

**Программа утверждена на заседании кафедры вычислительной математики
Протокол № 4 от 27 ноября 2014 г.**

Рабочая программа дисциплины (модуля)

1. Код и наименование дисциплины (модуля): Основы параллельных вычислений 1.
2. Уровень высшего образования – подготовка научно-педагогических кадров в аспирантуре.
3. Направление подготовки: 09.06.01 Информатика и вычислительная техника. Направленность программы: Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (научная специальность 05.13.11).
4. Место дисциплины (модуля) в структуре ООП: вариативная часть ООП, элективный курс по выбору кафедры, обязателен для освоения не позднее второго года обучения.
5. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

Формируемые компетенции (код компетенции)	Планируемые результаты обучения по дисциплине (модулю)
УК-1 УК-4	У1 (УК-1) анализировать альтернативные варианты решения исследовательских и практических задач и оценивать потенциальные выигрыши/проигрыши реализации этих вариантов З1 (УК-4) знать методы и технологии научной коммуникации на государственном и иностранном языках
ОПК-1	В2 (ОПК-1) владеть навыками решения теоретических и практических задач при помощи современных программных средств
ПК-1	З1 (ПК-1) ЗНАТЬ: строение и методы программирования современных параллельных ЭВМ У1 (ПК-1) УМЕТЬ:

	разрабатывать алгоритмы и программы для параллельных ЭВМ с общей и распределенной памятью В2 (ПК-1) ВЛАДЕТЬ: навыками тестирования и отладки программ на параллельных ЭВМ с общей и распределенной памятью
--	--

6. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся:

Объем дисциплины (модуля) составляет 2 зачетных единиц, всего 72 часов, из которых 44 часов составляет контактная работа аспиранта с преподавателем (36 часов занятия лекционного типа, 8 часов мероприятия промежуточной аттестации), 28 часов составляет самостоятельная работа аспиранта.

7. Входные требования для освоения дисциплины (модуля), предварительные условия:

Для того чтобы формирование данной компетенции было возможно, обучающийся, приступивший к освоению программы аспирантуры, должен

Знать: основ работы на ЭВМ и программирования.

Уметь: решать стандартные задачи разработки структур данных и реализации численных методов на языке C++.

Владеть: навыками решения задач из следующих разделов современной математики: математический анализ, линейная алгебра и дифференциальные уравнения.

8. Формат обучения:

спецкурс по выбору кафедры

9. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических или астрономических часов и виды учебных занятий

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе								
		Контактная работа (работа во взаимодействии с преподавателем), часы из них					Самостоятельная работа обучающегося, часы из них			
		Занятия лекционного типа	Занятия семинарского типа	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости коллоквиумы, практические контрольные занятия и др)	Всего	Выполнение домашних заданий	Подготовка рефератов и т.п..	Всего
Тема 1: Архитектуры современных процессоров. Конвейеры, независимо работающие устройства. Кэш память и ее	6	4					4	2		2

организация. Оптимизация кода для современных процессоров.										
Тема 2: Виды многопроцессорных архитектур. Общее строение современных кластерных систем. Типы топологий соединений узлов. Графические и вычислительные сопроцессоры.	6	4					4	2		2
Тема 3: Процессы, состояния, механизмы взаимодействия и управления. Виды ресурсов, приоритеты.	6	4					4	2		2
Тема 4: Межпроцессное взаимодействие стандарта IPC: разделяемая память, семафоры, очереди сообщений.	6	4					4	2		2
Тема 5: Потoki исполнения. Управление и объекты синхронизации типов mutex и condvar.	8	4					4	4		4
Тема 6: Влияние	8	4					4	4		4

дисциплины доступа к оперативной памяти на эффективность работы.										
Тема 7: Message Passing Interface (MPI). Сообщения и их виды. Коммуникаторы. Попарный обмен сообщениями.	8	4					4	4		4
Тема 8: Коллективный обмен сообщениями. Передача строк и столбцов матриц при коллективных обменах.	8	4					4	4		4
Тема 9: Ограничение коллективного обмена на подмножество процессов.	8	4					4	4		4
Промежуточная аттестация: экзамен	8						8			
Итого	72	36					44	28		28

10. Перечень учебно-методического обеспечения для самостоятельной работы аспирантов по дисциплине (модулю).
Списка литературы, см. 12.

11. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю).

- Перечень компетенций: ПК-1

- Описание шкал оценивания: экзамен с оценкой по пятибалльной шкале
- Критерии и процедуры оценивания результатов обучения по дисциплине (модулю), характеризующих этапы формирования компетенций.

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	ШКАЛА оценивания					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
	1	2	3	4	5	
<i>У1 (УК-1)</i>	Отсутствие умений	Частично освоенное умение анализировать альтернативные варианты решения исследовательских и практических задач и оценивать потенциальные выигрыши/проигрыши реализации этих вариантов	В целом успешно, но не систематически осуществляемые анализ альтернативных вариантов решения исследовательских и практических задач и оценка потенциальных выигрышей/проигрышей реализации этих вариантов	В целом успешно, но содержащие отдельные пробелы анализ альтернативных вариантов решения исследовательских задач и оценка потенциальных выигрышей/проигрышей реализации этих вариантов	Сформированное умение анализировать альтернативные варианты решения исследовательских и практических задач и оценивать потенциальные выигрыши/проигрыши реализации этих вариантов	экзамен в форме индивидуального собеседования
<i>31 (УК-4)</i>	Отсутствие знаний	Фрагментарные знания методов и технологий научной коммуникации на государственном и иностранном языках	Неполные знания методов и технологий научной коммуникации на государственном и иностранном языках	Сформированные, но содержащие отдельные пробелы знания методов и технологий научной коммуникации на государственном и иностранном языках	Сформированные и систематические знания методов и технологий научной коммуникации на государственном и иностранном языках	экзамен в форме индивидуального собеседования
<i>В2 (ОПК-1)</i>	Отсутствие навыков	Фрагментарное применение навыков построения	В целом успешное, но не систематическое	В целом успешное, но содержащее отдельные пробелы применение	Успешное и систематическое применение навыков	экзамен в форме индивидуального собеседования

		и анализа математических моделей, решения задач при помощи современных программных средств	применение навыков построения и анализа математических моделей, решения задач при помощи современных программных средств	навыков построения и анализа математических моделей, решения задач при помощи современных программных средств	построения и анализа математических моделей, решения задач при помощи современных программных средств	
З1 (ПК-1)	Не имеет базовых знаний	Допускает существенные ошибки	Демонстрирует частичные знания	Демонстрирует знания с небольшими пробелами	Раскрывает полное строение и методы программирования современных параллельных ЭВМ	экзамен в форме индивидуального собеседования
У1 (ПК-1)	Не умеет и не готов формулировать	Фрагментарное умение разрабатывать программы для параллельных ЭВМ с общей и распределенной памятью	В целом успешное, но не систематическое умение разрабатывать программы для параллельных ЭВМ с общей и распределенной памятью	В целом успешное, но содержащее отдельные пробелы умение разрабатывать программы для параллельных ЭВМ с общей и распределенной памятью	Готов и умеет разрабатывать программы для параллельных ЭВМ с общей и распределенной памятью	экзамен в форме индивидуального собеседования
В2 (ПК-1)	Не владеет методами и навыками.	Владеет отдельными приемами	Владеет приемами и навыками решения основных стандартных задач	В целом успешное, но содержащее отдельные пробелы владение навыками тестирования и отладки программ на параллельных ЭВМ с общей и распределенной памятью	Полностью владеет навыками тестирования и отладки программ на параллельных ЭВМ с общей и распределенной памятью	экзамен в форме индивидуального собеседования

- Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций.

Контрольные вопросы и задания по обязательной и вариативной частям дисциплины для промежуточной аттестации по итогам освоения дисциплины:

1. CISC и RISC процессоры. Основные черты RISC архитектуры.
2. Повышение производительности процессоров за счет конвейеризации. Условия оптимального функционирования конвейера.
3. Суперконвейерные и суперскалярные процессоры. Выделение независимо работающих устройств: IU, FPU, MMU, BU.
4. Методы уменьшения негативного влияния инструкций перехода на производительность процессора.
5. Повышение производительности процессоров за счет введения кэш памяти. Кэши: единый, Гарвардский, с прямой записью, с обратной записью.
6. Организация кэш-памяти. Алгоритмы замены данных в кэш памяти. Специальные кэши.
7. Согласование кэшей в мультипроцессорных системах с общей памятью.
8. Виды многопроцессорных архитектур. Системы с разделяемой и распределенной памятью. Гибридные системы.
9. Типы топологий соединений узлов кластерных систем. Взаимодействие Infiniband и Ethernet каналов. Коммуникационный сопроцессор платы ввода-вывода, его взаимодействие с основной системой.
10. Общее строение современных кластерных систем, их подсистем коммуникации и ввода-вывода. Проблемы масштабирования в больших системах.
11. Графические и вычислительные сопроцессоры. Методы программирования и взаимодействия с основной вычислительной системой.
12. Методы повышения производительности и емкости оперативной и дисковой памяти.
13. Поддержка многозадачности и многопроцессорности специальными инструкциями процессора. Организация данных во внешней памяти.
14. Программа, процессор, процесс. Основные составляющие процесса, состояния процесса. Стек, виртуальная память, механизмы трансляции адреса.
15. Механизмы взаимодействия процессов. Разделяемая память, семафоры, сигналы, почтовые ящики, события. Задачи (threads). Сравнение с процессами. Ресурсы, приоритеты. Параллельные процессы. Связывание. Статическое и динамическое связывание.
16. Виды ресурсов: аппаратные, программные, активные, пассивные, локальные, разделяемые, постоянные, временные, не критичные, критичные.

17. Типы взаимодействия процессов: сотрудничающие и конкурирующие процессы. Критические секции, взаимное исключение процессов (задач).
18. Проблемы, возникающие при синхронизации задач и идеи их разрешения.
19. Состояния процесса и механизмы перехода из одного состояния в другое.
20. Стандарты на UNIX системы. Стандарты на работу с объектами в разделяемой и распределенной памяти.
21. Управление процессами. Основные этапы создания нового процесса. Отношения между родительским и порожденным процессами (функции `fork`, `execl`, `execv`, `waitpid`).
22. Работа с сигналами. Идентификация сигналов, методы реакции на сигналы и инициации сигналов (функции `signal`, `kill`). Оповещение процесса о наступлении аварийной ситуации.
23. Межпроцессное взаимодействие стандарта IPC: разделяемая память (функции `shmget`, `shmat`, `shmctl`); семафоры (функции `semget`, `semop`, `semctl`); очереди сообщений (функции `msgget`, `msgsnd`, `msgrcv`, `msgctl`). Роль ядра ОС в механизме IPC.
24. Управление задачами (threads). Функции `pthread_create`, `pthread_join`, `sched_yield`.
25. Объекты синхронизации типа `mutex`. Функции `pthread_mutex_init`, `pthread_mutex_lock`, `pthread_mutex_trylock`, `pthread_mutex_unlock`, `pthread_mutex_destroy`.
26. Объекты синхронизации типа `condvar`. Функции `pthread_cond_init`, `pthread_cond_signal`, `pthread_cond_broadcast`, `pthread_cond_wait`, `pthread_cond_destroy`.
27. Влияние дисциплины доступа к оперативной памяти на эффективность работы.
28. Message Passing Interface (MPI). Общая структура MPI-программы. Функции `MPI_Init`, `MPI_Finalize`. Сообщения и их виды.
29. Коммуникаторы. Функции `MPI_Comm_size`, `MPI_Comm_rank`.
30. Попарный обмен сообщениями. Функции `MPI_Send`, `MPI_Recv`.
31. Операции ввода-вывода в MPI программах.
32. Дополнительные возможности для попарного обмена сообщениями. Функции `MPI_Sendrecv`, `MPI_Sendrecv_replace`.
33. Дополнительные возможности для попарного обмена сообщениями. Функции `MPI_Isend`, `MPI_Irecv`, `MPI_Test`, `MPI_Testany`, `MPI_Wait`, `MPI_Waitany`.
34. Коллективный обмен сообщениями. Функции `MPI_Barrier`, `MPI_Abort`, `MPI_Bcast`.
35. Коллективный обмен сообщениями. Функции `MPI_Reduce`, `MPI_Allreduce`, `MPI_Op_create`, `MPI_Op_free`.
36. Время в MPI программах. Астрономическое и процессорное время. Функции `MPI_Wtime`, `MPI_Wtick`.
37. Дополнительные возможности для коллективного обмена массивами данных. Функции `MPI_Gather`, `MPI_Allgather`, `MPI_Scatter`.

38. Дополнительные возможности для коллективного обмена массивами данных. Передача строк и столбцов матриц. Функции `MPI_Type_vector`, `MPI_Type_commit`.
39. Ограничение коллективного обмена на подмножество процессов. Функции `MPI_Comm_group`, `MPI_Group_incl`, `MPI_Comm_create`, `MPI_Comm_free`.

Билеты формируются в виде трех вопросов (А, Б, В) из указанного списка и одной задачи (Г), примеры задач см. далее.

Образцы билетов:

Билет №1.

- А. Организация кэш-памяти. Алгоритмы замены данных в кэш памяти. Специальные кэши.
- Б. Межпроцессное взаимодействие стандарта IPC: разделяемая память (функции `shmget`, `shmat`, `shmctl`); семафоры (функции `semget`, `semop`, `semctl`); очереди сообщений (функции `msgget`, `msgsnd`, `msgrcv`, `msgctl`). Роль ядра ОС в механизме IPC.
- В. Дополнительные возможности для коллективного обмена массивами данных. Функции `MPI_Gather`, `MPI_Allgather`, `MPI_Scatter`.
- Г. Написать многопоточную функцию, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и возвращающую ненулевое значение, если массив A симметричен (т.е. $A_{\{ij\}} = A_{\{ji\}}$), 0 в противном случае. При этом должна быть обеспечена равномерная загрузка всех потоков. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту функцию и выводить на экран результат ее работы.

Билет №2

- А. Согласование кэшей в мультипроцессорных системах с общей памятью.
- Б. Управление задачами (threads). Функции `pthread_create`, `pthread_join`, `sched_yield`.
- В. Дополнительные возможности для коллективного обмена массивами данных. Передача строк и столбцов матриц. Функции `MPI_Type_vector`, `MPI_Type_commit`.
- Г. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и заменяющую матрицу A на ее транспонированную. При этом должна быть обеспечена равномерная загрузка всех потоков. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.

Билет №3

- А. Виды многопроцессорных архитектур. Системы с разделяемой и распределенной памятью. Гибридные системы.
- Б. Объекты синхронизации типа mutex. Функции `pthread_mutex_init`, `pthread_mutex_lock`, `pthread_mutex_trylock`, `pthread_mutex_unlock`, `pthread_mutex_destroy`.
- В. Ограничение коллективного обмена на подмножество процессов. Функции `MPI_Comm_group`, `MPI_Group_incl`, `MPI_Comm_create`, `MPI_Comm_free`.
- Г. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и заменяющую матрицу A на матрицу $(A+A^t)/2$, где A^t - транспонированная матрица A . При этом должна быть обеспечена равномерная загрузка всех задач. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.

- Методические материалы, определяющие процедуры оценивания результатов обучения: листки с задачами для самостоятельного решения.

Примеры задач:

1. Написать программу, вычисляющую сумму элементов N последовательностей вещественных чисел, находящихся в N файлах, имена которых заданы массивом A . Ответ должен быть записан в файл `res.txt`. Программа запускает N процессов, вычисляющих ответ для каждого из файлов и прибавляющих его к результату, находящемуся в выходном файле. Взаимное исключение при доступе к файлу обеспечивается с помощью семафора.
2. Написать функцию, получающую в качестве аргументов целое число N и массив длины N с именами файлов, содержащих единую последовательность вещественных чисел неизвестной длины, и возвращающую количество участков постоянства этой последовательности. Функция возвращает -1, -2 и т.д., если она не смогла открыть какой-либо файл, прочитать элемент и т.д.. Функция должна запускать N процессов, обрабатывающих свой файл и передающих результаты в основной процесс через очередь сообщений для формирования ответа для последовательности в целом.
3. Написать программу, осуществляющую мониторинг и перезапуск в случае завершения работы заданного количества приложений. Приложения задаются массивом строк, являющихся их полным путевым именем, и не имеют аргументов.
4. Написать реализацию стека строк в разделяемой памяти. При запуске программа создает блок разделяемой памяти (если его еще нет) или присоединяется к существующему блоку. Программа должна обеспечивать основные функции работы со стеком (добавить и удалить элемент) и возможность запуска себя во многих экземплярах.

5. Написать реализацию набора конфигурационных параметров для многих одновременно работающих экземпляров программы. Набор параметров задается некоторой структурой данных и хранится в разделяемой памяти. Блок разделяемой памяти создается при запуске первого экземпляра программы и заполняется из файла. Перед окончанием работы последнего экземпляра набор параметров сохраняется в файле, а блок разделяемой памяти удаляется. Программа должна обеспечивать основные функции работы с набором параметров (прочитать и изменить элемент), причем в случае изменения данных одним из экземпляров он оповещает все остальные экземпляры с помощью сигнала.
6. Написать многопоточную функцию, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и возвращающую ненулевое значение, если массив A симметричен (т.е. $A_{ij} = A_{ji}$), 0 в противном случае. При этом должна быть обеспечена равномерная загрузка всех потоков. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту функцию и выводить на экран результат ее работы.
7. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и заменяющую матрицу A на ее транспонированную. При этом должна быть обеспечена равномерная загрузка всех потоков. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.
8. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, целое число N , номер потока K , общее количество потоков P , и заменяющую матрицу A на матрицу $(A + A^t)/2$, где A^t - транспонированная матрица A . При этом должна быть обеспечена равномерная загрузка всех задач. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.
9. Написать многопоточную подпрограмму, получающую в качестве аргументов массив A вещественных чисел, целое число N , являющееся длиной этого массива, номер потока K , общее количество потоков P , и заменяющую каждый элемент массива (для которого это возможно) на среднее арифметическое соседних элементов. При этом должна быть обеспечена равномерная загрузка всех задач. Основная программа должна вводить числа P , N и массив A (из файла или по заданной формуле), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.
10. Написать многопоточную функцию, получающую в качестве аргументов массив A вещественных чисел, целое число N , являющееся длиной этого массива, номер потока K , общее количество потоков P , и заменяющую элементы массива, принадлежащие участку строго возрастания, на среднее арифметическое элементов во всех таких участках массива. При этом должна быть обеспечена равномерная загрузка всех потоков. Основная программа должна вводить числа P , N и массив A (из

файла или по заданной формуле), запускать потоки, в которых вызывается эта функция, выводить на экран результат, а также процессорное время, затраченное на каждый поток и суммарно на все потоки.

11. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, вспомогательный массив B вещественных чисел длины N (в каждом потоке свой), целое число N , номер потока K , общее количество потоков P , и заменяющую каждый элемент A_{ij} матрицы A (для которого это возможно) на $(A_{i+1,j} + A_{i-1,j} + A_{i,j+1} + A_{i,j-1} - 4A_{i,j})$. При этом должна быть обеспечена равномерная загрузка всех задач. Основная программа должна вводить числа P , N и массив A (из файла или по заданным формулам), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.
12. Написать многопоточную подпрограмму, получающую в качестве аргументов $N \times N$ массив A вещественных чисел, вспомогательный массив B вещественных чисел длины $2N$ (в каждом потоке свой), целое число N , номер потока K , общее количество потоков P , и заменяющую каждый элемент A_{ij} матрицы A (для которого это возможно) на $(A_{i+2,j} + A_{i-2,j} + A_{i,j+2} + A_{i,j-2} - 4A_{i,j})$. При этом должна быть обеспечена равномерная загрузка всех задач. Основная программа должна вводить числа P , N и массив A (из файла или по заданным формулам), запускать потоки, вызывать эту подпрограмму и выводить на экран результат ее работы.
13. Написать MPI функцию, получающую в качестве аргументов соответствующую часть (блок) $N \times N$ массива A вещественных чисел, целое число N , номер процесса K , общее количество процессов P , и возвращающую ненулевое значение, если массив A симметричен (т.е. $A_{ij} = A_{ji}$), 0 в противном случае. При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту функцию и выводить на экран результат ее работы.
14. Написать MPI подпрограмму, получающую в качестве аргументов соответствующую часть (блок) $N \times N$ массива A вещественных чисел, целое число N , номер процесса K , общее количество процессов P , и заменяющую матрицу A на ее транспонированную. При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту подпрограмму и выводить на экран результат ее работы.
15. Написать MPI подпрограмму, получающую в качестве аргументов соответствующую часть (блок) $N \times N$ массива A вещественных чисел, целое число N , номер процесса K , общее количество процессов P , и заменяющую матрицу A на матрицу $(A + A^t)/2$, где A^t -- транспонированная матрица A . При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту подпрограмму и выводить на экран результат ее работы.

16. Написать MPI подпрограмму, получающую в качестве аргументов соответствующую часть (блок) массива A вещественных чисел, целое число N , являющееся длиной этого массива, номер процесса K , общее количество процессов P , и заменяющую каждый элемент массива (для которого это возможно) на среднее арифметическое соседних элементов. При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту подпрограмму и выводить на экран результат ее работы.
17. Написать MPI подпрограмму, получающую в качестве аргументов соответствующие части (блоки) $N \times N$ массива A вещественных чисел, вспомогательный массив B вещественных чисел длины N , целое число N , номер процесса K , общее количество процессов P , и заменяющую каждый элемент A_{ij} матрицы A (для которого это возможно) на $(A_{i+1,j} + A_{i-1,j} + A_{i,j+1} + A_{i,j-1}) - 4A_{i,j}$. При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту подпрограмму и выводить на экран результат ее работы.
18. Написать MPI подпрограмму, получающую в качестве аргументов соответствующие части (блоки) $N \times N$ массива A вещественных чисел, вспомогательный массив B вещественных чисел длины $2N$, целое число N , номер процесса K , общее количество процессов P , и заменяющую каждый элемент A_{ij} матрицы A (для которого это возможно) на $(A_{i+2,j} + A_{i-2,j} + A_{i,j+2} + A_{i,j-2}) - 4A_{i,j}$. При этом должна быть обеспечена равномерная загрузка всех процессов. Основная программа должна начинать работу с MPI, вводить число N и массив A (из файла или по заданной формуле), вызывать эту подпрограмму и выводить на экран результат ее работы.

12. Ресурсное обеспечение:

- Перечень основной и дополнительной учебной литературы:

1. К.Ю. Богачев. Основы параллельных вычислений. Москва: ЦПИ при механико-математическом ф-те МГУ им. М.В.Ломоносова, 2002. 352 с.
2. К.Ю. Богачев. Основы параллельного программирования. Москва: Бином, 2003. 342 с. ISBN: 5-94774-037-0.
3. В.В.Воеводин, Вл.В.Воеводин. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.

- Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

- Перечень используемых информационных технологий, используемых при осуществлении образовательного процесса, включая программное обеспечение, информационные справочные системы (при необходимости):
 1. Справочная система для библиотек POSIX threads и MPI
- Описание материально-технической базы:
 1. аудиторный фонд механико-математического факультета

13. Язык преподавания.
русский

14. Преподаватель (преподаватели).
доцент Кирилл Юрьевич Богачев

Заведующий кафедрой
Вычислительной математики
профессор, д.ф.-м.н.

/Кобельков Г.М./