МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по естественнонаучному образованию

УТВЕРЖДЕНО
Первым заместителем Министра
образования Республики Беларусь
И.А. Старовойтовой
20.04.2020 г.

Регистрационный № **ТД-G.640/тип.**

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Типовая учебная программа по учебной дисциплине

для направления специальности

1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем)

СОГЛАСОВАНО
Председатель Учебно-
методического объединения по
естественнонаучному образованию
_____ О.А. Ивашкевич
«____» _____ 2018 г.

СОГЛАСОВАНО
Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь
_____ С.А. Касперович
«____» _____ 2020 г.

СОГЛАСОВАНО
Директор ИООО «Эксадел»
_____ В.Л. Черницкий
«____» _____ 2020 г.

СОГЛАСОВАНО
Проректор по научно-методической
работе Государственного
учреждения образования
«Республиканский институт высшей
школы»
_____ И.В. Титович
«____» _____ 2020 г.

Эксперт-нормоконтролер
_____ _____
«____» _____ 2020 г.

Минск 2020

СОСТАВИТЕЛИ:
А.Н.Курбацкий, заведующий кафедрой технологий программирования Белорусского государственного университета, доктор технических наук профессор;
В.М.Гошко, ассистент кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета

РЕЦЕНЗЕНТЫ:

Кафедра информационных радиотехнологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

И.А.Король, заместитель начальника управления стратегических проектов Министерства связи и информатизации, кандидат физико-математических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой технологий программирования Белорусского государственного университета
(протокол № 12 от 17 мая 2018 г.);

Научно-методическим Советом Белорусского государственного университета (протокол № 6 от 16 июня 2018 г.);

Научно-методическим Советом по прикладной математике и информатике учебно-методического объединения по естественнонаучному образованию (протокол № 16 от 29 мая 2018 г.).

Ответственный за редакцию: В.М.Гошко

Ответственный за выпуск: А.Н.Курбацкий

Пояснительная записка

Типовая учебная программа по учебной дисциплине «Системное программирование» разработана в соответствии требованиями образовательного стандарта первой ступени высшего образования по специальности 1- 31 03 07 «Прикладная информатика (по направлениям)».

Учебная дисциплина «Системное программирование» знакомит студентов с основными принципами построения и организации работы операционных систем семейства Windows.

Подробно рассматриваются вопросы системного программирования с использованием интерфейса Win32 API. Описываются управление потоками и процессами, включая их диспетчеризацию; передача данных между процессами, с использованием анонимных и именованных каналов, а также почтовых ящиков; структурная обработка исключений; управление виртуальной памятью; управление файлами и каталогами; асинхронная обработка данных; создание динамически подключаемых библиотек; разработка сервисов.

Особое внимание уделено вопросам отладки программного обеспечения. Дается обзор существующих инструментов поиска и устранения дефектов, приводится ряд практических рекомендаций по настройке отладчиков, рассматриваются различные сценарии исследования программного обеспечения.

Рассматриваются основы взаимодействия приложений по сети с использованием библиотеки WinSock.

Отдельно рассматриваются методы перехвата вызовов функций и модификации возвращаемых значений. Главной задачей данных тем является формирование у студентов четкого представления функционирования программного обеспечения, передачи управления между функциями, использование стека потока. Предусмотрены соответствующие лабораторные задания.

Цели учебной дисциплины «Системное программирование»:

– изучение принципов организации операционных систем на примере операционных систем семейства Windows;

– изучение способов и принципов создания системного программного обеспечения.

Основные задачи:

– изучение архитектуры операционной системы Windows, способов и принципов организации ее работы;

– ознакомление с возможностями, предоставленными интерфейсом прикладного программирования Win32 API;

– изучение принципов использования средств разработки и отладки, предоставляемых операционной системой, для создания эффективного и безопасного программного обеспечения.

В результате изучения учебной дисциплины обучаемый должен:

знать:

– основные функции операционной системы;
– основные компоненты операционной системы
– методы взаимодействия процессов;
– методы синхронизации потоков;
– модель памяти в защищенном режиме;
– методы управления виртуальной памятью;
– методы управления файлами;
– принципы построения клиент-серверных приложений с использованием библиотеки WinSock;
– принципы обработки исключительных ситуаций в ОС Windows;
– методы перехвата вызовов функций;
– основные виды уязвимостей программного обеспечения;
– механизмы защиты программ, предоставляемые операционной системой;
– методы отладки и поиска дефектов в программном обеспечении;

уметь:

– программировать многопоточные приложения;
– организовать обмен данными между двумя процессами;
– создавать приложения, взаимодействующие по сети;
– создавать и использовать динамически подключаемые библиотеки;
– пользоваться отладчиком, исследовать аварийные дампы памяти, проблемы утечки памяти;

владеть:

– методами и инструментами отладки и поиска дефектов в системном и прикладном программном обеспечении;
– языком программирования низкого уровня С.

В результате изучения учебной дисциплины «Системное программирование» формируются следующие компетенции:

– Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.
– Владеть системным и сравнительным анализом.
– Владеть исследовательскими навыками.
– Уметь работать самостоятельно.
– Быть способным порождать новые идеи (обладать креативностью).
– Владеть междисциплинарным подходом при решении проблем.
– Проектировать, разрабатывать и тестировать программное обеспечение различных видов.
– Разрабатывать техническую документацию на программное обеспечение.

– Работать с научно-технической информацией с использованием современных информационных технологий.

– На основе технической документации выполнять внедрение и сопровождение программного обеспечения, в том числе разработанного сторонними организациями.

На изучение дисциплины отведено 104 учебных часа, из них аудиторных – 68 часов. Примерное распределение аудиторных часов по видам занятий: лекций – 34 часа, лабораторных занятий - 34 часа.

Примерный тематический план

Название раздела, темы	Количество аудиторных часов		
	Всего	В том числе	
		Лекции	Лабораторные занятия
Раздел I. Язык программирования C.	**2**	**2**	**-**
Тема 1.1. Особенности языка программирования C.	2	2	-
Раздел II. Операционная система Windows	**38**	**18**	**20**
Тема 2.1. Функции и архитектура операционных систем.	4	2	2
Тема 2.2. Управление процессами и потоками.	4	2	2
Тема 2.3. Управление памятью.	6	2	4
Тема 2.4. Управление файлами.	6	2	4
Тема 2.5. Динамически подключаемые библиотеки.	6	2	4
Тема 2.6. Сервисы и драйверы Windows.	6	4	2
Тема 2.7. Программирование сети.	2	2	-
Тема 2.8. Перехват API вызовов.	4	2	2
Раздел III. Безопасное программирование	**28**	**14**	**14**
Тема 3.1. Уязвимости ПО. Безопасное программирование.	6	2	4
Тема 3.2. Структурная обработка исключений.	4	2	2
Тема 3.3. Отладка ПО.	8	4	4
Тема 3.4. Утилиты SysInternals.	4	2	2
Тема 3.5. Инструменты статического анализа кода.	6	4	2
Итого	**68**	**34**	**34**

Содержание учебного материала

Раздел I. Язык программирования С

Тема 1.1. Особенности языка программирования С.

Особенности языка. Отличия от С++. Массивы, строки, адресная арифметика, указатели на функции, функции с переменным числом аргументов. Построение программы. Построение основных структур данных. Стандартная библиотека языка С. Безопасные аналоги стандартных функций языка С.

Раздел II. Операционная система Windows

Тема 2.1. Функции и архитектура операционных систем.

Операционные системы. Функции. Архитектура. Выполнение задач. Процессы, потоки. Многозадачность. Windows API. UNICODE.

Тема 2.2 .Управление процессами и потоками.

Понятие процесса. Ресурсы, принадлежащие процессу. Создание и завершение процессов.3. Дескрипторы процесса. Взаимодействие процессов (файлы, командная строка, разделяемая память). Безопасность. Маркер доступа. Понятие потока. Контекст потока.

Тема 2.3. Управление памятью.

Модель памяти в защищенном режиме. Виды памяти (стек, куча, пулы памяти режима ядра, тегирование пула).

Тема 2.4. Управление файлами.

Обзор средств управления файлами. Представление файлов на жестком диске. Открытие, закрытие, чтение, запись, управление курсором. Чтение и изменение атрибутов, создание и удаление каталогов, наблюдение за изменениями. Копирование и перемещение файлов.

Тема 2.5. Динамически подключаемые библиотеки.

Назначение динамически подключаемых библиотек. Варианты использования библиотек. Зависимости библиотек. Формат файла PE. Таблицы импорта и экспорта. Загрузка библиотек. Внедрение кода в другой процесс.

Тема 2.6. Сервисы и драйверы Windows.

Сервисы Windows. Создание сервиса. Регистрация сервиса в системе. Менеджер сервисов. Запуск и остановка сервисов. Драйверы, точки входа в драйвер. Объект, описывающий драйвер. Объект, описывающий файл. Взаимосвязь объектов. Запрос ввода-вывода. Менеджер ввода-вывода. Стек драйверов. Прерывания, уровни прерываний. Подпрограммы обработки прерываний. Отложенные вызовы процедур. Асинхронные вызовы процедур. Типы асинхронных процедур.

Тема 2.7. Программирование сети.

Обзор модели OSI. Обзор сетевых API Windows. Использование Windows Sockets.

Тема 2.8. Перехват API вызовов.

Выполнение кода. Стек потока. Соглашения о вызовах. Перехват функций путем модификации исходного кода. Перехват функций путем модификации таблиц импорта/экспорта. Перехват функций путем модификации системных таблиц. Использование драйверов-фильтров.

Раздел III. Безопасное программирование

Тема 3.1. Уязвимости ПО. Безопасное программирование.

Классификация уязвимостей ПО. Ошибка переполнения буфера. Ошибка переполнения переменных. Ошибки форматирования строк. Механизмы защиты программ, предоставляемые операционной системой (ASLR, DEP). Безопасное программирование (переполнение чисел, буфера, неправильное использование памяти, проверка возвращаемых значений). Проверка входных данных.

Тема 3.2. Структурная обработка исключений.

Обработчики завершения. Фильтры и обработчики исключений. Необработанные исключения и исключения C++.

Тема 3.3. Отладка ПО

Обзор отладчиков. Обзор пакета Debugging Tools for Windows. Отладочные символы. Исследование аварийных завершений приложений. Исследование ошибок синхронизации. Исследование утечки памяти. Настройка отладки драйверов в режиме ядра. Настройка аварийного дампа памяти операционной системы.

Тема 3.4. Утилиты SysInternals.

Processmonitor. Processexplorer. Autoruns. Handle

Тема 3.5. Инструменты статического анализа кода.

Использование аннотации исходного кода (SAL). PREFast. Code analysis для VisualStudio.

Информационно-методическая часть

Литература

Основная

1. *Руссинович, М.* Внутреннее устройство Windows / М. Руссинович [и др.]. – 7-е изд. – СПб. : Питер, 2018. – 944 с.
2. *Рихтер, Дж.* Windows для профессионалов. Создание эффективных WIN32-приложений с учетом специфики 64-разрядной версии Windows / Дж. Рихтер – СПб. : Питер, 2001. – 752 с.
3. *Побегайло, А. П.* Системное программирование в Windows / А.П. Побегайло – СПб. : БХВ-Петербург, 2006. – 1056 с.
4. *Hewardt, M.* Advanced Windows Debugging / Hewardt M., D. Pravat – Boston : Addison-Wesley Professional, 2007. – 840 с.
5. *Шилдт, Г.* Полный справочник по C++ / Г. Шилдт. – 4-е изд. – М. : Издательский дом «Вильямс», 2006. – 800 с.
6. *Таненбаум, Э.* Современные операционные системы / Э. Таненбаум, Х. Бос. – 4-е изд. – СПб. : Питер, 2015. – 1120 с.

Дополнительная

7. *Беляев, А.* Централизованная обработка исключений / А. Беляев // RSDN Magazine, 25.09.2004. – [б.м.]
8. *Лохас П.* Debugging: Развертывание сервера отладочной информации / П. Лохас // Habrahabr [Электронный ресурс]. – 2010. – Режим доступа : http://habrahabr.ru/blogs/development/89094/. – Дата доступа : 29.11.2019.
9. *Лохас П.* Debugging: Введение в postmortem debugging / П. Лохас // Habrahabr [Электронный ресурс]. – 2010. – Режим доступа : http://habrahabr.ru/blogs/development/89220/. – Дата доступа : 29.11.2019.
10. *Харт, Дж. М.* Системное программирование в среде Windows / Дж. М. Харт – 3-е изд. – М. : Издательский дом «Вильямс», 2005. – 592 с.

Перечень рекомендуемых средств диагностики

Текущий контроль по дисциплине «Системное программирование» рекомендуется осуществлять в течение процесса обучения в виде вопросов для самоконтроля и проведения коллоквиумов.

Для проверки знаний и умений студентов рекомендуется выполнение всех предусмотренных лабораторных работ, совместное рассмотрение наиболее удачных решений и типовых ошибок.

Методические рекомендации по организации и выполнению самостоятельной работы

Условия для самостоятельной работы студентов, в частности, для развития навыков самоконтроля, способствующих интенсификации учебного процесса, обеспечиваются наличием и полной доступностью электронных (и бумажных) учебно-методических пособий по основным разделам учебной дисциплины.

Необходимо учитывать отсутствие у студентов опыта программирования на языке ассемблера. Это требует дополнительных пояснений, особенно при рассмотрении тем безопасного программирования и перехвата API вызовов.

В силу различного уровня готовности студентов к восприятию новых понятий на практических занятиях по дисциплине рекомендуется проводить регулярные самостоятельные работы и при необходимости проводить дополнительные консультации для объяснения и закрепления сложного материала.

Рекомендуемая форма текущей аттестации – зачет.