

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

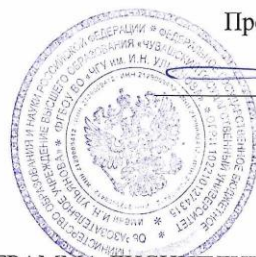
Кафедра компьютерных технологий

«УТВЕРЖДАЮ»

Проректор по учебной работе

И.Е. Поверинов

«31» августа 2017 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
«АРХИТЕКТУРА ЭВМ»

Направление подготовки (специальность) 09.03.04 Программная инженерия

Квалификация (степень) выпускника – бакалавр

Профиль (направленность) Управление разработкой программных проектов

Прикладной бакалавриат

Чебоксары – 2017

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 «Программная инженерия», утвержденного приказом Министерства образования и науки Российской Федерации № 229 от 12.03.2015 г.

*СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):*

доцент, к.т.н.

 П.В. Желтов

старший преподаватель

 Е.А. Мытникова

*ОБСУЖДЕНО:*

на заседании кафедры компьютерных технологий « 30 » 08 2017г., протокол № 1

заведующий кафедрой

 Т.А. Лавина

*СОГЛАСОВАНО:*

Методическая комиссия факультета информатики и вычислительной техники  
« 30 » 08 2017г., протокол № 1

Декан факультета

 А.В. Щипцова

Директор научной библиотеки

 Н. Д. Никитина

Начальник управления информатизации

 И. П. Пивоваров

Начальник учебно-методического управления

 В. И. Маколов

## Оглавление

<b>1. Цель и задачи обучения по дисциплине</b> .....	4
<b>2. Место дисциплины в структуре основной образовательной программы (ООП)</b> .....	4
<b>3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП</b> .....	4
<b>4. Структура и содержание дисциплины</b> .....	4
4.1. Содержание дисциплины .....	5
4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения .....	6
<b>5. Содержание разделов дисциплины</b> .....	7
5.1. Лекции .....	7
5.2. Лабораторные работы .....	13
5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины .....	13
<b>6. Образовательные технологии</b> .....	14
<b>7. Формы аттестации и оценочные материалы</b> .....	15
7.1. Вопросы к зачету .....	15
7.2. Вопросы и задачи к экзамену .....	17
7.3. Выполнение и примерная тематика курсового проекта .....	19
7.4. Выполнение и примерные задания расчетно-графической работы .....	19
7.5. Выполнение и примерная тематика контрольной работы .....	19
<b>8. Учебно-методическое и информационное обеспечение дисциплины</b> .....	19
8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе) .....	20
8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе).....	20
8.3. Программное обеспечение, профессиональные базы данных, информационно-справочные системы. ....	20
8.4. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы .....	20
<b>9. Материально-техническое обеспечение дисциплины</b> .....	21
<b>10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями</b> .....	21
<b>11. Методические рекомендации по освоению дисциплины</b> .....	21

## 1. Цель и задачи обучения по дисциплине

Целью изучения дисциплины является усвоение базовой совокупности знаний о принципах организации современных вычислительных систем, на основе которой студенты могли бы самостоятельно оценивать возможности различных вычислительных систем, принимать решения о выборе типа вычислительной системы и особенностях разработки программного обеспечения в зависимости от класса решаемых задач.

Задачи изучения дисциплины:

- ознакомление студентов с основными принципами организации и функционирования вычислительных систем; важнейшими этапами и тенденциями в развитии вычислительных систем; методами оценки параметров компонент и систем в целом.

## 2. Место дисциплины в структуре основной образовательной программы (ООП)

Блок дисциплин – Б1.В Дисциплины (модули). Вариативная часть

Изучение данной дисциплины базируется на курсах «Информатика и программирование», «Организация ЭВМ и систем», «Основы программирования».

Последующие дисциплины и практики, для успешного изучения которых необходима данная дисциплина: «Преддипломная практика (практика для выполнения выпускной квалификационной работы)», «Выпускная квалификационная работа».

## 3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП

Процесс обучения по дисциплине направлен на формирование следующих компетенций:

профессиональных (ПК), соответствующими виду (видам) профессиональной деятельности, на который (которые) ориентирована ООП:

- владением навыками использования различных технологий разработки программного обеспечения (ПК-3);

В результате обучения по дисциплине, обучающийся должен (ЗУН):

**знать:**

- 31 - методы представления числовой информации в вычислительных системах;
- 32 – методы обработки числовой информации в вычислительных системах;
- 33 - методы обмена информацией между компонентами вычислительных систем;

**уметь:**

- У1 – оценивать компьютерные архитектуры вычислительных систем в точки зрения комплексных критериев качества;
- У2 - планировать эксперимент, проводить экспериментальное исследование с помощью натуральных или имитационных моделей вычислительных систем;

**владеть навыками:**

- Н1 - анализа потоков данных в вычислителях различной архитектуры;
- Н2 – замера времени в программных реализациях алгоритмов;
- Н3 - технологий разработки программ для заданных архитектур вычислителей.

## 4. Структура и содержание дисциплины

Образовательная деятельность по дисциплине проводится:

- в форме контактной работы обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (далее – контактная работа);
- в форме самостоятельной работы.

Контактная работа включает в себя занятия лекционного типа, занятия семинарского типа (лабораторные работы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, п/р – практические занятия, КСР – контроль самостоятельной работы, СРС – самостоятельная работа студента, ИФР – интерактивная форма работы, К – контроль.

#### 4.1. Содержание дисциплины

Содержание	Формируемые компетенции	Формируемые ЗУН
Раздел 1. Архитектура вычислительных систем	ПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 31 - методы представления числовой информации в вычислительных системах;</li> <li>- 32 – методы обработки числовой информации в вычислительных системах;</li> <li>- 33 - методы обмена информацией между компонентами вычислительных систем;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У1 – оценивать компьютерные архитектуры вычислительных систем в точки зрения комплексных критериев качества;</li> <li>- У2 - планировать эксперимент, проводить экспериментальное исследование с помощью натуральных или имитационных моделей вычислительных систем;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н1 - анализа потоков данных в вычислителях различной архитектуры;</li> <li>- Н2 – замера времени в программных реализациях алгоритмов;</li> <li>- Н3 - технологий разработки программ для заданных архитектур вычислителей.</li> </ul>
1.1. История машинного счёта		
1.2. Определения понятия "Архитектура" в применении к вычислительным системам		
1.3. Уровни управления процессом вычислений. Архитектура процессоров x86 фирмы Intel		
1.4. Общие требования к программному коду. Поточковые (DATA-FLOW) вычислители. Машинные команды, язык программирования Ассемблер		
1.5. Недостаток процесса вычислений в позиционной системе счисления и альтернативные решения. Простейшие программы на Ассемблере - простые типы и описания данных		
1.6. Архитектура параллельных вычислительных систем. Ассемблер - целочисленная арифметика		
1.7. Суперкомпьютеры. Ассемблер - программирование условных и безусловных переходов		
1.8. Нейронные сети и нейрокомпьютеры		
1.9. Ассемблер - практика разработки программ. Интерактивные среды разработки. Макроассемблер. Процессор x86 и программирование под DOS. Ассемблер для 32-х и 64-бит архитектур. Поддержка чисел с плавающей запятой		
1.10. Транспьютеры. Отладка программ на языке Ассемблер		
1.11. Метакомпьютинг и концепция GRID. Использование ассемблерных программ при программировании на языках высокого уровня		
1.12. Архитектура GPU фирмы NVIDIA и технология CUDA		
1.13. Аналоговые вычислительные системы		
1.14. Вычислители с программи-		

руемой архитектурой, пути усовершенствования архитектуры		
1.15. Квантовые вычислители и системы передачи данных		
1.16. Архитектура систем поиска информации в сети InterNet		
Зачеты/Экзамены	ПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 31 - методы представления числовой информации в вычислительных системах;</li> <li>- 32 – методы обработки числовой информации в вычислительных системах;</li> <li>- 33 - методы обмена информацией между компонентами вычислительных систем;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У1 – оценивать компьютерные архитектуры вычислительных систем в точки зрения комплексных критериев качества;</li> <li>- У2 - планировать эксперимент, проводить экспериментальное исследование с помощью натуральных или имитационных моделей вычислительных систем;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н1 - анализа потоков данных в вычислителях различной архитектуры;</li> <li>- Н2 – замера времени в программных реализациях алгоритмов;</li> <li>- Н3 - технологий разработки программ для заданных архитектур вычислителей.</li> </ul>

#### 4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения

Содержание	Всего, час	Контактная работа, час				СРС, час	ИФР, час	К, час
		Л	л/р	п/р	КСР			
Раздел 1. Архитектура вычислительных систем								
1.1. История машинного счёта	8	4	4				4	
1.2. Определения понятия "Архитектура" в применении к вычислительным системам	8	4	4				4	
1.3. Уровни управления процессом вычислений. Архитектура процессоров x86 фирмы Intel	23	4	4			15	4	
1.4. Общие требования к программному коду. Поточные (DATA-FLOW) вычислители. Машинные команды, язык программирования Ассемблер	23	4	4			15	4	
1.5. Недостаток процесса вычислений в позиционной системе счисления и альтернативные ре-	20	4	8			8	8	

шения. Простейшие программы на Ассемблере - простые типы и описания данных								
1.6. Архитектура параллельных вычислительных систем. Ассемблер - целочисленная арифметика	20	4	8			8	8	
1.7. Суперкомпьютеры. Ассемблер - программирование условных и безусловных переходов	20	4	8			8	8	
1.8. Нейронные сети и нейрокомпьютеры	21	4	8			9	8	
1.9. Ассемблер - практика разработки программ. Интерактивные среды разработки. Макроассемблер. Процессор x86 и программирование под DOS. Ассемблер для 32-х и 64-бит архитектур. Поддержка чисел с плавающей запятой	17	4	4			9	4	
1.10. Транспьютеры. Отладка программ на языке Ассемблер	18	4	4			10	4	
1.11. Метакомпьютинг и концепция GRID. Использование ассемблерных программ при программировании на языках высокого уровня	18	4	4			10	4	
1.12. Архитектура GPU фирмы NVIDIA и технология CUDA	18	4	4			10	4	
1.13. Аналоговые вычислительные системы	13	2	6			5	6	
1.14. Вычислители с программируемой архитектурой, пути совершенствования архитектуры	13	2	6			5	6	
1.15. Квантовые вычислители и системы передачи данных	13	2	6			5	6	
1.16. Архитектура систем поиска информации в сети InterNet	13	2	6			5	6	
Зачет(1)	10					10		
Экзамен(1)	36							36
Зачет(2)	10					10		
Экзамен(2)	38				2			36
<b>Итого</b>	<b>360</b>	<b>56</b>	<b>88</b>		<b>2</b>	<b>142</b>	<b>88</b>	<b>72</b>
<b>Зачетных единиц</b>	<b>10</b>							

## 5. Содержание разделов дисциплины

### 5.1. Лекции

#### Раздел 1. Архитектура вычислительных систем

##### *Тема 1.1. История машинного счёта*

Предпосылки необходимости счёта. Мысли великих людей по поводу счёта. От пальцевого счёта к механизации вычислений. Примитивные орудия вычислений. Абак. Барон Джон Непери его "счёт на палочках". Блез Паскаль и "Паскалина". Лейбниц и его арифмометр. Чарльз Беббидж, его механические вычислители и фактическое рождение

понятия "Архитектура" в применении к вычислительным системам. Леди Байрон-Лавлейс как первая программистка. Гипотетическая машина Тьюринга. Фон-Неймановские принципы построения процессоров. Гарвардская и Принстонская архитектуры. Электромеханические и электронные вычислительные машины. "Из ряда вон выходящие" вычислительные системы (вычислители с трёхзначной логикой, вычислители на основе арифметики остаточных классов). Суперкомпьютеры и задачи класса GRAND CHALLENGES.

### *Тема 1.2. Определения понятия "Архитектура" в применении к вычислительным системам*

Общее и разница между понятиями структура и архитектура. Современные определения понятия архитектуры вычислительных систем. Машина Тьюринга как классика архитектуры вычислителей. Пять принципов фон-Неймана построения вычислителей. Классическая фон-Неймановская (Принстонская) архитектура, Гарвардская архитектура, их сравнительные преимущества и недостатки. Основные архитектуры многопроцессорных и многокомпьютерных вычислительных систем. Сравнительные достоинства и недостатки SMP- и MPP-архитектур. Классификации архитектур вычислителей. Классификация Флинна. Методы управления процессом вычислений. Понятие регистра - счётчика команд (Set Counter). Форматы машинных команд. Методы повышения производительности процессоров. CISC- и RISC-процессоры. Конвейерная и суперскалярная обработка данных. Истинный параллелизм.

### *Тема 1.3. Уровни управления процессом вычислений*

Управление последовательностью вычислений. Процессор как синтез операционного и управляющего автоматов. Системы счисления современных процессоров. Выбор рационального основания позиционной системы счисления и форматов представления чисел в ЭВМ. Точность представления чисел. Стандарт IEEE 754 для представления вещественных чисел. Выполнение арифметических действий над числами с фиксированной запятой, проблемы "размножения ошибки" вследствие переноса значений битов и невозможности распараллеливания на битовом уровне. Использование конвейерной архитектуры для повышения производительности процессора. Особенности функционирования конвейера.

Классика - архитектура процессора Intel x86. История разработки, внутренние регистры процессора, их разрядность, обозначение и назначение. Связь с внешними устройствами. Принцип определения последовательности исполнения машинных команд.

### *Тема 1.4. Общие требования к программному коду. Потокосые (DATA-FLOW) вычислители*

Условия корректного выполнения программного кода. Принципы программного управления последовательностью выполнения операций (CONTROL-FLOW) и управления порядком выполнения операций самими данными (DATA-FLOW). Отрицательная роль регистра - счётчика команд на возможность распараллеливания вычислений. Исторические попытки модернизации классической фон-Неймановской архитектуры. Принципиальная возможность распараллеливания процесса вычислений по произвольному алгоритму без априорного указания последовательности действий. Использование ЯПФ (Ярусно-Параллельной Формы) информационного графа алгоритма с целью выявления параллельно исполняемых блоков (гранул) программы. Структурная схема вычислителя с управлением последовательностью вычислений потоком данных. Реализации потокосых вычислителей. Проблема ассоциативной памяти, особенности программирования DATA-FLOW машин. Компьютерная модель (симулятор) потокосого вычислителя и её использование для моделирования и оптимизации процесса вычислений. Понятие интенсивности вычислений и возможность целенаправленного управления ею. Кумулятивная кривая количества исполненных операций.



Язык Ассемблер и машинные команды. Компилятор ассемблера, макроассемблер. Отличие уровня программирования на ассемблере от программирования непосредственно в командах процессора. Машинные команды - основные группы, синтаксис описания, адресность команд, время выполнения. Понятие микропрограммирования.

*Тема 1.5. Недостаток процесса вычислений в позиционной системе счисления и альтернативные решения*

Принцип поразрядного последовательного выполнения булевых операций при использовании ПСС (Позиционной Системы Счисления). Распространение ошибки булевых операций от младших к старшим разрядам вследствие "переноса в старший разряд", затруднение выявления ошибки выполнения арифметического действия вследствие этого. Невозможность распараллеливания собственно арифметической операции в ПСС. Методы ускорения вычислений в рамках ПСС. История непозиционных систем счисления. Китайская теорема об остатках. Теория вычетов. Модулярная алгебра. История применения СОК (Системы Остаточных Классов) при разработке арифметических устройств, персоналии. Архитектура арифметических устройств на основе СОК. Табличный метод определения результатов арифметической операции по заданному основанию vs вычислительный метод, условия выбора одного из этих методов. Реальные ЭВМ, использующие вычисления на основе СОК. Преимущества и недостатки арифметических устройств на основе СОК.

Программное обеспечение разработки программ на Ассемблере. Интегрированная среда emu8086, её возможности. Макроассемблеры. Секции программы на ассемблере - описание формата исполняемого файла, данных, собственно ассемблерный код. Знаковые и беззнаковые целые числа. Возможности структурирования программ на ассемблере - макросы, процедуры. Использование регистров процессора и стека при использовании процедур. Простейшая программа на ассемблере, последовательность компиляции ассемблерной программы в машинный код.

*Тема 1.6. Архитектура параллельных вычислительных систем*

Цель параллелизации обработки информации. Ускорение вычислений vs надёжность. Физические ограничения повышения производительности процессоров на едином кристалле. Зависимость тепловыделения процессора от его тактовой частоты, закон Рэлея. Доказательство возможности полного распараллеливания вычислений для конкретного алгоритма. Ярусно-Параллельная Форма (ЯПФ) информационного графа алгоритма. Параллелизация внешняя и истинная. Технологии параллелизации. Конвейерный принцип. Векторные процессоры. Контр-фон-Неймановские архитектуры вычислителей. Типы параллелизаций - параллелизация вычислений и параллелизация по данным. Абстрактные модели параллельных вычислений. Концепция неограниченного параллелизма. Понятие тонкой информационной структуры алгоритма. Формальное определение гранулы (зерна, блока) параллелизации. Глубина распараллеливания.

Целочисленная арифметика при программировании на ассемблере. Синтаксис команд целочисленной арифметики. Анализ данных и трассировка ассемблерной программы в эмуляторе emu8086. Анализ выполнения машинной команды с помощью регистра состояния процессора, битовые флаги состояния.

*Тема 1.7. Суперкомпьютеры*

Определение суперкомпьютера. Психоэмоциональное состояние "механетикс" (шутл.). Задачи и области применения суперкомпьютеров. Проблемы класса GRAND CHALLENGES (ЗАДАЧИ БОЛЬШОГО ВЫЗОВА). Методы определения производительности суперкомпьютеров, требования к методам. История - вычислительные смеси Гибсона. Тесты LinPACK, HPL (High-Performance Linpack benchmark). Диапазон производительности современных су-

перкомпьютеров. Вычислительные кластеры. Реальное и пиковое быстродействие. Проблемы пета- и эксафлопса. Энергетическая стоимость одной арифметической операции. Оценка погрешности вычислений в зависимости от точности представления данных и числа выполненных операций. Обоснование использования вещественной арифметики двойной точности (IEEE 754) при суперкомпьютерных вычислениях. Топологии коммуникационных сред суперкомпьютеров. Концепция неограниченного параллелизма. Закон Амдала и сетевой закон Амдала.

Принципы организации ветвлений при программировании на Ассемблере. анализ конкретных битовых флагов регистра состояния и команды условного перехода. Реализации "ближних" и "дальних переходов", дополнительные возможности реализации циклов.

#### *Тема 1.8. Нейронные сети и нейрокомпьютеры*

Биологический нейрон. Дерево входов (дендриты), выход (аксон), понятия возбуждения и торможения нейрона. Связь нейронов в коре головного мозга. Искусственные нейроноподобные структуры. Нейронная сеть (НС). Функция активации нейрона. Множества входов и выходов НС. Перцептрон. Понятие "решить задачу" в применении к НС. Обучение нейронной сети. Нейронные сети Кохонена, Хопфилда. Когнитрон. НС с обратным распространением информации. Обучающая, тестовая, рабочая последовательности. Признак окончания обучения НС. Одно- и многослойные НС. Процедуры обучения НС. Метод обратного распространения ошибки (back propagation), условие применимости этого метода. Самообучающиеся НС. Программные реализации НС. Аппаратная реализация НС. Применения НС. Нейронная сеть как решатель, функционирующий на основе теории нечёткой логики.

#### *Тема 1.9. Зависимость производительности вычислительного кластера MPP-архитектуры от параметров оборудования и решаемой задачи*

Вычислительный кластер как представитель многопроцессорных вычислительных систем

(МВС) архитектуры MPP. Вычислительные кластеры BEOWULF. МВС как симбиоз вычислительных узлов (ВУ) и коммуникационной сети (КС). Потоки данных в МВС. Необходимость синхронизации времени отдельных ВУ кластера. Задержка данных при обмене информацией через КС как источник простоя ВУ. Латентность (инерционность) сети передачи данных. Понятие тонкой информационной структуры программы и гранулы (зёрна, блока) параллелизма. Коэффициент гранулярности. технологии разработки параллельных программ. Параллельное программирование с использованием передачи сообщений MPI (Messages Passing Interface). Многозначность параллельных реализаций единого математического алгоритма. Пример - ленточный алгоритм параллельного умножения матриц классическим способом. Системные скрипты компиляции и запуска на исполнение MPI-программ. Определение зависимости времени обмена по компьютерной сети и производительности сети от размера сообщения (при обменах типа "точка - точка"). Исследование зависимости производительности МВС от размера обрабатываемых данных и количества вычислительных узлов. Верификация кубической зависимости времени выполнения параллельной программы от размерности данных. Интерпретация результатов.

Разница между форматами исполняемых файлов для 16, 32 и 64-х битовых архитектур. программные сегменты и их реализация для 16-ти битовых процессоров Intel и для "плоской" (flat) организации памяти. Реализация многозадачности, разграничение доступа к памяти. Двоично-десятичная арифметика как способ представления чисел с фиксированной точкой. Арифметический сопроцессор действий с плавающей запятой - его организация, машинные команды.

#### *Тема 1.10. Транспьютеры*

Предпосылки перехода к распределённой архитектуре вычислений. Энтони Хоар как инициатор разработки транспьютерной архитектуры. Транспьютер как специализированный процессор с большим количеством каналов связи с подобными. Роль компании Inmos в разработке первых транспьютеров. Архитектура транспьютера T805 фирмы Inmos. Топологии объединения транспьютеров. Языки программирования для транспьютеров. Современное состояние транспьютерной индустрии. Транспьютеры как элементы реализации архитектур систолических массивов, вычислителей с реконфигурируемой структурой.

Отладка ассемблерных программ. Режим эмуляции и пошагового исполнения в интегрированных средах разработки программ на Ассемблере. Повышение надёжности ассемблерпрограммирования путём использования готовых библиотек. Частые ошибки при программировании на ассемблере - некорректность работы со стеком, неверные манипуляции со счётчиком команд, "затирание" регистров процессора. Полезность применения известных схем (шаблонов) работы с подпрограммами и тщательного документирования (в т.ч. комментирования) ассемблерных программ.

### *Тема 1.11. Метакомпьютинг и концепция GRID*

Понятия метакомпьютера и метакомпьютинга. Вычислительная сеть (GRID). Архитектура метакомпьютера. Отличия метакомпьютера от традиционного компьютера. Понятие добровольных вычислений. Инструментальные системы организации и управления метакомпьютинга. Реализации метакомпьютерных вычислительных систем. Облачные вычисления как частный случай метакомпьютинга. Концепция доверительных отношений между заказчиком и фирмой, предоставляющей услугу облачного сервиса. Облачный сервис с точки зрения конечного пользователя. Доводы за и против облачных технологий. Ричард Столлман против использования проприетарных программ и ресурсов.

Принцип "90/10 - 90% времени тратится на выполнение всего 10% кода, причём обычно даже на меньшую часть кода". Определение критических по времени выполнения участков программы методом анализа исходного кода на языках программирования высокого уровня и вставка ассемблер-фрагментов с помощью ключевого слова `asm`. Дизассемблирование исполняемого файла программы с дальнейшим выявлением неоптимально скомпилированных участков кода и заменой его ассемблерного представления.

### *Тема 1.12. Архитектура GPU фирмы NVIDIA и технология CUDA*

История совершенствования графических карт. Шейдерный механизм вычислений как основа использования карт в качестве графических процессоров (GPU - Graphics Processing Unit). GPU как процессор массового параллелизма архитектуры SIMD. Графические процессоры фирм NVIDIA и AMD. направление GPGPU (General-Purpose computing on Graphics Processing Units). Поточковый процессор, вычислительные ядра потокового процессора. Программирование графических процессоров. Модель программирования в CUDA (Compute Unified Device Architecture). Понятие устройства (device), центрального процессора (host), потока (нити, `ththread`). Компилятор NVCC. Значение Compute Capability как показатель вычислительных возможностей GPU. Принципиальные различия между нитями GPU и нитями CPU. Бесплатность SDK CUDA и уровень поддержки пользователей и разработчиков фирмой NVIDIA. Типовой шаблон работы с GPU на языках высокого уровня. Поддерживаемые языки высокого уровня с CUDA. Программирование с использованием низкоуровневого CUDA driver API и высокоуровневого CUDA runtime API. Идентификация вычислительных ядер и распределение блоков данных для обработки на конкретных ядрах. Типовые примеры (язык C++ интегрированной среды Microsoft Visual Studio) программ для CUDA. Экспериментальная оценка производительности GPU vs CPU.

### *Тема 1.13. Аналоговые вычислительные системы*

Аналоговый и цифровой способы представления и хранения информации. История механических аналоговых вычислителей. Аналоговая вычислительная машина (АВМ) - механический дифференциальный анализатор (Уильям Томсон, лорд Кельвин). Классификация АВМ по используемому рабочему телу. Интегрирование (суммирование) на гидрорегистраторах. Нейронные сети как аналоговые вычислители. Архитектура аналоговых вычислителей. Базовые элементы электронных АВМ. Операционный усилитель как основа электронной АВМ. Устройства ввода и вывода АВМ. Модели аналоговых вычислителей и их функциональные возможности. Метод электрогидродинамических аналогий (ЭГДА) и его применение к решению уравнений Лапласа. Области применения АВМ. Достоинства и недостатки АВМ. Гибридные вычислительные системы, преимущества перед цифровыми вычислительными системами. Аналогоцифровые (АЦП) и цифро-аналоговые (ЦАП) преобразователи.

*Тема 1.14. Вычислители с программируемой архитектурой. Пути совершенствования архитектур вычислителей*

Предпосылки разработки вычислительных систем с программируемой архитектурой. Реконфигурация как средство создания вычислительной структуры, максимально эффективной для заданного алгоритма. Связь реконфигурируемых вычислительных систем с идеей клеточных автоматов фон-Неймана (1948). Понятие процессорного элемента (ПЭ), универсальной коммутационной среды (УКС). Два этапа настройки реконфигурируемых вычислительных систем на выполнение конкретной задачи. Программируемые логические интегральные схемы (ПЛИС) как основа создания реконфигурируемых систем. Эффективность систем с программируемой архитектурой. Перспективные пути совершенствования архитектур вычислителей - переход к принципу перемещения кода к данным (а не наоборот, как принято сейчас), обработка данных по мере их готовности (поточковый принцип), теговая память, домены санкционированного доступа в память, реализация односторонней памяти.

*Тема 1.15. Квантовые вычислители и системы передачи данных*

Определение квантового компьютера, история идеи. Принципиальная возможность сверхбыстрого выполнения вычислений. Понятие кубита, квантовой связанности (спутанности). Число линейно независимых состояний системы  $N$  кубитов. Физическая реализация кубитов. Квантовые проблемы задания исходных данных и считывания результата. Фундаментально вероятностный характер квантовых вычислений. Известные алгоритмы для квантовых компьютеров (алгоритмы Гровера, Залки-Визнера, Шора). Современные реализации квантовых вычислителей. Неэффективность известных методов шифрования при использовании квантового компьютера для дешифрации. Недостатки современных линий передачи информации. Использование квантовых эффектов для детектирования перехвата данных копированием.

*Тема 1.16. Архитектура систем поиска информации в сети InterNet*

Краткая история и современные параметры сети InterNet. Проблемы и технологии поиска информации в сети InterNet. Технологии Data Mining. Классификация поисковых систем в сети InterNet. Каталоги и специализированные базы данных сетевых ресурсов. Активные поисковые системы. Закон Зипфа (George Kingsley Zipf) в применении к системам поиска информации. Архитектура и принципы функционирования машин поиска информации в сети InterNet. Принципы ранжирования информации в поисковых системах. Понятие релевантности. Поисковая машина Google. История развития компании Google. Отличия принципов ранжирования и поиска информации Google существующих. Архитектура аппаратной части поисковых машин Google. Принцип построения серверов Google – простота и дублирование.

## 5.2. Лабораторные работы

Примерный перечень тем лабораторных работ:

1. Углубленное изучение специальных вопросов по темам лекций (с использованием информационных материалов преподавателя и/или ресурсов сети Интернет).
2. Практика подготовки, отладки и выполнения программ на языке Ассемблер для процессоров семейства x86.
3. Практика подготовки, отладки и выполнения CUDA-программ (Microsoft Visual Studio2010, CUDA SDK ver. 5.0).
4. Подготовка, отладка и выполнение параллельных программ в технологии MPI (Microsoft Visual Studio 2010, MPICH) в режиме удалённого доступа.
5. Ознакомление с практикой работы на программном симуляторе потокового (DATA-FLOW) вычислителя, подготовка простейших программ, их отладка и выполнение в режиме интерпретации; импорт данных с целью графической интерпретации результатов.

На лабораторных работах языком программной реализации алгоритмов является язык программирования C++, в связи с этим на занятиях повторяются основы языка C++ в объёме, необходимом для выполнения практических заданий. Дополнительно изучается язык программирования Ассемблер в версии процессоров семейства x86 фирмы Intel.

## 5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины

1. Конвейерная и суперскалярная обработка данных.
2. Истинный параллелизм.
3. Понятие микропрограммирования.
4. Верификация кубичной зависимости времени выполнения параллельной программы от размерности данных.
5. Арифметический сопроцессор действий с плавающей запятой - его организация, машинные команды.
6. Транспьютеры как элементы реализации архитектур систолических массивов, вычислителей с реконфигурируемой структурой.
7. Полезность применения известных схем (шаблонов) работы с подпрограммами и тщательного документирования (в т.ч. комментирования) ассемблерных программ.
8. Дизассемблирование исполняемого файла программы с дальнейшим выявлением неоптимально скомпилированных участков кода и заменой его ассемблерного представления.
9. Компилятор NVCC.
10. Значение Compute Capability как показатель вычислительных возможностей GPU.
11. Принципиальные различия между нитями GPU и нитями CPU.
12. Бесплатность SDK CUDA и уровень поддержки пользователей и разработчиков фирмой NVIDIA.
13. Типовой шаблон работы с GPU на языках высокого уровня.
14. Поддерживаемые языки высокого уровня с CUDA.
15. Реальные ЭВМ, использующие вычисления на основе СОК.
16. Преимущества и недостатки арифметических устройств на основе СОК.
17. Использование регистров процессора и стека при использовании процедур.
18. Абстрактные модели параллельных вычислений.
19. Концепция неограниченного параллелизма.
20. Понятие тонкой информационной структуры алгоритма.

21. Формальное определение гранулы (зерна, блока) параллелизации.
22. Глубина распараллеливания.
23. Топологии коммуникационных сред суперкомпьютеров.
24. Концепция неограниченного параллелизма.
25. Закон Амдаля и сетевой закон Амдаля.
26. Применения НС.
27. Нейронная сеть как решатель, функционирующий на основе теории нечёткой логики.
28. Метод электрогидродинамических аналогий (ЭГДА) и его применение к решению уравнений Лапласа.
29. Аналогоцифровые (АЦП) и цифро-аналоговые (ЦАП) преобразователи.
30. Неэффективность известных методов шифрования при использовании квантового компьютера для дешифрации.
31. Недостатки современных линий передачи информации.
32. Использование квантовых эффектов для детектирования перехвата данных копированием.
33. Принцип построения серверов Google – простота и дублирование.

## 6. Образовательные технологии

В соответствии со структурой образовательного процесса по дисциплине применяются следующие технологии:

- диагностики;
- целеполагания;
- управления процессом освоения учебной информации;
- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, научно-исследовательской, частично-поисковой, репродуктивной, творческой и пр.);

В соответствии с требованиями ФГОС ВО для реализации компетентного подхода при обучении дисциплине предусмотрено широкое использование в учебном процессе активных и интерактивных методов проведения занятий:

Интерактивные технологии:

№ темы	Вид занятия	Используемые интерактивные технологии	Всего часов
1-16	Лабораторная работа	Метод проектов	88

При обучении дисциплине применяются следующие формы занятий:

- лекции, направленные на получение новых и углубление научно-теоретических знаний, в том числе вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция, лекции-дискуссии, лекции-беседы и др.;
- лабораторные занятия, проводимые под руководством преподавателя в учебной лаборатории с использованием компьютеров и учебного оборудования, направленные на закрепление и получение новых умений и навыков, применение знаний и умений, полученных на теоретических занятиях, при решении практических задач и др.

Все занятия обеспечены мультимедийными средствами (SMART доски, проекторы, экраны) для повышения качества восприятия изучаемого материала. В образовательном процессе широко используются информационно-коммуникационные технологии.

Самостоятельная работа студентов – это планируемая работа студентов, выполняемая по заданию при методическом руководстве преподавателя, но без его непосредственного участия. Формы самостоятельной работы студентов определяются содержанием учебной дисциплины, степенью подготовленности студентов. Они могут

иметь учебный или учебно-исследовательский характер: подготовка к лабораторным работам.

Формами контроля самостоятельной работы выступают: проверка письменных отчётов по результатам выполненных лабораторных работ. Результаты самостоятельной работы учитываются при оценке знаний на зачете и экзамене.

## 7. Формы аттестации и оценочные материалы

Текущий контроль предназначен для проверки хода и качества усвоения учебного материала, стимулирования учебной работы обучающихся и совершенствования методики проведения занятий. Он проводится в ходе всех видов занятий в форме, избранной преподавателем.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по учебной дисциплине и проводится в форме зачета, экзамена. Принимается экзамен и зачет преподавателем, читающим лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся:

### 7.1. Вопросы к зачету

*Вопросы (5 семестр):*

1. Чарльз Беббидж, его механические вычислители и фактическое рождение понятия "Архитектура" в применении к вычислительным системам.
2. Гипотетическая машина Тьюринга.
3. Фон-Неймановские принципы построения процессоров.
4. Гарвардская и Принстонская архитектуры.
5. Электромеханические и электронные вычислительные машины.
6. Суперкомпьютеры и задачи класса GRAND CHALLENGES.
7. Современные определения понятия архитектуры вычислительных систем.
8. Основные архитектуры многопроцессорных и многокомпьютерных вычислительных систем.
9. Сравнительные достоинства и недостатки SMP- и MPP-архитектур.
10. Классификации архитектур вычислителей.
11. Понятие регистра - счётчика команд (Set Counter).
12. Форматы машинных команд.
13. CISC- и RISC-процессоры.
14. Системы счисления современных процессоров.
15. Стандарт IEEE 754 для представления вещественных чисел.
16. Выполнение арифметических действий над числами с фиксированной запятой, проблемы "размножения ошибки" вследствие переноса значений битов и невозможности распараллеливания на битовом уровне.
17. Использование конвейерной архитектуры для повышения производительности процессора.
18. Классика - архитектура процессора Intel x86. История разработки, внутренние регистры процессора, их разрядность, обозначение и назначение.
19. Принцип определения последовательности исполнения машинных команд.
20. Принципы программного управления последовательностью выполнения операций (CONTROL-FLOW) и управления порядком выполнения операций самими данными (DATA-FLOW).
21. Структурная схема вычислителя с управлением последовательностью вычислений потоком данных.
22. Проблема ассоциативной памяти, особенности программирования DATA-FLOW машин.
23. Язык Ассемблер и машинные команды.

## 24. Компилятор ассемблера, макроассемблер.

*Вопросы (7семестр):*

1. Вычислительный кластер как представитель многопроцессорных вычислительных систем (МВС) архитектуры MPP.
2. Вычислительные кластеры BEOWULF.
3. Латентность (инерционность) сети передачи данных.
4. Понятие тонкой информационной структуры программы и гранулы (зёрна, блока) параллелизма.
5. Коэффициент гранулярности. технологии разработки параллельных программ.
6. Параллельное программирование с использованием передачи сообщений MPI (Messages Passing Interface).
7. Мнозначность параллельных реализаций единого математического алгоритма.
8. Разница между форматами исполняемых файлов для 16, 32 и 64-х битовых архитектур.
9. Программные сегменты и их реализация для 16-ти битовых процессоров Intel и для "плоской" (flat) организации памяти.
10. Реализация многозадачности, разграничение доступа к памяти.
11. Двоично-десятичная арифметика как способ представления чисел с фиксированной точкой.
12. Транспьютер как специализированный процессор с большим количеством каналов связи с подобными.
13. Архитектура транспьютера T805 фирмы Inmos.
14. Топологии объединения транспьютеров.
15. Языки программирования для транспьютеров.
16. Современное состояние транспьютерной индустрии.
17. Повышение надёжности ассемблерпрограммирования путём использования готовых библиотек.
18. Понятия метакомпьютера и метакомпьютинга.
19. Вычислительная сеть (GRID).
20. Архитектура метакомпьютера.
21. Облачные вычисления как частный случай метакомпьютинга.
22. Шейдерный механизм вычислений как основа использования карт в качестве графических процессоров (GPU - Graphics Processing Unit).
23. GPU как процессор массового параллелизма архитектуры SIMD.
24. Графические процессоры фирм NVIDIA и AMD.
25. Направление GPGPU (General-Purpose computing on Graphics Processing Units).
26. Поточковый процессор, вычислительные ядра потокового процессора.
27. Программирование графических процессоров.

Зачет проводится по окончании занятий по дисциплине до начала экзаменационной сессии.

Билет для проведения промежуточной аттестации в форме зачета включают вопросы для проверки сформированности знаний, умений и навыков.

- Оценка «зачтено» проставляется студенту, выполнившему и защитившему в полном объеме лабораторные работы в течение семестра, чей уровень знаний, умений и навыков соответствует уровню оценок «отлично», «хорошо» или «удовлетворительно».
- Оценка «не зачтено» проставляется студенту, не выполнившему и (или) не защитившему в полном объеме практические задания и лабораторные работы в



течение семестра, либо чей уровень знаний, умений и навыков соответствует уровню оценки «неудовлетворительно».

Общими критериями, определяющими оценку знаний, умений и навыков являются:

- для оценки «отлично» - наличие глубоких и исчерпывающих знаний в объёме пройденного программного материала правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы;
- для оценки «хорошо» - наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильны действия по применению знаний на практике, четкое изложение материала;
- для оценки «удовлетворительно» - наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике;
- для оценки «неудовлетворительно» - наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

## 7.2. Вопросы и задачи к экзамену

### *Вопросы (6 семестр)*

1. Принцип поразрядного последовательного выполнения булевых операций при использовании ПСС (Позиционной Системы Счисления).
2. Методы ускорения вычислений в рамках ПСС.
3. Китайская теорема об остатках.
4. Программное обеспечения разработки программ на Ассемблере.
5. Интегрированная среда emu8086, её возможности.
6. Макроассемблеры.
7. Секции программы на ассемблере - описание формата исполняемого файла, данных, собственно ассемблерный код.
8. Возможности структурирования программ на ассемблере - макросы, процедуры.
9. Ярусно-Параллельная Форма (ЯПФ) информационного графа алгоритма.
10. Параллелизация внешняя и истинная.
11. Технологии параллелизации.
12. Конвейерный принцип.
13. Векторные процессоры.
14. Контр-фон-Неймановские архитектуры вычислителей.
15. Типы параллелизаций - параллелизация вычислений и параллелизация по данным
16. Целочисленная арифметика при программировании на ассемблере.
17. Синтаксис команд целочисленной арифметики.
18. Анализ данных и трассировка ассемблерной программы в эмуляторе emu8086.
19. Анализ выполнения машинной команды с помощью регистра состояния процессора, битовые флаги состояния.
20. Определение суперкомпьютера.
21. Вычислительные кластеры.
22. Реальное и пиковое быстродействие.
23. Проблемы пета- и эксафлопса.
24. Принципы организации ветвлений при программировании на Ассемблере, анализ конкретных битовых флагов регистра состояния и команды условного перехода.

25. Реализации "ближних" и "дальних переходов", дополнительные возможности реализации циклов.
26. Нейронная сеть (НС).
27. Обучение нейронной сети.
28. Нейронные сети Кохонена, Хопфилда.
29. НС с обратным распространением информации.
30. Одно- и многослойные НС.
31. Процедуры обучения НС.
32. Метод обратного распространения ошибки (back propagation), условие применимости этого метода.
33. Самообучающиеся НС.

*Вопросы (8 семестр):*

1. Нейронные сети как аналоговые вычислители.
2. Архитектура аналоговых вычислителей.
3. Базовые элементы электронных АВМ.
4. Операционный усилитель как основа электронной АВМ.
5. Устройства ввода и вывода АВМ.
6. Модели аналоговых вычислителей и их функциональные возможности.
7. Области применения АВМ.
8. Достоинства и недостатки АВМ.
9. Гибридные вычислительные системы, преимущества перед цифровыми вычислительными системами.
10. Реконфигурация как средство создания вычислительной структуры, максимально эффективной для заданного алгоритма.
11. Связь реконфигурируемых вычислительных систем с идеей клеточных автоматов фон-Неймана (1948).
12. Понятие процессорного элемента (ПЭ), универсальной коммутационной среды (УКС).
13. Программируемые логические интегральные схемы (ПЛИС) как основа создания реконфигурируемых систем.
14. Эффективность систем с программируемой архитектурой.
15. Определение квантового компьютера, история идеи.
16. Понятие кубита, квантовой связанности (спутанности).
17. Число линейно независимых состояний системы N кубитов.
18. Физическая реализация кубитов.
19. Квантовые проблемы задания исходных данных и считывания результата.
20. Фундаментально вероятностный характер квантовых вычислений.
21. Известные алгоритмы для квантовых компьютеров (алгоритмы Гровера, Залки-Визнера, Шора).
22. Современные реализации квантовых вычислителей.
23. Проблемы и технологии поиска информации в сети InterNet.
24. Технологии Data Mining.
25. Закон Зипфа (George Kingsley Zipf) в применении к системам поиска информации.
26. Архитектура и принципы функционирования машин поиска информации в сети InterNet.
27. Принципы ранжирования информации в поисковых системах.
28. Поисковая машина Google.
29. Архитектура аппаратной части поисковых машин Google.

*Примерная тематика задач:*

Разработка программы на языке Ассемблер для процессоров архитектуры x86 фирмы Intel.

По заданному алгоритму требуется:

Разработать программу на языке ассемблера для нахождения наибольшего общего делителя (НОД, GCD - Greatest Common Divisor) двух положительных чисел по методу Эвклида вычитанием. Программу разработать в среде эмулятора emu8086 с использованием 16-ти битовых регистров и оформить в виде ассемблер-процедуры (исходные данные в регистрах AX, BX, выходное значение поместить в AX).

Формат входных и выходных данных:

1. Для разработки программы применяется интегрированная среда разработки Ассемблерных программ emu8086 (исполняемый файл emu8086.exe).

2. Команды программы находятся в текстовом файле GCD\_Euclid\_16.asm, скомпилированный бинарный файл (версия 16 бит для x86) - GCD\_Euclid\_16.com. Бинарный файл исполняется в эмуляторе emu8086 или в DOS-виртуальной машине (напр., в DosBox) или при загрузке ЭВМ в режиме DOS (загрузка с FLASH-накопителя).

Формат выходных данных:

Выходные данные - исполняемый бинарный 16-битный файл GCD\_Euclid\_16.com (размер 39 байт).

Билет для проведения промежуточной аттестации в форме экзамена включает вопросы и задачи для проверки сформированности знаний, умений и навыков.

Общими критериями, определяющими оценку знаний, умений и навыков являются:

– для оценки «отлично» - наличие глубоких и исчерпывающих знаний в объёме пройденного программного материала правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы;

– для оценки «хорошо» - наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильны действия по применению знаний на практике, четкое изложение материала;

– для оценки «удовлетворительно» - наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике;

– для оценки «неудовлетворительно» - наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

7.3. Выполнение и примерная тематика курсового проекта

Курсовой проект не предусмотрен.

7.4. Выполнение и примерные задания расчетно-графической работы

Расчетно-графическая работа не предусмотрена.

7.5. Выполнение и примерная тематика контрольной работы

Контрольная работа не предусмотрена.

## **8. Учебно-методическое и информационное обеспечение дисциплины**

Электронный каталог и электронные информационные ресурсы, предоставляемые научной библиотекой ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://library.chuvsu.ru/>

8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Параллельные вычисления на GPU. Архитектура и программная модель CUDA [Электронный ресурс] : учебное пособие / А.В. Боресков [и др.]. — Электрон. текстовые данные. — М. : Московский государственный университет имени М.В. Ломоносова, 2015. — 336 с. - Режим доступа: <a href="http://www.iprbookshop.ru/54647.html">http://www.iprbookshop.ru/54647.html</a>
2.	Гергель В.П. Теория и практика параллельных вычислений [Электронный ресурс] / В.П. Гергель. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 480 с. - Режим доступа: <a href="http://www.iprbookshop.ru/57385.html">http://www.iprbookshop.ru/57385.html</a>
3.	Симаков А. Л. Кросс-ассемблер: учебное пособие [для 1 курса направления подготовки бакалавров "Информатика и вычислительная техника"] / Симаков А. Л., [отв. ред. И. А. Обломов] ; Чуваш. гос. ун-т им. И. Н. Ульянова - Чебоксары: Изд-во Чуваш. ун-та, 2015. - 59с.

8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Васильев В.Н. Основы программирования на языке C++ [Электронный ресурс] : учебное пособие / В.Н. Васильев. — Электрон. текстовые данные. — Волгоград: Волгоградский институт бизнеса, Вузовское образование, 2010. — 72 с. - Режим доступа: <a href="http://www.iprbookshop.ru/11341.html">http://www.iprbookshop.ru/11341.html</a>
2.	Разработка Windows-приложений в среде программирования Visual Studio.Net [Электронный ресурс] : учебно-методическое пособие по дисциплине Информатика и программирование / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 20 с. - Режим доступа: <a href="http://www.iprbookshop.ru/61536.html">http://www.iprbookshop.ru/61536.html</a>
3.	Орлов С. А. Организация ЭВМ и систем: [фундаментальный курс по архитектуре и структуре современных компьютерных средств : учебник для вузов по направлению "Информатика и вычислительная техника"] / Орлов С. А., Цилькер Б. Я. - 2-е изд. - СПб.: Питер, 2011. - 686с.

8.3. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.

Доступное программное лицензионное и свободно распространяемое программное обеспечение, предоставляемое студенту университетом возможно для загрузки и использования по URL: <http://ui.chuvsu.ru/> \*.

№ п/п	Наименование Рекомендуемого ПО	Условия доступа/скачивания
		свободное лицензионное соглашение:
1.	Microsoft Visual Studio	<a href="https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017">https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017</a>
2.	DevC++	<a href="https://sourceforge.net/projects/orwelldevcpp/">https://sourceforge.net/projects/orwelldevcpp/</a>
3.	Linux/ Ubuntu	<a href="http://ubuntu.ru/">http://ubuntu.ru/</a>
4.	LibreOffice	<a href="https://ru.libreoffice.org/">https://ru.libreoffice.org/</a>
1.	Microsoft Windows	из внутренней сети университета (договор)*
2.	Microsoft Office	
1.	Гарант	из внутренней сети университета (договор)*
2.	Консультант +	

8.4. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы

№ п/п	Наименование интернет ресурса	Режим доступа
1	Открытое образование. Параллельное программирование с использованием OpenMP и MPI	URL: <a href="https://openedu.ru/course/tgu/PROGR/">https://openedu.ru/course/tgu/PROGR/</a>
2	Национальный открытый университет. Язык програм-	URL: <a href="http://www.intuit.ru/studies/courses/17/17/info">http://www.intuit.ru/studies/courses/17/17/info</a>

	мирования C++	
3	Национальный открытый университет. Архитектура микропроцессоров	URL: <a href="http://www.intuit.ru/studies/courses/604/460/info">http://www.intuit.ru/studies/courses/604/460/info</a>
4	Национальный открытый университет. Архитектура ЭВМ и язык ассемблера	URL: <a href="http://www.intuit.ru/studies/courses/535/391/info">http://www.intuit.ru/studies/courses/535/391/info</a>
5	Национальный открытый университет. Архитектуры и топологии многопроцессорных вычислительных систем	URL: <a href="http://www.intuit.ru/studies/courses/45/45/info">http://www.intuit.ru/studies/courses/45/45/info</a>

## 9. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине оснащены автоматизированным рабочим местом (АРМ) преподавателя, обеспечивающим тематические иллюстрации и демонстрации, соответствующие программе дисциплины в составе:

- ПЭВМ с доступом в интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением;
- настенный экран;

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены АРМ преподавателя и пользовательскими АРМ по числу обучающихся, объединенных локальной сетью («компьютерный» класс), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова».

## 10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

## 11. Методические рекомендации по освоению дисциплины

В ходе лекционных занятий студенту рекомендуется вести конспектирование учебного материала. Следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. При составлении конспекта желательно оставлять в рабочих конспектах поля, на которых в дальнейшем можно делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. В ходе

лекционных занятий рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к лабораторным работам рекомендуется изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях. Основой для выполнения лабораторной работы являются разработанные кафедрой методические указания. Рекомендуется дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной рабочей программой дисциплины.

Форма организации студентов на лабораторных работах фронтальная. При фронтальной форме организации занятий все студенты выполняют одновременно одну и ту же работу.

Если в результате выполнения лабораторной работы запланирована подготовка письменного отчета, то отчет о выполненной работе необходимо оформлять в соответствии с требованиями методических указаний. Качество выполнения лабораторных работ является важной составляющей оценки текущей успеваемости обучающегося.