

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

**«ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ»**

Специальность

27.05.01 Специальные организационно-технические системы

Специализация

Информационные технологии и программное обеспечение в специальных  
организационно-технических системах

Квалификация (степень) выпускника — инженер-системотехник

Форма обучения — очная, очно-заочная

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общепрофессиональных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета и экзамена. Форма проведения – тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

## 2. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
Тема 1. Общая характеристика параллельных вычислительных систем	ПК-1.1, ПК-1.2, ПК-1.3	Экзамен, зачет
Тема 2. Параллельные вычисления	ПК-1.1, ПК-1.2, ПК-1.3	Экзамен, зачет
Тема 3. Технология параллельного программирования OpenMP	ПК-1.1, ПК-1.2, ПК-1.3	Экзамен, зачет
Тема 4. Технология параллельного программирования для графических процессоров CUDA	ПК-1.1, ПК-1.2, ПК-1.3	Экзамен, зачет
Тема 5. Перспективные направления в параллельном программировании	ПК-1.1, ПК-1.2, ПК-1.3	Экзамен, зачет

## 3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

### Уровень освоения компетенций, формируемых дисциплиной:

#### Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

### Описание критериев и шкалы оценивания теоретического вопроса:

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

### Описание критериев и шкалы оценивания практического задания:

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла (эталонный уровень)	Задача решена верно
2 балла (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На промежуточную аттестацию (зачет) выносится тест, два теоретических вопроса и 2 задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено» и «не зачтено».

**Оценки «зачтено»** заслуживает обучающийся, продемонстрировавший полное знание материала изученной дисциплины, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; показавший систематический характер знаний по дисциплине, ответивший на все вопросы билета или допустивший погрешности в ответах на вопросы, но обладающий необходимыми знаниями для их устранения под руководством преподавателя.

Дополнительным условием получения оценки «зачтено» могут стать успехи при выполнении лабораторных работ, систематическая активная работа на практических работах.

Оценка «зачтено» выставляется студенту, набравшему 8 и более баллов при промежуточной аттестации.

**Оценки «не зачтено»** заслуживает обучающийся, продемонстрировавший серьезные пробелы в знаниях основного материала изученной дисциплины, не ответивший на все вопросы билета и дополнительные вопросы. Как правило, оценка «не зачтено» ставится обучающимся, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной).

Оценка «не зачтено» выставляется студенту, набравшему менее 8 баллов при промежуточной аттестации.

На промежуточную аттестацию (экзамен) выносится тест, два теоретических вопроса и 2 задачи. Максимально студент может набрать 15 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

**Оценка «отлично»** выставляется студенту, который набрал в сумме 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «хорошо»** выставляется студенту, который набрал в сумме от 12 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием

является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «удовлетворительно»** выставляется студенту, который набрал в сумме от 8 до 11 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

**Оценка «неудовлетворительно»** выставляется студенту, который набрал в сумме менее 8 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

#### 4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

##### 4.1. Промежуточная аттестация

Коды компетенций	Результаты освоения ОПОП Содержание компетенций
ПК-1	Способен руководить процессом разработки, проверки работоспособности и интеграцией программного обеспечения
ПК-1.1	Осуществляет руководство разработкой программного обеспечения
ПК-1.2	Осуществляет руководство проверкой работоспособности программного обеспечения
ПК-1.3	Осуществляет руководство интеграцией программного обеспечения

##### а) типовые тестовые вопросы:

###### Вопрос 1

**Вычислительная система называется многопроцессорной, если:**

- а) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляется разными операционными системами;
- б) содержит несколько процессоров;
- +в) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляется одной общей операционной системой;
- г) она содержит несколько процессоров, работающих с отдельной оперативной памятью, и управляется одной общей операционной системой.

###### Вопрос 2

**Вычислительная система, которая содержит несколько ЭВМ, каждая из которых имеет свою ОП и работает под управлением своей операционной системы:**

- а) многопроцессорной;
- б) однородной;
- в) специализированной;
- +г) многомашинной.

###### Вопрос 3

**Вычислительная машина какой архитектуры позволяют выполнять одну арифметическую операцию сразу над многими данными — элементами вектора.**

- а) SISD;
- б) MISD;
- +в) SIMD;
- г) MIMD.

###### Вопрос 4

**В сильно связанных системах класса MIMD:**

- а) вся память распределена между процессорными элементами;
- +б) мультипроцессоры симметричные.
- в) существуют кластерные системы;
- +г) общая память данных и команд;

###### Вопрос 5

**В вычислительных системах с общей памятью...**

- а) значение, записанное в память одним из процессоров недоступно для другого процессора;
- б) память физически распределена по различным частям системы, но логически разделяема;
- +в) значение, записанное в память одним из процессоров, напрямую доступно для другого процессора;
- г) каждый процессор имеет свою локальную память с локальным адресным пространством.

#### **Вопрос 6**

*Под симметричным доступом к памяти понимается следующее (допускается множественный выбор):*

- +а) равные права всех процессоров на доступ к памяти;
- +б) одна и та же адресация для всех элементов памяти;
- в) разное время доступа всех процессоров системы;
- +г) равное время доступа всех процессоров системы;

#### **Вопрос 7**

*Какой уровень параллелизма осуществляется путем одновременного выполнения нескольких независимых заданий на разных процессорах, которые, практически, не взаимодействуя друг с другом. Этот уровень реализуется на многопроцессорных и многомашинных ВС.*

- а) уровень потоков;
- +б) уровень заданий;
- в) микроуровень;
- г) уровень команд

#### **Вопрос 8**

*Какой уровень параллелизма обеспечивается операционной системой.*

- а) среднезернистый параллелизм;
- +б) крупнозернистый параллелизм;
- в) мелкозернистый параллелизм;
- г) среднезернистый и крупнозернистый параллелизм.

#### **Вопрос 9**

*Отношение времени, затрачиваемого на проведение вычислений на однопроцессорной ВС, ко времени решения той же задачи на параллельной n-процессорной системе):*

- а) ускорение; +
- б) индекс параллелизма;
- в) эффективность;
- г) утилизация.

#### **Вопрос 10**

*Избыточность это ...*

- а) целесообразность наращивания числа процессоров через ту долю ускорения, достигнутого за счет параллельных вычислений, которая приходится на один процессор;
- б) средняя скорость параллельных вычислений, выраженная через количество выполненных операций;
- +в) отношение объема параллельных вычислений к объему эквивалентных последовательных вычислений;
- г) показатель эффективной скорости вычислений.

#### **Вопрос 11**

*Назовите преимущества SMP-систем.*

- +а) отказ одного из компонентов не ведет к отказу системы;
- +б) производительность;
- в) число процессоров в системе изменить нельзя;
- г) все перечисленное.

#### **Вопрос 12**

**Назовите основные причины появления систем с массовой параллельной обработкой.**

- +а) необходимость построения ВС с гигантской производительностью;
- б) стремление раздвинуть границы производства ВС в большом диапазоне производительности и стоимости;
- в) нет верного ответа.

### **Вопрос 13**

**По каким особенностям вычислительную систему причисляют к классу MPP?**

- а) система плохо масштабируется;
- +б) вычислительный узел обладает всеми средствами для независимого функционирования;
- в) один узел содержит сетевой адаптер, используемый для объединения остальных узлов.
- +г) работа системы координируется главной ВМ (хост-компьютером).

### **Вопрос 14**

**Что общего у кластерной вычислительной системы с MPP?**

- а) узлы кластерной системы — это не специализированные устройства, приспособленные под использование в ВС;
- +б) тот же принцип распределенной памяти;
- +в) большой потенциал для масштабирования системы;
- г) в единую систему могут объединяться узлы разного типа.

### **Вопрос 15**

**Благодаря чему группа взаимосвязанных узлов становится кластером?**

- а) благодаря межузловым связям;
- +б) благодаря связующему слою;
- в) благодаря сетевым адаптерам;
- г) благодаря подключению к шине.

### **Вопрос 16**

**С чем в современных условиях связан переход к активному распараллеливанию вычислений? Допускается множественный выбор.**

- +а) с увеличением выделяемой теплоты (рост температуры);
- б) со сложностью изготовления;
- +в) с нецелесообразностью наращивать потребление;
- +г) с нецелесообразностью увеличивать частоту.

### **Вопрос 17**

**Может ли программа быть полностью распараллеленной?**

- а) программа всегда не может быть распараллелена;
- +б) программа может быть распараллелена, но не полностью;
- в) программа всегда может быть распараллелена.

### **Вопрос 18**

**Что такое OpenMP?**

- а) технология организации параллельных вычислений в системах с распределённой памятью;
- +б) технология организации параллельных вычислений в системах с общей памятью;
- в) технология организации последовательных вычислений на графическом процессоре;
- г) технология организации последовательных вычислений на мобильном процессоре.

### **Вопрос 19**

**Какие преимущества даёт программисту использование OpenMP?**

- а) предлагает встроенную реализацию популярных паттернов;
- б) предлагает статический анализ кода;
- +в) предлагает модель инкрементального программирования;
- г) предлагает средства рефакторинга.

### **Вопрос 20**

**Каким образом осуществляется программирование с помощью OpenMP?**

- а) с помощью новых встроенных функций;
- +б) с помощью новых встроенных функций и директив;
- в) с помощью новых встроенных функций, директив и типов.

**Вопрос 21**

**Что такое поток? В чём заключается разница между master и slave потоками?**

- а) поток – это экземпляр программы, все потоки равны;
- б) поток – это экземпляр программы, master поток управляет slave потоками;
- в) поток – это обособленная копия набора команд вместе со своим стеком, все потоки равны;
- +г) поток – это обособленная копия набора команд вместе со своим стеком, master поток управляет slave потоками.

**Вопрос 22**

**Директива parallel в технологии OpenMP:**

- а) запускает цикл на нескольких параллельных потоках;
- +б) запускает код в нескольких параллельных потоках;
- в) обозначает критическую секцию;
- г) обозначает барьер синхронизации параллельных потоков.

**Вопрос 23**

**Какое количество одновременно работающих потоков в OpenMP устанавливается по умолчанию; можно установить максимальным?**

- а) оба равны количеству процессоров;
- б) равное количеству процессоров, не ограничено;
- в) оба равны количеству вычислительных ядер;
- +г) равное количеству вычислительных ядер, не ограничено;

**Вопрос 24**

**Выберите все объекты OpenMP, которые позволяют управлять настройками количества потоков.**

- а) опция threadprivate;
- +б) опция num\_threads;
- +в) функция omp\_set\_num\_thread;
- г) директива single.

**Вопрос 25**

**Чем модификатор firstprivate отличается от модификатора lastprivate?**

- а) firstprivate задаёт значение переменной только в потоке-мастере, а lastprivate – только в подчинённом потоке;
- +б) firstprivate задаёт значение переменной при входе в поток, а lastprivate – на выходе из потока;
- в) firstprivate обнуляет значение переменной только в потоке-мастере, а lastprivate – только в подчинённом потоке;
- г) ничем не отличаются;

**Вопрос 26**

**Выберите верные утверждения об опции reduction.**

- а) опция распараллеливает циклы;
- б) опция распараллеливает условные операторы;
- +в) опция выполняет указанную операцию на выходе из параллельной области;
- г) опция выполняет очистку переменных на выходе из параллельной области.

**Вопрос 27**

**Какие функции выполняет критическая секция?**

- +а) впускает не более одного потока одновременно;
- б) впускает только мастер поток;

- в) включает режим высокоточных вычислений;
- г) является параллельным аналогом try-catch.

### **Вопрос 28**

*Можно ли в OpenMP регулировать необходимость распараллеливания программы по условию?*

- а) нет, нельзя;
- б) нет, нельзя, реализовать вручную невозможно;
- в) да, можно, но нужно реализовать вручную;
- +г) да, можно, существуют встроенные средства.

### **Вопрос 29**

*Какие существуют способы настройки OpenMP вне директив?*

- а) иных способов не существует;
- б) иных встроенных способов не существует, но можно реализовать самому;
- +в) существуют встроенные функции;
- г) существуют встроенные классы.

### **Вопрос 30**

*Можно ли распараллелить цикл? Какой цикл? Какими способами? Выберите верные утверждения.*

- а) циклы распараллеливать нельзя;
- б) можно распараллелить любой цикл встроенными средствами;
- в) можно распараллелить любой цикл собственной реализацией;
- +г) можно распараллелить любой цикл собственной реализацией, если итерации цикла не зависят друг от друга;
- +д) можно распараллелить цикл for встроенными средствами.

### **Вопрос 31**

*Как работает барьер?*

- а) впускает только первый поток;
- б) впускает только мастер поток;
- в) впускает первый поток, ждёт завершения, впускает следующий;
- +г) ждёт, пока все потоки подойдут к нему, после впускает всех одновременно.

### **Вопрос 32**

*Каким образом осуществляется синхронизация памяти в OpenMP?*

- а) синхронизация памяти не предусмотрена;
- б) память синхронизируется автоматически;
- в) существует встроенная функция;
- +г) существует специальная директива.

### **Вопрос 33**

*Какое поведение определяет опция `copyin`?*

- а) копирует значения всех указанных переменных в параллельную область;
- +б) копирует значения всех `threadprivate` переменных в параллельную область со значениями из нити мастера;
- в) создаёт внутри параллельных областей `threadprivate` переменные со значениями по умолчанию;
- г) возвращает из параллельной области значения указанных переменных.

### **Вопрос 34**

*В чём сложность организации параллельных вычислений? Выберите все правильные варианты.*

- +а) циклы имеют зависимости между итерациями;
- б) невозможно создать локальные копии переменных;
- +в) потоки пытаются изменять одни и те же области памяти;

г) высокая стоимость создания потока. +

### Вопрос 35

*Всегда ли распараллеливание программ даёт выигрыш в скорости вычислений?*

- а) да, всегда даёт выигрыш;
- б) даёт выигрыш только для небольших задач;
- +в) даёт выигрыш только для больших задач;
- г) не даёт выигрыш.

### Вопрос 36

*Укажите основные отличия архитектур CPU и GPU.*

- +а) CPU заточены под скорость выполнения, GPU – под параллельное выполнение;
- б) ядра CPU проще и дешевле, ядра GPU – дороже и сложнее;
- +в) ядра CPU сложнее и больше в размерах, ядра GPU – простые, маленькие, заточены под параллельное выполнение;
- г) только размеры ядер.

### Вопрос 37

*Выберите верные утверждения о программной архитектуре CUDA.*

- +а) CUDA Host API – это функции, выполняющиеся только на CPU;
- б) программа компилируется и выполняется полностью на GPU;
- в) допускается одновременное использование Driver API и Runtime API;
- +г) Runtime API – это высокоуровневая реализация с помощью Driver API;

### Вопрос 38

*Что такое устройство в терминах CUDA?*

- а) ядро GPU;
- +б) графический ускоритель;
- в) устройство, управляющее процессом выполнения и компиляции;

### Вопрос 39

*Что такое хост?*

- а) ядро GPU;
- б) графический ускоритель;
- +в) устройство, управляющее процессом выполнения и компиляции;

### Вопрос 40

*Что такое ядро (kernel)?*

- а) ядро GPU;
- б) графический ускоритель;
- в) CPU;
- +г) программа на CUDA.

### Вопрос 41

*Что будет, если для выполнения ядра не будет достаточного количества вычислительных ресурсов?*

- а) произойдёт ошибка времени выполнения;
- б) ядра, на которых не хватило ресурсов, будут проигнорированы;
- +в) ядра, на которых не хватило ресурсов, будут поставлены в очередь.

### Вопрос 42

*Какие существуют ограничения на размеры гридов, блоков?*

- а) ограничений нет, это логические понятия;
- б) ограничения заданные статично и не изменяются в версиях архитектуры;
- +в) каждое поколение GPU имеет собственные ограничения.

### Вопрос 43

*Почему ограничения на размер грида/блока важно учитывать при разработке программ?*

- а) потому что превышение установленного уровня может вызвать отказ устройства;
- б) потому что превышение установленного уровня на порядок снижает производительность GPU;

+в) потому что превышение установленного уровня может приводить к небольшому снижению производительности и ошибкам устройства.

### Вопрос 44

*Выберите все основные объекты программного интерфейса CUDA.*

- +а) встроенные типы;
- +б) встроенные функции;
- +в) встроенные переменные;
- г) встроенные директивы;
- +д) атрибуты переменных;
- +е) атрибуты функций;
- ж) атрибуты директив.

### Вопрос 45

*Выберите верные утверждения касательно модификаторов `__global__`, `__host__`, `__device__`.*

- +а) `__global__` - модификатор функции, делающий её ядром;
- б) `__global__` - модификатор переменной, делающий её видимой на устройстве и хосте;
- +в) `__device__` может быть как модификатором функции, так и модификатором переменной;
- г) функция с модификатором `__host__` компилируется на GPU, а выполняется на CPU;
- +д) функция с модификатором `__device__` выполняется на GPU и может вызываться там же.

### Вопрос 46

*Выберите участок кода, который определяет и запускает ядро.*

- а) `void Kernel(...) { } Kernel(param1, ..., paramN);`
- б) `__device__ void Kernel(...) { } Kernel(param1, ..., paramN);`
- +в) `__global__ void Kernel(...) { } Kernel<<<k, l>>>(param1, ..., paramN);`
- г) `__device__ int Kernel(...) { } Kernel<<<k, l>>>(param1, ..., paramN);`

### Вопрос 47

*Какие параметры используются при запуске ядра?*

- а) `grid`, `block` – количество блоков в гриде и потоков в блоке;
- +б) `grid`, `block`, `mem`, `stream` - количество блоков в гриде, потоков в блоке, объём дополнительной `shared`-памяти, поток вызова;
- в) `mem`, `stream` - объём дополнительной `shared`-памяти и поток вызова;
- г) `dev`, `mem`, `block` – номер устройства, количество выделяемой памяти на блок, количество потоков в блоке.

### Вопрос 48

*Выберите необходимые шаги, которые должны быть выполнены любой программой, выполняющей расчёты на GPU с помощью CUDA.*

- +а) выделение памяти на устройстве;
- б) выделение памяти на хосте;
- +в) копирование данных на устройство;
- +г) копирование данных на хост;
- +д) вызов ядра;
- +е) проверка наличия ошибок выполнения;
- +ж) освобождение памяти на устройстве;
- з) освобождение памяти на хосте.

### Вопрос 49

Какой из перечисленных способов распараллеливания наиболее оптимальный в большинстве случаев?

- а) по гридам
- б) по блокам;
- в) по тредам;
- +г) по блокам и тредам;
- д) всё вместе.

### Вопрос 50

Какие существуют модификаторы переменных в CUDA?

- +а) `__device__` - переменная находится на устройстве;
- б) `__host__` - переменная находится на хосте;
- +в) `__const__` - переменная является константой;
- +г) `__shared__` - переменная разделена между тредами в блоке;
- д) `__local__` - переменная является локальной у треда.

### Вопрос 51

Какая операция не является операцией по работе с глобальной памятью CUDA.

- а) `cudaMalloc`;
- б) `cudaFree`;
- в) `cudaMemcpy`;
- +г) `cudaMemclr`.

### б) типовые практические задания:

Выполняются 2 задания одно с использованием технологии OpenMP, второе – с использованием технологии CUDA.

#### Технология OpenMP

1) В параллельной области программы вывести номер исполняющейся в данный момент нити.

2) С помощью опции `reduction` произвести суммирование элементов вектора, длина которого равна количеству нитей, создаваемых при старте параллельной области (значение по умолчанию либо устанавливаемое вами).

3) Создать вектор из четырёх элементов. Заполнить его случайными значениями. С помощью директивы `firstprivate` передать его в параллельную область. Вывести на экран элемент по индексу, равному номеру исполняющейся нити. Заменить значение этого элемента номером исполняющейся нити. С помощью `lastprivate` передать изменённый вектор в последовательную область. Показать на экране элементы изменённого вектора.

4) Произвести параллельное суммирование значений строк матрицы. Полученные значения записать в вектор.

#### Технология CUDA

1) Создать матрицу размера  $[10 \times 10]$ . В первый элемент каждой строки записать номер исполняющегося блока, остальные элементы равномерно заполнить числами от 0 до 8.

2) Произвести параллельное суммирование значений строк матрицы. Полученные значения записать в вектор.

3) Создать вектор из тысячи целых чисел, заполнить его значениями  $[1; 10]$ . Параллельно посчитать гистограмму распределений чисел. В решении использовать буферный вектор из десяти элементов.

4) Распараллелить любой простой расчёт в цикле `while` при условии, что все итерации цикла независимы друг от друга.

5) Произвести параллельное суммирование элементов матрицы  $[100 \times 100]$ .