

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Саровский физико-технический институт -

филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(СарФТИ НИЯУ МИФИ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЭЛЕКТРОНИКИ
Кафедра «Вычислительной и информационной техники»

УТВЕРЖДАЮ

Декан ФИТЭ, к.ф-м.н., доцент

_____ В.С. Холушкин

«__» _____ 2020 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

наименование дисциплины

Направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Наименование образовательной программы	Вычислительные машины, комплексы, системы и сети
Квалификация (степень) выпускника	бакалавр
Форма обучения	очная

Программа одобрена на заседании кафедры Зав. кафедрой ВИТ

Протокол № _____ от _____ 2020 г. _____ В.С. Холушкин

«__» _____ 2020 г.

г. Саров, 2020 г.

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с Семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ

В.С. Холушкин

Семестр	В форме практической подготовки	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	КР/КП	Форма(ы) контроля, экз./зач./ЗСО/
4	32	4	144	32	-	32	53	-	Э
ИТОГО	32	4	144	32	-	32	53	-	27

АННОТАЦИЯ

Курс посвящен изучению теоретических и практические основ системного программного обеспечения. Изучаются способы и методы разработки эффективных алгоритмов и программных продуктов в рамках создания системного программного обеспечения. Главная цель преподавания дисциплины – подготовка специалиста, владеющего фундаментальными знаниями и практическими навыками в области разработки и успешного применения прикладного системного обеспечения.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цели и задачи освоения учебной дисциплины

Целью дисциплины «Системное программное обеспечение» является ознакомление студентов с основными понятиями, методами построения, способами использования, инструментами системного программного обеспечения.

Задачи дисциплины:

- сформировать у студентов системное представление о программном обеспечении компьютерных средств, сетей и информационных систем;
- рассмотреть на примерах виды системного программного обеспечения;
- научить студентов анализировать основные подходы к проектированию и реализации операционных систем;
- научить основам проектной деятельности.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Системное программное обеспечение» входит в вариативную часть профессионального цикла дисциплин, определяемых вузом; она непосредственно связана с такими дисциплинами как «Организация ЭВМ», «Операционные системы», «Сети ЭВМ и телекоммуникации», «Компьютерная графика».

Содержание дисциплины носит интегративный характер, в котором используются достижения целого ряда дисциплин: «Информатика», «Языки программирования», «Программирование».

Дисциплина «Системное программное обеспечение» является основой для изучения дисциплин профессионального цикла: «Технология программирования», «Параллельное программирование», «Параллельные вычисления», «Мультипроцессорные системы» Изучение данной дисциплины позволяет дать студентам базу, необходимую для успешного усвоения материала учебных дисциплин, связанных с использованием современных информационных технологий и ВТ.

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Общепрофессиональные компетенции (ОПК)

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	З-ОПК-8 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения У-ОПК-8 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули В-ОПК-8 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	З-ОПК-2 Знать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых при решении задач профессиональной деятельности У-ОПК-2 Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности В-ОПК-2 Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы					Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС			
			32		32	53			
Семестр 4									
Раздел 1.									
1.1.	Тема 1. . Краткая история эволюции вычислительных систем. Структура и их основные компонент. Структура и виды прикладного и системного программного обеспечения	1	4		4	7	УО, Защита ЛР	4	

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы					Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС			
			32		32	53			
1.2	Тема 2. Программное обеспечение для Internet. Системы баз данных/ Информационно-поисковые системы.	2	4			4	7	Защита ЛР	4
Раздел 2.									
2.1	Тема 1 Состав и схема работы систем программирования. Программное, языковое и информационное обеспечение системы программирования	3,4	4			4	7	Защита ЛР	4
2.2	Тема 2. Классификация, назначение и основные функции операционных систем. Управление задачами	5,6	4			4	7	Защита ЛР	4
2.3	Тема 3 Управление памятью. Управление вводом-выводом	7,8	4			4	7	Защита ЛР	3
2.4	Тема 4. Организация внешней памяти. Организация параллельных Взаимодействующих вычислений	9,10	4			4	9	Защита ЛР	3
Рубежный контроль		11						СР	5
Раздел 3.									
3.1	Тема 1 Сети ЭВМ	12,13	4			4	6	Защита ЛР	4
3.2	Тема 2. Искусственные нейронные сети	14,15	8			8	6	Защита ЛР	4
Рубежный контроль		16						СР	10
Промежуточная аттестация							9	-	50
Посещаемость									5
Итого:			32			32	53	-	100

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

СР – самостоятельная работа(решение задачи на заданную тему)

РГР – расчетно – графическая работа

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
4 семестр		
Раздел 1		
1.1	Тема 1. Краткая история эволюции вычислительных систем. Структура и их основные компоненты. Структура и виды прикладного и системного программного обеспечения	Введение. Краткая история эволюции вычислительных систем. Структура и основные компоненты вычислительной машины. Структура программного обеспечения. Понятие о системном и прикладном программном обеспечении. Структура системного программного обеспечения. Виды прикладного программного обеспечения.
1.2	Тема 2. Программное обеспечение для Internet. Системы баз данных. Информационно-поисковые системы.	Программное обеспечение для Internet. Разработка собственных Web страниц. Основные понятия баз данных. Обзор информационно-поисковых систем, систем искусственного интеллекта, экспертных систем.
Раздел 2		
2.1	Тема 1 Состав и схема работы систем программирования. Программное, языковое и информационное обеспечение системы программирования.	Основные понятия, связанные с системами программирования. Состав и схема работы систем программирования. Программное, языковое и информационное обеспечение систем программирования.
2.2	Тема 2. Классификация, назначение и основные функции операционных систем. Управления задачами.	Основные понятия, связанные с операционными системами. Основные характеристики операционных систем. Состав и схема работы операционных систем. Классификация операционных систем. Назначение и их основные функции. Управляющая программа. Управление задачами. Планирование и диспетчеризация процессов и задач
2.3	Тема 3. Управление па-	Управление памятью. Виртуальная память, мультипро-

	мятью. Управление вводом-выводом.	граммный режим. Управление вводом-выводом в операционных системах. Основные концепции и режимы работы.
2.4	Тема 4. Организация внешней памяти. Организация параллельных взаимодействующих вычислений	Организация внешней памяти. Кэширование операций ввода-вывода. Организация параллельных взаимодействующих вычислений. Независимые и взаимодействующие вычислительные процессы. Средства синхронизации. Мониторы Хоара.
Раздел 3.		
3.1	Тема 1. Сети ЭВМ	Основные понятия, связанные с сетями ЭВМ. Сетевое программное обеспечение. Протоколы передачи информации.
3.2	Тема 2. Искусственные нейронные сети	Основные понятия, связанные с искусственными нейронными сетями (ИНС). Программное обеспечение искусственных нейросетей. Область применения ИНС.

Лабораторные занятия

Лабораторный практикум предполагает выполнение лабораторных работ по основным разделам дисциплины. Темы лабораторных работ приведены в следующей таблице. Инструментальная среда для выполнения лабораторных работ: QtCreator.

№	Примерные темы лабораторных занятий
1	Разработка консольных приложений
2	Разработка служб ОС
3	Разработка клиентских и серверных приложений

4.3 Перечень учебно-методического обеспечения для самостоятельной работы студентов

При изучении дисциплины используются следующие виды самостоятельной работы:

- самостоятельный поиск литературы по разделам и темам курса;
- изучение материала по дополнительным разделам дисциплины;
- изучение литературы и подготовка к выполнению лабораторных работ, курсовых работ;
- подготовка к тестированию, контрольным работам, написанию рефератов;

- подготовка к зачету, экзаменам.

Форма контроля: отчет по лабораторным работам и их защита, защита курсовых работ.

Учебно-методические пособия:

1. Донован, Дж. Системное программирование [Текст]. Мир. 2005.
2. Дональд, Э. Кнут. Искусство программирования. том 1. Основные алгоритмы, 3-е изд.: Пер. с англ. [Текст] / Э. Кнут Дональд. М.: Издательский дом «Вильямс», 2006. - 720 с.
3. Камерон, Хьюз. Параллельное и распределенное программирование с использованием C++ [Текст] / Хьюз Камерон. - Вильямс. - 2004. - 672 с.
4. Орлов, С. А. Технологии разработки программного обеспечения [Текст] / — СПб.: Питер, 2003.
5. Орлов, С.А. Организация ЭВМ и систем [Текст] / С.А. Орлов, Б.Я. Цилькер. Питер, 2006.
6. Гагарина, Л. Г., Кокорева Е. В., Виснадул Б. Д. Технология разработки программного обеспечения [Текст] / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. М.: ИД «ФОРУМ»; ИНФРА-М, 2008.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

5.1. Паспорт фонда оценочных средств по дисциплине

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Раздел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 4				

Раздел 1	Тема 1. Краткая история эволюции вычислительных систем. Структура и их основные компоненты. Структура и виды прикладного и системного программного обеспечения	ОПК-2К-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	УО 1
	Тема 2. Программное обеспечение для Internet. Системы баз данных. Информационно-поисковые системы.		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР3
Раздел 2	Тема 1 Состав и схема работы систем программирования. Программное, языковое и информационное обеспечение системы программирования	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР5
	Тема 2. Классификация, назначение и основные функции операционных систем. Управление задачами		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР7
	Тема 3 Управление памятью. Управление вводом-выводом		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР9
	Тема 4. Организация внешней памяти. Организация параллельных взаимодействующих вычислений		3-ОПК-8; У-ОПК-8; В-ОПК-8 3-ОПК-9; У-ОПК-9; В-ОПК-9	Защита ЛР10
Рубежный контроль		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	СР 11
Раздел 3	Тема 1 Сети ЭВМ	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР13
	Тема 2. Искусственные нейронные сети		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР 15
Рубежный контроль		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	
Промежуточная аттестация		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Экзамен

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

5.2.1. Оценочные средства для текущего контроля

5.2.1.1. Примерные вопросы для устного опроса (УО)

- Основные команды ОС
- Создание Shell-скриптов, основные конструкции
- Практическое создание Shell-скриптов
- Работа с информационным наполнением ОС
- Команды работы с файлами
- Команды работы с каталогами
- Свободное программное обеспечение в среде Unix

5.2.1.2. Примерные темы и вопросы для самостоятельной работы (СР)

- Разработать Shell-скрипт, выполняющий заданные действия
- Реализовать сборку свободно распространяемого пакета ПО
- Разработать Shell-скрипт, выполняющий запуск процессов в основном и в фоновом режимах
- Разработать Shell-скрипт, осуществляющий выборку данных и сортировку из текстовых файлов каталога
- Разработать Shell-скрипт, создающий файлы с заданными режимами доступа
- Спроектировать и реализовать набор «команд» согласно заданным функциям

5.2.2. Оценочные средства для рубежного контроля

5.2.2.1. Примерные задания для решения задач по заданной теме

Лабораторная работа №1

Написать консольное приложение Win32 на языке C++, выполняющее действия, описанные в конкретном задании. При отсутствии необходимых аргументов командной строки программа должна выдавать информацию о себе, включая подсказку по требуемым аргументам командной строки. Реализовать в программе корректную обработку ошибок при вызове любых API функций: в случае возникновения ошибки выдать код ошибки, имя вызвавшей ошибку функции и текстовое описание ошибки, после чего завершить работу программы. Программа должна осуществлять вывод всех сообщений на стандартный выходной поток. Работа с текстовыми строками в программах должна осуществляться с использованием Unicode.

Варианты заданий на лабораторную работу

1. Программа копирования файлов. Осуществляет копирование одного файла в другой файл. Имена файлов указываются в командной строке. Программа должна проверять наличие конечного файла, в который осуществляется копирование, и выдавать пользователю запрос на его перезапись. В случае успешного копирования программа должны выводить соответствующее сообщение.
2. Программа записи дерева каталогов в файл. Осуществляет рекурсивный поиск файлов по маске, начиная с заданного каталога, результаты поиска записываются в файл. Начальный каталог, маска для поиска и имя файла, в который осуществляется запись результатов, указываются в командной строке. Программа должна записывать следующую информацию в файл: имя и путь найденного файла, его размер и атрибуты.

3. Программа поиска текстовой строки в файле. Осуществляет поиск текстовой строки в файлах, заданных каталогом и маской (нерекурсивный поиск). Каталог файлов, маска поиска и текстовая строка указываются в командной строке. Текстовая строка может содержать пробелы, поэтому при указании в командной строке ее следует заключать в кавычки. Программа должна выводить имена файлов, в которых найдена строка.
4. Программа сравнения двоичных файлов. Осуществляет побайтовое сравнение двух файлов и выдает их различия. Имена файлов указываются в командной строке. Если файлы имеют разный размер, программа должна выдать запрос на их дальнейшее сравнение и в случае положительного ответа пользователя сравнивать файлы по длине меньшего из них. Алгоритм работы программы заключается в побайтовом сравнении двух файлов и выводе двух байт в случае их различия. Байты выводятся в столбец в шестнадцатеричной системе счисления.
5. Программа преобразования двоичного файла в текстовый. Осуществляет преобразование двоичного файла в текстовый файл, содержащий коды байтов первого файла в шестнадцатеричной форме. Имена файлов указываются в командной строке. Шестнадцатеричные коды в выходном файле пишутся в столбец по 16 байт в строке. Между кодами соседних байт ставится пробел. Пример вывода: 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00.
6. Программа преобразования текстового файла в двоичный. Осуществляет преобразование описанного выше текстового файла обратно в двоичный. Имена файлов указываются в командной строке. Программа считывает шестнадцатеричные коды байт из текстового файла и записывает сами байты в двоичный файл. При считывании из текстового файла необходимо осуществлять простейшую проверку синтаксиса.
7. Программа разбиения двоичного файла на несколько частей. Осуществляет разбиение произвольного двоичного файла на несколько равных частей, каждая из которых записывается в отдельный файл. Имя файла для разбиения и количество частей указываются в командной строке. Программа формирует набор новых файлов с информацией из исходного файла. Имя нового файла формируется из имени исходного файла, в конце которого добавляется суффикс “.NNN”, где NNN – номер части файла. Например, если имеется файл размером 1000 байт и его разбивают на 10 частей, то формируется 10 новых файлов по 100 байт каждый. Если размер файла не делится нацело, последняя часть должна содержать остаток.

8. Программа «склеивания» нескольких файлов. Осуществляет последовательную запись информации из нескольких файлов, задаваемых маской, в один указанный файл. Маска файлов и имя выходного файла вводятся в командной строке. Для обеспечения корректной последовательности записи в выходной файл, имена найденных по маске файлов должны быть отсортированы по алфавиту.
9. Программа перекодировки текстового файла с русским текстом в транслит. Осуществляет перекодировку текста на русском языке из одного файла в текст на транслите, записываемый в другой файл. Имена файлов указываются в командной строке.
10. Программа переименовывания файлов с русскими именами. Осуществляет переименовывание файлов, имена которых содержат русские буквы в файлы с именами на транслите. Файлы задаются маской поиска в каталоге (нерекурсивный поиск). Имя каталога и маска поиска указываются в командной строке. Если в процессе переименования файла обнаруживается уже существующий файл с таким именем, программа должна выдавать имя файла и запрос на перезапись.

Лабораторная работа №2

Написать службу Windows NT, выполняющую действия, указанные в конкретном задании. Служба должна воспринимать следующие сигналы управления: останов, приостановка, возобновление выполнения. Реализовать корректную обработку ошибок при вызове API функций: при возникновении ошибки служба должна создать файл error.log в каталоге с исполняемым модулем службы. В файл записывается код ошибки и ее текстовое описание, после чего служба должна корректно перейти в состояние останова. Все задания ориентированы на работу с файлами и процессами. Настройки службы хранятся в виде текстового файла config.txt в каталоге с исполняемым модулем службы. При старте служба должна производить считывание настроек из этого файла.

Варианты заданий на лабораторную работу

1. Отслеживание изменений в каталоге и запись их в файл. Служба должна сканировать изменения в заданном каталоге с интервалом в 1 сек., и производить запись при появлении нового файла(ов) в заданный текстовый файл. Имя файла и каталог с записями определяются в файле настройки. Формат записей текстового файла: имя созданного файла, его размер.
2. Отслеживание изменений в списке запущенных процессов и запись их в файл. Служба должна сканировать изменения в списке запущенных в системе процессов

- с интервалом в 0,1 сек., и производить запись при создании нового процесса(ов) в заданный текстовый файл. Имя файла с записями определяется в файле настройки. Формат записей текстового файла: имя созданного процесса, его идентификатор.
3. Отслеживание изменений в каталоге и запись их в файл. Служба должна сканировать изменения в заданном каталоге с интервалом в 1 сек., и производить запись при удалении какого-либо файла(ов) в заданный текстовый файл. Имя файла и каталог с записями определяются в файле настройки. Формат записей текстового файла: имя удаленного файла, его размер.
 4. Отслеживание изменений в списке запущенных процессов и запись их в файл. Служба должна сканировать изменения в списке запущенных в системе процессов с интервалом в 0,1 сек., и производить запись при завершении какого-либо процесса(ов) в заданный текстовый файл. Имя файла с записями определяется в файле настройки. Формат записей текстового файла: имя завершенного процесса, его идентификатор.
 5. Отслеживание изменений в файле и перекодировка его в транслит. Служба должна отслеживать изменение заданного текстового файла с интервалом в 1 сек. и при наличии изменений выполнять его перекодировку в транслит. Изменения в файле характеризуются датой и временем его последней модификации. Имя файла определяется в файле настройки. Запись перекодированного файла осуществлять в новый файл с таким же именем + расширение “.trans”.
 6. Сканирование каталога и запись его структуры в файл. Служба должна производить сканирование каталога с интервалом в 1 сек. и производить запись структуры каталога в заданный текстовый файл (файл каждый раз заново перезаписывается). Имя файла и каталог с записями определяются в файле настройки. Формат записей текстового файла: имя файла в каталоге, его размер.
 7. Сканирование списка запущенных процессов и запись его в файл. Служба должна производить сканирование списка запущенных в системе процессов с интервалом в 0,1 сек. и производить запись списка процессов в заданный текстовый файл (файл каждый раз заново перезаписывается). Имя файла с записями определяется в файле настройки. Формат записей текстового файла: имя процесса, его идентификатор.
 8. Сканирование списка процессов и завершение определенного процесса. Служба должна производить сканирование списка запущенных в системе процессов с интервалом в 0,1 сек. и производить завершение заданного процесса. Имя процесса задается в файле настройки. При невозможности завершения процесса служба

должна дописывать код ошибки и ее текстовое описание в файл error.log, но при этом продолжать работу.

9. Отслеживание работы процесса. Служба должна производить сканирование списка запущенных в системе процессов с интервалом в 0,1 сек. и производить запуск заданного процесса, если он отсутствует в списке. Имя процесса задается в файле настройки. При невозможности запуска процесса служба должна дописывать код ошибки и ее текстовое описание в файл error.log, но при этом продолжать работу.
10. Отслеживание изменений в файле и копирование его в заданный каталог. Служба должна отслеживать изменение заданного файла с интервалом в 1 сек. и при наличии изменений выполнять его копирование в заданный каталог. Изменения в файле характеризуются датой и временем его последней модификации. Имя файла и каталог определяется в файле настройки. При копировании файла осуществлять его перезапись в папке назначения.

Лабораторная работа №3

Написать клиентское и серверное приложения в виде консольных приложений Win32 на языке C++, выполняющие действия, описанные в конкретном задании. Оба приложения должны использовать интерфейс socket API для обмена данными по протоколу TCP. Клиентское приложение должно подключаться к серверному при запуске, причем имя хоста и порта для подключения указываются в командной строке. Хост может указываться как DNS-именем, так и IP-адресом. Серверное приложение должно при запуске переходить в режим ожидания клиентских подключений. Порт, на котором происходит ожидание подключений, вводится в командной строке при запуске серверного приложения. Для упрощения задачи положим, что серверное приложение может обрабатывать запросы только одного клиента в каждый конкретный момент времени. При отсутствии необходимых аргументов командной строки приложение (клиентское и серверное) должно выдавать информацию о себе, включая подсказку по требуемым аргументам командной строки. Реализовать в приложениях корректную обработку ошибок при вызове любых API функций (Win32 API и socket API): в случае возникновения ошибки выдать код ошибки, имя вызвавшей ошибку функции и текстовое описание ошибки, после чего завершить работу приложения. Приложения должны осуществлять вывод всех сообщений на стандартный выходной поток. Работа с текстовыми строками в приложениях должна осуществляться с использованием Unicode.

Варианты заданий на лабораторную работу

1. Пользователь вводит имя папки удаленной машины, серверное приложение присылает список файлов этой папки и их размеры, клиентское приложение отображает список. Если заданная папка не найдена, серверное приложение должно прислать код ошибки и ее текстовое описание.
2. Пользователь вводит одну из двух возможных команд: list и kill <id_процесса>. По первой команде серверное приложение присылает список запущенных на удаленной машине процессов, список отображается клиентским приложением. По второй команде серверное приложение завершает процесс с заданным идентификатором. Если процесс с заданным идентификатором не найден или его невозможно завершить, серверное приложение должно прислать код ошибки и ее текстовое описание.
3. Пользователь вводит имя текстового файла удаленной машины, серверное приложение присылает содержимое файла, которое отображается клиентским приложением. Если указанный файл не найден, серверное приложение должно прислать код ошибки и ее текстовое описание.
4. Пользователь вводит имя файла, возможно с маской. Серверное приложение ищет указанный файл рекурсивно во всех каталогах диска C: и присылает список найденных файлов с указанием полного пути и размера. Этот список отображается клиентским приложением.
5. Пользователь вводит имена двух файлов на удаленной машине. Серверное приложение копирует первый файл во второй и присылает результат выполнения этой операции: признак успешного копирования или код ошибки с текстовым описанием.

5.2.3. Оценочные средства для промежуточной аттестации

5.2.3.2. Примерные вопросы к экзамену

1. Структура программного обеспечения.
2. Виды системного программного обеспечения.
3. Текстовые редакторы, назначение, возможности. Наиболее распространенные в настоящее время текстовые редакторы.
4. Текстовый редактор TeX. Основные возможности.
5. Текстовый редактор Word. Основные возможности.
6. Табличные редакторы, назначение, возможности. Наиболее распространенные в настоящее время табличные редакторы.

7. Табличный редактор EXCEL. Основные возможности.
8. Области применения компьютерной графики. Наиболее распространенные в настоящее время графические системы.
9. Графические редакторы.
10. Системы автоматизированного проектирования (САПР).
11. Геоинформационные системы (ГИС).
12. Издательские системы, обзор.
13. Интегрированные системы.
14. Автоматизированные рабочие места (АРМ). Примеры АРМ.
15. Пакеты прикладных программ (ППП). Системное, функциональное и информационное наполнение ППП. Примеры ППП.
16. Системы баз данных. Язык определения (описания) и язык манипулирования данными.
17. Иерархические, сетевые и реляционные базы данных. Уровни представления данных.
18. СУБД. Наиболее распространенные в настоящее время СУБД.
19. Информационно-справочные системы. Информационный образ документа, ключевые слова, тезаурус. Поиск нужных документов, релевантность.
20. Проблемы искусственного интеллекта (ИИ). Классификация систем ИИ.
21. Экспертные системы, назначение, возможности.
22. Методы представления знаний в системах ИИ, базы знаний. Взаимодействие пользователя с системами ИИ.
23. Программное обеспечение для Internet: почтовые программы
24. Программное обеспечение для Internet: браузеры и HTML-редакторы.
25. Системы программирования, состав и схема работы.
26. Системы программирования: редакторы текстов, компиляторы и интерпретаторы, отладчики.
27. Системы программирования: редактор внешних связей, загрузчик.
28. Наиболее популярные системы программирования.
29. Операционные системы: исторический обзор, назначение и основные функции.
30. В чём состоит иерархическая организация памяти ЭВМ?
31. Для чего нужны регистры, кэш-память?
32. Что означает термин «расслоение оперативной памяти»?
33. Каковы методы организации и управления основной памятью?

34. Как происходит статическое и динамическое распределение оперативной памяти между задачами?

35. Что такое фрагментация памяти?

36. Что такое релокация программ?

37. Для чего нужна виртуальная память?

38. В чём заключается механизм подкачки?

39. Как организована виртуальная память?

40. Как происходит работа с внешними устройствами?

41. Понятие канала.

42. Какие Вам известны структуры управления дисками?

43. Что такое файловая система? Файлы, их структура, типы.

44. Логические и физические записи файлов. Что такое блокирование?

45. Как происходит совмещение работы центрального процессора и периферийных устройств ЭВМ?

46. Для чего нужна система прерываний? Какие классы прерываний Вам известны?

47. В чём состоит защита памяти?

48. Как происходит вызов супервизора?

49. Дайте понятие процесса. Какие Вам известны состояния процесса?

Изменения состояний.

50. Как осуществляется синхронизация параллельных процессов?

51. Дайте понятие критической секции. Что такое взаимоисключение?

52. Что такое семафоры и примитивы управления параллельными процессами?

53. Пакетный и диалоговый режимы работы.

54. Работа ЭВМ в режиме мультипрограммирования, деления времени и реального времени.

55. Наиболее распространенные современные операционные системы для ЭВМ различных классов.

56. Управляющая программа операционной системы. Язык директив. Командные файлы.

57. Иерархическая организация памяти ЭВМ: регистры, кэш-память, оперативная и внешняя память.

58. Расслоение оперативной памяти.

59. Методы организации и управления основной памятью.

60. Статическое и динамическое распределение оперативной памяти между задачами.
61. Фрагментация памяти. Релокация программ.
62. Виртуальная память, механизм подкачки. Страничная организация виртуальной памяти.
63. Организация работы с внешними устройствами. Понятие канала.
64. Внешняя память. Стратегии управления дисками.
65. Файловая система. Файлы, их структура, типы и способы организации.
66. Логические и физические записи файлов, блокирование.
67. Совмещение работы центрального процессора и периферийных устройств ЭВМ. Мультипрограммный режим работы.
68. Система прерываний, классы прерываний. Защита памяти, команды прерывания (вызов супервизора).
69. Понятие процесса. Состояния процесса, изменения состояний.
70. Параллельные процессы, их синхронизация.
71. Понятие критической секции. Семафоры, примитивы управления параллельными процессами.
72. Задача о поставщике и потребителе. Взаимодействие параллельных процессов.
73. Проблема тупиков и методы ее решения.
74. Сетевое программное обеспечение.

5.3. Шкалы оценки образовательных достижений

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – «отлично»	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно,

			четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69	3 – «удовлетворительно»	E	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
60-64			
Ниже 60	2 – «неудовлетворительно»	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ТОВАРИЩЕСТВО ДИСЦИПЛИНЫ

6.1. Рекомендуемая литература

а) основная литература

1. Донован, Дж. Системное программирование [Текст]. Мир. 2005.
2. Дональд, Э. Кнут. Искусство программирования. том 1. Основные алгоритмы, 3-е изд.: Пер. с англ. [Текст] / Э. Кнут Дональд. М.: Издательский дом «Вильямс», 2006. - 720 с.
3. Камерон, Хьюз. Параллельное и распределенное программирование с использованием C++ [Текст] / Хьюз Камерон. - Вильямс. - 2004. - 672 с.
4. Орлов, С. А. Технологии разработки программного обеспечения [Текст] / — СПб.: Питер, 2003.
5. Орлов, С.А. Организация ЭВМ и систем [Текст] / С.А. Орлов, Б.Я. Цилькер. Питер, 2006.

6. Гагарина, Л. Г., Кокорева Е. В., Виснадул Б. Д. Технология разработки программного обеспечения [Текст] / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. М.: ИД «ФОРУМ»; ИНФРА-М, 2008.
7. Малыхина, М.П. Базы данных. Основы, проектирование, использование [Текст] / М.П. Малыхина. - БВХ - Петербург. 2006. - 528 с.
8. Рудикова, Л.В. Базы данных. Разработка приложений [Текст] / Л.В. Рудикова. - БХВ - Петербург. - 2006. - 496.
9. Таненбаум, Э. Современные операционные системы [Текст] / Э. Таненбаум. - Питер. - 2011. - 1120 с.
10. Фуфаев, Э.В. Пакеты прикладных программ [Текст] / Э.В. Фуфаев, Л.И. Фуфаева. - Академия, 2006. - 352 с.

б) Дополнительная литература

1. Галушкин, А.И. Нейронные сети. Основы теории [Текст] / А.И. Галушкин. - Горячая Линия - Телеком. - 2012. - 496 с.
2. Гордеев, А.В. Операционные системы [Текст] / А.В. Гордеев. СПб.: Питер. 2005.
3. Дейтел, Г. Введение в операционные системы [Текст] / Г. Дейтел. Т.1 - 2. Мир. 2007.
4. Редько, В.Г. Эволюция, нейронные сети, интеллект. Модели и концепции эволюционной кибернетики [Текст] / В.Г. Редько. -Либроком. - 2011. - 224 с.
5. Таненбаум, Э. Архитектура компьютера [Текст] / Э. Таненбаум. - Питер. 2011. 848 с

6.2. Средства обеспечения освоения дисциплины

Учебно-методический материал в библиотеке института, ресурсы Интернета, ресурсы электронной библиотеки

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Изучение дисциплины проводится в лабораториях кафедры «Вычислительная и информационная техника». Лабораторные работы проводятся с использованием ресурсов компьютерных классов, позволяющих работать в различных операционных системах и средах программирования, таких как интегрированные пакеты разработки программ C++, Visual C++, Delphi и др.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В соответствии с требованиями ОС ВО по направлению подготовки 09.03.01 «Информатика и вычислительная техника» реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов. В рамках учебного курса студенты работают с лекциями, рекомендованной литературой, выполняют лабораторные работы, готовятся к экзамену и зачету. В процессе подготовки студенты используют программные продукты, инструментальные среды, информационно-справочные системы, информационные источники, размещенные в сети Интернет (официальные сайты, веб-порталы, тематические форумы и телекоммуникации), электронные учебники и учебно-методические пособия.

9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Предлагается

- Самостоятельно прорабатывать лекционный материал для более полного усвоения материала;
- В учебном процессе при выполнении лабораторного практикума эффективно использовать методические пособия и методический материал по темам лабораторных работ;
- Активно использовать Интернет-ресурсы для получения актуального материала по изучаемой дисциплине;
- Активно использовать Интернет-ресурсы для обновления инструментальной базы (систем программирования, инструментальных сред и т.д.) при выполнении лабораторных работ.

Программа составлена в соответствии с требованиями ОС ВО НИЯУ МИФИ к обязательному минимуму содержания основной образовательной программы по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Автор(ы) _____ Романова М.Д.

Рецензенты _____ Мельцас А.В.

Согласовано:

Зав. кафедрой ВИТ _____ В.С.Холушкин

Руководитель ОП _____ В.С.Холушкин