УТВЕРЖДАЮ

Зав. кафедрой СП ЮУрГУ

\_\_\_\_\_\_\_\_\_Л.Б. Соколинский

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Фонд оценочных средств

ООП «Инженерия информационных и интеллектуальных систем»

по направлению 09.03.04 – Программная инженерия

Дисциплина «Теория, методы и средства параллельной обработки информации»

| **№ КМ** | **Вид КМ** | **Наименование КМ** | **Оценочные средства** |
| --- | --- | --- | --- |
|  | Текущий контроль | Защита лабораторной работы №1 | Вопросы:   1. Привести определения процесса и потока. 2. Чем определяется порядок выполнения потоков программы? Какая дисциплина используется? 3. Описать все смены состояний потоков в ходе работы приложения. 4. Какова дисциплина обслуживания буфера и почему выбрана именно она? Какие операции доступа к буферу должны синхронизироваться и почему? 5. Какой механизм выбран для реализации синхронизации и почему? Чем он отличается от классического семафора Дейкстры? |
|  | Текущий контроль | Защита лабораторной работы №2 | Вопросы:   1. Привести определение файла, отображаемого в память, пояснить причину появления этого механизма и способы его использования. 2. К какому из видов IPC относится этот механизм? 3. Прокомментировать порядок использования функций, необходимых для реализации этого механизма, и их интерфейсы. 4. Как эти функции связаны с используемой на Вашем компьютере организацией основной памяти? 5. Как реализуется синхронизация операций чтения/записи? |
|  | Текущий контроль | Защита лабораторной работы №3 | Вопросы:   1. Привести определение канала, описать способ его использования. 2. К какому из видов IPC относится этот механизм? 3. Прокомментировать порядок использования функций, необходимых для реализации этого механизма, и их интерфейсы. 4. Чем канал отличается от предыдущего механизма - файла, отображаемого в память? 5. Какой из использованных механизмов IPC Вам показался наиболее удобным, почему? |
|  | Текущий контроль | Защита лабораторной работы №4 | Вопросы:   1. Что такое OpenMP? Какие модели он реализует? Опишите модели и их связь. 2. Как идентифицируются нити в OpenMP? Для чего это нужно? Приведите содержательный пример. Совпадают ли эти идентификаторы с идентификаторами потоков в ОС? 3. Для чего нужны частные переменные? Не противоречит ли их существование реализуемой OpenMP модели программирования в общей памяти? Приведите содержательный пример частной переменной. Какие новые области видимости появляются в параллельной программе? Как они задаются? 4. Продемонстрируйте конфликт обращений к переменной rank в написанной программе? Всегда ли он возникает? Как его предотвратить? 5. Какие дополнительные действия выполняет директива, если она имеет параметр reduction? Чем инициализируются частные переменные, создаваемые параметром reduction? 6. Когда и в каком порядке назначаются итерации цикла нитям для выполнения? |
|  | Текущий контроль | Защита лабораторной работы №5 | Вопросы:   1. Что такое MPI? Какую модель параллельного программирования он реализует, на какую архитектуру ориентирован?  Как Вы подключили его в используемой системе программирования? 2. Чем в MPI реализованы параллельно выполняемые подзадачи? Как и в какой момент они запускаются? До каких пор существуют? Чем идентифицируются? 3. Что такое коммуникатор? Как учитываются входящие в него процессы? 4. Что делают функции MPI\_Init() и MPI\_Finalize()? Какую роль играют? Сколько раз могут быть вызваны? 5. Рассказать общую схему межпроцессного обмена (перечислить участников, указать порядок действий). Привести общий формат функции передачи сообщений. |
|  | Текущий контроль | Защита лабораторной работы №6 | Вопросы:   1. Для каких устройств предназначен стандарт CUDA? 2. Какую модель параллельного программирования реализует этот стандарт? 3. Как организуется программа в стандарте CUDA? 4. Какую роль играет команда kernel, каковы ее параметры? |
|  | Текущий контроль | Защита лабораторной работы №7 | Вопросы:   1. Расскажите о функциях управления памятью на графическом ускорителе. 2. Какой размер грида для данной задачи даст наибольшее ускорение решения? |
|  | Текущий контроль | Защита лабораторной работы №8 | Вопросы:   1. Как будут распределены итерации по нитям? |
|  | Текущий контроль | Тест 1 | Примеры вопросов теста:   1. Суперкомпьютерные системы необходимы для:  * работы большого количества пользователей в одной системе * решения сверхбольших задач * обработки большого количества поступающих запросов  1. Выберите существующие виды параллелизма:  * многопроцессорная обработка * конвейерная обработка * векторная обработка * гетерогенная обработка * полупоследовательная обработка  1. К классу систем с общей памятью относятся:  * UMA-системы * SMP-системы * ccNUMA системы * nccNUMA-системы * MPP-системы |
|  | Текущий контроль | Тест 2 | Примеры вопросов теста:   1. Идеальной технологией распараллеливания программ являлась бы (если бы существовала):  * распараллеливающий компилятор * низкоуровневая коммуникационная библиотека * библиотека параллельных алгоритмов * параллельный язык программирования  1. Что делает директива #pragma omp parallel  * Определяет блок кода, который будет выполнен всеми созданными на входе в этот блок нитями * Определяет цикл, итерации которого должны выполняться одновременно несколькими нитями * Определяет блок кода, который будет выполнен только главной нитью  1. Процессы MPI-программы взаимодействуют в рамках  * коммуникаторов * пространства общей памяти * специальных переменных - семафоров * сокетов  1. Стандарт CUDA реализует модель  * SISD * SIMD * SIMT * MIMD * MIMT |
|  | Промежуточный аттестация | Итоговый тест | Вопросы для подготовки к экзамену:   1. Большие задачи. Ускорение расчетов при использовании параллелизма. 2. Виды параллельной обработки 3. Классификации параллельных вычислительных систем: классификация Флинна, классификация MIMD-систем 4. Способы оценки производительности многопроцессорных систем 5. Инструменты параллельного программирования: расширения существующих языков, языки параллельного программирования, низкоуровневые интерфейсы, библиотеки параллельных алгоритмов, инженерные пакеты, инструментальные среды параллельной разработки 6. Технологический цикл разработки: разбиение, установление связей, агрегирование и привязка. Методы реализации каждого из этапов 7. Ускорение и эффективность параллельного алгоритма. 8. Стандарт OpenMP. Принципы использования. Основные директивы и функции. 9. Общие и частные переменные в OpenMP. Гонка потоков. 10. Стандарт передачи сообщений MPI. Принципы использования. Основные функции. 11. Идентификация процессов и задач в MPI. Коммуникаторы. 12. Коммуникации в MPI. 13. Архитекура GPU NVIDIA. 14. Стандарт CUDA. Структура программы. Основные директивы и функции. |

Паспорт фонда оценочных средств приведен в п. 6.3 РПД.

Разработчик Т.Ю. Маковецкая

ФГАОУ ВО «Южно-Уральский государственный университет   
(национальный исследовательский университет)»

Кафедра системного программирования

Дисциплина «Теория, методы и средства параллельной обработки информации»

ИТОГОВЫЙ ТЕСТ

| № | Вопрос | Варианты ответа |
| --- | --- | --- |
|  | Выберите существующие виды параллелизма | * многопроцессорная обработка * конвейерная обработка * векторная обработка * гетерогенная обработка * полупоследовательная обработка * распределенные вычисления |
|  | Сколько времени понадобится для выполнения сложения двух векторов длины 100\*{n}  на конвейерном устройстве содержащем {x} одинаковых сумматоров? | 100\*{n}+{x}-1 |
|  | Выберите верное определение масштабируемого алгоритма: | * при росте числа процессоров обеспечивает увеличение ускорения при сохранении постоянного уровня эффективности использования процессоров * при росте числа процессоров обеспечивает увеличение ускорения при увеличении уровня эффективности использования процессоров * при росте числа процессоров обеспечивает сохранение постоянного уровня ускорения при увеличении эффективности использования процессоров |
|  | Технология OpenMP реализует модель | * Fork-Join для систем с общей памятью * Fork-Join для систем с распределенной памятью * Мастер-Рабочие для систем с общей памятью * Мастер-Рабочие для систем с распределенной памятью |
|  | Укажите верное определение термина «частная переменная» OpenMP-программы. | * Переменная, которая существует в нескольких независимых друг от друга экземплярах и доступна всем нитям программы под одним и тем же именем. * Переменная, которая существует в единственном экземпляре для всей программы и доступна всем нитям под одним и тем же именем. * Переменная, экземпляр которой создается в момент создания всех нитей программы и уничтожается в момент завершения этих нитей. |
|  | Дана OpenMP-программа: #include "omp.h" #include <stdio.h> void main() {    int r;    #pragma omp parallel    {        r=omp\_get\_thread\_num();        printf("%d\n",r);    } } Что будет выдано на экран в результате работы данной программы? | * Числа (возможно, повторяющиеся) из диапазона от 0 до N, где N - число создаваемых нитей * Числа от 0 до N без повторений, где N - число создаваемых нитей * Число из диапазона от 0 до N, где N - число создаваемых нитей |
|  | Процессы MPI-программы взаимодействуют в рамках | * коммуникаторов * пространства общей памяти * пространства общей памяти * одного процессора |
|  | Функция MPI-Init | * должна вызываться в MPI-программе только один раз * может вызываться в MPI-программе многократно, но только после функции MPI-Finalize() * необязательна для использования |
|  | Единицей масштабирования графических ускорителей NVIDIA является | * потоковый мультипроцессор * вычислительное ядро * многопоточное ядро * процессор * контроллер |
|  | Стандарт CUDA реализует модель | * SIMT * SISD * SIMD * MIMD * MIMT |