

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УТВЕРЖДАЮ

Проректор по учебной работе
и менеджменту качества

_____ Е.Н. Живицкая
24.12.2014г.

Регистрационный № УД-5-148/р.

«Операционные системы и системное программирование»

Учебная программа учреждения высшего образования по учебной дисциплине
для специальности
1-40 01 01 Программное обеспечение информационных технологий

Кафедра Программное обеспечение информационных технологий

Всего часов по
дисциплине 380

Зачетных единиц 10

2014 г.

Группа составителей:

П.Ю. Бранцевич, доцент кафедры ПОИТ

К.А. Сурков, старший преподаватель кафедры ПОИТ

И.Г.Алексеев, ассистент кафедры ПОИТ

Учебная программа учреждения высшего образования составлена на основе типовой учебной программы «Операционные системы и системное программирование», утвержденной Министерством образования Республики Беларусь “_____” _____ 201__ г., регистрационный номер № ТД – _____ /тип. и учебных планов специальности 1- 40 01 01 «Программное обеспечение информационных технологий»

Рассмотрена и рекомендована к утверждению на заседании кафедры программного обеспечения информационных технологий

протокол № 8 от 24.11.2014

Заведующая кафедрой ПОИТ

Н.В. Лапицкая

Одобрена и рекомендована к утверждению Научно-методической комиссией факультета компьютерных систем и сетей Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

протокол № 5 от 2.12.2014

Председатель _____ М.М. Лукашевич

СОГЛАСОВАНО

Эксперт-нормоконтролер

Декан ФНиДО

В.М. Бондарик

Директор ИИТ БГУИР

В.Г. Назаренко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

План учебной дисциплины в дневной форме обучения:

Код специальности	Название специальности	Курс	Семестр	Аудиторных часов				Академ. часов на курс. работу (проект)	Форма текущей аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия, семинары		
1-40 01 01	Программное обеспечение информационных технологий	2	4	102	70	32	-	-	экзамен
		3	5	80	48	32	-	40	экзамен

План учебной дисциплины в дистанционной форме обучения:

Код специальности	Название специальности	Курс	Семестр	Всего	Количество работ			Академ. часов на курс. работа (проект)	Форма текущей аттестации
					Контрольные работы	Лабораторные занятия	Индивидуальная практическая работа		
1-40 01 01	Программное обеспечение информационных технологий	3	5	200	2	-	2		экзамен
		3	6	180	2	-	2	40	экзамен

План учебной дисциплины в вечерней форме обучения для получения высшего образования, интегрированного со средним специальным образованием:

Код специальности	Название специальности	Курс	Семестр	Аудиторных часов				Академ. часов на курс. работу (проект)	Форма текущей аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия, семинары		
1-40 01 01	Программное обеспечение информационных технологий	2	4	24	8	16	-	40	экзамен

План учебной дисциплины в заочной форме обучения для получения высшего образования, интегрированного со средним специальным образованием:

Код специальности	Название специальности	Курс	Семестр	Аудиторных часов				Академ. часов на курс. работу (проект)	Форма текущей аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия, семинары		
1-40 01 01	Программное обеспечение информационных технологий	2	4	8	2	4	2	40	экзамен

Место дисциплины.

Развитие аппаратных компьютерных средств требует создания системного программного обеспечения, направленного на их эффективное использование. Это обуславливает место курса «Операционные системы и системное программирование» в основе формирования компетенций специалиста в области программного обеспечения отрасли информационных технологий.

Цель учебной дисциплины:

изучение организации и принципов построения современных операционных систем и системных программ; аппаратных и программных средств современных процессоров, предназначенных для поддержки многозадачных операционных систем; методологии разработки системно-ориентированных программ с использованием современных алгоритмических языков и систем программирования.

Задачи изучения учебной дисциплины:

- формирование базовых знаний в области организации и функционирования современных операционных систем;
- изучение способов разработки системного программного обеспечения с учетом особенностей современных операционных систем;
- овладение методами разработки, тестирования, отладки и документирования программ, направленных на решение системных задач, с использованием современных инструментальных средств.

В результате изучения учебной дисциплины студент должен:

знать:

- принципы построения операционных систем;

- типовые алгоритмы организации взаимодействия между процессами;
- задачи, решаемые при управлении виртуальной памятью;
- основные системные вызовы стандарта POSIX;
- систему прав доступа в файловой системе ОС Windows, UNIX/Linux;
- способы взаимодействия между процессами в ОС Windows, UNIX/Linux;
- способы потокового взаимодействия в ОС Windows, UNIX/Linux;
- архитектуру и подсистемы ОС Windows, UNIX/Linux;
- модель виртуальной памяти процесса в пользовательском режиме и в режиме ядра операционной системы Windows;
- программный интерфейс графических устройств GDI;
- понятие динамически-загружаемой библиотеки и средства построения таких библиотек в операционной системе Windows;
- программный интерфейс прикладных программ Win32;
- средства поддержки многозадачности в ОС Windows, UNIX/Linux и виды синхронизации задач;
- системные механизмы операционной системы Windows: прерывания, исключения и системные вызовы;
- модель памяти в ядре ОС Windows;
- принципы построения драйверов для операционной системы Windows;
- средства синхронизации потоков в ОС семейства Windows;

уметь:

- разрабатывать программы, в ОС Windows, UNIX/Linux;
- разрабатывать программы, по организации взаимодействия между процессами в ОС Windows, UNIX/Linux;
- разрабатывать программы, по организации потокового взаимодействия в ОС Windows, UNIX/Linux;
- разрабатывать программы в ОС Windows с графическим пользовательским интерфейсом;
- разрабатывать многопоточные программы с синхронизацией данных ОС Windows, UNIX/Linux;
- разрабатывать динамически подключаемые библиотеки;
- использовать проецируемые в память файлы ОС Windows, UNIX/Linux;
- перехватывать вызовы к операционной системе;

владеть:

- структурой ОС Windows/UNIX;
- системными WinAPI в ОС Windows и системными вызовами ОС Windows, UNIX/Linux;
- средствами синхронизации процессов и потоков в ОС Windows, UNIX/Linux.

иметь представление:

- о тенденциях развития современных операционных систем для различных аппаратных платформ;
- о научных исследованиях, проводимых в данной области.

**Перечень учебных дисциплин, усвоение которых необходимо
для изучения данной учебной дисциплины**

№ п.п.	Название дисциплины	Раздел, темы
1.	Языки программирования	Язык программирования высокого уровня.
2.	Основы алгоритмизации и программирования	Алгоритм и его свойства. Структурное программирование. Основные типы данных. Структура программы.

1. Содержание учебной дисциплины

№ тем	Наименование разделов, тем	Содержание тем
Раздел 1. Операционная система UNIX (LINUX)		
1	Разработка программ в ОС UNIX	Основные принципы организации и построения ОС UNIX. Стандарты UNIX. Пользователи системы, атрибуты пользователя. Создание программы, исходный текст, заголовки. Системные вызовы и функции стандартных библиотек. Обработка ошибок. Форматы выполняемых файлов
2	Файловая система ОС UNIX	Типы файлов. Владельцы файлов. Управление правами доступа в файловой системе. Атрибуты файлов. Управление свойствами файлов. Работа с файлами. Структура файловой системы.
3	Процессы в ОС UNIX	Понятие процесса. Типы процессов. Атрибуты процессов. Создание и управление процессами. Перегрузка процессов. Завершение процессов. Синхронизация процессов. Системные средства для управления свойствами процессов.
4	Взаимодействие между процессами	Сигналы. Обработка сигналов. Неименованные каналы. Именованные каналы. Дополнительные средства взаимодействия между процессами. Сообщения, семафоры, разделяемая память.
5	Потоки в ОС UNIX	Понятие потока. Атрибуты потоков. Создание и управление потоками. Завершение потоков. Синхронизация потоков. Мьютексы. Блокировки. Переменные состояния. Потоки и процессы.
Раздел 2. Основы операционных систем		
6	Введение в операционные системы	Назначение, основные этапы развития операционных систем. Определение ОС. Особенности современных операционных систем. Принципы построения ОС.
7	Основы операционных систем	Понятие процесса, потока, ресурса, свойства, классификация. Концепция виртуализации. Концепция прерывания.
8	Процессы	Состояния процессов. Описание процессов. Взаимодействие процессов. Задача взаимного исключения. Решение задачи взаимного исключения. Алгоритм Деккера. Алгоритм Петерсона. Семафоры. Применение семафоров для решения задачи взаимного исключения. Задача производителя-потребителя и ее решения. Мониторы. Задача «читатели-писатели». Процессы и потоки.
9	Ресурсы	Распределение ресурсов, проблема тупиков. Алгоритм банкира. Применение алгоритма банкира.
10	Память. Управление памятью	Требования к управлению памятью. Схемы распределения памяти. Страничная организация памяти. Сегментация памяти.
11	Организация вирту-	Структуризация адресного пространства виртуальной памяти.

	альной памяти	Задачи управления виртуальной памятью: задача размещения, задача перемещения, задача преобразования адресов, задача замещения.
12	Планирование в операционных системах	Типы планирования. Алгоритмы планирования. Примеры реализации алгоритмов планирования в современных операционных системах.
13	Управление вводом-выводом и файлами	Организация функций ввода-вывода. Логическая структура функций ввода-вывода. Буферизация операций ввода-вывода. Дисковое планирование. Система управления файлами. Организация файлов, доступ к файлам. Управление внешней памятью.
14	Аппаратно-программные особенности современных процессоров, ориентированные на поддержку многозадачных систем	Управление памятью в реальном и защищенном режимах. Дескрипторные таблицы и дескрипторы сегментов. Организация защиты по данным и кодам. Команды и структуры данных для управления в многозадачных режимах. Поддержка многозадачности. Обработка прерываний.
Раздел 3. Операционная система Windows в пользовательском режиме		
15	Введение в операционную систему Windows	Архитектура и основные подсистемы ОС Windows. Системный реестр ОС Windows, его назначение и использование. Средства программирования для ОС Windows. Понятие объекта и дескриптора. Нотация программного интерфейса ОС Windows.
16	Оконный пользовательский интерфейс	Основные элементы программ с оконным пользовательским интерфейсом. Минимальная программа для ОС Windows с окном на экране. Создание и отображение окна. Понятие оконного сообщения. Источники сообщений. Очередь сообщений. Цикл приема и обработки сообщений. Процедура обработки сообщений. Синхронные и асинхронные сообщения, их передача и обработка. Ввод данных с манипулятора «мышь». Обработка сообщений мыши. Ввод данных с клавиатуры. Понятие фокуса ввода. Обработка сообщений от клавиатуры. Вывод информации в окно. Механизм перерисовки окна.
17	Ресурсы программ	Понятие ресурсов программ. Виды ресурсов. Работа с ресурсами. Меню Окна диалога.
18	Интерфейс графических устройств	Принципы построения графической подсистемы ОС Windows. Понятие контекста устройства. Вывод графической информации на физическое устройство. Рисование геометрических фигур. Графические инструменты. Управление цветом. Палитры цветов. Растровые изображения. Вывод растровых изображений. Значки и курсоры. Вывод растровых изображений с эффектом прозрачного фона. Вывод текста. Логические и физические шрифты. Системы координат. Трансформации. Метафайлы.
19	Многозадачность в	Организация многозадачности в ОС Windows. Понятие про-

	ОС Windows	цесса и потока. Контекст потока. Создание и завершение процессов и потоков. Синхронизация потоков одного и того же процесса. Критические секции. Синхронизация потоков разных процессов. Объекты синхронизации: флаги, семафоры, события, ожидаемые таймеры, трубы.
20	Динамически подключаемые библиотеки	Понятие динамически подключаемой библиотеки. Структура DLL-библиотеки. Создание DLL-библиотеки. Использование DLL-библиотеки в программе. Статический и динамический импорт. Технология COM.
Раздел 4. Операционная система Windows в режиме ядра		
21	Отладка программ в режиме ядра	Отладчики для режима ядра. Режимы отладки. Компоненты отладчика. Представление в памяти строк формата Unicode. Представление в памяти двусвязных списков. Создание дампа памяти. Анализ дампов памяти.
22	Системные механизмы ядра	Прерывания. Уровни прерываний. Подпрограммы обработки прерываний. Отложенные процедуры. Асинхронные процедуры. Типы асинхронных процедур. Рабочие элементы. Переход из пользовательского режима в режим ядра. Таблицы дескрипторов функций ОС Windows.
23	Виртуальное адресное пространство	Пулы памяти. Пул подкачиваемой памяти, пул неподкачиваемой памяти, пул сессии, особый пул. Тегирование пулов. Структура данных пула. Представление объекта в памяти ядра. Менеджер объектов. Ассоциативные списки. Блокирование страниц в памяти. Списки описателей памяти и их использование для работы с аппаратурой.
24	Драйверы ОС Windows	Структура драйвера. Точки входа в драйвер. Объект, описывающий драйвер. Объект, описывающий устройство. Объект, описывающий файл. Взаимосвязь объектов. Запрос ввода-вывода. Менеджер ввода-вывода. Стек драйверов. Организация сетевых драйверов.
25	Перехват функций ОС Windows	Перехват функций ОС Windows API в пользовательском режиме. Интерфейсный модуль NTDLL.DLL. Функции ОС Windows API в режиме ядра. Модуль NTOSKRNL.EXE. Технология перехвата функций в ядре за счет модификации таблицы дескрипторов функций ОС Windows. Технология перехвата функций в ядре за счет исправления кода функции ОС Windows. Правила разработки функций-перехватчиков. Альтернативные стандартные методы перехвата функций: мини-фильтры файловой системы, обратные вызовы системного реестра, обратные вызовы менеджера объектов, обратные вызовы процессов, обратные вызовы потоков, обратные вызовы загрузчика модулей.

2. Информационно-методический раздел

2.1 Литература

2.1.1 Основная

2.1.1.1. Гордеев, А.В. Системное программное обеспечение / А.В. Гордеев, А.Ю. Молчанов А.Ю. –СПб.: Питер, 2002. –736 с.

2.1.1.2. Столингс, В. Операционные системы / Столингс, В. – М.: Издательский дом "Вильямс", 2002. –848 с.

2.1.1.3. Таненбаум, Э. Современные операционные системы /Э. Таненбаум. – СПб.: Питер, 2010. –1120 с.

2.1.1.4. Хэвиленд, К. Системное программирование в UNIX. Руководство программиста по разработке ПО / К. Хэвиленд, Д. Грэй, Б. Салама.–М.: ДМК Пресс, 2000. –368 с.

2.1.1.5. Робачевский, А.М. Операционная система UNIX / А.М. Робачевский. – СПб.: BHV–Санкт-Петербург, 1997. –528 с.

2.1.1.6. Стивенс, У. UNIX: взаимодействие процессов /У. Стивенс. –СПб.: Питер, 2002. – 576 с.

2.1.1.7. Стивенс, У. UNIX: разработка сетевых приложений / У.Стивенс.СПб.: Питер, 2003. – 1088 с.

2.1.1.8. Моли, Б. Unix/Linux: теория и практика программирования / Б. Моли – М.: КУДИЦ-ОБРАЗ, 2004, –576 с.

2.1.1.9. Роббинс, А. Linux: программирование в примерах / А. Роббинс– М.: КУДИЦ-ОБРАЗ, 2005, –656 с.

2.1.1.10. Петерсен, Р. LINUX: руководство по операционной системе / Р. Петерсен; пер. с англ. – К.: Издательская группа BHV, 1997. – 688 с.

2.1.1.11. Чан, Т. Системное программирование на C++ для UNIX / Т. Чан; Пер. с англ. – К.: Издательская группа BHV, 1997. – 592 с.

2.1.1.12. Лав, Р. Linux. Системное программирование / Р. Лав. – СПб.: Питер, 2008. – 416 с.

2.1.1.13. Мэтью, Н. Основы программирования в Linux: Пер. с англ./ Н.Мэтью, Р. Стоунс. – СПб.: БХВ-Петербург, 2009. –896 с.

2.1.1.14. Колисниченко, Д.Н. Linux на ноутбуке /Д.Н. Колисниченко. – СПб.: БХВ-Петербург, 2009. –704 с.

2.1.1.15. Маслаков, В.Г. Linux на 100% / В.Г. Маслаков. – СПб.: Питер, 2009. – 336 с.

2.1.1.16. Стивенс, Р. UNIX. Профессиональное программирование / Р. Стивенс, С. Раго. –СПб.: Символ-Плюс, 2010. – 1040 с.

2.1.1.17. Иванов, Н. Программирование в Linux. Самоучитель / Н. Иванов. – СПб.:BHV, 2012. –400 с.

2.1.1.18. Руссинович, М. Внутреннее устройство Microsoft Windows / М. Руссинович, Д. Соломон; пер. с англ. – 4-е изд. – М: Издательско-торговый дом «Русская редакция»; СПб.: «Питер», 2005. – 992 с.

2.1.1.19. Рихтер, Дж. Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64-разрядной версии Windows / Дж. Рихтер; пер. с англ. – 4-е изд. – СПб; Питер; М.: Издательско-торговый дом "Русская Редакция", 2001. – 752 с.

2.1.1.20. Dabak, P. Undocumented Windows NT / P.Dabak, S.Phadke, M. Borate. – IDG Books Worldwide, Inc.; M&T Books. – 327 с.

2.1.1.21. Nebbett, G. Windows NT/2000 Native API Reference / G.Nebbett. – МТР – 482 с.

2.1.1.22. Солдатов, В.П. Программирование драйверов Windows / В.П. Солдатов. – М.: Бином-Пресс, 2006. – 576 с.

2.1.1.23. Комиссарова, В. Программирование драйверов для Windows / В. Комиссарова. – СПб.: БХВ-Петербург, 2007. – 256 с.

2.1.1.24. Driver development with Visual Studio Express / Microsoft Corporation [Электронный ресурс]. – 2014. – Режим доступа <http://www.microsoft.com/whdc/default.msp>

2.1.1.25. The Undocumented Functions Microsoft Windows NT/2000 / NTAPI Undocumented Functions [Электронный ресурс]. – 2014. – Режим доступа <http://undocumented.ntinternals.net>

2.1.2 Дополнительная

2.1.2.1. Дейтел, Р. Введение в операционные системы: в 2 т. / Р. Дейтел. – М.: Мир, – Т.1, Т.2. – 1987.

2.1.2.2. Кейслер, С. Проектирование операционных систем для малых ЭВМ / С. Кейслер; пер. с англ. – М.: Мир, 1986. – 680 с.

2.1.2.3. Немеет, Э. UNIX: руководство системного администратора / Э. Немеет, Г. Снайдер, С. Сибасс, Т. Хейн.; Пер. с англ. – К.: ВНУ, 1996. – 832 с.

2.1.2.4. Керниган, Б.В. UNIX – универсальная среда программирования / Б.В. Керниган, Р. Пайк; пер. с англ., – М.: Финансы и статистика, 1992. – 304 с.

2.1.2.5. Краковяк, С. Основы организации и функционирования ОС ЭВМ / С. Краковяк. – М.: Мир, – 1988 с.

2.1.2.6. Бек, Л. Введение в системное программирование / Л. Бек. – М.: Мир. – 1988 с.

2.1.2.7 Григорьев, В.Л. Микропроцессор i486. Архитектура и программирование: в 4 кн. Кн. 1. Программная архитектура / В.Л. Григорьев – М.: ГРАНАЛ, 1993. – 346 с.

2.2 Перечень компьютерных программ, наглядных и других пособий, методических указаний и материалов, технических средств обучения, оборудования для выполнения лабораторных работ

2.2.1. Классы ПЭВМ, объединенных в сеть с многозадачной операционной системой UNIX(Linux), WINDOWS.

2.2.2. Система программирования для разработки программ для многозадачной операционной системы на алгоритмическом языке типа GNUC/C++версии не ниже 4.

2.2.3. Среда программирования VisualStudio версии не ниже 2008.

2.3. Перечень тем практических занятий, их название

(для студентов заочной формы обучения для получения высшего образования, интегрированного со средним специальным образованием)

Целью практического занятия является закрепление теоретического курса, приобретение навыков решения задач, активизация самостоятельной работы студентов.

№ темы по п.1	Название практического занятия	Содержание	Обеспеченность по пункту 2.2
1	2	3	4
3	Основные принципы программирования, процессы в ОС Unix/Linux	Изучение файловой системы ОС Linux и основных функций для работы с каталогами и файлами, способов создания процессов в ОС Linux, основные функции создания и управления процессами, обмен данными между процессами.	2.2.1, 2.2.2

2.4. Перечень тем лабораторных занятий, их название

Основная цель проведения лабораторных занятия состоит в закреплении теоретического материала курса, приобретении навыков выполнения эксперимента, обработки экспериментальных данных, анализа результатов, грамотного оформления отчетов.

№ темы по п.1	Наименование лабораторной работы	Содержание	Обеспеченность по пункту 2.2
1	2	3	4
1	Работа в ОС UNIX/Linux, интерпретатор BASH	Изучить теоретическую часть лабораторной работы. В консольном режиме создать в <i>домашней папке</i> подкаталог: <i>/номер_группы/ФИО_студента</i> , где в дальнейшем будут храниться все файлы студента. Перейти в корневой каталог и вывести его со-	2.2.1, 2.2.2

		<p>держимое используя команды <i>dir</i> <i>ls -all</i>, проанализировать различия.</p> <p>Проверить действие команд <i>ps</i>, <i>ps -x</i>, <i>top</i>, <i>htop</i>.</p> <p>Найти в справочной системе используя команду <i>man</i> справку по функциям <i>fprintf</i>, <i>fputc</i> и команде <i>ls</i>.</p> <p>В текстовом редакторе <i> joe</i> (вызов: <i> joe 1.c</i>) написать программу <i> 1.c</i>, выводящую на экран фразу “HELLOSUSELinux”. Компилировать полученную программу компилятором <i> gcc</i>: <i> gcc1.c -o 1.exe</i>. Запустить полученный файл <i> 1.exe</i> на выполнение: <i> ./1.EXE</i></p> <p>Написать скрипт, выводящий на консоль и в файл все аргументы командной строки.</p> <p>Написать скрипт, выводящий в файл (имя файла задаётся пользователем в качестве первого аргумента командной строки) имена всех файлов с заданным расширением (третий аргумент командной строки) из заданного каталога (имя каталога задаётся пользователем в качестве второго аргумента командной строки).</p> <p>Написать скрипт, компилирующий и запускающий программу (имя исходного файла и <i> exe</i>- файла результата задаётся пользователем в качестве аргументов командной строки). В случае ошибок при компиляции вывести на консоль сообщение об ошибках и не запускать программу на выполнение.</p>	
2	Работа с файлами и каталогами ОС UNIX	<p>Изучить теоретическую часть лабораторной работы.</p> <p>Написать программу вывода сообщения на экран.</p> <p>Написать программу ввода символов с клавиатуры и записи их в файл (в качестве аргумента при запуске программы вводится имя файла).</p> <p>Для чтения или записи файла использовать только функции посимвольного ввода-вывода <i> getc ()</i>, <i> putc ()</i>, <i> fgetc ()</i>, <i> fputc ()</i>. Предусмотреть выход после ввода определённого символа (например: <i> ctrl-F</i>). Предусмотреть контроль ошибок открытия/закрытия/чтения файла.</p> <p>Написать программу вывода содержимого текстового файла на экран (в качестве аргумента при запуске программы передаётся имя файла, второй аргумент (<i> N</i>) устанавливает вывод по группам строк (по <i> N</i> –строк) или сплошным текстом (<i> N=0</i>)). Для вывода очередной группы строк необходимо ожидать нажатия пользова-</p>	2.2.1, 2.2.2

		<p>телем любой клавиши. Для чтения или записи файла использовать только функции посимвольного ввода-вывода <code>getc ()</code>, <code>putc ()</code>, <code>fgetc ()</code>, <code>fputc ()</code>. Предусмотреть контроль ошибок открытия/закрытия/чтения/записи файла.</p> <p>Написать программу копирования одного файла в другой. В качестве параметров при вызове программы передаются имена первого и второго файлов. Для чтения или записи файла использовать только функции посимвольного ввода-вывода <code>getc ()</code>, <code>putc ()</code>, <code>fgetc ()</code>, <code>fputc ()</code>. Предусмотреть копирование прав доступа к файлу и контроль ошибок открытия/закрытия/чтения/записи файла.</p> <p>Написать программу вывода на экран содержимого текущего и корневого каталогов. Предусмотреть контроль ошибок открытия/закрытия/чтения каталога.</p>	
3	Процессы в ОС UNIX/Linux	<p>Написать программу нахождения массива K последовательных значений функции $y[i]=\sin(2*\pi*i/N)$ ($i=0,1,2\dots K-1$) с использованием ряда Тейлора. Пользователь задаёт значения K, N и количество n членов ряда Тейлора. Для расчета каждого члена ряда Тейлора запускается отдельный поток. Каждый поток выводит на экран свой <code>id</code> и рассчитанное значение ряда. Головной процесс суммирует все члены ряда Тейлора, и полученное значение $y[i]$ записывает в файл.</p> <p>Написать программу синхронизации двух каталогов, например, <code>Dir1</code> и <code>Dir2</code>. Пользователь задаёт имена <code>Dir1</code> и <code>Dir2</code>. В результате работы программы файлы, имеющиеся в <code>Dir1</code>, но отсутствующие в <code>Dir2</code>, должны скопироваться в <code>Dir2</code> вместе с правами доступа. Процедуры копирования должны запускаться в отдельном процессе для каждого копируемого файла. Каждый процесс выводит на экран свой <code>pid</code>, имя копируемого файла и число скопированных байт. Число одновременно работающих процессов не должно превышать N (вводится пользователем).</p> <p>Написать программу поиска одинаковых по содержимому файлов в двух каталогов, например, <code>Dir1</code> и <code>Dir2</code>. Пользователь задаёт имена <code>Dir1</code> и <code>Dir2</code>. В результате работы программы файлы, имеющиеся в <code>Dir1</code>, сравниваются с файлами в <code>Dir2</code> по их содержимому. Процеду-</p>	2.2.1, 2.2.2

		<p>ры сравнения должны запускаться в отдельном процессе для каждой пары сравниваемых файлов. Каждый процесс выводит на экран свой pid, имя файла, общее число просмотренных байт и результаты сравнения. Число одновременно работающих процессов не должно превышать N (вводится пользователем).</p> <p>Написать программу поиска заданной пользователем комбинации из m байт ($m < 255$) во всех файлах текущего каталога. Пользователь задаёт имя каталога. Главный процесс открывает каталог и запускает для каждого файла каталога отдельный процесс поиска заданной комбинации из m байт. Каждый процесс выводит на экран свой pid, имя файла, общее число просмотренных байт и результаты поиска. Число одновременно работающих процессов не должно превышать N (вводится пользователем).</p> <p>Разработать программу «интерпретатор команд», которая воспринимает команды, вводимые с клавиатуры, (например, ls -l /bin/bash) и осуществляет их корректное выполнение. Для этого каждая вводимая команда должна выполняться в отдельном процессе с использованием вызова exec (). Предусмотреть контроль ошибок.</p> <p>Создать дерево процессов по индивидуальному заданию. Каждый процесс постоянно, через время t, выводит на экран следующую информацию: номер процесса/потока pid ppid текущее время (мсек). Время t = ((номер процесса/потока по дереву) * 200 (мсек)).</p>	
4	Взаимодействие между процессами	<p>Изучить теоретическую часть лабораторной работы.</p> <p>Организовать функционирование процессов следующей структуры: отец-сын-сын.</p> <p>Процессы определяют свою работу выводом сообщений вида: N pid ppid текущее время (мсек) (N – текущий номер сообщения) на экран. “Отец” одновременно, посылает сигнал SIGUSR1 “сыновьям”. “Сыновья”, получив данный сигнал, посылают в ответ “Отцу” сигнал SIGUSR2. “Отец”, получив сигнал SIGUSR2, через время t=100 мсек одновременно, посылает сигнал SIGUSR1 “сы-</p>	2.2.1, 2.2.2

		<p>новьям”. И так далее... Написать функции-обработчики сигналов, которые при получении сигнала выводят сообщение о получении сигнала на экран. При получении/посылке сигнала они выводят соответствующее сообщение: N pid ppid текущее время (мсек) сын такой-то get/putSIGUSRm. Предусмотреть механизм для определения “Отцом”, от кого из “Сыновей” получен сигнал. Написать программу, создающую дочерний процесс. Родительский процесс создаёт семафор (sem1) и общий файл. Дочерний процесс записывает в файл по одной строке всего 100 строк вида: номер_строкиpid_процессатекущее_время (мсек). Родительский процесс читает из файла строки и выводит их на экран в следующем виде: pid строка_прочитанная_из_файла. Семафор sem1 используется процессами для разрешения, кому из процессов получить доступ к файлу. Написать программу, создающую дочерний процесс. Родительский процесс создаёт неименованный канал. Дочерний процесс записывает в канал 100 строк вида: номер_строкиpid_процессатекущее_время (мсек). Родительский процесс читает из канала строки и выводит их на экран в следующем виде: pid строка прочитанная из файла.</p>	
5	Потоки в ОС UNIX	<p>Изучить теоретическую часть лабораторной работы. Написать программу, создающую два дочерних потока. Родительский процесс и два дочерних потока должны выводить на экран свой id и pid родительского процесса и текущее время в формате: часы: минуты: секунды: миллисекунды. Написать программу, создающую дочерний процесс. Родительский процесс создаёт семафор (sem1) и общий файл. Дочерний процесс записывает в файл по одной строке всего 100 строк вида: номер строкиpid_процессатекущее_время (мсек). Родительский процесс читает из файла строки и выводит их на экран в следующем виде: pid строка_прочитанная_из_файла. Семафор sem1 используется процессами для разрешения, кому из процессов получить доступ к файлу.</p>	2.2.1, 2.2.2

		Написать программу, создающую дочерний процесс. Родительский процесс создаёт семафор (sem1) и общую память. Дочерний процесс записывает в память по 70 строк всего 1000 строк вида: номер_строки pid_процесса текущее_время (мсек). Родительский процесс читает из файла по 81 строке и выводит их на экран в следующем виде: pid строка, прочитанная_из_файла. Семафор sem1 используется процессами для разрешения, кому из процессов получить доступ к файлу.	
16,18	Создание программ с графическим пользовательским интерфейсом.	Рисование в окне с помощью мыши. Изображение должно состоять из прямых, ломаных, прямоугольников, эллипсов, многоугольников, текстовых надписей. Сохранение рисунка на диск в векторном формате - в виде enhanced-метафайла. Загрузка картинки с диска и продолжение рисования. Перемещение (Pan) и масштабирование (Zoom) рисунка с помощью колесика мыши (Pan - Wheel и Shift+Wheel, Zoom - Ctrl+Wheel и Ctrl+Shift+Wheel). Печать любой прямоугольной части векторного изображения.	2.2.1, 2.2.3
19	Многозадачность в операционной системе Windows.	Разработать переносимый (portable) класс ThreadPool. Объект этого класса содержит несколько потоков, находящихся в состоянии ожидания. К объекту можно обратиться с запросом выполнить некоторый метод некоторого объекта в потоке. Для выполнения процедуры объект класса ThreadPool подключает поток из пула ожидающих потоков. (Доп. задание: класс ThreadPool должен работать на Windows и Linux).	2.2.1, 2.2.3
20	Динамически-загружаемые библиотеки – DLL.	Существует телефонный справочник Минска в текстовом формате Телефон, Фамилия, Имя, Отчество, Улица, Дом, Корпус, Квартира. Разработать базу данных, которая используется методом проецирования файлов в память. Загрузить в базу данных телефоны и адреса из текстового файла. Разработать программу гибкого поиска данных в телефонном справочнике. Несколько копий программы должны работать с одной и той же базой данных в памяти. Расположить ядро БД и функции поиска данных в DLL-библиотеке.	2.2.1, 2.2.3
21	Механизмы синхронизации в операци-	Разработать программу, которая выполняет параллельный поиск телефонных адресов из	2.2.1, 2.2.3

	онной системе Windows.	задачи 4 и формирует отчет. Данные для поиска берутся из текстового файла. Результаты поиска (отчет) помещаются в текстовый файл. Параллельный поиск должен выполняться в нескольких потоках. Распределение данных по рабочим потокам и сбор результатов следует выполнять через трубы (Pipes).	
22	Перехват системных вызовов операционной системы Windows.	Разработать программу, которая запускает приложение, переданное программе на вход. После запуска приложения выполняется журнализация всех обращений приложения к реестру и файловой системе. Для этого в запущенное приложение подгружается DLL-библиотека, перехватывающая функции работы с файловой системой (NtXxxx). Перехват обращений к реестру можно осуществить любым известным способом.	2.2.1, 2.2.3
24,25	Разработка драйверов для операционной системы Windows.	Разработать драйвер (виртуального устройства), который отслеживает запуск некоторого процесса X. При запуске этого процесса драйвер запускает другой процесс Y. Как только процесс X завершается по какой-то причине, драйвер выгружает процесс Y.	2.2.1, 2.2.3

2.5 Курсовой проект, его характеристика

Курсовой проект по данному предмету - это самостоятельная программная разработка студента по заданной теме в области системного программирования.

Целью курсового проектирования является приобретение навыков в реализации цикла разработки программного средства, начиная от технического задания на проект, анализа предметной области и заканчивая оформлением пояснительной записки и чертежа форматом А1, в соответствии с требованиями действующих стандартов. Особое внимание следует уделить разработке алгоритмов и реализации программного средства.

В состав курсового проекта входят:

- пояснительная записка;
- графическая часть;
- работающее программное средство.

Пояснительная записка должна отражать основные этапы разработки программного средства.

Перечень тем курсовых проектов

1. Проектирование системы поддержки функционирования параллельных процессов при обработке данных в многозадачной операционной системе.
2. Разработка средств управления функционированием параллельных процессов для однозадачной операционной системы.
3. Исследование взаимодействия процессов в системах реального времени.
4. Разработка грамматики проблемно-ориентированного языка и транслятора с данного языка на промежуточный язык.
5. Разработка операционной среды для поддержания многопрограммного режима работы с разделением времени.
6. Разработка системных средств поддержки функционирования системы реального времени.
7. Разработка системных сервисов поддержки тестирования программ.
8. Разработка системных сервисов мониторинга работы программ.
9. Разработка драйверов внешних устройств.
10. Разработка драйверов виртуальных устройств.

2.6 Контрольная работа, ее характеристика (для дистанционной формы обучения)

Основная цель выполнения контрольных работ состоит в закреплении знаний и навыков, полученных при изучении теоретического материала.

№ темы по п.1	Наименование контрольной работы	Содержание	Обеспеченность по пункту 2.2
1	2	3	4
4	Контрольная работа №1 Средства межпроцессного взаимодействия (сигналы)	Цель работы – изучить методы и средства взаимодействия процессов с использованием сигналов в ОС Linux. Пример задания. Организовать функционирование процессов следующей структуры: отец-сын-сын. Процессы определяют свою работу выводом сообщений вида: N pid ppid текущее время (мсек) (N – текущий номер сообщения) на экран. “Отец” одновременно, посылает сигнал SIGUSR1 “сыновьям”. “Сыновья”, получив данный сигнал, посылают в ответ “Отцу” сигнал SIGUSR2. “Отец”, получив сигнал SIGUSR2, через время $t=100$ мсек одновременно, посылает сигнал SIGUSR1 “сыновьям”. И так далее... Написать функции-обработчики сигналов, которые при получении сигнала выводят сообщение о получении сигнала на экран. При получении/посылке сигнала они выводят соответствующее сообщение: N pid ppid текущее время (мсек) сын такой-то get/putSIGUSRm. Предусмотреть механизм для определения “Отцом”, от кого из “Сыновей” получен сигнал.	2.2.1, 2.2.2
4	Контрольная работа №2 Средства межпроцессного взаимодействия (каналы, разделяемая память, семафоры)	Цель работы – изучить методы и средства взаимодействия процессов с использованием каналов, разделяемой памяти и семафоров в ОС Linux. Пример задания. Создать два дочерних процесса. Родительский процесс создаёт семафоры (sem1), (sem2) и 2 неименованных канала (кан1 и кан2). Оба дочерних процесса непрерывно записывают в каналы по 100 строк вида: номер стро-	2.2.1, 2.2.2

		ки, pid_процесса, текущее_время (мсек). Всего процессы должны записать 1000 строк. Семафоры (sem1), (sem2) используются процессами для разрешения кому из процессов получить доступ к каналу. Родительский процесс читает из каждого канала по 75 строк и выводит их на экран. Дочерние процессы начинают операции с каналами после получения сигнала SIGUSR2 от родительского процесса.	
15,16, 17,18	Контрольная работа № 3 Пользовательский интерфейс операционной системы Windows. Событийная архитектура, структура сообщения, цикл обработки сообщений Windows. Обработка сообщений от мыши и клавиатуры. Понятие фокуса ввода. Механизм прорисовки окон. Области отсечения. Особенности обработки сообщения WM_PAINT. Принципы построения графической системы GDI. Понятие контекста устройства. Объекты контекста устройства. Рисование геометрических фигур.	Цель работы – изучить: пользовательский интерфейс операционной системы Windows, событийную архитектуру, структуру сообщения, цикл обработки сообщений Windows, обработку сообщений от мыши и клавиатуры, понятие фокуса ввода, механизм прорисовки окон, особенности обработки сообщения WM_PAINT, принципы построения графической системы GDI, понятие контекста устройства, вывод геометрических фигур. Пример задания. Рисование в окне с помощью мыши. Изображение должно состоять из прямых, ломаных, прямоугольников, эллипсов, многоугольников, текстовых надписей. Сохранение рисунка на диск в векторном формате - в виде enhanced-метафайла. Загрузка картинки с диска и продолжение рисования. Перемещение (Pan) и масштабирование (Zoom) рисунка с помощью колесика мыши (Pan - Wheel и Shift+Wheel, Zoom - Ctrl+Wheel и Ctrl+Shift+Wheel). Печать любой прямоугольной части векторного изображения.	2.2.3
21,22, 23,24	Контрольная работа № 4 Разработка драйверов для операционной системы Windows	Цель работы – изучить структуру драйвера Windows, точки входа в драйвер, объект, описывающий драйвер, объект, описывающий устройство, объект, описывающий файл, запрос ввода-вывода, менеджер ввода-вывода, стек драйверов, организация сетевых драйверов. Пример задания. Разработать драйвер (виртуального устройства), который отслеживает запуск некоторого процесса X. При запуске этого процесса драйвер за-	2.2.3

		пускает другой процесс Y. Как только процесс X завершается по какой-то причине, драйвер выгружает процесс Y.	
--	--	--	--

2.7 Индивидуальная практическая работа (для дистанционной формы обучения)

№ темы по п.1	Наименование индивидуальной практической работы	Содержание	Обеспеченность по пункту 2.2
1	2	3	4
2	ИПР №1 Система команд, файловая структура, работа с файлами и управление ОС Unix/Linux с помощью интерпретатора BASH	Цель работы – изучение команд ОС UNIX для работы с файлами, каталогами, дисками, системной датой и временем; текстового редактора и файлового менеджера, исследовать основные объекты, команды, типы данных и операторы управления интерпретатора bash; создать скрипт-файл. Пример задания. Написать программу копирования одного файла в другой, имена файлов передаются в качестве аргументов командной строки при запуске программы. Предусмотреть копирование прав доступа к файлу.	2.2.1, 2.2.2
3	ИПР № 2 Основные принципы программирования, процессы и потоки в ОС Unix/Linux	Цель работы – изучение файловой системы ОС Linux и основных функций для работы с каталогами и файлами, исследовать методы создания процессов в ОС Linux, основные функции создания и управления процессами, обмен данными между процессами. Пример задания. Разработать программу «интерпретатор команд», которая воспринимает команды, вводимые с клавиатуры, и осуществляет их корректное выполнение. Для этого каждая вводимая команда должна выполняться в отдельно запускаемом процессе с использованием вызова exec (). Предусмотреть контроль ошибок.	2.2.1, 2.2.2
19,20	ИПР № 3 Разработка многопоточных приложений	Цель работы – изучить создание многопоточных и многопроцессных приложений Windows, разработку и под-	2.2.3

	приложений Windows	<p>ключение DLL-библиотек, синхронизацию многопоточных приложений.</p> <p>Пример задания. Разработать переносимый (portable) класс ThreadPool. Объект этого класса содержит несколько потоков, находящихся в состоянии ожидания. К объекту можно обратиться с запросом выполнить некоторый метод некоторого объекта в потоке. Для выполнения процедуры объект класса ThreadPool подключает поток из пула ожидающих потоков. класс ThreadPool должен работать на Windows и Linux.</p>	
25	ИПР № 4 Перехват системных вызовов операционной системы Windows	<p>Цель работы – изучить перехват API функций ОС Windows в пользовательском режиме, интерфейсный модуль NTDLL.DLL, API функции ОС Windows в режиме ядра, модуль NTOSKRNL.EXE, технологии перехвата функций в ядре за счет модификации таблицы дескрипторов функций ОС Windows, технология перехвата функций в ядре за счет исправления кода функции ОС Windows, правила разработки функций-перехватчиков.</p> <p>Пример задания. Разработать программу, которая запускает приложение, переданное программе на вход. После запуска приложения должна выполняться журнализация всех обращений приложения к реестру и файловой системе. Для этого в запущенное приложение должна подгружаться DLL-библиотека, перехватывающая функции работы с файловой системой (NtXxxx). Перехват обращений к реестру можно осуществить любым известным способом</p>	2.2.3

3. 1 Учебно-методическая карта учебной дисциплины в дневной форме обучения

Номер раздела, темы по п.1	Название раздела, темы	Количество аудиторных часов			Самостоятельная работа, часы	Форма контроля знаний студентов
		ЛК	ПЗ	Лаб. зан.		
1	2	3	4	5	6	7
4 семестр						
	Раздел 1. Операционная система UNIX (LINUX)	40		32	70	
1	Разработка программ в ОС UNIX	6		2	6	Отчет по ЛР
2	Файловая система	12		6	18	Отчет по ЛР
3	Процессы в ОС UNIX	6		8	14	Отчет по ЛР
4	Взаимодействие между процессами	10		8	18	Отчет по ЛР
5	Потоки в ОС UNIX	6		8	14	Отчет по ЛР
	Раздел 2. Основы операционных систем	30			28	
6	Введение в операционные системы	2			2	Текущий опрос
7	Основы операционных систем	4			4	Текущий опрос
8	Процессы	6			6	Текущий опрос
9	Ресурсы	2			2	Текущий опрос
10	Память. Управление памятью	2			2	Текущий опрос
11	Организация виртуальной памяти	4			2	Текущий опрос
12	Планирование в операционных системах	2			2	Текущий опрос
13	Управление вводом-выводом и файлами.	2			2	Текущий опрос
14	Аппаратно-программные особенности современных процессоров, ориентированные на поддержку многозадачных систем.	6			6	Текущий Опрос
	Текущая аттестация					экзамен
	Итого	70		32	98	
5 семестр						
	Раздел 3. Операционная система Windows в пользовательском режиме	24		20	56	
15	Введение в операционную систему Windows	2			4	Текущий опрос
16	Оконный пользовательский интерфейс	4		4	12	Отчет по ЛР
17	Ресурсы программ	2			4	Текущий опрос
18	Интерфейс графических устройств	6		4	12	Отчет по ЛР
19	Многозадачность в ОС Windows	6		8	14	Отчет по ЛР
20	Динамически подключаемые библиотеки	4		4	10	Отчет по ЛР

	Раздел 4. Операционная система Windows в режиме ядра	24		12	44	
21	Отладка программ в режиме ядра	2		2	6	Отчет по ЛР
22	Системные механизмы ядра	6		2	10	Отчет по ЛР
23	Виртуальное адресное пространство	6			8	Текущий опрос
24	Драйверы ОС Windows	6		6	14	Отчет по ЛР
25	Перехват функций ОС Windows	4		2	6	Отчет по ЛР
	Текущая аттестация					экзамен
	Итого	48		32	100	
	Всего	118		64	198	

3.2 Учебно-методическая карта учебной дисциплины в дистанционной форме обучения

Номер раздела, темы по п.1	Название раздела, темы	Количество работ			Самостоятельная работа, часы	Форма контроля знаний студентов
		КР	ИПР	Лаб. зан.		
1	2	3	4	5	6	7
	5 семестр					
	Раздел 1. Операционная система UNIX (LINUX)				142	
1	Разработка программ в ОС UNIX				14	
2	Файловая система		ИПР №1		36	Защита ИПР
3	Процессы в ОС UNIX		ИПР №2		28	Защита ИПР
4	Взаимодействие между процессами	КР №1 КР №2			36	Защита КР
5	Потоки в ОС UNIX				28	
	Раздел 2. Основы операционных систем				58	
6	Введение в операционные системы				4	
7	Основы операционных систем				8	
8	Процессы				12	
9	Ресурсы				4	
10	Память. Управление памятью				4	
11	Организация виртуальной памяти				8	
12	Планирование в операционных системах				4	
13	Управление вводом-выводом и файлами.				4	
14	Аппаратно-программные особенности современных процессоров, ориентированные на поддержку многозадачных систем.				10	
	Текущая аттестация					экзамен
	Итого	2	2		200	

	6 семестр						
	Раздел 3. Операционная система Windows в пользовательском режиме				86		
15	Введение в операционную систему Windows	КР №3			10	Защита КР	
16	Оконный пользовательский интерфейс	КР №3			10	Защита КР	
17	Ресурсы программ	КР №3			10	Защита КР	
18	Интерфейс графических устройств	КР №3			10	Защита КР	
19	Многозадачность в ОС Windows		ИПР №3		22	Защита ИПР	
20	Динамически подключаемые библиотеки		ИПР №3		24	Защита ИПР	
	Раздел 4. Операционная система Windows в режиме ядра				94		
21	Отладка программ в режиме ядра	КР №4			6	Защита КР	
22	Системные механизмы ядра	КР №4			20	Защита КР	
23	Виртуальное адресное пространство	КР №4			20	Защита КР	
24	Драйверы ОС Windows	КР №4			28	Защита КР	
25	Перехват функций ОС Windows		ИПР №4		20	Защита ИПР	
	Текущая аттестация					экзамен	
	Итого	2	2		180		
	Всего	4	4		380		

3.3 Учебно-методическая карта учебной дисциплины в вечерней форме обучения для получения высшего образования, интегрированного со средним специальным образованием

Номер раздела, темы по п.1	Название раздела, темы	Количество аудиторных часов			Самостоятельная работа, часы	Форма контроля знаний студентов
		ЛК	ПЗ	Лаб. зан.		
1	2	3	4	5	6	7
	4 семестр					
	Раздел 1. Операционная система UNIX (LINUX)				40	
1	Разработка программ в ОС UNIX				4	
2	Файловая система			4	8	Отчет по ЛР
3	Процессы в ОС UNIX	2			10	Текущий опрос
4	Взаимодействие между процессами	2		4	10	Отчет по ЛР
5	Потоки в ОС UNIX				8	

	*Раздел 2. Основы операционных систем					
6	Введение в операционные системы					
7	Основы операционных систем					
8	Процессы					
9	Ресурсы					
10	Память. Управление памятью					
11	Организация виртуальной памяти					
12	Планирование в операционных системах					
13	Управление вводом-выводом и файлами.					
14	Аппаратно-программные особенности современных процессоров, ориентированные на поддержку многозадачных систем.					
	*Раздел 3. Операционная система Windows в пользовательском режиме					
15	Введение в операционную систему Windows					
16	Оконный пользовательский интерфейс					
17	Ресурсы программ					
18	Интерфейс графических устройств					
19	Многозадачность в ОС Windows					
20	Динамически подключаемые библиотеки					
	Раздел 4. Операционная система Windows в режиме ядра				44	
21	Отладка программ в режиме ядра	2			8	Текущий опрос
22	Системные механизмы ядра			4	8	Отчет по ЛР
23	Виртуальное адресное пространство				8	
24	Драйверы ОС Windows	2		4	10	Отчет по ЛР
25	Перехват функций ОС Windows				10	
	Текущая аттестация					экзамен
	Итого	8		16	84	

* Разделы 2 и 3 перечислены на основе программы обучения в колледже.

3.4 Учебно-методическая карта учебной дисциплины в заочной форме обучения для получения высшего образования, интегрированного со средним специальным образованием

Номер раздела, темы по п. 1	Название раздела, темы	Количество аудиторных часов			Самостоятельная работа, часы	Форма контроля знаний студентов
		ЛК	ПЗ	Лаб. зан.		
1	2	3	4	5	6	7
	4 семестр					
	Раздел 1. Операционная система UNIX (LINUX)				40	
1	Разработка программ в ОС UNIX				4	

2	Файловая система				8	
3	Процессы в ОС UNIX		2		10	Текущий опрос
4	Взаимодействие между процессами				10	
5	Потоки в ОС UNIX				8	
	*Раздел 2. Основы операционных систем					
6	Введение в операционные системы					
7	Основы операционных систем					
8	Процессы					
9	Ресурсы					
10	Память. Управление памятью					
11	Организация виртуальной памяти					
12	Планирование в операционных системах					
13	Управление вводом-выводом и файлами.					
14	Аппаратно-программные особенности современных процессоров, ориентированные на поддержку многозадачных систем.					
	*Раздел 3. Операционная система Windows в пользовательском режиме					
15	Введение в операционную систему Windows					
16	Оконный пользовательский интерфейс					
17	Ресурсы программ					
18	Интерфейс графических устройств					
19	Многозадачность в ОС Windows					
20	Динамически подключаемые библиотеки					
	Раздел 4. Операционная система Windows в режиме ядра				60	
21	Отладка программ в режиме ядра	2			12	Текущий опрос
22	Системные механизмы ядра				12	
23	Виртуальное адресное пространство				12	
24	Драйверы ОС Windows			4	12	Отчет по ЛР
25	Перехват функций ОС Windows				12	
	Текущая аттестация					экзамен
	Итого	2	2	4	100	

* Разделы 2 и 3 перечислены на основе программы обучения в колледже.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ С ДРУГИМИ
УЧЕБНЫМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ

Перечень учебных дисциплин	Кафедра, обеспечивающая учебную дисциплину по п.1	Предложения об изменениях в содержании по изучаемой учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)	Подпись заведующего кафедрой, обеспечивающей учебную дисциплину по п.1
1	2	3	4	5
Базы данных	ПОИТ	нет	Рекомендовать к утверждению Протокол № <u>8</u> от <u>24.11.2014</u>	

Заведующая кафедрой ПОИТ

Н.В.Лапицкая