

## Н5Р элементы

Н5Р элемент № 1 – лекция, в которой подпункты представлены в компонента «Аккордеон». Лекция № 2 «Исследование коллективного типа передачи данных, групп и коммуникаторов в MPI» делится на следующие подтемы, которые по умолчанию скрыты, но при нажатии открываются:

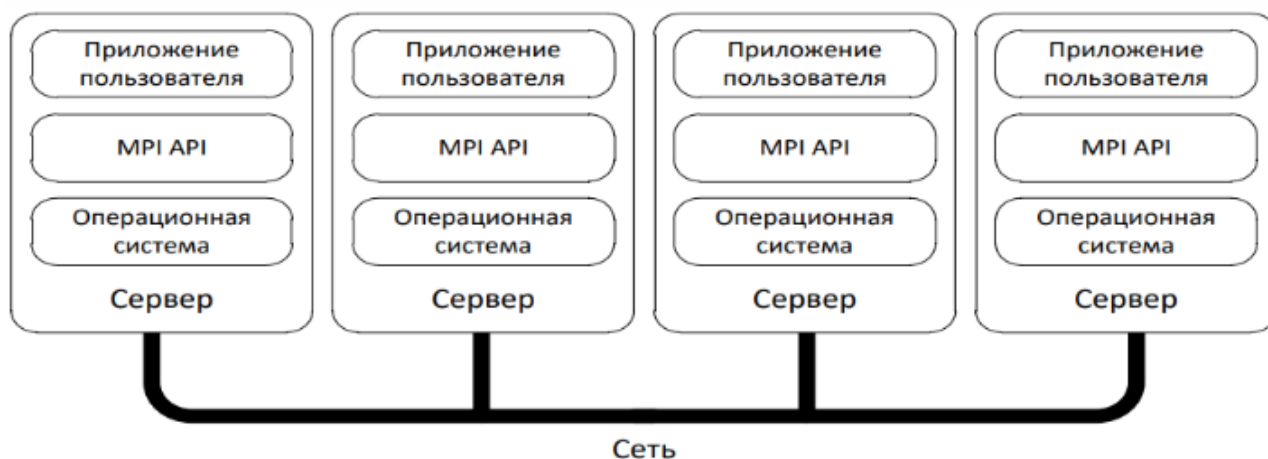
1. понятие вычислительной модели (модели вычислений) и понятие процесса;
2. ресурсы процессов;
3. синхронные и асинхронные процессы;
4. разделение программы на задачи для параллельного выполнения;
5. различие между процессами и потоками (при реализации модели с общей памятью);
6. взаимоотношение между синхронизируемыми задачами;

<b>▼ Ресурсы процессов</b>
Три типа ресурсов:
<ul style="list-style-type: none"> <li>◦ аппаратные ресурсы – физические устройства, которые могут совместно использоваться несколькими процессами. Аппарат процессам целиком).</li> <li>◦ информационные ресурсы – к ним относятся данные (объекты, переменные), системные данные (файлы), глобальные переменные.</li> <li>◦ программные ресурсы – общий набор процедур, которые могут использоваться различными процессами.</li> </ul>
<b>&gt; Синхронные и асинхронные процессы</b>
<b>&gt; Разделение программы на задачи для параллельного выполнения</b>

Рисунок 1 – Элемент «Аккордеон»

Н5Р элемент № 2 – лекция, представленная «Агамотто» элементом – весь контент расположен последовательно друг за другом сверху вниз.

MPI-программа состоит из независимых процессов, выполняющих свой собственный код, т.е. реализует MIMD (Multiple Instruction Multiple Data) подход к организации параллельных программ. При реализации MPI-программ операционная система для каждого процесса размещает копию выполняемой программы в оперативной памяти отдельной компоненты (хоста, рабочей станции в сети) вычислительной системы. Таким образом, независимо от конфигурации системы (Рис. 1.2–1. 4) код программы располагается на всех процессах. Для запуска требуемого количества копий программы необходимо в исполняемой среде указать необходимое количество процессов (запускаемых копий программы). Каждый процесс начинает выполнять свою собственную копию кода. Различные процессы могут выполнять различные участки кода посредством ветвления внутри программы, т.е. коды, выполняемые процессом, не обязательно должны быть идентичными (как правило, они являются различными). Для определения участков кода в программе, которые должны выполнять процессы (участка кода, который должен выполнять каждый процесс) используются ранги процессов, но могут быть созданы и более сложные структуры распределения заданий для каждого процесса.



◀ Лекция 3 - Технология

Рисунок 2 – Элемент «Агамотто»

### SCORM элементы

Добавим два упражнения по алгоритмам распределенных систем. Первое упражнение: необходимо составить алгоритм операции «Сравнить и переставить» построчно. Строки алгоритма приведены заранее и необходимо только переставить их местами. Реализация упражнения приведена на рисунке 3



Рисунок 3 – Упражнение на сортировку

Второе упражнение: необходимо составить правильное соответствие между функциями MPI и их назначениями. Для этого нужно перетянуть соответствующие элементы друг на друга:

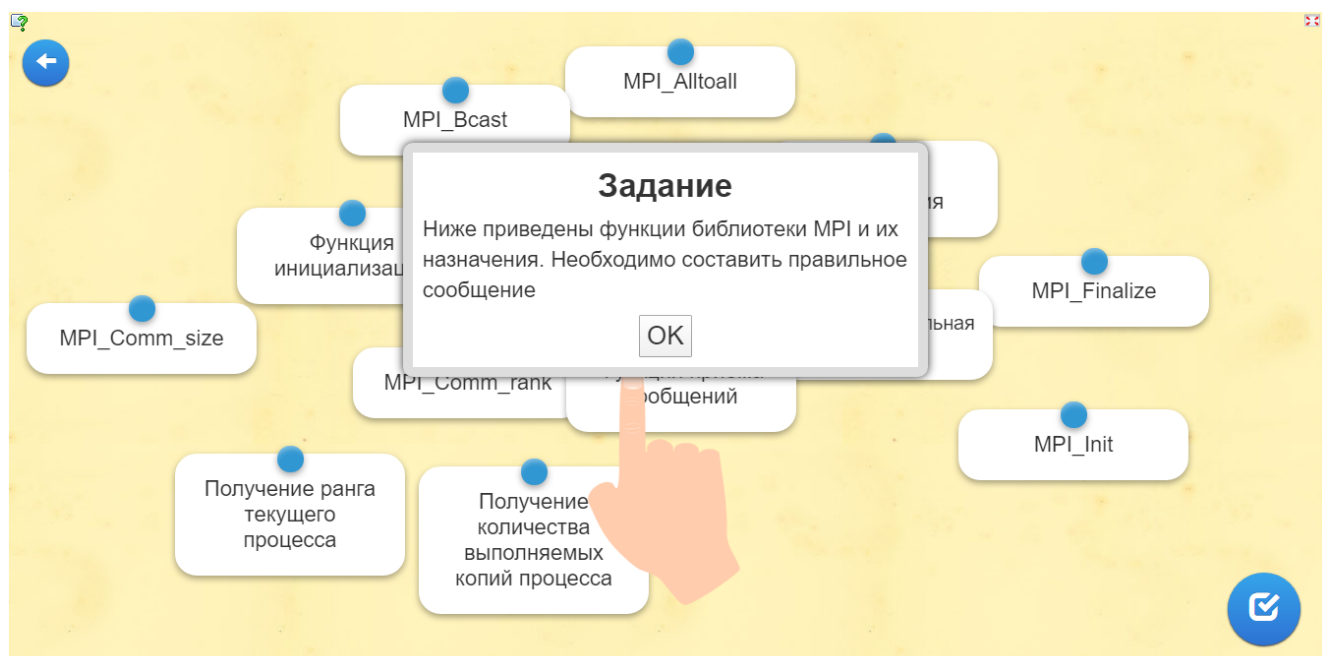


Рисунок 4 – Условие упражнения на соответствие

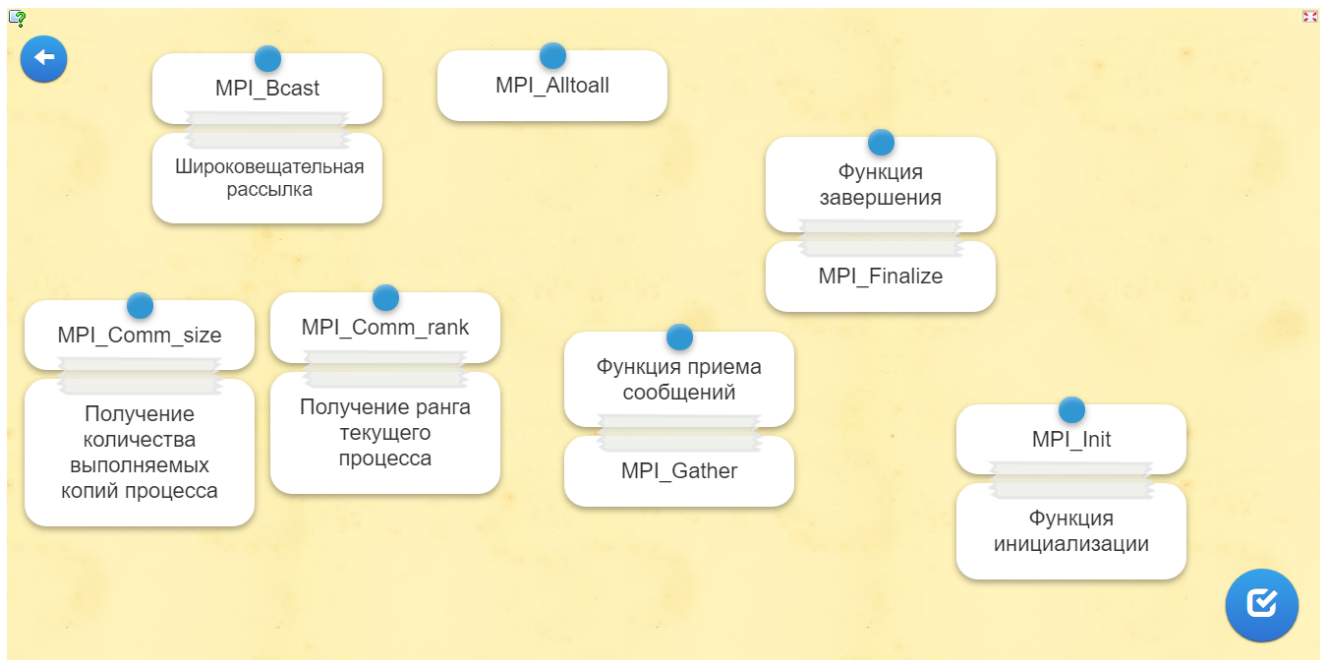


Рисунок 5 – Ход выполнения упражнения