

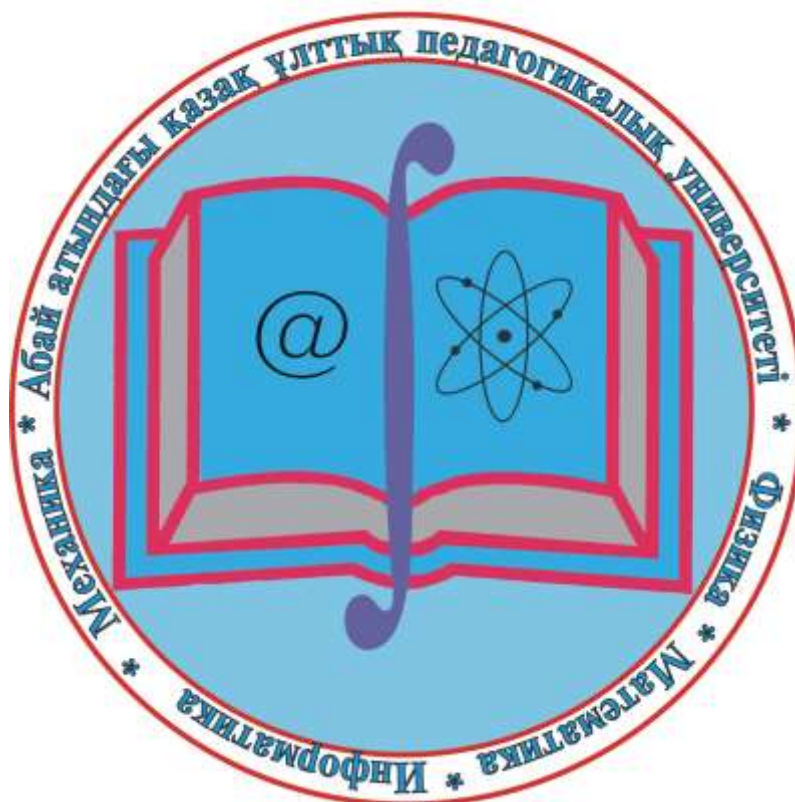


Абай атындағы
Қазақ ұлттық педагогикалық университеті

Казахский национальный педагогический
университет имени Абая

Серия «Физико-математические науки» • «Физика-математика ғылымдары» сериясы

ХАБАРШЫ ВЕСТНИК



Алматы

№ 1 (53)

2016

Казахский национальный педагогический университет имени Абая ВЕСТНИК серия “Физико-математические науки” № 1 (53) Главный редактор <i>Академик НАН РК Г.У. Уалиев</i> Редакционная коллегия: зам.главного редактора: <i>д.п.н. Е.Ы. Бидайбеков,</i> <i>к.ф.-м.н. М.Ж. Бекпатшаев</i> ответ.секретарь <i>к.п.н. Г.А. Абдулкаримова</i> члены: <i>Dr.-ing. Holm Altenbach(Germany),</i> <i>Dr. S.A.Hasan (Pakistan),</i> <i>Dr. YasuhideFukumoto(Japan),</i> <i>Phd.d Shuo-Hung Chang, (Taiwan),</i> <i>д.п.н., академик РАО А.Е. Абылкасымова,</i> <i>д.ф.-м.н. М.А. Бектемесов,</i> <i>д.ф.-м.н. А.С.Бердышев,</i> <i>д.п.н. В.В. Гриншкун (Россия),</i> <i>к.ф.-м.н. Ф.Р. Гусманова,</i> <i>д.т.н. А.Д.Джураев(Узбекистан),</i> <i>д.ф.-м.н., чл.-кор. РАН С.И. Кабанихин (Россия),</i> <i>д.ф.-м.н. Б.А. Кожамкулов,</i> <i>д.ф.-м.н., чл.-кор. НАН РК В.Н. Косов,</i> <i>д. ф.-м.н. К.К. Коксалов,</i> <i>д.т.н. М.К. Кулбеков,</i> <i>д.п.н., академик РАО М.П. Лапчик (Россия),</i> <i>д.ф.-м.н. Қ.М. Мукашев,</i> <i>д.ф.-м.н. С.Т. Мұхамбетжанов,</i> <i>д.т.н. Г.Я. Пановко (Россия),</i> <i>д.п.н. Б.Д. Сыдыков,</i> <i>д.ф.-м.н., академик НАН РК Н.Ж. Такибаев,</i> <i>д.ф.-м.н. К.Б. Тлебаев,</i> <i>д.т.н. А.К. Тулешов,</i> <i>д.ф.-м.н. З.Г. Уалиев,</i> <i>д.ф.-м.н., чл.-кор. НАН РК Л.М. Чечин,</i> <i>к.ф.-м.н. Е.Б. Шалбаев,</i> <i>к.т.н.Ш.И. Хамраев</i>	©Казахский национальный педагогический университет им. Абая, 2016 Зарегистрирован в Министерстве информации Республики Казахстан, № 4824 - Ж - 15.03.2004 (периодичность—4 номера в год) Выходит с 2000 года Редакторы:Ф.Р. Гусманова, Г.А. Абдулкаримова Компьютерная верстка: Г.А. Абдулкаримова Ф.Р. Гусманова Подписано в печать 18.03.2016 г. Формат 60х84 1/8. Об 9,95 уч.-изд.л. Тираж 300 экз. 050010, г.Алматы, пр.Достык, 13, КазНПУ им.Абая <i>Отпечатано в типографии “ТОО Palitra Press”</i> <i>г.Алматы, ул.Хамиди 4а</i>
	Д.А. Кинжебаева, А.С. Сарсекеева Анализ движения системы «электрический двигатель – механизм IV класса с выстом ведомых звеньев» при разгоне 117 В.Н. Косов, С.А. Красиков, О.В. Федоренко, Г. Акылбекова Создание модуля опытного устройства для разделения природных газов 123 М. Көшербай, Д.А. Кинжебаева «КОМПАС - 3D V15» бағдарламасы арқылы шахмат фигураны жасау технологиясы 128 Д.Р. Сатывалдиев, Ж. Сағындыков, К.М. Мукашев Новые образовательные технологии при изучении физических и химических понятий с применением анимационно - электронных учебников 135 Г. Уалиев, З.Г. Уалиев Динамический анализ механизмов независимого движения 139 ИНФОРМАТИКА. ИНФОРМАТИКАНЫ ОҚЫТУ ӘДІСТЕМЕСІ. БІЛІМ БЕРУДІ АҚПАРАТТАНДЫРУ ИНФОРМАТИКА, МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ. ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ О.С. Ахметова, С.А. Исаев, М.Т. Шуйнбаева Организация проектно-исследовательской деятельности при изучении информатики в младших классах 145 К.В. Баскаков, А.Ю. Пыркова Технология фрагментированного программирования 150 И.У. Бекбулатова, С.А. Бахтыбаева, Г.П. Мейрбекова, Г.Ж. Ниязова Бұлттық технологиялардың білім берудегі мүмкіндіктері 158 D.A. Berdysheva Automated analysis in education 163 К.М. Беркимбаев, И.У. Бекбулатова, Г.Ж. Ниязова, Б.Т. Керимбаева Болашақ мұғалімнің коммуникативтік әлеуетін қалыптастыру мәселелері 168 М.О. Беркинбаев, Ш.И. Байметова, Ә.Х. Сарыбаева, С.А. Бактыбаева Электрондық оқыту жүйесі құралдарын қолдану әлеуеті 174 К.С. Бертаева, С.А. Исаев, О.С. Ахметова Влияние повышения уровня ИКТ-компетентности учителя на развитие функциональной грамотности учащихся 180 Ф.Р. Гусманова, Г.А. Абдулкаримова Ойындар теориясының концепциясының негізінде оңтайлы шешімдерді қабылдаудағы анықталмағандықтың қауіп-қатермен өзара байланысы 185 М.Н. Қалимолдаев, А.А. Абдилдаева, М.А. Ахметжанов, М.Ж. Сақыпбекова Электр энергетикалық жүйесіне ақпараттық жүйе құру 190 С.Н. Конева Особенности применения Intranet-технологий в условиях кредитной системы обучения 196 Р.Р. Мусабасев, П.Б. Сейсенбекова Қазақ тілін синтездеу үрдісіне интонациялық модельдердің қолданылуы 203 Л.Б. Рахимжанова, Д.Н. Исабаева Ролевая игра при обучении алгоритму электронной цифровой подписи RSA 208 Н.С. Уалиев, А.А. Сакабаев Обеспечение защиты информации при интеграции информационно-образовательных сред в мобильные Web-приложения 213 К.З. Халықова Болашақ информатикфа мамандарын даярлау процесінде студенттердің зерттеу іс-әрекетін ұйымдастыру 219 К.З. Халықова, Р.А. Ильясова, М. Булатаева Action Script 3 объектіге бағытталған программалау тілін оқытудың ерекшеліктері 225 Ш.Т. Шекербекова, Ұ.Қ. Амантаева Электрондық оқу құралын жасау технологиялары мен әдістері 231

**ИНФОРМАТИКА. ИНФОРМАТИКАНЫ ОҚЫТУ ӘДІСТЕМЕСІ.
БІЛІМ БЕРУДІ АҚПАРАТТАНДЫРУ
ИНФОРМАТИКА, МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ
ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ**

начального образования: материалы Всероссийской заочной научной конференции (Барнаул, 18 февраля 2010): в 2 ч. / под ред. Г. Ф. Свиридовой, Е. Н. Ставской. Барнаул: АлтГПА, 2010. Ч. 1. С. 37–39.

Аңдатпа. Мақалада бастауыш мектепте информатиканы оқытуда жобалап-зерттеу әрекетін қолданудың мүмкіндіктері қарастырылған. Жобалап-зерттеу әрекетінің анықтамасы беріледі. Бастауыш мектеп оқушыларын жобалап-зерттеу әрекетіне нәтижелі дайындауға қойылатын негізгі талаптар анықталған. 2-сыныпта «Ақпарат» тақырыбын оқытуда жоба жұмысынан мысал келтірілген. Жобамен жұмыс барысында оқушылардың бағалануы тиіс іскерлігі анықталған.

Түйін сөздер: оқыту, жобалап-зерттеу әрекеті, информатика, бастауыш мектеп, оқу жетістіктерін бағалау.

Abstract. The possibilities of using project and research activities in the study of Computer Science in primary school are considered in this article. And in this article we defined the term “design and research activity”. As well as were identified the basic requirements for readiness of primary school students to the productive project and training activities. We showed an example of a project for the students grade 2 when they studied the theme "Information". We defined the skills of students which to be evaluated when they work at the project.

Keywords: training, project and research activity, Computer Science, primary school, assessment of educational achievements

УДК 004.05

К.В. Баскаков*, А.Ю. Пыркова

ТЕХНОЛОГИЯ ФРАГМЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

(г. Алматы Казахский национальный университет имени аль-Фараби, *-магистрант)

Аннотация. Для моделирования с высокой точностью необходимы огромные вычислительные ресурсы, поэтому для большинства задач разрабатываются параллельные программы, позволяющие выполнять моделирование на суперкомпьютерах. С повсеместным распространением многоядерных процессоров задача разработки параллельной программы ещё более усложняется, поскольку эффективно распределить доступные ресурсы между крупными взаимодействующими последовательными процессами затруднительно, поскольку программа должна обладать рядом динамических свойств, которые существующими системами программирования в полной мере не обеспечиваются. Одним из новых подходов, альтернативных программированию на основе параллельно-последовательной модели вычислений, является фрагментированное программирование.

В данной статье производится сравнительный анализ параллельного программирования и фрагментированного программирования на примере умножения матриц.

Ключевые слова: фрагментированное программирование, параллельное программирование, умножение матриц, *LiNA*, *MPI*.

Введение. Роль численного моделирования в решении научных и прикладных задач продолжает возрастать [1]. Для моделирования с высокой точностью необходимы огромные вычислительные ресурсы, поэтому для большинства задач разрабатываются параллельные программы, позволяющие выполнять моделирование на

суперкомпьютерах.

Разработка хорошей параллельной программы в настоящее время остаётся сложной задачей. Параллельная программа должна обладать рядом динамических свойств (динамической настройкой на доступные ресурсы, динамической балансировкой загрузки, динамическим учётом поведения численной модели), которые существующими системами программирования в полной мере не обеспечиваются, а их ручная реализация требует серьёзных знаний теории параллельного программирования, архитектуры ЭВМ, программных особенностей средств разработки. В параллельных программах встречаются специфические ошибки, которые сложно обнаружить и устранить в силу не детерминизма исполнения таких программ.

С повсеместным распространением многоядерных процессоров задача разработки параллельной программы ещё более усложняется, поскольку эффективно распределить доступные ресурсы между крупными взаимодействующими последовательными процессами затруднительно. Это породило множество исследовательских проектов, направленных на поиск более подходящей модели вычислений [1].

Одним из новых подходов, альтернативных программированию на основе параллельно-последовательной модели вычислений, является фрагментированное программирование [2]. К его основным преимуществам можно отнести возможность автоматической генерации параллельной программы, возможность автоматически обеспечить программам ряд динамических свойств, лучшую переносимость между вычислителями различной архитектуры, а также возможность выполнить формальный анализ параллельной программы.

Разработка фрагментированного программирования и реализующие ее язык и система программирования LuNA [2] — разработаны в ИВМиМГ СО РАН.

Технология параллельного программирования MPI. Наиболее распространенной технологией программирования для параллельных компьютеров с распределенной памятью в настоящее время является MPI. Основным способом взаимодействия параллельных процессов в таких системах является передача сообщений друг другу. Это и отражено в названии данной технологии - Message Passing Interface (интерфейс передачи сообщений). Стандарт MPI фиксирует интерфейс, который должен соблюдаться как системой программирования на каждой вычислительной платформе, так и пользователем при создании своих программ.

Под параллельной программой в рамках MPI понимается множество одновременно выполняемых процессов. Все процессы порождаются один раз, образуя параллельную часть программы. Каждый процесс работает в своем адресном пространстве, никаких общих переменных или данных в MPI нет. Процессы могут выполняться на разных процессорах, но на одном процессоре могут располагаться и несколько процессов (в этом случае их исполнение осуществляется в режиме разделения времени). В предельном случае для выполнения параллельной программы может использоваться один процессор — как правило, такой способ применяется для начальной проверки правильности параллельной программы.

Интерфейс MPI [3, 4] поддерживает создание параллельных программ в стиле MIMD (Multiple Instruction Multiple Data), что подразумевает объединение процессов с различными исходными текстами. Однако писать и отлаживать такие программы очень сложно, поэтому на практике программисты гораздо чаще используют SPMD-модель (Single Program Multiple Data) параллельного программирования, в рамках которой для всех параллельных процессов используется один и тот же код. В настоящее время все больше и больше реализаций MPI поддерживают работу с нитями

Каждый процесс параллельной программы порождается на основе копии одного и того же программного кода (модель SPMP - Single Program Multiple Processes). Данный

**ИНФОРМАТИКА. ИНФОРМАТИКАНЫ ОҚЫТУ ӘДІСТЕМЕСІ.
БІЛІМ БЕРУДІ АҚПАРАТТАНДЫРУ
ИНФОРМАТИКА, МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ
ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ**

программный код, представленный в виде исполняемой программы, должен быть доступен в момент запуска параллельной программы на всех используемых процессорах. Исходный программный код для исполняемой программы разрабатывается на алгоритмических языках C или Fortran с использованием той или иной реализации библиотеки MPI.

Количество процессов и число используемых процессоров определяется в момент запуска параллельной программы средствами среды исполнения MPI-программ и в ходе вычислений меняться не может (в стандарте MPI-2 предусматривается возможность динамического изменения количества процессов). Все процессы программы последовательно перенумерованы от 0 до $p-1$, где p есть общее количество процессов. Номер процесса именуется рангом процесса.

При этом для того, чтобы избежать идентичности вычислений на разных процессорах, можно, во-первых, подставлять разные данные для программы на разных процессорах, а во-вторых, с помощью средств для идентификации процесса, на котором выполняется программа и тем самым, предоставляется возможность организовать различия в вычислениях в зависимости от используемого программой процессора.

Это позволяет загружать ту или иную подзадачу в зависимости от “номера” процессора. При этом исходная задача разбивается на подзадачи (декомпозиция). Обычная техника состоит в следующем: каждая из подзадач оформляется в виде отдельной структурной единицы (функции, модуля), на всех процессорах запускается одна и та же программа “загрузчик”, которая, в зависимости от “номера” процессора загружает ту или иную подзадачу.

Шаги разработки фрагментированной программы. Фрагментированная программа разрабатывается в несколько последовательных этапов[2, 5].

Разработка исходного алгоритма решения задачи.

- Фрагментация исходного алгоритма. Фрагментация рассматривается здесь как универсальный способ распараллеливания алгоритмов, что позволяет поддержать его реализацию системой программирования. Фрагментация нередко является очень сложной работой.

- Описание фрагментированной программы на языке LuNA. Входной язык системы LuNA устроен просто, в него в основном включены средства для определения множества фрагментов данных и вычислений, синтаксис языка не очень интересен.

- Компиляция и анализ фрагментированного алгоритма.

- Исполнение фрагментированной программы

Пример фрагментированного алгоритма. В качестве примера возьмем простейший алгоритм перемножения двух квадратных матриц. Рассматриваться будут технология параллельного вычисления MPI и фрагментированное программирование. Умножение матрицы A размера $m \times n$ и матрицы B размера $n \times l$ приводит к получению матрицы C размера $m \times l$, каждый элемент которой определяется в соответствии с выражением:

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} \cdot b_{kj}, \quad 0 \leq i \leq m, \quad 0 \leq j \leq l \quad (1)$$

Каждый элемент результирующей матрицы C есть скалярное произведение соответствующих строки матрицы A и столбца матрицы B :

$$c_{ij} = (a_i b_j^T), \quad a_i = (a_{i0}, a_{i1}, \dots, a_{i,n-1}), \quad b_j^T = (b_{0j}, b_{1j}, \dots, b_{n-1,j}) \quad (2)$$

Этот алгоритм предполагает выполнение $m \times n \times l$ операций умножения и столько же операций сложения элементов исходных матриц. При умножении квадратных матриц размера $n \times n$ количество выполненных операций имеет порядок $O(n^3)$.

Последовательный алгоритм умножения двух квадратных матриц

```
// Последовательный алгоритм умножения матриц
double MatrixA[Size][Size];
double MatrixB[Size][Size];
double MatrixC[Size][Size];
int i,j,k;
...
for (i=0; i<Size; i++){
  for (j=0; j<Size; j++){
    MatrixC[i][j] = 0;
    for (k=0; k<Size; k++){
      MatrixC[i][j] = MatrixC[i][j] + MatrixA[i][k]*MatrixB[k][j];
    }
  }
}
```

Этот алгоритм является итеративным и ориентирован на последовательное вычисление строк матрицы C. Действительно, при выполнении одной итерации внешнего цикла (цикла по переменной i) вычисляется одна строка результирующей матрицы.

На первой итерации цикла по переменной i используется первая строка матрицы A и все столбцы матрицы B для того, чтобы вычислить элементы первой строки результирующей матрицы C (Рис 1).

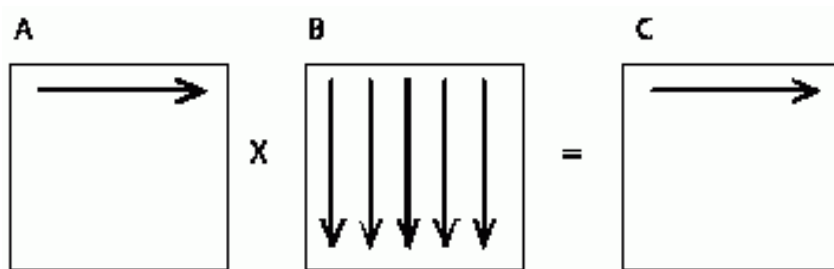


Рис 1. Вычисление элементов первой строки матрицы C.

Поскольку каждый элемент результирующей матрицы есть скалярное произведение строки и столбца исходных матриц, то для вычисления всех элементов матрицы C размером $n \times n$ необходимо выполнить $n^2 \times (2n-1)$ скалярных операций и затратить время

$$T_1 = n^2 \cdot (2n - 1) \quad (3)$$

Из определения операции матричного умножения следует, что вычисление всех элементов матрицы C может быть выполнено независимо друг от друга. Как результат, возможный подход для организации параллельных вычислений состоит в использовании в качестве базовой подзадачи процедуры определения одного элемента результирующей матрицы C. Для проведения всех необходимых вычислений каждая подзадача должна содержать по одной строке матрицы A и одному столбцу матрицы B. Общее количество получаемых при таком подходе подзадач оказывается равным n^2 (по числу элементов матрицы C).

Рассмотрев предложенный подход, можно отметить, что достигнутый уровень параллелизма является в большинстве случаев избыточным [1]. Обычно при проведении практических расчетов такое количество сформированных подзадач превышает число имеющихся процессоров и делает неизбежным этап укрупнения базовых задач. В этом

**ИНФОРМАТИКА. ИНФОРМАТИКАНЫ ОҚЫТУ ӘДІСТЕМЕСІ.
БІЛІМ БЕРУДІ АҚПАРАТТАНДЫРУ
ИНФОРМАТИКА, МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ
ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ**

плане может оказаться полезной агрегация вычислений уже на шаге выделения базовых подзадач. Возможное решение может состоять в объединении в рамках одной подзадачи всех вычислений, связанных не с одним, а с несколькими элементами результирующей матрицы C . Для дальнейшего рассмотрения определим базовую задачу как процедуру вычисления всех элементов одной из строк матрицы C . Такой подход приводит к снижению общего количества подзадач до величины n .

Для выполнения всех необходимых вычислений базовой подзадаче должны быть доступны одна из строк матрицы A и все столбцы матрицы B . Простое решение этой проблемы – дублирование матрицы B во всех подзадачах – является, как правило, неприемлемым в силу больших затрат памяти для хранения данных. Поэтому организация вычислений должна быть построена таким образом, чтобы в каждый текущий момент времени подзадачи содержали лишь часть данных, необходимых для проведения расчетов, а доступ к остальной части данных обеспечивался бы при помощи передачи данных между процессорами.

Алгоритм представляет собой итерационную процедуру, количество итераций которой совпадает с числом подзадач. На каждой итерации алгоритма каждая подзадача содержит по одной строке матрицы A и одному столбцу матрицы B . При выполнении итерации проводится скалярное умножение содержащихся в подзадачах строк и столбцов, что приводит к получению соответствующих элементов результирующей матрицы C . По завершении вычислений в конце каждой итерации столбцы матрицы B должны быть переданы между подзадачами с тем, чтобы в каждой подзадаче оказались новые столбцы матрицы B и могли быть вычислены новые элементы матрицы C . При этом данная передача столбцов между подзадачами должна быть организована таким образом, чтобы после завершения итераций алгоритма в каждой подзадаче последовательно оказались все столбцы матрицы B .

Возможная простая схема организации необходимой последовательности передач столбцов матрицы B между подзадачами состоит в представлении топологии информационных связей подзадач в виде кольцевой структуры. В этом случае на каждой итерации подзадача i , $0 \leq i < n$, будет передавать свой столбец матрицы B подзадаче с номером $i+1$ (в соответствии с кольцевой структурой подзадач $n-1$ передает свои данные подзадаче с номером 0). После выполнения всех итераций алгоритма необходимое условие будет обеспечено – в каждой подзадаче поочередно окажутся все столбцы матрицы B .

Ниже представлены итерации алгоритма матричного умножения (Рис. 2) для случая, когда матрицы состоят из четырех строк и четырех столбцов ($n=4$). В начале вычислений в каждой подзадаче i , $0 \leq i < n$, располагаются i -я строка матрицы A и i -й столбец матрицы B . В результате их перемножения подзадача получает элемент c_{ii} результирующей матрицы C . Далее подзадачи осуществляют обмен столбцами, в ходе которого каждая подзадача передает свой столбец матрицы B следующей подзадаче в соответствии с кольцевой структурой информационных взаимодействий. Далее выполнение описанных действий повторяется до завершения всех итераций параллельного алгоритма.

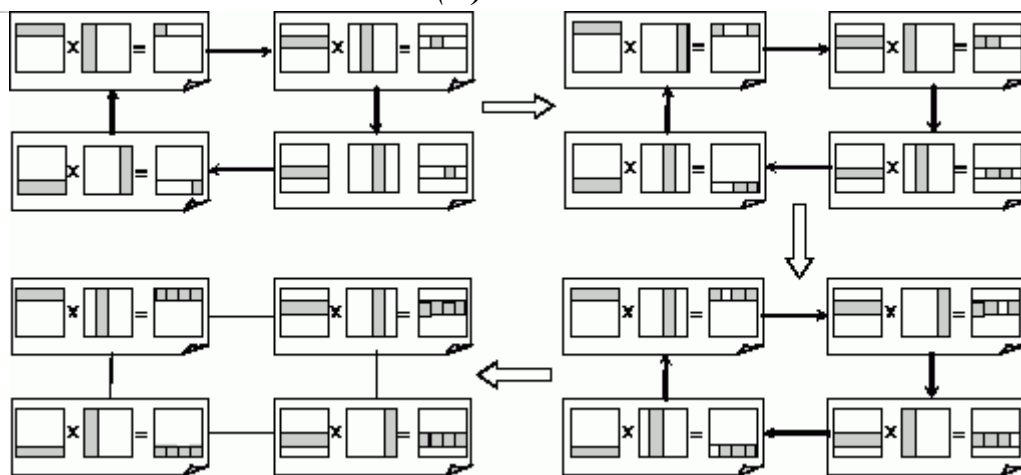


Рис 2. Итерации алгоритма матричного умножения

Исходный алгоритм умножения квадратных матриц (Рис. 3). Вычисления проводятся по формулам:

$$c_{i,j} = \sum_{k=1}^N a_{i,k} \times b_{k,j}, \quad i, j = 1, 2, \dots, N.$$

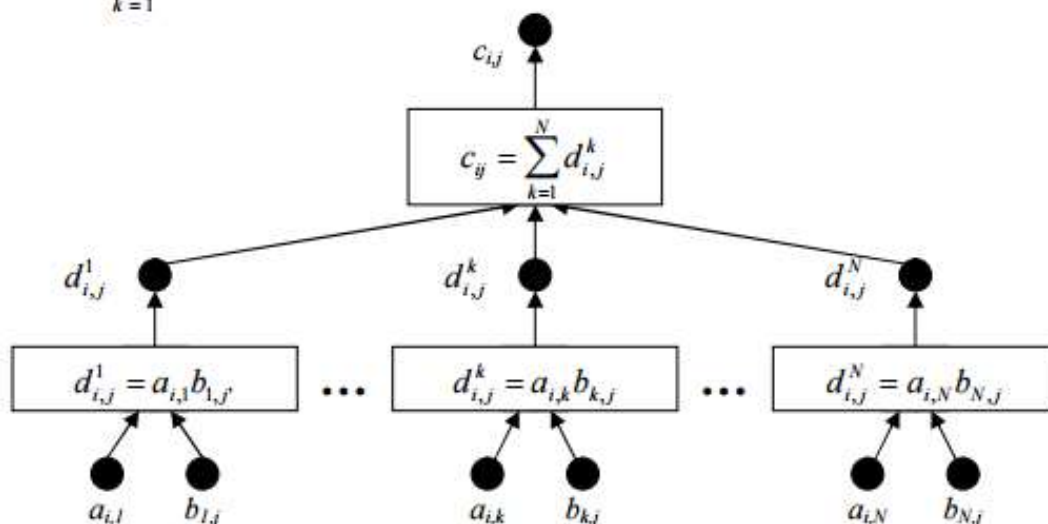


Рис 3. Исходный алгоритм умножения квадратных матриц

Так как множество функциональных термов хранить нехорошо, то в языке LuNA вместо множества функциональных термов определяются множества фрагментов вычислений и фрагментов данных, а нужные термы конструируются из них по мере необходимости алгоритмом вывода

Фрагментированный алгоритм умножения квадратных матриц.

В системе LuNA исходный алгоритм может быть запрограммирован как фрагментированный, т.е. каждая операция $c_{ij} = \sum_{k=1}^N d_{ij}^k$ объявлена фрагментом вычислений (и будет реализоваться как независимый процесс программы), переменные a_{i1} , b_{1j} , c_{ij} , d_{ij}^k объявлены фрагментами данных [2, 5]. Но такая фрагментированная программа будет исполняться с большим замедлением, примерно с 1000 кратным, по сравнению с программой с использованием MPI из-за больших расходов на реализацию

**ИНФОРМАТИКА. ИНФОРМАТИКАНЫ ОҚЫТУ ӘДІСТЕМЕСІ.
БІЛІМ БЕРУДІ АҚПАРАТТАНДЫРУ
ИНФОРМАТИКА, МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ
ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ**

управления. Поэтому для численных алгоритмов, отличающихся высокой регулярностью, в процессе фрагментации проводится агрегация и переменных и операций. Для всех I и J выполнить сначала все, кроме K -го, фрагменты вычислений $D_{i,j}^L = A_{iL}B_{LJ}$ а потом исполнить все K -ые фрагменты, то память вычислителя должна будет хранить все выработанные, но своевременно не потребленные, промежуточные данные. Объем хранимых промежуточных данных в K раз больше чем может понадобиться при хорошей организации вычислений. В результате будет ограничен размер решаемой задачи и замедлится исполнение программы. Поэтому основными задачами, решаемыми runtime системой, являются распределение ресурсов и выбор наиболее подходящего очередного фрагмента вычислений на исполнение.

В качестве переменных фрагментированной программы используются агрегаты переменных исходного алгоритма, составляющих подматрицы исходной матрицы. Аналогично, код фрагмента вычислений является агрегатом кодов операций исходного алгоритма. В системе LuNA конструируются программы, в которых размер фрагментов данных является входным параметром. Ниже в качестве примера приведена часть фрагментированной программы умножения матриц, записанная в языке LuNA: Множество фрагментов данных:

```
df a[i,k] := block(4*M*M) | i=0..K-1, k=0..K-1;
df c[i,j] := block(4*M*M) | i=0..K-1, j=0..K-1;
df d[i,j,k] := block(4*M*M) | i=0..K-1, j=0..K-1, k=0..K-1;
df b[k,j] := block(4*M*M) | k=0..K-1, j=0..K-1;
Множество фрагментов вычислений
cf initc[i,j] := proc_zero (out: c[i,j])
| i=0..K-1, j=0..K-1;
cf mul[i,j,k] := proc_mmul (in: a[i,k],b[k,j];
out: d[i,j,k]) | i=0..K-1, j=0..K-1, k=0..K-1;
cf sum[i,j,k] := proc_add (in: d[i,j,k],c[i,j]; out: c[i,j])
| i=0..K-1, j=0..K-1, k=0..K-1;
```

Входной язык LuNA - теоретико-множественный, единственного присваивания и единственного исполнения фрагментов вычислений. Фрагменты данных и вычислений задаются рекурсивно перечислимыми множествами с использованием индексных выражений. Управление в LuNA-программе задается отношением частичного порядка на множестве фрагментов вычислений.

Отношение соседства на множестве фрагментов данных определяет, какие фрагменты данных должны размещаться в одном либо в соседних узлах. Дополнительно имеются операторы рекомендации по распределению ресурсов вычислителя, по определению требуемого порядка исполнения фрагментов вычислений. Средства задания приоритетов исполнения фрагментов вычислений используются runtime системой для выбора наиболее подходящего фрагмента на исполнение.

LuNA имеет три уровня преобразования фрагментированного алгоритма в программу: Компиляция, Генерация и Исполнение [1]:

- Компилятор принимает решения, которые можно принять, используя только информацию о свойствах алгоритма.

- Генератор принимает решения, которые зависят от свойств конкретного вычислителя на котором программа должна исполняться.

Runtime система принимает те решения, которые могут быть сделаны только динамически, в ходе вычислений. В их числе выбор фрагмента вычислений на

исполнение, динамическая балансировка нагрузки узлов вычислителя, динамическое распределение ресурсов, включая назначение процессора для исполнения фрагмента вычислений и многое другое.

Заключение. В системе LuNA есть еще один компонент – профилировщик, который собирает реальную информацию о ходе исполнения фрагментированной программы. Эта информация используется затем для улучшения последующих исполнений программы. При помощи фрагментированного программирования достигается высокое качество динамического распределения ресурсов и гибкая настройка параллельной программы на все доступные ресурсы. Фрагментация программы позволяет единообразно решить проблему организации межпроцессорных обменов данными на фоне вычислений.

1. Арыков С.Б. Язык и система фрагментированного параллельного программирования задач численного моделирования: диссертация кандидата физико-математических наук, 05.13.11. - Новосибирск, 2010. - 195с.
2. Малышкин В. Э. Технология фрагментированного программирования. // Вестник ЮУрГУ, серия "Вычислительная математика и информатика", 2012, № 46 (305). – С.45-55.
3. Сералин, Г.А. Технологии распараллеливания для решения сложных вычислительных задач. - Алматы, 2008. – 90с.
4. Антонов А.С. Введение в параллельные вычисления. // Методическое пособие. – М.: Изд-во Физфака МГУ, 2002. – 70с.
5. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. - М.: Изд-во МГУ, 2004. - 71с.

Аңдатпа. Жогары дәлдікпен пішімдеу үшін үлкен есептеу қорлары керек, сондықтан көптеген тапсырмалар үшін пішімдеуді суперкомпьютерлерде орындауы мүмкіндік беретін қосарлас бағдарламалар жетілдірілуде. Көпядролы процессорлардың көп таралуымен қосарлас бағдарламаларды жетілдіру одан әрі қиындайды, себебі өзара әсер ететін жүйелі үрдістер арасында қол жетімді ресурстарды тиімді бөлу қиындық туғызады. Себебі бағдарламалар қазір бар бағдарламалау жүйелерімен толық қамтамасыз етілмейтін динамикалық қасиеттермен сипатталуы керек. Фрагменттелген бағдарламалау есептеудің қосарлас-жүйелі пішімдеу негізіндегі бағдарламалаудың баламасы бола алатын жаңа көзқарастардың бірі болып табылады.

Бұл мақалада матрицаларды көбейту мысалында қосарлас бағдарламалау мен фрагменттелген бағдарламалауды салыстырмалы талдау жүргізілген.

Түйін сөздер: фрагменттелген бағдарламалау, қосарлас бағдарламалау және матрицаларды көбейту, LuNA, MPI.

Abstract. To simulate with high accuracy are required enormous computing power, so for the majority of tasks are developed parallel programs that let you perform simulations on supercomputers. With the ubiquitous multi-core processors the task of developing a parallel program even more complicated, because effectively allocate available resources between large communicating sequential processes is difficult because the program must have a range of dynamic properties that existing programming systems are not fully supported. One of the new approaches, alternative programming, based on a series-parallel computing model is fragmented programming.

This article performs a comparative analysis of parallel programming and fragmented programming example of matrix multiplication.

Keywords: fragmented programming, parallel programming, matrix multiplication, LuNa, MPI.

ҚҰРМЕТТІ ОҚЫРМАНДАР!

ҚР БЖҒМ Білім және ғылым саласындағы бақылау комитетінің 2012 жылы 10 шілдедегі № 1082 Шешімімен Қазақстан Республикасының Білім және Ғылым министрлігінің Білім және ғылым саласындағы бақылау комитетінің ұсынысымен Абай атындағы Қазақ ұлттық педагогикалық университетінің «Физика-математика ғылымдары» сериясы бойынша Хабаршы журналы ғылыми қызметтерінің негізгі нәтижелерін жариялау үшін келесі мамандықтар бойынша ұсынылатын басылымдар Тізіміне енгізілгенін хабарлаймыз:

- 01.01.00 топ мамандықтары бойынша – физика-математика ғылымдары (математика);
- 01.02.00 (01.02.01, 01.02.04, 01.02.06) мамандықтары бойынша – физика-математика ғылымдары (механика);
- 05.00.00 (05.02.18) мамандықтары бойынша – техникалық ғылымдар;
- 13.00.00 (13.00.02) мамандықтары бойынша – педагогика (оқыту және тәрбиелеу теориясы мен әдістемесі /математика, физика, информатика/) ғылымдары;
- 01.04.00 топ мамандықтары бойынша - физика-математика ғылымдары (физика).

2009 жылдан бастап Инженеринг және Технология Институтымен (Ұлыбритания) ақпараттық-қолдау қызмет көрсетуге жасалған келісім-шарттың (№2, 12.01.2009ж.) негізінде Абай атындағы ҚазҰПУ «Физика-математика сериясы» бойынша Хабаршы журналында жарияланатын мақалалардың реферативті ақпараты INSPEC электронды мәлімтер қорына енгізіледі.

Редакция ғылыми мақала материалдарын толығымен немесе жекелеп ғылыми дәйексөз индексімен қамтамасыз ететін қазақстандық және шетелдік мекемелерге беруге, сонымен қатар осы материалдарды журналдың интернет сайтына орналастыруға құқылы.

«ХАБАРШЫ. ФИЗИКА-МАТЕМАТИКА ҒЫЛЫМДАРЫ СЕРИЯСЫ» ЖУРНАЛЫНА БАСЫЛАТЫН МАҚАЛАЛАРДЫ БЕЗЕНДІРІЛУГЕ ҚОЙЫЛАТЫН ТАЛАПТАР

I. Қажетті материалдар

- 1.Парақтары төменгі жағында қарандашпен нөмірленген мақаланың қатты көшірмесі (2 дана), барлық авторлар мақала мәтінінің бір данасына қол қоюлары керек;
2. Мақаланың аты, мақалаға үш түрде (қазақ, орыс, ағылшын тілдерінде) жазылған аңдатпалардың, түйін сөздердің қатты көшірмесі (1 дана). Аңдатпада зерттеудің мақсаты мен әдіснамасын, негізгі жетістігі мен қорытындысын көрсету қажет. Аңдатпада 5-7 сөйлем болуы керек. Түйін сөздер саны электрондық іздеу жүйесінде мақала табылатындай 5-10 сөзден (сөз тіркесінен) тұруы керек.
- 3 Автор (авторлар – 4 адамнан артық болмау керек) жайында мәліметтердің қатты көшірмесі (1 дана): Ф.А.Ә. толығымен, жұмыс орны (ұжым аты, жоғары оқу орынның толық аты), атқаратын қызметі, жұмыс телефоны, үйдің мекен-жайы, үй телефоны, ғылыми дәрежесі, ғылыми атағы, e-mail.
4. Мақаланың, мақалаға жазылған аңдатпаның және түйін сөздердің электрондық түрі;

II. Мақаланы безендіру ережесі

- 1.Мақала мәтіні Word редакторында бірлік интервал арқылы терілу керек;
- 2.Парақ пішімі : 210 x 297 mm (A4);
3. Жоғары, төменгі, оң жақтағы, сол жақтағы өрістер: – 2,5 см;
- 4.Мақала беттері нөмірленбейді;
- 5.Шрифт: Times New Roman (қазақ, орыс, ағылшын тілдері үшін), өлшемі - 12 пт; Абзацтық жылжу– 1 см.;
6. Кестелер мен графикалық материалдар (суреттер, графикалар, фотосуреттер және т.б.) толассыз нөмірленулері және атаулары (кегль №11) болуы керек. Барлық кестелер мен графикалық материалдарға мақала мәтінінде сілтемелер болуы керек. Берілген нысандар сілтемелерден кейін орналасуы қажет. Word редакторында орындалған суреттер нысан ретінде қойылуы керек;
7. Мақала мәтіні ені бойынша форматталуы керек.

III. Формула жазуға қойылатын талаптар

Формулалардағы (Equation) символдар өлшемдері : Обычный – 11 пт, Крупный индекс – 6 пт, Мелкий индекс – 5 пт, Крупный символ – 24 пт, Мелкий символ – 4 пт; формулалар беттің ортасы бойынша орналастырылады. Сілтеме жасалатын формулаларды ғана нөмірлеу керек. Формулалар толассыз нөмірленеді. Формулалар нөмірлері оң жақ шетімен тегістеледі де, дөңгелек жақшаға алынады. Мәтінде де формулаларға сілтемелерді дөңгелек жақшаға жазады.

IV. Әдебиеттер тізімі

мақалада пайдаланылған әдебиеттер мәтінде пайдалану ретіне сәйкес қолжазбаның соңында келтіріледі. Мақаладағы әдебиетке сілтеу квадраттық жақшада беріледі, мысалы, [1], [2,3], [4-7].

V. Мақаланың түрі

1. Сол жақ жоғарғы бұрышта бас әріптермен ӘОК (кегль №13);
2. Жартылай қарайтылған кіші әріптермен (кегль №13) ортада автордың (авторлардың саны 4 адамнан аспауы керек) фамилиясы және аты-жөні;
3. Бір бос жолдан кейін жартылай қарайтылған бас әріптермен (кегль №13) мақала аты;
4. Бір бос жолдан кейін дөңгелек жақшада автор (авторлар) жұмыс істейтін немесе оқитын қала және ұжымның аты; егер автор (авторлар) білім алушы болса, * (жұлдызша) белгісінен кейін білім алушының статусы (студент, магистрант, докторант) жазылады (кегль №11);
5. Бос жолдан кейін мақала жазылған тілде 5-7 сөйлем көлемінде аңдатпа және 5-10 сөздер болатындай түйін сөздер (кегль №11, курсив);
6. . Бір бос жолдан кейін мақала мәтіні (кегль №12);
7. Пайдаланылған әдебиеттер тізімі мәтіннен кейін екі бос жол тастап (кегль №11) кіші әріптермен келтіріледі. Бірлік интервал.
8. Бос жолдан кейін 2 тілде мақалаға аңдатпа және 5-10 сөздер (сөз тіркесі) болатындай түйін сөздер (кегль №11, курсив).

1-мысал. Мақаланы безендіру

ӘОЖ

ФИО, ФИО*

МАҚАЛАНЫҢ АТЫ

(қала, мекеменің аты, * - студент/магистрант/ докторант)

Аңдатпа. Мақала тілінде (5-7 сөйлем).

Түйін сөздер: Мақала тілінде (5-10сөз, сөз тіркесі)

Мақала мәтіні (5-7 бет)

Пайдаланылған әдебиеттер тізімі

Аңдатпа. Мақала тілінен өзге 2 тілде (5-7 сөйлем).

Түйін сөздер: Мақала тілінен өзге 2 тілде (5-10 сөз, сөз тіркесі)

VI. Мақалаларды жариялау тілдері – қазақ, орыс, ағылшын тілдері.

Редакцияға түскен мақалаларға білім саласы бойынша мамандар мен ғылымдар пікір береді. Пікір негізінде редакция алқасы авторға мақаланы тағы да толықтыруға (түзетуге) ұсыныс жасауы, не мүлдем қайтарып беруі мүмкін. Бұрын жарияланған немесе басқа баспаға жіберілген мақалалар қабылданбайды.

Мақала көлемі 5-7 бет. Көлемі 7 беттен артық болған жағдайда журнал редакциясымен хабарласып келісулері қажет. Мақала мәтініне енетін иллюстрациялардың, сұлбалардың және кестелердің көлемі мәтіннің жалпы көлеміне кіреді.

Мақаланы дайындау және жариялау бойынша пайда болған барлық сұрақтар бойынша журнал редакциясына хабарласыңыздар.

Мекен-жайы: Алматы қаласы, Төле би 86 көшесі, Абай атындағы ҚазҰПУ, Математика, физика және информатика институты, жұмыс телефоны - 8(727) -2 61-68-69, e-mail Vestnik_KazNPU@mail.ru

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

Информируем Вас, что Решением Комитета по контролю в сфере образования и науки МОН РК от «10» июля 2012 года № 1082 «Вестник Казахского национального педагогического университета имени Абая, Серия физико-математические науки» включен в Перечень изданий, рекомендуемых Комитетом по контролю в сфере образования и науки Министерства образования и науки Республики Казахстан для публикации основных результатов научной деятельности по следующим направлениям:

- по специальностям группы 01.01.00 – физико-математические науки (математика);
- по специальности 01.02.00 (01.02.01, 01.02.04, 01.02.06) – физико-математические науки (механика);
- по специальности 05.00.00 (05.02.18) – технические науки;
- по специальности 13.00.00 (13.00.02) – педагогические науки (теория и методика обучения и воспитания /математика, физика, информатика);
- по специальностям группы 01.04.00 – физико-математические науки (физика).

С 2009 г. действует Договор с Институтом Инжиниринга и Технологий (Великобритания), (№2 от 12.01.2009 г.) на оказание информационно-сопроводительных услуг, согласно которому реферативная информация о статьях, публикуемых в Вестнике КазНПУ имени Абая, вносится в электронную базу данных INSPEC.

Редакция имеет право частично или полностью предоставлять материалы научных статей в казахстанские и зарубежные организации, обеспечивающие индекс научного цитирования, а также размещать данные материалы на Интернет-сайте журнала.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ СТАТЕЙ ПУБЛИКУЕМЫХ В ЖУРНАЛЕ «ВЕСТНИК. СЕРИЯ ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ»

I. Представление необходимых материалов

1. Текст статьи в твердой копии с пронумерованными, внизу по центру карандашом, страницами (2 экз.), один экземпляр текста статьи должен быть подписан всеми авторами.
2. Название статьи, аннотации к статье, ключевые слова (на 3-х языках: казахском, русском, английском) в твердой копии (1 экз.). В аннотации необходимо указывать цель и методологию исследования, основные достижения и выводы. Объем аннотации – 5-7 предложений. Ключевые слова: 5–10 слов (словосочетаний), по которым статьи могут быть найдены в электронных поисковых системах.
3. Сведения об авторе (авторах – не более 4 человек) в твердой копии на 3-х языках (1 экз.): фамилия, имя, отчество, место работы (город, название организации/вуза без сокращений), ученая степень и звание, должность, e-mail, контактный телефон.
4. Статья, аннотации к статье, ключевые слова в электронном виде.

II. Правила оформления статей.

1. Текст статьи должен быть набран в редакторе Word через одинарный интервал;
2. Формат листа: 210 x 297 mm (A4);
3. Поля: верхнее, нижнее, правое, левое – 2,5 см;
4. Страницы статьи не нумеруются;
5. Шрифт: Times New Roman (для каз., рус. и англ. языков), размер - 12 пт; Абзацный отступ – 1 см.
6. Таблицы и графические материалы (рисунки, графики, фотографии и др.) должны иметь сквозную нумерацию и названия (кегль №11). На все таблицы и графические материалы должны быть сделаны ссылки в тексте статьи. При этом расположение данных объектов должно быть после ссылок на них. Рисунки, выполненные в редакторе Word, должны быть вставлены как объект.
7. Текст статьи должен быть отформатирован по ширине.

III. Требования к написанию формул

Размеры символов в формулах (Equation): обычный – 11 пт, крупный индекс – 6 пт, мелкий – 5 пт, крупный символ – 24, мелкий – 4 пт; формулы располагают по центру страницы, Нумеровать следует только те формулы, на которые приводятся ссылки. Нумерация формул должна быть сквозной. Номер формулы располагают по правому краю страницы и заключают в круглые скобки. В тексте ссылку на формулу также приводят в круглых скобках.

IV. Список литературы, использованной в статье, составляется по ходу упоминания ее в тексте и приводится в конце рукописи. Ссылки на литературу в тексте указываются в квадратных скобках, например, [1], [2,3], [4-7].

V. Вид статьи

1. УДК в левом верхнем углу прописными буквами (кегель №13);
2. Полуужирными строчными буквами (кегель №13) по центру инициалы и фамилия автора (авторов не более 4 человек);
3. Через пустую строку полуужирными прописными буквами (кегель №13) название статьи;
4. Через пустую строку в круглых скобках, город и название организации, где работает или учится автор (авторы). Если автор (авторы) обучающиеся, то после знака * (звездочка) – указать статус обучающегося (студент, магистрант, докторант) (кегель №11);
5. Через пустую строку аннотация на языке публикации в 5-7 кратких предложениях (кегель №11);
6. Через пустую строку текст статьи (кегель №12);
7. Используемые источники, указываются после текста статьи, через две пустые строки строчными буквами (кегель №11). Интервал – одинарный;
8. Через пустую строку аннотации к статье на 2-х языках, ключевые слова: 5–10 слов (словосочетаний) (кегель №11, курсив).

Образец. Оформление статьи

УДК

ФИО, ФИО*

НАЗВАНИЕ СТАТЬИ

(город, название организации, *- студент/магистрант/докторант)

Аннотация. На языке статьи (5-7 предложений).

Ключевые слова: на языке статьи

Текст статьи (не менее 5 и не более 7 стр.)

Список литературы

Библиографический список

Aңдатпа.

Түйін создер:

Abstract.

Keywords:

VI. Языки издания статей – казахский, русский, английский.

Поступившие в редакцию статьи рецензируются ведущими специалистами и учеными по отраслям знаний. На основании рецензии редколлегии может рекомендовать автору доработать статью или отказать в публикации. Рукописи статей, опубликованных ранее или переданных в другие издания, не принимаются.

Рекомендуемый объем статьи – не менее пяти и не более семи страниц. В ином случае вопрос по объему статьи необходимо согласовать с редакцией журнала. Иллюстрации, схемы, таблицы, включаемые в текст статьи, учитываются в общем объеме текста.

По всем вопросам, связанным с подготовкой, представлением и публикацией материалов, необходимо обращаться в редакцию журнала.

Адрес: г.Алматы, ул.Толе би 86, КазНПУ им.Абая, Институт математики, физики и информатики
рабочий телефон 8(727) -2 61-68-69, e-mail Vestnik_KazNPU@mail.ru