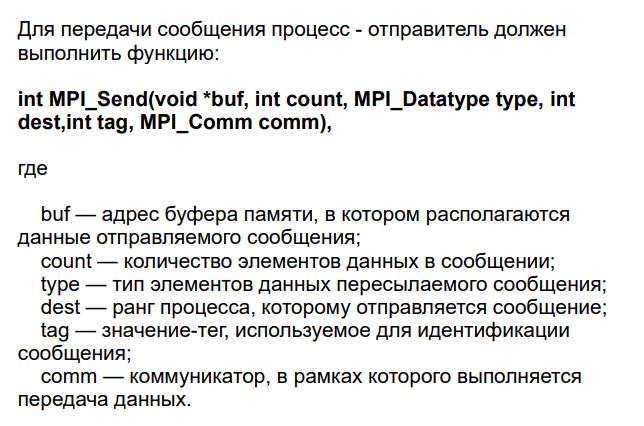


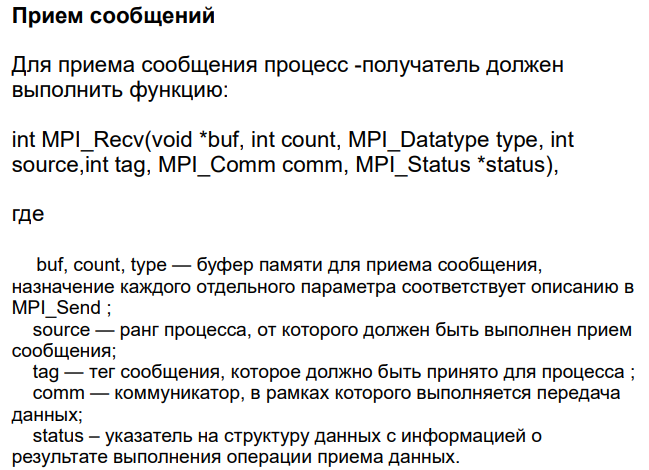




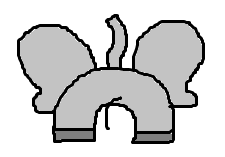
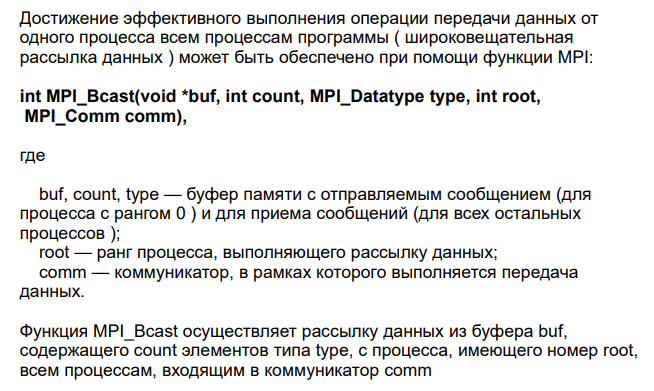


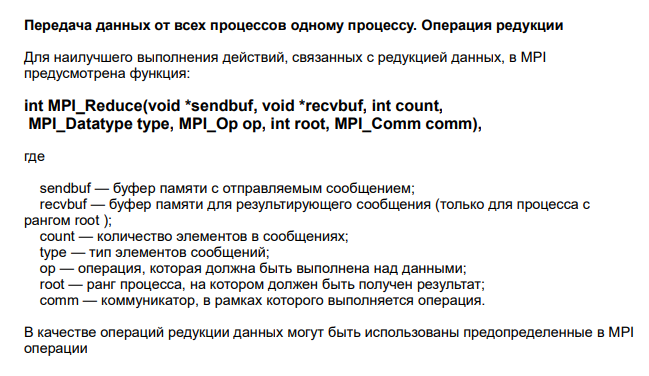
1. **Основные операции передачи и приема в MPI.**

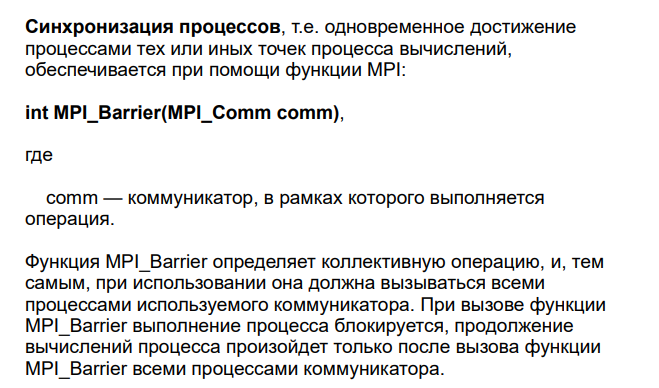


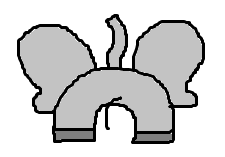


1. **Коллективные функции обмена в MPI.**

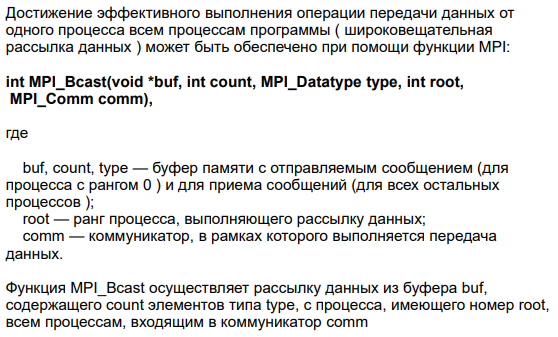




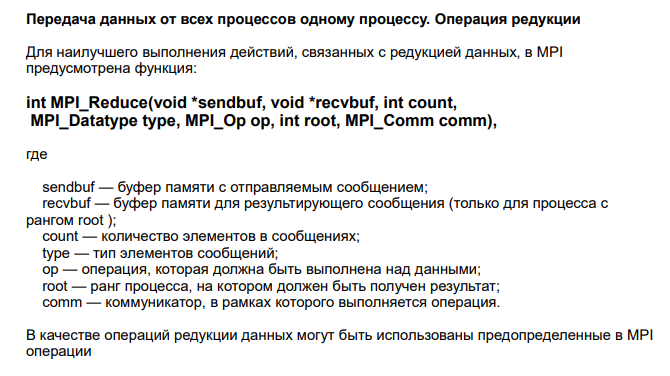


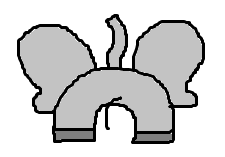


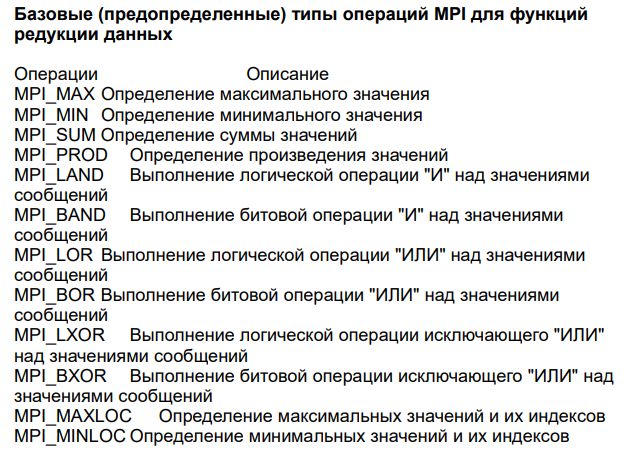
1. **Широковещательная рассылка данных в MPI.**



1. **Передача данных от всех процессов одному. Операция редукции.**







1. **Организация и работа с суперкомпьютерами.**

