

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего
профессионального образования «Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Одобрено УМС ИАТЭ НИЯУ МИФИ,
Протокол №2-8/2021 От 30.08.2021

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Системное программирование

Шифр, название дисциплины

для студентов специальности/направления подготовки

09.03.01 «Информатика и вычислительная техника»

Шифр, название специальности/направления подготовки

профиль

«Вычислительные машины, комплексы, системы и сети»

Шифр, название специализации/профиля

Форма обучения: очная

г. Обнинск 2021г.

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 09.03.01 – Информационные системы и технологии (уровень бакалавриата).

Программу составил:

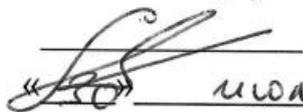
_____ В.И. Тищенко, доцент, к.ф.-м.н., доцент

Рецензент:

_____ С.Г. Сивачок, зам. директора ФГБУ «ВИИГМИ-МЦД», к.ф.-м.н.

Программа рассмотрена на заседании отделения интеллектуальных кибернетических систем (ОИКС) ИАТЭ НИЯУ МИФИ (протокол № 5/7 от «30» июля 2021 г.)

Руководитель образовательной программы
09.03.01 Информатика и вычислительная техника

 _____ С.О. Старков
«30» июля 2021 г.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенций	Результаты освоения ООП <i>Содержание компетенций*</i>	Перечень планируемых результатов обучения по дисциплине**
ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	<p>Знать: связь архитектуры вычислительных систем с программным обеспечением, состав и назначение программного обеспечения, основные машинно-зависимые и машинно-независимые функции системных компонентов, их эксплуатационные характеристики, архитектурные особенности компьютера, влияющие на работу системных компонент.</p> <p>Уметь: ставить задачу; разрабатывать алгоритм ее решения на языке, наиболее подходящем для ее решения; создавать многомодульные комплексы, оптимизировать программу посредством опций компилятора.</p> <p>Владеть: технологией разработки программ с использованием опций современных трансляторов в среде программирования для оптимизации программ; навыками самостоятельной работы с технической литературой по системному программному обеспечению для дальнейшего совершенствования навыков программирования в современных средах программирования.</p>

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина реализуется в рамках вариативной части профессионального цикла. Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: «Программирование», «Программирование на языке Ассемблер». Дисциплина «Системное программирование» является одной из основ для изучения дисциплин «Операционные системы», «Технология программирования. Изучение дисциплины возможно при условии, что студент освоил: основы программирования на ассемблере (форматы команд и данных; способы адресации данных в командах ассемблера; основные машинные команды процессора; структуру программы на ассемблере; средства разработки и отладки программ на языке ассемблер; основные функции прерываний DOS и BIOS для организации ввода-вывода информации; основные директивы ассемблера, используемые для организации сегментов в программе и для реализации отдельной трансляции многомодульной программы); основы программирования на языке C++.

Дисциплина изучается на третьем курсе в 6 семестре.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (з.е.), 180 академических часа.

3.1. Объём дисциплины по видам учебных занятий (в часах)

Объем дисциплины	Всего часов	
	Очная форма обучения	Заочная форма обучения
Общая трудоемкость дисциплины	144	
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	90	
Аудиторная работа (всего):	68	
<i>в том числе:</i>		
лекции	16	
семинары, практические занятия	16	
лабораторные работы	16	
Самостоятельная работа обучающихся (всего)	60	
Курсовая работа	16	
Вид промежуточной аттестации обучающегося экзамен	36	

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела /темы дисциплины	Общая трудоём- кость всего (в часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоёмкость (в часах)				Формы текущего контроля успевае- мости
			Аудиторные учебные занятия			СРО	
			Лек	Сем/Пр	Лаб		
1.	Введение	6	6				
2.	Ассемблеры	15	8		6		Тест 1, отчет
3.	Макропроцессоры	1	1				
4.	Загрузчики	12	8		4		Тест 1, отчет
5.	Формальные языки и грамматики	4	4				
6.	Компиляторы	14	6		7		Тест 2, отчет
7.	Заключение	1	1				
8.	Консультации по курсовой работе	39		17		22	

4.2. Содержание дисциплины, структурированное по разделам
Лекционный курс

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Введение в системное программирование	
1.1	Этапы развития системного программного обеспечения.	Цели, задачи, место дисциплины в учебном процессе. Программное и информационное обеспечение. Классификация программного обеспечения ЭВМ. Связь системных компонентов с архитектурой ЭВМ. Состав системных компонентов.
1.2.	Операционная система как интерфейс между	Функции и сервисы операционной системы.

	пользователем и ЭВМ	
2.	Ассемблеры	
2.1.	Основные функции ассемблеров	Определение ассемблера как системной компоненты. Основные функции ассемблера. Основные функции двухпросмотрового ассемблера. Структура объектного модуля.
2.2.	Машинно-зависимые функции ассемблера и машинно-независимые функции ассемблера	Машинно-зависимые характеристики ассемблера: генерация машинных команд, выбор способа адресации, перемещение программ. Машинно-независимые характеристики ассемблера: средства определения имен, вычисление выражений, сегментирование и связывание программ. Пример ассемблирования программы. Транслятор TASM - встроенные имена, операции, опции. Оверлейный вызов программ на ассемблере. Формирование ассемблером адреса перехода. Способы передачи аргумента в модуль ассемблера.
2.3.	Структура заголовка obj-файла	Типы записей заголовка файла. Структура записей для всех типов. Формат записи по типам.
3.	Макропроцессоры	Основные функции. Машинно-независимые особенности. Конструктивное решение макропроцессоров.
4.	Загрузчики	
4.1.	Машинно-зависимые функции загрузчика	Основные функции абсолютного загрузчика. Машинно-зависимые функции относительного загрузчика: перемещение и связывание программ. Атрибуты

		директивы SEGMENT и их влияние на структуру загрузочного модуля.
4.2.	Машинно-независимые функции	Машинно-независимые функции: библиотечный автопоиск и подключение библиотек, управление процессом загрузки, создание программ оверлейной структуры. Заголовочный файл (header) и файл проекта в языке C++. Опции турбо-компоновщика TLINK. Особенности COM - файлов.
4.3.	Настраивающий и динамический загрузчик. Структура EXE - файлов	Настраивающий загрузчик. Динамический загрузчик. Статическое и динамическое связывание программ. Структура exe-файла в DOS и их загрузка в память. Структура Win - файлов. Заголовок файла. Форматы NE и PE. Принципы создания динамических библиотек DLL.
5.	Формальные языки и грамматики	
5.1.	Формальные языки и грамматики – основные понятия	Понятия: алфавит, лексика, синтаксис, семантика, грамматика языка, лексема, транслятор, компилятор, интерпретатор. Формальные языки и грамматики. Формальное определение грамматики. Форма Бекуса-Наура.
5.2	Распознаватели и их виды. Классификация грамматик	Распознаватели (компоненты и операции). Задача распознавания текста в трансляторах. Формальное определение грамматики. Типы грамматик. Классификация грамматик по Холмскому. Понятие вывода. Цепочки вывода. Классификация языков Способы задания правил грамматики. Виды

		распознавателей. Классификация распознавателей. Конечный и магазинный автоматы. Сентенциальная форма грамматики
6.	Компиляторы	
6.1.	Основные функции компилятора	Структура компиляторов и интерпретаторов. Отличительные особенности. Основные функции компилятора: лексический анализ, синтаксический разбор, семантический контроль, генерация объектного кода. Виды переменных. Распределение памяти под переменные. Методы грамматического разбора.
6.2.	Машинно-зависимые особенности компилятора	Машинно-зависимая оптимизация как функция компилятора. Генерация кода как функция компилятора.
6.3.	Машинно-независимые особенности компилятора. Интерпретаторы	Распределение памяти как функция компилятора. Классы памяти переменных; Диапазон доступа переменных. Обработка структурированных переменных как функция компилятора. Машинно-независимая оптимизация кода как функция компилятора. Обработка блочных конструкций языка программирования как функция компилятора. Интерпретаторы – основные функции.
7.	Заключение	Основные направления эксплуатации и перспективы развития системного программного обеспечения.

		Мобильность программного обеспечения. Эффективность программ.
--	--	---

Лабораторные занятия

№	Наименование раздела /темы дисциплины	Название лабораторной работы
1.	Ассемблеры	
1.1.	Машинно-зависимые функции ассемблера и машинно-независимые функции ассемблера	Изучение функций транслятора TASM на примере отдельной трансляции программ
1.2.	Структура заголовка obj-файла	Структуры объектного файла
2.	Загрузчики	
2.1.		Создание оверлейных программ на ассемблере
3.	Компиляторы	
3.1.		Опции компилятора ВСС для оптимизация программ С++

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

Магда Ю.С. Использование ассемблера для оптимизации программ на С++. – СПб.: БХВ-Петербург, 2004. – 496 с.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

1.	Ассемблеры	ОПК-5 (знать, уметь)	Тест 1
2.	Загрузчики	ОПК-5 (знать, владеть)	Тест 1
3.	Компиляторы	ОПК-5 (знать, уметь)	Тест 2

6.2. Типовые контрольные задания или иные материалы

6.2.1. Экзамен

а) типовые вопросы (задания):

Билет 1

1. Вопрос для проверки уровня обученности ЗНАТЬ:
Форма Бэкуса – Наура.
2. Вопрос для проверки уровня обученности УМЕТЬ:
Анализ структуры заголовка obj-файла.
3. Вопрос (задача/задание) для проверки уровня обученности ВЛАДЕТЬ:
Определение классов памяти переменных в исходной программе на C++.

Билет 2

1. Вопрос для проверки уровня обученности ЗНАТЬ:
Машинно-зависимые функции компоновщика.
2. Вопрос для проверки уровня обученности УМЕТЬ:
Способы передачи параметров в модуль ассемблера.
3. Вопрос (задача/задание) для проверки уровня обученности ВЛАДЕТЬ:
Оптимизация кода программы как опция компилятора.

б) критерии оценивания компетенций (результатов):

К экзамену допускаются студенты, отчитавшиеся по всем формам контроля (два теста, контрольная работа, курсовая работа). За курсовую работу ставится отдельно зачет с оценкой. Общая оценка за экзамен складывается из баллов за две точки промежуточного контроля в течение семестра и баллов итогового контроля.

в) описание шкалы оценивания:

Подробный ответ с примерами на экзамене каждый вопрос (всего три вопроса в билете) оценивается в 30% и один дополнительный вопрос в 10%. Ответ оцениваются максимально в 100%, это дает 40 баллов итогового контроля. Полученные проценты проставляются в рейтинг, где они суммируются с баллами промежуточного контроля. Максимальная оценка составляет 100 баллов.

6.2.2. Тесты

а) типовые задания

Применяется компьютерное тестирование закрытого типа с использованием выбора правильного ответа из множества предлагаемых вариантов. В примерах тестовых заданий из базы вопросов, используемых при компьютерном тестировании бакалавров, приводится вопрос и пять альтернативных ответов, из которых только один правильный.

1. Что делает опция Tasm: /a?

- Задает алфавитный порядок сегментов

- Разрешает перекрестные ссылки в листинге
- Устанавливает максимальный размер стека
- Генерирует файл листинга
- Генерирует .map-файл

2. Что не содержится в листинге?

- Запись - заголовок
- Машинный (объектный) код команды
- Смещение в кодовом сегменте
- Таблица символов
- Сообщения об ошибках

3. Что не делает TASM?

- Связывание или объединение двух программ
- Преобразование мнемонических кодов в их машинный эквивалент
- Преобразование мнемонических операндов в машинные адреса
- Построение машинных команд
- Преобразует константы в исходной программе во внутреннее машинное представление

5. Что такое автопоиск в библиотеках?

- Способ разрешения внешних ссылок
- Способ разрешения перекрёстных ссылок
- Создание перекрёстных ссылок
- Определение внешних ссылок
- Определение реальных адресов внешних ссылок

6. Каким процессом является перемещение?

- Процессом модификации программы так, что она может загружаться с адреса, отличного от первоначального
- Процессом размещения программы в ОП для выполнения
- Процессом, обеспечивающим объединение 2-х и более отдельно транслируемых программ
- Процессом компиляции загружаемой программы
- Процессом трансляции загружаемой программы

б) критерии оценивания компетенций (результатов):

для сдачи теста необходимо набрать не менее 80% правильных ответов на вопросы, заданные в тесте. Если результат хуже, испытуемый будет проходить тестирование в другое время повторно.

в) описание шкалы оценивания:

процент правильных ответов будет заноситься в рейтинговую систему по точке контроля в графу «контроль лекций», например, при 20 заданных студенту вопросов применяется следующая шкала:

Число правильных ответов	Процент в рейтинге
20	100
19	95
18	90
17	85
16	80

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков

Оценивание знаний с использованием балльно-рейтинговой системы заключается в занесении данных в рейтинговую таблицу в Excel. Заполняются графы по точкам контроля Т1 и Т2. В колонку «лекции» записывается процент выполнения теста, в колонку «лаб» – процент выполнения работ на указанную дату. Итоговый контроль записывается в отдельную графу. Полученное суммарное значение является итоговым баллом по дисциплине. Оно проставляется в зачетную книжку.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная литература:

1. Молчанов А.Ю. Системное программное обеспечение. Учебник для вузов. 3-е изд.– СПб.: Питер, 2010. -400 с. (5 экз.)
2. Тищенко В.И. Основные функции системных компонентов. Учебное пособие по курсу «Системное программирование».- Обнинск: ИАТЭ.- 2014.- 48с. (25 экз.)

б) дополнительная литература:

3. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации: учебник для вузов.- СПб.: Питер, 2011. -560 с. (1 экз.,ч,з.)
4. Магда Ю.С. Использование ассемблера для оптимизации программ на C++.- СПб.: БХВ-Петербург, 2004. – 496 с. (1 экз.)
5. Молчанов А.Ю. Системное программное обеспечение. Лабораторный практикум. – СПб.: Питер, 2005. -284 с. (1 экз. ч.з.)
6. Прата С. Язык программирования C++. Лекции и упражнения. Учебник. – СПб.: ООО «ДиалогСофтЮП», 2003. – 1104 с. (1 экз. ч.з.)
7. Столлингс В. Операционные системы. - М.: Издательский дом «Вильямс», 2004. -848с.
8. Тищенко В.И. Основы системного программирования для студентов технического вуза. Учебное пособие. - Обнинск: ИАТЭ, 2006. - 72с. (50 экз.)
9. Тищенко В.И. Разработка резидентных и оверлейных программ. Лабораторный практикум по курсу “Системное программирование” . - Обнинск: ИАТЭ, 2002. - 56 с. (200 экз.)

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- 1 Научная электронная библиотека: <http://eLIBRARY.RU>.
2. Издательство «Лань». Электронно-библиотечная система. <http://e.lanbook.com>.

9. Методические указания для обучающихся по освоению дисциплины

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины.. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации или в процессе выполнения лабораторного практикума.
Лабораторная работа	Методические указания по выполнению лабораторных работ.
Курсовая работа	Изучение научной, учебной, нормативной и другой литературы. Отбор необходимого материала; формирование выводов и разработка конкретных рекомендаций по решению поставленной цели и задачи; проведение практических исследований по данной теме. Инструкция по выполнению требований к оформлению курсовой работы находится в методических материалах по дисциплине.
Консультации по курсовой работе (практические занятия)	На практических занятиях проводятся индивидуальные консультации студента по выбору темы, содержанию работы, обсуждается план работы, состав реферативной и практической частей работы, порядок оформления работы.
Подготовка к экзамену	При подготовке к экзамену необходимо ориентироваться на конспекты лекций, слайд–презентации лекций, рекомендуемую литературу и методические пособия.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. Электронный учебник установлен на сайте кафедры «Компьютерные системы, сети и технологии» ИАТЭ НИЯУ МИФИ и доступен для работы студентов во всех дисплейных классах кафедры по ссылке в Internet explorer: <http://ksstlocalsite.campus.iate.obninsk.ru/>, затем в главном меню выбрать «Ресурсы», затем– «Учебник».
2. Система компьютерного тестирования.
3. Использование слайд – презентаций, разработанных автором для проведения лекционных и лабораторных.

11. Описание материально-технической базы, необходимой для

осуществления образовательного процесса по дисциплине

Лекционные занятия проводятся в аудитории, оборудованной проекционной установкой в комплекте с компьютером. Лабораторные работы выполняются в дисплейном классе на компьютерах типа IBM PC с установленной операционной системой Windows 7, эмулятор для работы с Dos программами DosBox, программный пакет фирмы Borland, в составе которого есть транслятор с языка Ассемблер TASM, транслятор с языка C++ BCC, компоновщик программ TLINK.

12. Иные сведения и материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Для данной дисциплины автором программы разработаны: электронный учебник и система компьютерного тестирования, которые связаны в единый комплекс средств обучения. Система тестирования может использоваться в трех режимах: преподавателем для контроля знаний студентов, студентами в процессе обучения совместно с электронным учебником или для самоконтроля знаний при самостоятельной работе. Система установлена на сайте кафедры «Компьютерные системы, сети и технологии» ИАТЭ НИЯУ МИФИ и доступна для работы студентов во всех дисплейных классах кафедры по ссылке: <http://ksstlocalsite.campus.iate.obninsk.ru/>, затем в главном меню выбрать «Ресурсы», затем – «Тестер» или «Учебник». Режимы работы системы устанавливаются преподавателем.

В интерактивных режимах по дисциплине применяются:

1. Решение ситуационных задач (лабораторные работы № 1, 2) – 8 часов.

Формируются навыки применения полученных знаний для решения ситуационной задачи студентами индивидуально или в группах. Организуется обсуждение преподавателя и студентов полученных результатов и способов их применения на практике.

2. Поисковая лабораторная работа (лабораторные работы № 3, 4) – 8 часов.

Формируются умения делать практические выводы на основе данных, полученных в результате трансляции и компоновки программ, навыки использования методов сопряжения программ и методов оптимизации программ для эффективного использования вычислительных ресурсов, умения формулирования обобщений и закономерностей.

Всего аудиторных занятий в интерактивной форме – 16 часов (23,5 % от аудиторных занятий).

12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Примерная тематика для выполнения курсовых работ:

- Средства профилирования программ.
- Набор компиляторов GNU Compiler Collection для различных языков программирования.
- Сравнительные характеристики автоматических дизассемблеров.
- Организация интерфейса с процедурой на ассемблере.

- Связь ассемблера с другими языками программирования.

Формы организации самостоятельной работа студента:

1. Консультации по подбору литературы по теме курсовой работы.
2. Обсуждение с преподавателем плана работы.
3. Консультации по реферативной части работы.
4. Консультации по написанию практической части курсовой работы.
5. Консультации по оформлению курсовой работы.
6. Комиссионная защита работы.

12.3. Краткий терминологический словарь

Перечень ключевых слов и понятий учебной дисциплины:

Абсолютный загрузчик
Автопоиск в библиотеках
Ассемблер как системная компонента
Встроенные имена
Генерация кода
Грамматика языка
Грамматический разбор
Диапазон доступа переменных
Динамический загрузчик
Заголовок obj-файла
Заголовок файла в MZ-формате
Загрузчик
Классы памяти переменных
Конечный автомат
Лексический анализ
Макропроцессор
Настраивающий загрузчик
Оверлейная структура программы
Оптимизация программ
Опции редактора связей
Опция транслятора
Передачи аргумента в модуль
Перемещение программы
Распознаватель