

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 математического и аппаратного обеспечения информационных систем

«УТВЕРЖДАЮ»
Проректор по учебной работе

И.Е. Поверинов

31 августа 2017 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Системное программное обеспечение»

Направление подготовки – 09.03.01 «Информатика и вычислительная техника»

Квалификация (степень) выпускника – бакалавр

Профиль «Вычислительные машины, комплексы, системы и сети»

Академический бакалавриат

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Министерства образования и науки №5 от 12.01.2016 г.

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):

старший преподаватель



С.О. Иванов

ОБСУЖДЕНО:

на заседании кафедры математического и аппаратного обеспечения информационных систем «30» августа 2017г., протокол №1

заведующий кафедрой



Д.В. Ильин

СОГЛАСОВАНО:

Методическая комиссия факультета информатики и вычислительной техники «30» августа 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

1. Цель и задачи обучения по дисциплине	4
2. Место дисциплины в структуре основной образовательной программы (ООП)	4
3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП	4
4. Структура и содержание дисциплины	4
4.1. Содержание дисциплины	5
4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения.....	5
5. Содержание разделов дисциплины	6
5.1. Лекции и практические занятия.....	6
5.2. Лабораторные работы	9
5.3. Вопросы для самостоятельной работы студента	10
6. Образовательные технологии.....	11
7. Формы аттестации и оценочные материалы.....	12
7.1. Вопросы к зачету.....	12
7.2. Оценивание результатов зачета.....	13
7.3. Вопросы к экзамену.....	14
7.4. Оценивание результатов экзамена	15
7.5. Выполнение и примерная тематика курсовой работы.....	16
7.6. Оценивание курсовой работы	17
8. Учебно-методическое и информационное обеспечение дисциплины	17
8.1. Рекомендуемая основная литература	17
8.2. Рекомендуемая дополнительная литература.....	18
8.3. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.	18
8.4. Рекомендуемые интернет-ресурсы и открытые онлайн курсы.....	18
9. Материально-техническое обеспечение дисциплины.....	19
10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями.....	19
11. Методические рекомендации по освоению дисциплины.....	19

1. Цель и задачи обучения по дисциплине

Целью дисциплины «Системное программное обеспечение» является изучение особенностей функционирования системного уровня компьютерной системы, принципов и методов его работы и организации.

Основными задачами дисциплины являются:

- проектирование программных и аппаратных средств (систем, устройств, деталей, программ, баз данных) в соответствии с техническим заданием с использованием средств автоматизации проектирования;
- применение современных инструментальных средств при разработке программного обеспечения;

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

Дисциплина «Системное программное обеспечение» относится к числу дисциплин вариативной части профессионального цикла. Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин: «Информатика», «Программирование».

Дисциплина является предшествующей для дисциплин: «ЭВМ и периферийные устройства», «Операционные системы», «Базы данных», «Микропроцессорные средства и системы», «Системы реального времени», «Архитектура вычислительных систем и компьютерных сетей», «Техническое обслуживание ЭВМ», прохождения практик, выполнения ВКР

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

Процесс обучения по дисциплине направлен на формирование следующих компетенций:

- способность участвовать в настройке и наладке программно-аппаратных комплексов (ОПК-4);
- способность проверять техническое состояние вычислительного оборудования и осуществлять необходимые профилактические процедуры (ПК-7);

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

знать:

- принципы и алгоритмы работы системного программного обеспечения (З1);
- назначение системного программного обеспечения в составе компьютерных систем (З2);

уметь:

- применять системное программное обеспечение для формирования компьютерных систем (У1);
- создавать системные средства для обеспечения работы вспомогательного и прикладного программного обеспечения (У2);

владеть навыками:

- системными утилитами и инструментами для настройки и обслуживания компьютерной системы (Н1).
- инструментальными средствами используемых для создания системного программного обеспечения (Н2).

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 (семинары, лабораторные работы, практикумы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, 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. Низкоуровневое программирование.	ОПК-4	31, У1
1.1. Введение в СПО.		
1.2. Вычислительная машина.		
1.3. Реализация структур данных и вычислительных конструкций.		
1.4. Макропроцессоры		
Раздел 2. Системные инструментальные средства.	ОПК-4, ПК-7	31, Н1, 32, Н2
2.1. Ассемблеры.		
2.2. Загрузчики.		
2.3. RunTime Interface		
2.4. Отладчики.		
2.5. Виртуальная машина.		
2.6. Компиляция.		
Зачет	31, 32, У1, Н1, Н2	31, 32, У1, Н1, Н2
Раздел 3. Методы трансляции.	ПК-7	У2, Н2
3.1. Вычислительные модели		
3.2. Трансляция		
3.3. Теория формальных грамматик		
3.4. Лексический анализ.		
3.5. Лексические анализаторы.		
3.6. Синтаксический анализ.		
3.7. Синтаксические анализаторы.		
3.8. Семантический анализ		
3.9. Оптимизация.		
3.10. Генерация кода и данных.		
Курсовая работа	ПК-7	У2, Н2
Экзамен	ОПК-4, ПК-7	31, У2, Н2

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	КСР			
Раздел 1. Низкоуровневое программирование.							
1.1. Введение в СПО.	6	2	2		2	2	
1.2. Вычислительная машина.	10	6	2		2	2	
1.3. Реализация структур данных и вычислительных конструкций.	12	6	4		2	2	
1.4. Макропроцессоры	6	2	2		2	2	
Раздел 2. Системные инструментальные средства.							
2.1. Ассемблеры.	10	4	4		2	2	
2.2. Загрузчики.	14	4	8		2	2	
2.3. RunTime Interface	6	2	2		2	2	
2.4. Отладчики.	6	2	2		2	2	

2.5. Виртуальная машина.	8	2	4		2	2	
2.6. Компиляция.	6	2	2		2	2	
Зачет	4				4	2	
Раздел 3. Методы трансляции.							
3.1. Вычислительные модели	9	2	2		5	2	
3.2. Трансляция	8	2	2		4	2	
3.3. Теория формальных грамматик	17	4	4		9	2	
3.4. Лексический анализ.	22	8	6		8	2	
3.5. Лексические анализаторы.	9	2	2		5	2	
3.6. Синтаксический анализ.	13	6	2		5	2	
3.7. Синтаксические анализаторы.	9	2	2		5	2	
3.8. Семантический анализ	12	2	6		4	2	
3.9. Оптимизация.	8	2	2		4	2	
3.10. Генерация кода и данных.	10	2	4		4	2	
Экзамен	47			2			45
Итого	252 7 з.е.	64	64	2	77	42	45

5. Содержание разделов дисциплины

5.1. Лекции

Раздел 1. Низкоуровневое программирование.

Тема 1.1. Введение в СПО.

Лекция 1. Введение в СПО.

1. Понятие СПО, виды.
2. Цели и задачи СПО.
3. Системный слой информационной системы. Среда программирования.

Тема 1.2. Вычислительная машина.

Лекция 2. Вычислительная машина.

1. Архитектура компьютера.
2. Система команд.
3. Способы указания операндов.

Лекция 3. Абстрактная вычислительная машина.

1. Архитектура и характеристик вычислительной машины.
2. Средства реализации вычислительных примитивов.

Лекция 4. Написание программ на машинном коде для вычислительной машины.

1. Процесс преобразования псевдокода в машинный код.
2. Управление метками.
3. Оформление блоков кода и данных.

Тема 1.3. Реализация структур данных и вычислительных конструкций.

Лекция 5. Управляющие конструкции.

1. Последовательности.
2. Ветвления.
3. Повторы.

Лекция 6. Группирующие конструкции.

1. Подпрограммы. Дисплей и фрейм процедуры.
2. Особенности и соглашения передачи аргументов.
3. Обработка ошибок.

Лекция 7. Абстрактные типы данных.

1. Понятие абстрактных типов данных, их виды.
2. Реализация абстрактных типов данных.
3. Операторы и выражения.

Тема 1.4. Макропроцессоры.

Лекция 8. Макропроцессор.

1. Назначение макропроцессора, примеры решаемых задач.
2. Алгоритм работы макропроцессора.
3. Расширенные возможности макропроцессоров.

Раздел 2. Системные инструментальные средства.

Тема 2.1. Ассемблеры.

Лекция 9. Ассемблер.

1. Синтаксис ассемблера, виды директив.
2. Примеры синтаксиса ассемблера: AT&T и Intel.

Лекция 10. Ассемблирование

1. Алгоритм работы ассемблера.
2. Проблемы ассемблирования.
3. Расширенные возможности ассемблера.

Тема 2.2. Загрузчики

Лекция 11. Загрузчик.

1. Многоуровневая трансляция.
2. Принцип работы резидентного загрузчика.
3. Компоновщик: задачи и принцип работы.

Лекция 12. Системный загрузчик.

1. Процесс загрузки IBM/PC-совместимого компьютера.
2. MBR: структура и ограничения, GPT.
3. Multiboot-спецификация.
4. UEFI: возможности и принцип работы.

Тема 2.3. RunTime Interface

Лекция 13. Интерфейс среды выполнения.

1. Понятие RTI, API, EEI, ABI, SPI.
2. Виды программных интерфейсов, способы реализации.
3. Характеристики и признаки «хорошего» интерфейса.

Тема 2.4. Отладчики.

Лекция 14. Отладчик.

1. Виды отладчиков.
2. Основные функции и принцип работы.
3. Антиотладочные приемы.

Тема 2.5. Виртуальная машина.

Лекция 15. Виртуальная машина.

1. Основные понятия и архитектура.
2. Достоинства и недостатки различных видов.

3. Принцип работы квази-эмуляции.

Тема 2.6. Компиляция.

Лекция 16. Компиляция.

1. Компиляция и кросс-компиляция.
2. Low Level Virtual Machine (LLVM).
3. Just-in-Time компиляция.

Раздел 3. Методы трансляции.

Тема 3.1. Вычислительные модели

Лекция 1. Вычислительные модели.

1. Наиболее распространённые вычислительные модели.
2. Формальное описание элементов языка программирования.

Тема 3.2. Трансляция

Лекция 2. Трансляция.

1. Цели и задачи трансляции.
2. Последовательная, многопроходная и синтаксически управляемые схемы трансляции.

Тема 3.3. Теория формальных грамматик

Лекция 3. Теория формальных грамматик.

1. Алфавит, цепочка, язык, грамматика.
2. Способы задания языка.
3. Иерархия формальных грамматик.

Лекция 4. Проблемы и задачи в теории грамматик.

1. Вывод строки по грамматике.
2. Распознавание структуры строки по грамматике.
3. Преобразование свойств грамматик.

Тема 3.4. Лексический анализ.

Лекция 5. Лексический анализ.

1. Цели и задачи лексического анализа.
2. Лексический анализатор: структура, принцип работы.
3. Регулярные грамматики: свойства, виды.

Лекция 6. Регулярные выражения.

1. Основные элементы регулярных выражений.
2. Стандарты регулярных выражений.
3. Особенности и способы обработки регулярных выражений.

Лекция 7. Конечные автоматы.

1. Структура и виды конечных автоматов.
2. Алгоритм работы конечного распознавателя.
3. Реализация конечных автоматов.

Лекция 8. Преобразования регулярных грамматик.

1. Преобразование регулярных грамматик в регулярные выражения и обратно.
2. Преобразование регулярных грамматик в конечный автомат и обратно.

Тема 3.5. Лексические анализаторы.

Лекция 9. Построение лексический анализаторов.

1. Преобразование регулярных выражений в конечный автомат.
2. Минимизация конечных автоматов.
3. Поиск совпадений.

Тема 3.6. Синтаксический анализ.

Лекция 10. Синтаксический анализ.

1. Цели и задачи синтаксического анализа.
2. Синтаксический анализатор: структура и принцип работы.
3. Генераторы трансляторов.

Лекция 11. Конечные автоматы с магазинной памятью.

1. Структура и виды стековых конечных автоматов.
2. Алгоритмы работы стековых конечных распознавателей.

Лекция 12. Приведение КС-грамматик.

1. КС-грамматики: свойства, виды.
2. Проверка и преобразования свойств контекстно-свободных грамматик.
3. Преобразование правил контекстно-свободных грамматик к нужному виду.

Тема 3.7. Синтаксические анализаторы.

Лекция 13. Построение синтаксических анализаторов.

1. Построение вспомогательных таблиц для синтаксического анализатора.
2. Формирование результата синтаксического разбора.

Тема 3.8. Семантический анализ

Лекция 14. Семантический анализ.

1. Цели и задачи семантического анализа.
2. Проверка и дополнение семантических особенностей языка.
3. Внутренний код.

Тема 3.9. Оптимизация.

Лекция 15. Оптимизация.

1. Цели и задачи оптимизации.
2. Принципы и методы оптимизации.

Тема 3.10. Генерация кода и данных.

Лекция 16. Генерация кода и данных.

1. Цели и задачи этапа генерации.
2. Распределение данных
3. Генерация вычислительных конструкций.

5.2. Лабораторные работы

Тема	Количество часов
Лабораторное занятие 1.1.1. Эмулятор SRM.	2
Лабораторное занятие 1.1.2. Программирование SRM.	2
Лабораторное занятие 1.1.3. SRM toolchain.	2
Лабораторное занятие 1.1.1.4. препроцессор SRM.	2
Лабораторное занятие 1.1.5. Ассемблер SRM.	4

Лабораторное занятие 1.1.6. Компоновщик SRM.	2
Лабораторное занятие 1.1.7. Загрузчик SRM.	2
Лабораторное занятие 1.1.8. Расширения SRM.	2
Лабораторное занятие 1.1.9. Отладчик SRM.	2
Лабораторное занятие 1.1.10. Монитор SRM.	2
Лабораторное занятие 1.1.11. Компиляция программы SRM.	2
Лабораторное занятие 1.2.1. Виртуальная машина.	2
Лабораторное занятие 1.2.2. Загрузочный диск.	2
Лабораторное занятие 1.2.3. Информация о системе.	2
Лабораторное занятие 2.1.1. Синтаксис паровозиков.	2
Лабораторное занятие 2.1.2. Генерация паровозиков.	2
Лабораторное занятие 2.1.3. Распознавание паровозиков.	2
Лабораторное занятие 2.1.4. Трансляция паровозиков.	2
Лабораторное занятие 2.2.1. Описание языка программирования АЛВЫР.	2
Лабораторное занятие 2.2.2. Интерпретатор для АЛВЫР.	2
Лабораторное занятие 2.2.3. Грамматика для АЛВЫР.	2
Лабораторное занятие 2.2.4. Сканер для АЛВЫР.	2
Лабораторное занятие 2.2.5. Парсер для АЛВЫР.	2
Лабораторное занятие 2.2.6. Семантический анализатор АЛВЫР.	2
Лабораторное занятие 2.2.7. Оптимизатор для АЛВЫР.	2
Лабораторное занятие 2.2.8. Генератор для АЛВЫР.	2
Лабораторное занятие 2.2.9. Построение машины Тьюринга на АЛВЫР.	2
Лабораторное занятие 2.3.1. Регулярные выражения.	2
Лабораторное занятие 2.3.2. Конечные автоматы.	2
Лабораторное занятие 2.3.3. Калькулятор выражений.	2
Лабораторное занятие 2.3.4. Универсальная машина Тьюринга.	2
Итого	64

5.3. Вопросы для самостоятельной работы студента.

Раздел 1. Низкоуровневое программирование.

1. Функции Linux для работы с задачами.
2. Функции Linux для работы с памятью.
3. Функции Linux для работы с дисками.
4. Функции Linux для работы с видеокартой.
5. Функции Linux для работы с клавиатурой, мышью.
6. Функции Linux для управления системой.

Раздел 2. Системные инструментальные средства.

1. Препроцессоры, макропроцессоры.
2. Ассемблеры.
3. Загрузчики, компоновщики.
4. Отладчики, дизассемблеры.
5. Эмуляторы, виртуальные машины.
6. Компиляторы, трансляторы.
7. Вспомогательные утилиты программирования.

Раздел 3. Методы трансляции.

1. Преобразование регулярной грамматики в автоматную
2. Преобразование регулярной грамматики в регулярное выражение
3. Преобразование регулярной грамматики в конечный автомат
4. Удаление недостижимых состояний ка
5. Поиск эквивалентных состояний ка
6. Преобразование недетерминированного в детерминированный конечный автомат
7. Преобразование конечного автомата в регулярное выражение
8. Преобразование конечного автомата в регулярную грамматику
9. Преобразование регулярного выражения в конечный автомат
10. Преобразование регулярного выражения в регулярную грамматику
11. Алгоритм Кока-Янгера-Касами
12. Генерация First, Follow
13. Построение управляющих таблиц LR(k)
14. Построение матрицы простого предшествования (алгоритм Вирта-Вебера)
15. Упрощение грамматики операторного предшествования
16. Проверка языка на существование
17. Устранение не-порождающих нетерминалов
18. Устранение недостижимых символов
19. Устранение eps-правил
20. Устранение цепных правил
21. Левая факторизация
22. Устранение левой рекурсии
23. Преобразование в НФХ/НФГ
24. Алгоритм преобразования синтаксического дерева в дерево операции
25. Алгоритм преобразования в триады
26. Алгоритм Дейкстры преобразования выражения в постфиксную форму
27. Алгоритм СУ-перевода
28. Методы оптимизации
29. Интерпретация (вычисление выражений) промежуточного кода
30. Примеры NP-полных задач

6. Образовательные технологии

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

- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, творческой).

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:

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 (проекторы) 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, подготовка тезисов к дискуссии, подготовка рецензий на изучаемые источники, разработка проекта.

Формами контроля самостоятельной работы выступают оценивание устного выступления студента на занятии, его доклада; проверка письменных отчётов по результатам выполненных заданий и лабораторных работ, курсовой работы. Результаты самостоятельной работы учитываются при оценке знаний на экзамене и зачёте.

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 экзамена и зачета, защиты курсовой работы (проекта). Принимается экзамен и зачет преподавателями, читающими лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся.

7.1. Вопросы к зачету

1. Что такое СПО?
2. Основные характеристики СПО?
3. Куда относятся системы программирования?
4. Какие основные задачи решает СПО?
5. Что такое трансляция?
6. Какие ограничения налагаются на трансляцию?
7. Какими особенностями обладают языки программирования?
8. Какие проблемы возникают при анализе программ?
9. Из чего состоит контекст выполнения?
10. Зачем нужна многоуровневая трансляция?
11. Из каких шагов состоит последовательная схема трансляции?
12. Что такое Гарвардская архитектура?
13. Из каких элементов состоит наиболее распространённая вычислительная машина?
14. Что такое RISC?
15. Структура команды.
16. Способы адресации операндов.
17. Что такое абстрактные типы данных?
18. Опишите АД для стека.
19. Что такое дисплей процедуры?
20. Структура фрейма процедуры
21. Особенности передачи аргументов.
22. Наиболее распространённые способы передачи параметров.
23. Опишите способы обработки ошибок, их достоинства и недостатки.
24. Из чего состоит программа на ассемблере.
25. Что такое директивы?

26. Особенности синтаксиса различных ассемблеров.
27. Что такое блок кода.
28. Особенности двухпроходного ассемблера.
29. Дополнительные возможности(абстракции) ассемблера.
30. Для чего нужны квомирование и эскейп последовательности?
31. Дополнительные возможности макропроцессоров?
32. Что такое точка входа?
33. Чем отличается блок кода от секции
34. Виды секций
35. Чем отличается загрузочный адрес от виртуального?
36. Принцип работы перемещающего загрузчика.
37. Что такое связывание кода.
38. Принцип работы компоновщика.
39. Дополнительные функции компоновщика
40. Что такое BIOS/EFI?
41. Чем отличается BootSector от PartitionBootRecord?
42. Что такое раскручивающийся загрузчик?
43. Этапы POST.
44. Что такое DiskLabel?
45. Содержимое Multiboot спецификации?
46. Что такое API?
47. Характеристики API?
48. Виды обратной совместимости?
49. Что такое SPI/ABI?
50. Чем отличается отладчик от тестера?
51. Что такое брейкпоинты?
52. Как отладчики останавливаются на брейкпоинтах?
53. Проблемы дизассемблирования.
54. Антиотладочные приёмы.
55. Какой тип VM лучший?
56. Что такое гипервизор, хост, гвест?
57. Примеры псевдоэмуляторов.
58. Достоинства и недостатки полных эмуляторов(симуляторов).
59. Принципы интерпретации команд.
60. Способы квази-эмуляции.
61. Что такое макропроцессор?
62. Что такое макроопределение?
63. Как осуществляется макроподстановка, макрогенерация?
64. Условия необходимые для кросс-компиляции?

7.2. Оценивание результатов зачета

Зачет проводится по окончании занятий по дисциплине до начала экзаменационной сессии в период недели контроля самостоятельной работы.

Билет для проведения промежуточной аттестации в форме зачета включают вопросы и задачи 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» (п.2.1). Оценка «не зачтено» проставляется студенту, не выполнившему и (или) не защитившему в полном объеме лабораторные работы в течение семестра, либо чей уровень знаний, умений и навыков соответствует уровню оценки «неудовлетворительно».

7.3. Вопросы к экзамену

1. СПО: определение, задачи, характеристики, виды.
2. Цели и задачи системного программирования. Системные инструментальные средства.
3. Трансляция. Особенности трансляции языков программирования. Этапы трансляции.
4. Необходимые условия трансляции. Лексика, синтаксис и семантика языков программирования.
5. Вычислительная машина. Виды, структура, принцип работы. Система команд и способы адресации.
6. Наиболее распространенные вычислительные машины. Архитектура компьютера.
7. Абстрактные типы данных: понятие, виды, способы реализации.
8. Способы реализации базовых структур данных: стеки, массивы, структуры, деревья, графы. Необходимый набор операций с базовыми структурами.
9. Базовые вычислительные конструкции: присвоение, выражение, ветвление, повторение, применение.
10. Реализация базовых вычислительных конструкций на низком уровне.
11. Ассемблер. Синтаксис программы на ассемблере. Структура и принцип работы.
12. Алгоритм работы ассемблера. Структура программы на ассемблере.
13. Загрузчик, компоновщик. Виды и принципы работы. Дополнительные возможности.
14. Программный модуль: виды, структура. Компоновка и загрузка программных модулей.
15. Среда выполнения. Прикладной и двоичный интерфейс программ (API, ABI).
16. Требования к интерфейсам среды выполнения. Характеристика «хорошего» API.
17. Системный загрузчик. Этапы загрузки. Multiboot-загрузчики.
18. Главная загрузочная запись. EFI: архитектура, принцип работы.
19. Отладчик. Основные компоненты и принцип их работы. Антиотладочные приемы.
20. Виды отладчиков. Отладочные приемы. Отладчики и Digital Rights Management.
21. Виртуальная машина. Структура. Виды.
22. Принцип работы виртуальной машины. Технология квазивиртуализации.
23. Макропроцессор. Синтаксис макросов. Структура и принцип работы.
24. Препроцессор: назначение, структура и принцип работы.
25. Виды вычислительных моделей. Формальное описание математических объектов.
26. Формальное описание структур данных и вычислительных конструкций.
27. Теория формальных грамматик. Определение основных операций с символами, алфавитами, языками. Способы определения языка.
28. Способы задания грамматики. Форма Бекуса-Наура. Автоматы для распознавания и вывода строк.
29. Иерархия грамматик. Проблемы и задачи в теории грамматик.
30. Вывод строки по грамматике. Распознавание свойств грамматик. Классы грамматик.
31. Последовательная схема трансляции. Лексический анализ. Сканер.
32. Лексический анализатор: основные функции, принцип работы. Таблицы символов.
33. Регулярные грамматики. Регулярные выражения. Регулярные распознаватели.
34. Базовые и расширенные регулярные выражения POSIX. Реализация регулярных выражений в языках программирования.
35. Конечные автоматы. Виды и структура. Алгоритм работы конечного распознавателя.
36. Способы реализации конечных автоматов. Расширенные возможности конечных автоматов.
37. Конвертирование регулярных грамматик, выражений и распознавателей.

38. Алгоритм преобразование регулярной грамматики в конечный автомат и регулярное выражение.
39. Упрощение регулярных грамматик, выражений и распознавателей.
40. Алгоритм преобразования недетерминированных конечных автоматов в детерминированные.
41. Последовательная схема трансляции. Синтаксический анализ. Парсер.
42. Синтаксический анализатор: основные функции, принцип работы. Синтаксическое дерево.
43. Конечный автомат с магазинной памятью: виды и структура. Восходящий и нисходящий алгоритмы работы синтаксических распознавателей.
44. Способы реализации конечных автоматов с магазинной памятью. Расширенные возможности конечных автоматов с магазинной памятью.
45. К-предсказывающие алгоритмы работы синтаксических распознавателей.
46. Алгоритмы построения таблиц FIRST, FOLLOW, управляющей таблицы.
47. Контекстно-свободные грамматики. Виды и свойства.
48. Классификация контекстно-свободных грамматик по виду алгоритма распознавания.
49. Приведение контекстно-свободных грамматик к заданной форме.
50. Нормальная форма Хомского. Нормальная форма Грейбах.
51. Оптимизация контекстно-свободных грамматик.
52. Преобразование правил контекстно-свободных грамматик. Устранение рекурсий, бесполезных символов.
53. Последовательная схема трансляции. Семантический анализ. Форматер.
54. Семантический анализатор: основные функции, принцип работы.
55. Теория алгоритмов. Машина Тьюринга. Проблемы вычислимости, классы задач.
56. Способы реализации машины тьюринга. Универсальная машина Тьюринга.
57. Синтаксически-управляемая трансляция. Внутренний код.
58. Представление промежуточного кода в форме триад, тетрад, инструкций.
59. Оптимизация: виды, принципы и методы.
60. Алгоритмы свёртки кода, разворачивания циклов, удаления недостижимых участков.
61. Генерация данных и кода.
62. Алгоритмы распределения данных, генерации выражений и управляющих конструкций.
63. Выполнение программ, среда поддержки выполнения. Системные библиотеки.
64. Среда выполнения программ. Системные функции.

7.4. Оценивание результатов экзамена

Экзаменационный билет для проведения промежуточной аттестации включают вопросы и задачи для проверки сформированности знаний, умений и навыков.

Общими критериями, определяющими оценку знаний, умений и навыков на экзамене, 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.

7.5. Выполнение и примерная тематика курсовой работы

Курсовая работа выполняется в процессе изучения дисциплины. Общее руководство и контроль за ходом выполнения курсовой работы осуществляет преподаватель соответствующей дисциплины. Курсовая работа выполняется в соответствии с методическими указаниями для обучающихся.

Основными функциями руководителя курсовой работы являются:

1. определение и формулирование темы курсовой работы совместно с обучающимся на основе примерной тематики или по предложенной обучающимся теме в рамках содержания дисциплины;
2. консультирование по вопросам содержания и последовательности выполнения курсовой работы;
3. оказание помощи студенту в подборе необходимой литературы;
4. контроль хода выполнения курсовой работы.

Примерная тематика курсовой работы:

1. Создание синтаксического анализатора для языка запросов SQL
2. Создание синтаксического анализатора для языка запросов XQuery
3. Создание синтаксического анализатора для языка описания ограничений Object Constraint Language (OCL)
4. Создание синтаксического анализатора для языка поисковых запросов google
5. Создание синтаксического анализатора для язык поисковых запросов yandex
6. Создание синтаксического анализатора для шаблонов дат systemd.time.
7. Создание синтаксического анализатора для языка хранения данных xml
8. Создание синтаксического анализатора для языка хранения данных CSV
9. Создание синтаксического анализатора для языка хранения данных json
10. Создание синтаксического анализатора для языка хранения данных yaml
11. Создание синтаксического анализатора для языка описания графики svg
12. Создание синтаксического анализатора для языка стилей css
13. Создание синтаксического анализатора для языка разметки документов tex
14. Создание синтаксического анализатора для языка разметки документов html
15. Создание синтаксического анализатора для языка разметки документов asciidoc
16. Создание синтаксического анализатора для языка разметки документов Markdown
17. Создание синтаксического анализатора для языка разметки документов creole
18. Создание синтаксического анализатора для хранения настроек apache
19. Создание синтаксического анализатора для хранения настроек nginx
20. Создание синтаксического анализатора для формата хранения настроек ini
21. Создание синтаксического анализатора для формата хранения настроек alsa.conf
22. Создание синтаксического анализатора для формата хранения настроек regedit
23. Создание синтаксического анализатора для обработки текстов с помощью m4
24. Создание синтаксического анализатора для обработки текстов с помощью sed
25. Создание синтаксического анализатора для обработки текстов с помощью awk
26. Создание синтаксического анализатора для скриптов bash
27. Создание синтаксического анализатора для скриптов bat/cmd
28. Создание синтаксического анализатора для скриптов autohotkey
29. Создание синтаксического анализатора для программ на языке INTERCAL
30. Создание синтаксического анализатора для программ на языке postscript
31. Создание синтаксического анализатора для программ на языке forth
32. Создание синтаксического анализатора для программ на языке lisp

7.6. *Оценивание курсовой работы*

Курсовая работа выполняется в процессе изучения дисциплины. Общее руководство и контроль за ходом выполнения курсовой работы осуществляет преподаватель соответствующей дисциплины. Курсовая работа выполняется в соответствии с методическими указаниями для обучающихся.

Оценивание курсовой работы осуществляется в соответствии с полнотой и качеством выполнения задания на курсовую работу, качеством защиты работы (ответы на вопросы, презентация и др.). Оценка курсовой работы отражает уровень сформированности соответствующих (п. 1.2) компетенций:

«отлично» - работа выполнена в соответствии с утвержденным планом и заданием, полностью раскрыто содержание каждого вопроса; студентом сформулированы собственные аргументированные выводы по теме работы; оформление работы соответствует предъявляемым требованиям; при защите работы обучающийся демонстрирует свободное владение материалом и верно отвечает на поставленные вопросы;

«хорошо» - работа выполнена в соответствии с утвержденным планом и заданием; полностью раскрыто содержание каждого вопроса; имеются незначительные замечания к оформлению работы; при защите работы обучающийся демонстрирует владение материалом, но отвечает на ряд поставленных вопросов не в достаточно полном объеме;

«удовлетворительно» - работа выполнена в соответствии с утвержденным планом и заданием, но не полностью раскрыто содержание каждого вопроса; обучающимся не сделаны собственные выводы по теме работы; допущены существенные недостатки в оформлении работы; при защите работы обучающийся демонстрирует владение материалом, но отвечает не на все поставленные вопросы, либо не в достаточно полном объеме;

«неудовлетворительно» - если работа не выполнена в соответствии с утвержденным планом и заданием, не раскрыто содержание каждого вопроса; обучающимся не сделаны выводы по теме работы, имеются существенные недостатки в оформлении работы; при защите работы обучающийся не демонстрирует владение материалом, не отвечает на поставленные вопросы.

В случае оценивания работы на «неудовлетворительно» работа направляется на дальнейшую доработку.

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 и условия доступа перечня представлены в Приложениях к рабочей программе.

№ п/п	Наименование
1.	Гунько А.В. Системное программное обеспечение [Электронный ресурс] : конспект лекций / А.В. Гунько. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2011. — 138 с. Режим доступа: http://www.iprbookshop.ru/45020.html
2.	Иванова Н.Ю. Системное и прикладное программное обеспечение [Электронный ресурс] : учебное пособие / Н.Ю. Иванова, В.Г. Маняхина. — Электрон. текстовые данные. — М. : Прометей, 2011. — 202 с. Режим доступа: http://www.iprbookshop.ru/58201.html
3.	Журавлёва И.А. Системное и прикладное программное обеспечение [Электронный ресурс] : лабораторный практикум / И.А. Журавлёва, П.К. Корнеев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 132 с. Режим доступа: http://www.iprbookshop.ru/69432.html
4.	Мартыничев, А. К. Программное обеспечение компьютерных систем управления оборудованием :

учебное пособие / А. К. Мартыничев ; [отв. ред. С. А. Лазарев] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2010. - 134с.
--

8.2. Рекомендуемая дополнительная литература

Ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе.

№ п/п	Наименование
1.	Учебно-методическое пособие по выполнению курсовой работы по дисциплине Системное программное обеспечение защищенных инфокоммуникационных систем [Электронный ресурс] / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2015. — 10 с. http://www.iprbookshop.ru/63359.html
2.	Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 1 [Электронный ресурс] : учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 104 с. http://www.iprbookshop.ru/45017.html
3.	Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 2 [Электронный ресурс] : учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2011. — 160 с. http://www.iprbookshop.ru/45018.html
4.	Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 3 [Электронный ресурс] : учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2012. — 120 с. http://www.iprbookshop.ru/45019.html

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

Программное обеспечение, профессиональные базы данных, информационно-справочные системы, предоставляемые управлением информатизации ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://ui.chuvsu.ru/>*

8.3.1 Программное обеспечение

№ п/п	Наименование	Условия доступа/скачивания
1.	MS Windows/Gentoo linux	лицензия университета/ свободное лицензионное соглашение (https://www.gentoo.org/downloads/)
2.	MS Office/ LibreOffice	лицензия университета/ свободное лицензионное соглашение (https://ru.libreoffice.org/)

8.3.2 Базы данных, информационно-справочные системы

№ п/п	Наименование программного обеспечения	Условия доступа/скачивания
1.	Гарант	из внутренней сети университета (договор)*
2.	Консультант +	

8.4. Рекомендуемые интернет-ресурсы и открытые онлайн курсы.

№ п/п	Наименование	Условия доступа
1.	Российская Государственная Библиотека	http://www.rsl.ru
2.	Государственная публичная научно-техническая библиотека России	http://www.gpntb.ru
3.	Фундаментальная библиотека Нижегородского государственного университета	http://www.unn.ru/library
4.	Научная библиотека Казанского государственного университета	http://lsl.ksu.ru

5.	Научная электронная библиотека	http://elibrary.ru
6.	Полнотекстовая библиотека учебных и учебно-методических материалов	http://window.edu.ru
7.	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru

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.

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.

