**Про 2 варианта параллельной программы**

При выполнении лабораторной №1 необходимо написать 3 программы: одну последовательную и 2 параллельных. Последовательную программу вы пишите без MPI, используя тот функционал языка C, который хорошо знаете с 1 семестра 1 курса. Параллельные программы, согласно задания, должны различаться тем, что в одном случае вектором *b* обладает каждый процесс, а во втором случае векторы *b* и *x* разрезаны между всеми процессами.

Естественен вопрос: чем принципиально отличается первый параллельный вариант от второго? Ведь во втором варианте вначале работы программы мы можем собрать весь вектор *b* со всех процессов и раздать полученный целый вектор всем процессам (кстати – как это сделать одной MPI-функцией?). А потом запустить тот же самый итерационный механизм, который запускали в первом варианте. Но такая схема работы второй параллельной программы меня не интересует.

Есть несколько способов решения задачи при разрезании векторов *b* и *x* между процессами. Здесь я предложу только один. Предлагаю разрезать матрицу *A* не по строкам, а по столбцам. В каждом варианте нужно вычислять разность (*Axn – b).* Если приглядитесь – в методе сопряженных градиентов эта разность с противоположным знаком. Давайте разберем как можно ее посчитать, если матрица *A* разрезана по столбцам, а векторы *x* и *b* распределены между процессами. При этом процессу с номером *i* принадлежат следующие столбцы матрицы и элементы векторов *x* и *b*: *i, n+i, 2\*n+i*…, где *n* – число процессов и их нумерация идет с единицы (можете реализовать вариант, когда каждому процессу принадлежат элементы и столбцы с *n\*i* по *n\*(i+1)*).

Предположим, что у нас порядок матрицы 6 и 3 процесса. Умножение p=*Ax* (для простоты не буду писать нижний индекс) можно выполнить так:

Процессу №1 (еще раз повторю – предположим, что нумерация процессов идет с единицы) принадлежат столбцы 1 и 4 и соответствующие элементы вектора *x*. Он производит умножение своих элементов *x* на свои столбцы *A*, после чего складывает получившиеся столбцы. Аналогичным образом поступает каждый процесс.

После этого у каждого процесса получается по одному столбцу. Для получения итогового произведения *p* эти столбцы необходимо сложить. Тут необходимо использовать функционал MPI.

Затем необходимо вычислить разницу полученного вектора *p* и вектора *b*. Но вектор *b* распределен аналогично вектору *x*. Значит нужно раздать полученный вектор *p* по процессам и вычесть соответствующие компоненты *b*. То есть наш процесс №1 в итоге посчитает:

*p1-b1*

*p4-b4*

Как совершить другие арифметические операции – догадаетесь сами ☺