Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт зачета

по дисциплине «Параллельное программирование», 8 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет состоит из 3 вопросов и формируется по следующему правилу:

- первый вопрос выбирается из диапазона вопросов 1 – 27;
- второй вопрос из диапазона вопросов 28- 54;
- второй вопрос из диапазона вопросов 55-83.

Таким образом, проверяются результаты обучения по, соотнесенные с установленными в программе индикаторами достижения компетенций.

На зачете преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____
к зачету по дисциплине «Параллельное программирование»

---

1. Каким ограничениям должен удовлетворять цикл для того, чтобы к нему можно было применить директиву for OpenMP?
2. С помощью какого расширения синтаксиса запускается ядро CUDA?
3. Что такое виртуальная топология в MPI? Как она связана с физической топологией вычислительной сети/кластера/комплекса?

Утверждаю: зав. кафедрой ____ _____ должность, ФИО
(подпись)
(дата)

2. Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Ответ на билет для зачета засчитывается на **продвинутом** уровне, если студент проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает

механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Установленные в программе компетенции сформированы в полном объеме. Оценка составляет *от 35 до 40 баллов.*

Ответ на билет для зачета засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Установленные в программе компетенции сформированы на базовом уровне. Оценка составляет *от 34 до 27 баллов.*

Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Установленные в программе компетенции сформированы на пороговом уровне. Оценка составляет *от 26 до 20 баллов.*

Ответ на билет для зачета считается **неудовлетворительным** (ниже порогового уровня), если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Установленные в программе компетенции не сформированы. Оценка составляет *менее 20 баллов.*

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине.

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

4. Вопросы к зачету по дисциплине «Параллельное программирование»

1. Перечислите составные части технологии OpenMP.
2. С помощью какой директивы (директив) создаются новые параллельные области программы?
3. Что такое критическая секция программы?
4. Каким образом можно установить нужное количество потоков для создания очередной параллельной области?
5. Как обеспечить выполнение фрагмента параллельной области только главным потоком?
6. Какая опция директивы OpenMP *for* используется для указания способа распределения итераций цикла между потоками?
7. Что такое *deadlock*? Каким правилам нужно следовать, чтобы избежать возможности попадания параллельной программы в *deadlock*?
8. Что такое сведение данных? Какие опции и в каких директивах используются для выполнения сведения?
9. Что делает директива OpenMP *threadprivate*?

10. Как обеспечить выполнение фрагмента параллельной области потоком с максимальным номером в данной параллельной области?
11. Для чего используется опция *firstprivate*? Чем она отличается от опций *private* и *lastprivate*?
12. Что такое вложенная параллельная область программы? В каких случаях ее нельзя создать?
13. Что такое неявная барьерная синхронизация? С помощью каких средств ее можно отменить?
14. Для чего используются директивы OpenMP *sections* и *section*? Что делает каждая из этих директив?
15. Перечислите все средства синхронизации потоков в OpenMP.
16. Перечислите возможные способы распределения итераций цикла между потоками.
17. Что делает директива OpenMP *atomic*?
18. В чем состоит различие между общими и локальными переменными потока?
19. С помощью каких средств можно ограничить глубину вложенности параллельных областей программы?
20. От чего зависит равномерность загрузки процессоров/ядер системы с общей памятью?
21. Каким образом функция, вызываемая из параллельной программы может выяснить, в последовательной или параллельной области она выполняется?
22. Как обеспечить выполнение фрагмента параллельной области в точности одним потоком?
23. Какое значение будет иметь переменная *count* в результате выполнения фрагмента параллельной программы:

```
int count = 0;
#pragma omp parallel for
for(int i = 0; i<10; i++){
count++;
}
```

24. Могут ли два потока одновременно захватить один множественный замок?
25. Для чего может использоваться опция *reduction* в некоторых директивах OpenMP?
26. Каково назначение директивы OpenMP single?
27. Каким ограничениям должен удовлетворять цикл для того, чтобы к нему можно было применить директиву OpenMP for?
28. Чем отличается простой замок от множественного?
29. Перечислите известные Вам механизмы синхронизации потоков в OpenMP.
30. Перечислите способы распределения итераций цикла между потоками параллельной области.
31. Что такое ядро в терминологии CUDA?
32. Как задается размерность и количество потоков ядра, выполняемого графическим процессором?
33. Перечислите виды и характеристики памяти, доступной из программы GPU.
34. Как указать требуемое размещение переменной в памяти GPU (регистровой, разделяемой, глобальной, памяти текстур, …)?
35. Перечислите виды и характеристики памяти, доступной из программы CPU при использовании архитектуры CUDA.
36. Какова максимальная размерность блока в потоках?
37. Перечислите встроенные векторные типы данных расширения языка C и объясните смысл их наименований.
38. Как в программе для CPU обеспечить синхронизацию с программой для GPU?
39. Если к одному CPU подключено несколько видеокарт NVidia архитектуры CUDA, то каким образом можно обеспечить их одновременную загрузку?

40. Перечислите виды и характеристики памяти GPU, доступной из программы основного процессора.
41. Какие ограничения наложены на функции, которые должны выполняться на GPU?
42. Что такое мультипроцессор в терминологии CUDA?
43. Какие ограничения наложены на функции, выполняемые в графическом процессоре?
44. Как определить версию и технические характеристики графического процессора NVIDIA?
45. Перечислите и охарактеризуйте группы библиотечных функций Run-time библиотеки CUDA.
46. Что такое сетка, блок, поток в терминологии CUDA?
47. Что такое тип данных dim3?
48. Какова максимальная размерность сетки в блоках?
49. Перечислите и охарактеризуйте функции компонент программного обеспечения CUDA.
50. Можно ли (и если да – то как) в программе для CPU выяснить, какую версию спецификации CUDA реализует видеокарта?
51. Какую модель параллелизма реализует архитектура CUDA?
52. Что делает препроцессор nvcc?
53. Можно ли получить указатель на функцию, выполняемую графическим процессором?
54. С помощью какого расширения синтаксиса запускается ядро CUDA?
55. Перечислите основные группы функций CUDA и охарактеризуйте их назначение.
56. Перечислите и охарактеризуйте дополнительные виды функций, существующие в расширении синтаксиса языка C для CUDA.
57. Какие виды памяти доступны из программ для GPU?
58. Как синхронизируются процессы в CPU и в GPU?
59. Можно ли одновременно использовать технологии OpenMP и CUDA?
60. Какие имена полей используются во встроенных векторных типах?
61. С данными какого типа оперируют арифметико-логические устройства GPU CUDA?
62. Как узнать максимальное количество потоков в блоке для видеокарты NVIDIA, подключенной к CPU?
63. Какие виды виртуальных топологий можно реализовывать с использованием MPI?
64. Что такое барьерная синхронизация в MPI? Какие еще виды синхронизации существуют в стандарте MPI-1?
65. Перечислите основные группы функций стандарта MPI-1.
66. Может ли MPI-программа, выполняющаяся на многоядерном узле, использовать все его ядра и если да, то каким образом?
67. Что такое пользовательская операция редукции?
68. Чем различаются размер и протяженность типов данных MPI?
69. Что такое буферизованная передача сообщений?
70. Что такое коммуникатор? Какие проблемы решаются с использованием этого понятия?
71. Сколько в MPI операций приема сообщений типа «точка-точка»? Перечислите и охарактеризуйте их.
72. Что такое коллективное взаимодействие ветвей программы?
73. Как путем вызова одной функции можно одновременно передать и принять блок данных?
74. Перечислите предопределенные операции редукции данных MPI.
75. Что такое виртуальная топология в MPI? Как она связана с физической топологией вычислительной сети/кластера/комплекса?
76. Для чего в MPI-программах можно использовать производные типы данных?
77. Что понимается под операциями сдвига и циклического сдвига данных? Какие функции обычно используются для их реализации?

78. Что такое исключающая редукция (MPI_Exscan), как она выполняется?
79. Может ли ветвь параллельной программы с помощью MPI передавать сообщения сама себе?
80. Какие операции рассылки и сбора данных реализованы в MPI?
81. Должны ли абсолютно все ветви параллельной программы принимать участие в коллективном взаимодействии?
82. Сколько конструкторов производных типов реализовано в MPI? Перечислите и охарактеризуйте их.
83. Чем различаются топологии «тор» и «решетка»?