

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Костромской государственный университет»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

АРХИТЕКТУРА ЭВМ

Направление подготовки: 09.03.02. Информационные системы и
технологии

Направленность: *Разработка программного обеспечения
информационных систем*

Квалификация (степень) выпускника: бакалавр

Кострома
2020

Рабочая программа дисциплины «Архитектура ЭВМ» разработана:


- в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки 09.03.02. «Информационные системы и технологии», утвержденным приказом Минобрнауки РФ №926 от 19.09.2017г

Разработал:  Орлов Александр Валерьевич, к.т.н., доцент

Рецензент: 
подпись Панин Игорь Григорьевич, д.т.н., доцент

УТВЕРЖДЕНО:

На заседании кафедры Информационных систем и технологий
Протокол заседания кафедры № 8 от 26.05.2020 г.
Заведующий кафедрой Информационных систем и технологий


Подпись Киприна Л.Ю., к.т.н., доцент

1. Цели и задачи освоения дисциплины

Цель дисциплины:

Обучение использованию основных механизмов платформы x86-64 и использующих её современных ОС, проектированию и прототипированию программного обеспечения, учитывающего особенности платформы.

Задачи дисциплины:

1. изучение основных средств вычислительных систем на платформе x86-64;
2. формирование базового понимания принципов работы современных ОС;
3. формирование практических навыков создания прототипов ПО, использующего возможности аппаратной платформы и ОС.

2. Перечень планируемых результатов обучения дисциплине

В результате освоения дисциплины обучающийся должен освоить компетенцию:

ПКрек-3: способен выполнять работы и управлять работами по созданию ИС, автоматизирующих задачи организационного управления и бизнес-процессы

Код и содержание индикаторов компетенции:

ПКрек-3.2. Разработка архитектуры, прототипов и баз данных ИС.

В результате освоения дисциплины обучающийся студент должен:

знать:

- основные возможности и механизмы платформы x86-64 и распространённых на ней современных операционных систем.

уметь:

- разрабатывать прототипы приложений;
- тестировать результаты прототипирования.

владеть:

- инструментами разработки, отладки и тестирования прототипов программного обеспечения.

3. Место дисциплины в структуре ОП ВО

Дисциплина относится к части учебного плана, формируемой участниками образовательных отношений. Изучается в 3 семестре.

Изучение дисциплины основывается на ранее освоенных дисциплинах/практиках:

- информационные технологии
- основы вычислительной техники
- программирование на языках высокого уровня

Изучение дисциплины является основой для освоения последующих дисциплин/практик:

- операционные системы
- подготовка и защита ВКР

4. Объем дисциплины (модуля)

4.1. Объем дисциплины в зачетных единицах с указанием академических (астрономических) часов и виды учебной работы

Виды учебной работы	Очная форма
Общая трудоемкость в зачетных единицах	4
Общая трудоемкость в часах	144
Аудиторные занятия в часах, в том числе:	72
Лекции	36
Лабораторные занятия	36
Практическая подготовка	22
КСР	
Самостоятельная работа в часах	72
Форма промежуточной аттестации	Зачет

4.2. Объем контактной работы на 1 обучающегося

Виды учебных занятий	Количество часов Очная форма
Лекции	36
Практические занятия	
Лабораторные занятия	36
Консультации	
Зачет/зачеты	
Экзамен/экзамены	
Курсовые работы	
Контрольные работы	
Всего	72

5. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием количества часов и видов занятий

5.1 Тематический план учебной дисциплины (очная форма)

№	Название раздела, темы	Всего час	Аудиторные занятия		Самостоятельная работа
			Лекции	Практика.	
1	Основные понятия и механизмы	16	4	4	8
1.1	Процессор и его основные возможности.	8	2	2	4
1.2	Основы Windows API.	8	2	2	4
2	Управление ресурсами	40	6	12	22
2.1	Механизмы управления памятью.	8	2	2	4
2.2	Файловый ввод/вывод.	14	2	4	8
2.3	Сетевые соединения и механизм сокетов.	18	2	6	10
3	Структурирование программы	34	6	8	20
3.1	Многопоточные приложения.	18	2	6	10
3.2	Динамически связываемые	8	2	2	4

	библиотеки.				
3.3	Обработка исключений.	8	2	0	6
4	Возможности ОС семейства Windows	50	16	8	26
4.1	Реестр Windows.	8	2	2	4
4.2	Службы Windows.	4	2	0	2
4.3	Оконная подсистема Windows.	10	2	4	4
4.4	Подсистема GDI.	4	2	0	2
4.5	Права доступа в Windows.	6	2	0	4
4.6	Функции отладки	4	2	0	2
4.7	Component Object Model.	8	2	2	4
4.8	Драйверы в среде Windows.	6	2	0	4
	Зачет	0			
	Итого:	144	36	36	72

5.2. Содержание

Раздел 1. Основные понятия и механизмы

1.1 Процессор и его основные возможности.

Основные архитектуры ЭВМ, их свойства и применимость. Режимы работы процессора. Понятие регистра. Виды регистров и их назначение. Понятие адреса. Виды адресации памяти. Прерывания и исключительные ситуации.

1.2 Основы Windows API.

История WinAPI. Основные типы данных. Понятие дескриптора. Приёмы обработки ошибок. Структура исполняемого файла Windows. Ресурсы приложения.

Раздел 2. Управление ресурсами

2.1 Механизмы управления памятью.

Понятие страничной адресации. Каталоги страниц. Исключение Page Fault. Организация памяти процесса. Секции адресного пространства процесса. Куча. Динамическое распределение памяти. Стек. Фреймы стека.

2.2 Файловый ввод/вывод.

Обзор функций ввода/вывода. Синхронный ввод/вывод. Асинхронный ввод/вывод. Отображение файлов в адресное пространство процесса.

2.3 Сетевые соединения и механизм сокетов.

Механизм сокетов. Подготовка соединения. Обмен данными. Завершение соединения. Настройка соединения. Контроль соединения.

Раздел 3. Структурирование программы

3.1 Многопоточные приложения.

Понятие потока. Планировщик выполнения потоков. Управление потоками. Обзор примитивов синхронизации. Функции ожидания. Свойства примитивов синхронизации.

3.2 Динамически связываемые библиотеки.

Назначение DLL. Секция экспорта. Статическое связывание. Динамическое связывание. Доступ к ресурсам. Точка входа в DLL. Рандомизация загрузки модулей. Декорирование имён функций.

3.3 Обработка исключений.

Понятие исключения. Генерация исключений. Поиск обработчика исключений. Кадровый метод обработки исключений. Векторный метод обработки исключений.

Обработка завершения блока кода. Исключения при работе с математическим сопроцессором. Система исключений языка C++.

Раздел 4. Возможности ОС семейства Windows

4.1 Реестр Windows.

Типы хранимых данных. Структура реестра. Права доступа к реестру. Разделы реестра. Доступ к ключам реестра. Доступ к значениям реестра. Перечисление ключей и значений. Отслеживание изменений в реестре.

4.2. Службы Windows.

Понятие службы. Диспетчер служб. Программный интерфейс диспетчера служб. Запуск службы. Работа службы.

4.3. Оконная подсистема Windows.

Понятие окна. Навигация по иерархии окон. Позиция, размер и класс окна. Оконная функция. Очереди сообщений.

4.4. Подсистема GDI.

Назначение подсистемы GDI. Контекст устройства. Объект "перо". Объект "кисть". Объект "растр". Объект "шрифт". Объект "регион". Подсистема GDI+.

4.5. Права доступа в Windows.

Модель контроля доступа. Токены доступа и дескрипторы безопасности. Списки контроля доступа. Функции работы с токенами доступа. Функции работы с дескрипторами безопасности. Функции, позволяющие использовать токены безопасности.

4.6. Функции отладки

Управление процессами. Подключение отладчика к процессу. Аппаратная отладка. Перехват вызовов API. Библиотека Detours. Перехват оконных событий. Механизм хуков.

4.7. Component Object Model.

Основы Component Object Model. Интерфейс IUnknown. Повторное использование объектов. Работа с библиотекой COM. Понятие псевдонима (moniker). Сохранение состояния объекта.

4.8. Драйверы в среде Windows.

Понятие драйвера. Дерево устройств Plug&Play. Запросы ввода/вывода. Модель ввода/вывода Windows. Особенности работы с памятью. Процесс выбора драйвера системой. Windows Driver Framework.

5.3 Практическая подготовка

Код компетенции	Индикатор компетенции	Содержание задания на практическую подготовку по выбранному виду деятельности	Число часов на практическую подготовку			
			Всего	Лекции	Курсовой проект	Лаб. раб.
ПКрек-3	ПКрек-3.2	Основы языка C++. Рассмотрение процесса исполнения кода процессором.	1	-	-	1

ПКрек-3	ПКрек-3.2	Работа с Windows API. Обработка ошибок API.	1	-	-	1
ПКрек-3	ПКрек-3.2	Получение и анализ информации о структуре адресного пространства процесса.	1	-	-	1
ПКрек-3	ПКрек-3.2	Приёмы работы с файлами. Отображение файлов на адресное пространство. Асинхронный ввод-вывод.	3	-	-	3
ПКрек-3	ПКрек-3.2	Приёмы работы с сетевыми соединениями. Прием и передача данных по UDP на примере NTP. Прием и передача данных по TCP на примере передачи файла.	5	-	-	5
ПКрек-3	ПКрек-3.2	Создание многопоточных приложений. Синхронизация между несколькими потоками.	4	-	-	4
ПКрек-3	ПКрек-3.2	Динамическая загрузка DLL. Получение и анализ списка загруженных процессом DLL.	1	-	-	1
ПКрек-3	ПКрек-3.2	Получение информации из реестра Windows.	1	-	-	1
ПКрек-3	ПКрек-3.2	Приёмы создания оконных программ. Манипуляция окнами других программ.	3	-	-	3
ПКрек-3	ПКрек-3.2	Приёмы работы с	2	-	-	2

		объектами COM на примере диалоговых окон выбора файла.				
--	--	--	--	--	--	--

6. Методические материалы по освоению дисциплины

6.1. Самостоятельная работа по дисциплине (модулю)

Очная форма

№п/п	Раздел (тема) дисциплины	Задание	Часы	Методические рекомендации	Форма контроля
1	Основные понятия и механизмы		8		
1.1	Процессор и его основные возможности	Сходства и различия между основными архитектурами ЭВМ. Понятия регистра, адреса, прерывания.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
1.2	Основы Windows API	Понятие декриптора. Приёмы обнаружения и обработки ошибок. Секции исполняемого файла и их роль.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
2	Управление ресурсами		22		
2.1	Механизмы управления памятью	Понятие виртуальных адресов. Понятие страницы. Секции адресного пространства процесса. Стек.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
2.2	Файловый ввод/вывод	Три подхода к вводу/выводу: синхронный, асинхронный, отображение в память.	8	Конспект лекции учебные пособия	Контрольные вопросы по лекции
2.3	Сетевые соединения и механизм сокетов	Понятие сокета. Различие между потоковыми и датаграммными сокетами.	10	Конспект лекции учебные пособия	Контрольные вопросы по лекции
3	Структурирование программы		20		

3.1	Многопоточные приложения	Понятие потока. Синхронизация потоков. Примитивы синхронизации.	10	Конспект лекции учебные пособия	Контрольные вопросы по лекции
3.2	Динамически связываемые библиотеки	Секции экспорта и импорта. Два типа связывания. Декорирование имен.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
3.3	Обработка исключений	Понятие исключения. Два метода обработки исключений. Алгоритм поиска кадрового обработчика исключений.	6	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4	Возможности ОС семейства Windows		26		Тест
4.1	Реестр Windows	Структура реестра. Хранимые типы данных. Основные разделы.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.2	Службы Windows.	Диспетчер служб. Запуск и работа службы.	2	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.3	Оконная подсистема Windows.	Понятие окна. Оконная функция. Работа очереди сообщений.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.4	Подсистема GDI.	Назначение подсистемы GDI. Понятие контекста устройства. Объекты подсистемы GDI.	2	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.5	Права доступа в Windows.	Понятие токена доступа и дескриптора безопасности. Списки контроля доступа.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.6	Функции отладки	Управление процессами. Отладка процесса. Перехват событий.	2	Конспект лекции учебные пособия	Контрольные вопросы по лекции
4.7	Component Object Model.	Понятие интерфейса. Интерфейс IUnknown,	4	Конспект лекции	Контрольные вопросы по

		его свойства. Понятие псевдонима.		учебные пособия	лекции
4.8	Драйверы в среде Windows.	Понятие драйвера. Дерево устройств Plug&Play. Windows Driver Framework.	4	Конспект лекции учебные пособия	Контрольные вопросы по лекции

6.2. Тематика и задания для практических занятий

Не предусмотрены

6.3. Тематика и задания для лабораторных занятий

Лабораторная работа №1. Исполнение кода. Работа стека

Задание: Написать на языке C++ программу, содержащую точку входа `main()` и подпрограмму `subproc()`. Используя стандартные средства ввода/вывода языка, вывести на экран следующие значения:

1. Содержимое регистра EIP в `main()` до вызова `subproc()`.
2. Содержимое регистра ESP в `main()` до вызова `subproc()`.
3. Содержимое регистра EIP в ходе вызова `subproc()`.
4. Содержимое регистра ESP в ходе вызова `subproc()`, и разность по отношению к п. 2.
5. Содержимое регистра EIP в `main()` после вызова `subproc()`.
6. Содержимое регистра ESP в `main()` после вызова `subproc()`.

Контрольные вопросы:

1. Каково назначение регистров EIP и ESP?
2. Почему значение EIP после вызова `subproc()` больше, чем до вызова?
3. Почему внутри процедуры `subproc()` значение ESP уменьшается?

Лабораторная работа №2. Основы работы с функциями WinAPI.

Задание 1: Написать на языке C++ программу, выводящую на стандартный вывод сообщение «Hello, world!». Для вывода текста следует использовать функцию `WriteFile()`. Получить дескриптор стандартного вывода можно с помощью функции `GetStdHandle()`.

Задание 2: Написать программу, в цикле выводящую текущее время, пока не нажата клавиша Esc. Для получения текущего времени используйте функцию `GetSystemTime()`. Узнать состояние клавиши, не блокируя работу программы, можно вызовом функции `GetKeyState()`. Для задержки выполнения программы используйте функцию `Sleep()`.

Контрольные вопросы

1. Как получить код ошибки для последней операции?
2. Какой синтаксис языка C++ используется для получения адреса переменной?

Лабораторная работа №3. Страничный механизм

Задание 1: Написать программу, использующую функцию `VirtualQuery()` для получения сведений о диапазоне страниц, и выводящую эти сведения на экран. Среди выводимых сведений должны присутствовать: начальный адрес, размер блока и/или конечный адрес, флаги защиты. Оформите получение и вывод сведений как отдельную подпрограмму, и используйте её для получения сведений о четырёх адресах:

- Нулевой адрес
- Текущее значение регистра EIP (секция кода программы)
- Текущее значение регистра ESP (стек программы)
- Адрес глобальной переменной (секция данных программы)

Задание 2: Модифицируйте вашу программу так, чтобы она выводила на экран сведения о блоках памяти в первом гигабайте адресного пространства. Для этого используйте сведения о стартовом адресе и размере блока, чтобы определить адрес следующего блока.

Контрольные вопросы

1. Как процессор определяет соответствие между логическими адресами программы и физическими адресами в банках памяти?
2. Какие состояния страниц виртуального адресного пространства вы знаете?
3. Опишите поля структуры `MEMORY_BASIC_INFORMATION`. В чем разница между полями `BaseAddress` и `AllocationBase`? `Protect` и `AllocationProtect`?

Лабораторная работа №4. Файловый ввод/вывод

Задание 1: Написать программу, подсчитывающую количество символов перевода строки «\n» в указанном файле размером не более 1 МБ. Имя анализируемого файла должно быть получено из командной строки. Для чтения файла должен использоваться механизм отображения файлов в адресное пространство. Следует предварительно проверить размер файла, используя функцию `GetFileSizeEx()`.

Задание 2: Написать программу, копирующую файл в асинхронном режиме, блоками по 1 КБ. Для ускорения работы чтение и запись должны производиться параллельно, в разные буферы. Имя исходного и целевого файла должны быть получены из командной строки.

Контрольные вопросы

1. Для каких целей в ОС семейства Windows применяется механизм отображения файлов на области адресного пространства?
2. Каковы преимущества и недостатки механизма отображения файлов в адресное пространство?
3. Каковы преимущества и недостатки синхронного режима работы с файлами.
4. Каковы преимущества и недостатки асинхронного режима работы с файлами.

Лабораторная работа №5. Механизм сокетов

Задание 1: Получить текущее время по протоколу NTP с сервера `pool.ntp.org:123` (или другого доступного). Используйте `getaddrinfo()` для получения структуры с адресом, и передавайте данные по протоколу UDP. Формат сообщения NTP, а также сведения о функциях форматирования даты/времени приведены в справочных материалах.

Задание 2: Создать пару программ (клиент и сервер), позволяющих передать файл по сети. Сервер должен ожидать TCP соединения, а клиент – устанавливать TCP соединение с указанным сервером. Имя отправляемого/принимаемого файла, а также адрес и порт для осуществления соединения должны задаваться в командной строке приложения. При разработке сервера можете исходить из предположения, что к вам не будут подключаться несколько клиентов одновременно.

Контрольные вопросы

1. В чем разница между потоковым (SOCK_STREAM) и датаграммным (SOCK_DGRAM) соединениями?
2. Что делает функция bind()?
3. Как сервер может различать клиентов (вне рамок передаваемых данных) при использовании протокола TCP? А при использовании UDP?

Лабораторная работа №6. Многопоточность

Задание 1: Создайте программу, выполняющую подсчёт числа строк в текстовом файле с помощью синхронного ввода/вывода и вывод этого числа в консоль. Операция должна выполняться в отдельном потоке, в то время как основной поток должен дожидаться его завершения. Допустимо использовать функцию Sleep() для имитации длительного процесса. Имя целевого файла должно быть передано потоку через параметр, а не через глобальные переменные.

Задание 2: Модифицируйте программу следующим образом: главный поток должен читать имена файлов для обработки из стандартного потока ввода. Не следует считать, что вы сможете прочитать весь этот файл заранее. Эти имена должны помещаться в коллекцию, откуда они будут извлекаться несколькими (количество должно задаваться константой) рабочими потоками. Вывод результатов обработки должен производиться в стандартный поток вывода. Вывод вспомогательных сообщений должен производиться в стандартный поток сообщений об ошибках.

По окончании входного файла программа должна дождаться завершения всех рабочих потоков, и завершиться.

Используйте примитивы синхронизации:

- мьютекс для регулирования доступа к коллекции заданий;
- критические секции для регулирования доступа к потоку вывода и потоку ошибок;
- события с ручным сбросом (одно на поток) для сигнализации о готовности данного потока к выполнению новой задачи;
- событие с автоматическим сбросом для сигнализации о наличии новой задачи;
- событие с ручным сбросом для сигнализации о завершении работы программы.

Контрольные вопросы

1. Какие примитивы синхронизации вы знаете?
2. Что общего у мьютекса и критической секции? Чем они отличаются?
3. В чем разница между событием с автосбросом и ручным сбросом?
4. Как дождаться перехода в сигнальное состояние одного из указанных примитивов? Всех указанных примитивов?

Лабораторная работа №7. Библиотеки DLL

Задание 1: Напишите программу, выводящую список подгруженных в её адресное пространство модулей. Для получения списка используйте функции EnumProcessModules() и GetModuleBaseName(). Оформите получение и вывод списка модулей в виде подпрограммы.

Задание 2: Модифицируйте программу следующим образом: используйте функции LoadLibrary() и GetProcAddress() чтобы динамически загрузить библиотеку User32.dll, найти и вызвать функцию MessageBoxA(). Затем выгрузите библиотеку вызовом FreeLibrary(). Программа должна вывести список загруженных модулей до загрузки библиотеки, после загрузки, и после выгрузки.

Контрольные вопросы

1. Какие секции исполняемого файла наиболее важны при работе с динамическими библиотеками?
2. Какие способы связать исполняемый файл с библиотекой вы знаете?
3. В каких случаях выполняется точка входа в DLL?

Лабораторная работа №8. Реестр Windows

Задание 1: Напишите программу, выводящую список USB-носителей, подключаемых к данному ПК. Этот список доступен в разделе реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR. Каждой модели носителя соответствует вложенный ключ, в котором находятся ключи для отдельных экземпляров (например, если один и тот же носитель вставлялся в разные USB-порты). Значение FriendlyName в ключе экземпляра содержит человеко-читаемое имя носителя. Рекомендуется выделить в отдельную функцию код получения списка вложенных ключей (например, представленный как std::vector<std::string>).

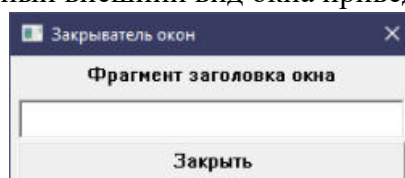
Контрольные вопросы

1. Какие ключи верхнего уровня (разделы) реестра вы знаете? Каково их назначение (в общих чертах)?
2. Какие права доступа к ключам реестра вы знаете?
3. Какие типы данных реестра позволяют хранить текстовые строки? В чём их различие?

Лабораторная работа №9. Оконная подсистема

Задание 1: На базе пустого проекта напишите программу, выводящую простое окно без содержимого. Программа должна завершать работу при закрытии окна.

Задание 2: Модифицируйте программу так, чтобы она создавала дочерние окна (элементы управления). Примерный внешний вид окна приведён на иллюстрации.



Задание 3: Модифицируйте программу следующим образом: по щелчку на кнопке, программа должна найти первое окно верхнего уровня, чей текст (заголовок) содержит строку, введённую в поле ввода. Затем она должна послать данному окну сообщение WM_CLOSE.

Контрольные вопросы

1. Что такое класс окна?
2. Какие функции позволяют осуществлять поиск/навигацию по дереву окон? Каково назначение каждой из них?

3. Какие оконные сообщения обрабатывает ваша программа? Каково их назначение?

Лабораторная работа №10. Component Object Model

Задание 1: Написать программу, выводящую на экран системный диалог открытия файла. Выбранное пользователем имя файла должно быть выведено в консоль.

Задание 2: Найдите и модифицируйте (либо напишите заново) программу копирования файла следующим образом:

- программа должна использовать диалоговое окно открытия файлов для выбора исходного файла;
- программа должна использовать диалоговое окно сохранения файлов для выбора целевого файла;
- программа должна отображать прогресс копирования в виде индикатора прогресса на панели задач;
- индикатор прогресса должен перейти в состояние ошибки, если таковая произошла, и оставаться в нём до завершения программы. В противном случае он должен быть отключён (состояние по умолчанию).

Контрольные вопросы

1. Что такое определение интерфейса, и чем оно отличается от его реализации?
2. Как идентифицируются классы и интерфейсы в подсистеме COM?
3. Для чего нужен интерфейс IUnknown? Какими свойствами должна обладать его реализация?

6.4. Методические рекомендации для выполнения курсовых работ

Не предусмотрены

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Основная

№	Наименование	Кол.
1	Архитектура ЭВМ: Учебное пособие / В.Д. Колдаев, С.А. Лупин. -М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. -384 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0373-5- http://znanium.com/catalog.php?bookinfo=375092	
2	Архитектура ЭВМ : учеб. пособие: Учебное пособие / Жмакин А.П., - 2-е изд., пере-раб. и доп. -СПб:БХВ-Петербург, 2010. -347 с. ISBN 978-5-9775-0550-5- http://znanium.com/catalog.php?bookinfo=351133	
3	Гуров, В.В. Архитектура и организация ЭВМ / В.В.Гуров, В.О.Чуканов. -2-е изд., испр. -Москва : Национальный Открытый Университет «ИНТУИТ», 2016. -184 с. : ил., схем. -(Основы информационных технологий). -Библиогр. в кн. -ISBN 5-9556-0040-X ; То же [Электронный ресурс]. -URL:	

http://biblioclub.ru/index.php?page=book&id=429021	
---	--

Дополнительная

№	Наименование	Кол
1	Архитектура корпоративных информационных систем/АстапчукВ.А., Терещен-коП.В. -Новосиб.: НГТУ, 2015. -75 с.: ISBN 978-5-7782-2698-2- http://znanium.com/catalog.php?bookinfo=546624	
2	Информационная безопасность конструкций ЭВМ и систем: учеб. пособие / Е.В. Глинская, Н.В. Чичварин. —М. : ИНФРА-М, 2018. — 118 с. + Доп. материалы [Электронный ресурс; Режим доступа http://www.znanium.com].—(Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/13571 .- http://znanium.com/catalog.php?bookinfo=925825	
3	Максимов, Н. В.Архитектура ЭВМ и вычислительных систем : учебник / Н. В. Максимов , И. И. Попов, Т. Л. Партыка. -5-е изд., перераб. и доп. -Москва : ФОРУМ; ИНФРА-М, 2013. -512 с.:ил. - (Проф. образование). -МО РФ. -ISBN 978-5-91134-742-0; 978-5-16-006732-2	
4	Новожилов, О. П.Архитектура ЭВМ и систем : учеб. пособие / Новожилов О. П. -Москва : Юрайт, 2016. -527 с.: рис. -(Бакалавр. Академ. курс). -УМО напр. 230100 "Информатика и вычислительная техника". -ISBN 978-5-9916-6882-8	
5	Прядкина, Н. О.Архитектура ЭВМ и систем : учеб. пособие. Ч. 3 / Прядкина Н. О. -Кострома : КГТУ, 2009. -63 с. -ISBN 978-5-8285-0256-1	1
6	Баула, В. Г.Архитектура ЭВМ и операционные среды : [учеб. для студ. высш. учеб. заведений] / В. Г. Баула, А. Н. Томилин, Д. Ю. Волканов. -М. : Академия, 2011. -335, [1] с. -(Бакалавриат) (Высшее профессиональное образование. Прикладная ма-тематика и информатика). -Библиогр.: с. 334. -ISBN 978-5-7695-8143-4	70

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

Информационно-образовательные ресурсы:

1. Федеральный портал «Российское образование». www.edu.ru
2. Официальный сайт министерства образования и науки Российской Федерации
<https://минобрнауки.рф>

Электронные библиотечные системы:

1. ЭБС «Лань»
2. ЭБС «Университетская библиотека online»
3. ЭБС «Znanium»

9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия проводятся в аудиториях с требуемым числом

посадочных мест и оборудованных мультимедиа.

Лабораторные занятия проводятся в компьютерных классах либо в аудиториях оборудованных мультимедиа. Необходимое программное обеспечение – офисный пакет, Microsoft Visual Studio версии не ниже 2015 с установленным модулем языка C++.