

Задание:

Написать параллельную функцию, выполняющую вычисление скалярного произведения двух векторов.

Условия выполнения:

Для выполнения задания необходимо скачать проект Example1 и реализовать параллельный алгоритм вычисления скалярного произведения в рамках функции `double scalmult (int n, double* a, double* b)`. В качестве имени функции на каждом процессе должно возвращаться вычисленное значение скалярного произведения.

К проекту прилагается dll-библиотека с процедурой Submit, в которой будет произведена корректность работы вашей функции. Для проверки задания раскомментируйте вызов функции Submit() в файле mail.cpp и запустите программу на выполнение. После завершения программы будет создан файл submit.bin, который необходимо загрузить на сайт Coursera для проверки задания.

В рамках проекта уже проведена инициализация MPI. Для каждого процесса определен размер подзадачи n1. Также в соответствии с размером подзадачи на каждом процессе выделена память под массивы соответствующей длины. Сами массивы на каждом процессе заполняются случайными числами.

В рамках функции `double scalmult (int n, double* a, double* b)` нужно выполнить операцию вычисления скалярного произведения на каждом процессе для заданной ему подобласти данных, затем воспользоваться функцией глобальной редукции, чтобы собрать полученные результаты. После этого необходимо получить результат вычисления скалярного произведения и на каждом процессе вернуть результат в качестве имени функции scalmult.

Проект настроен на платформу **x64** и выполнение в режиме компиляции **release**.

Задание считается выполненным правильно при выполнении следующих условий:

1. При запуске программы на выполнение более чем на одном процессе результат работы пользовательской функции совпадает с результатом работы эталонной функции в dll-библиотеке.
2. Эффективность выполнения пользовательской параллельной функции более 0,6.

Замечание: Эффективность - это отношение ускорения работы параллельной программы к количеству задействованных процессов. Максимальную эффективность можно получить при использовании двух или трех процессов.