Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Факультет непрерывного и дистанционного обучения

Кафедра информатики

Электронный учебно-методический комплекс  
по дисциплине

Операционные системы и среды

Часть 2

Для студентов специальности

1-31 03 04 Информатика

Минск 2011

# Общие сведения

## Сведения об ЭУМК

Электронный учебно-методический комплекс по дисциплине «Операционные системы и среды» предназначен для студентов специальности «Информатика» вузов, а также может быть использован магистрантами, аспирантами и специалистами соответствующего профиля.

Электронный учебно-методический комплекс составлен на основе рабочей учебной программы по курсу «Операционные системы и среды», утверждённой деканом факультета непрерывного и дистанционного обучения <дата утверждения>, регистрационный № УД 11‑XX‑YY/Р и рабочего учебного плана специальности 1-31 03 04 «Информатика».

**Составители:**

**С. И. Сиротко,** доцент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат физико-математических наук.

**В. В. Шендер**, ассистент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр технических наук.

Рассмотрен и рекомендован к изданию на заседании кафедры информатики, протокол № 3 от 26.09.2011.

Одобрен и рекомендован к изданию методической комиссией факультета компьютерных систем и сетей, протокол № \_\_ от \_\_.\_\_.2011.

## Методические рекомендации по изучению дисциплины

В соответствии с учебным планом студенты дистанционной формы обучения специальности «Информатика» изучают курс «Операционные системы и среды» в течение 8-го и 9-го семестров.

Учебным планом по данному курсу предусмотрены 2 части, включающие изучение теоретических вопросов и решение практических задач. Вторая часть изучается в течение 9‑го семестра, содержит 2 индивидуальных практические и 1 контрольную работу и заканчивается сдачей экзамена, к которому студенты допускаются только при условии выполненных и защищенных индивидуальных практических и контрольных работ. Также предусмотрено выполнение курсового проекта, направленное на закрепление пройденного материала и приобретение практических навыков.

Рекомендуется изучать курс «Операционные системы и среды» в соответствии с рабочей программой. Сначала необходимо ознакомиться с содержанием курса, затем изучить рекомендуемую литературу, обращая внимание на вопросы, выделенные в рабочей программе, после чего изучить теоретическое изложение курса по приведенным разделам, темам и вопросам, выполнить задачи для решения (выполнения контрольных работ) в соответствии с заданием.

## Рабочая учебная программа

**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

УТВЕРЖДАЮ

Декан факультета непрерывного и дистанционного обучения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. М. Бондарик

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2011 г.

Регистрационный № УД-11-23-\_\_\_/р.

**Операционные системы и среды**

Рабочая учебная программа

для специальности 1‑31 03 04

**Информатика**

Факультет **непрерывного и дистанционного обучения**

Кафедра **информатики**

Курс **4, 5**

Части **1, 2**

Контрольные работы **3 работы**

в т.ч.: часть 1 2 работы

часть 2 1 работы

Индивидуальные практические работы: **4 работы**

в т.ч.: часть 1 2 работы

часть 1 2 работы

Курсовой проект **9** **семестр**

Всего часов **268 часов**

в т.ч.: часть 1 122 часов

часть 2 146 часов

Зачет **8 семестр**

Экзамен **9 семестр**

Форма получения

высшего образования **дистанционная**

Минск 2011

Рабочая учебная программа составлена на основе базового учебного плана специальности 1-31 03 04 «Информатика», утвержденного Министерством Образования Республики Беларусь 18.04.2006 г., рег. № G.31-025/баз.

**Составитель:**

**С. И. Сиротко,** доцент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат физико-математических наук.

**В. В. Шендер,** ассистент кафедры информатики Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр технических наук.

Рассмотрена и рекомендована к утверждению на заседании кафедры информатики, протокол № 3 от 26.09.2011 г.

Зав. кафедрой информатики \_\_\_\_\_\_\_\_\_\_\_ Л. И. Минченко

Одобрена и рекомендована к утверждению Советом факультета компьютерных систем и сетей Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», протокол № \_\_ от \_\_.\_\_.2011.

Председатель \_\_\_\_\_\_\_\_\_\_\_ В. А. Прытков

СОГЛАСОВАНО

Начальник ОМОУП \_\_\_\_\_\_\_\_\_\_\_ Ц. С. Шикова

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

|  |  |  |  |
| --- | --- | --- | --- |
| Название дисцип­лины, с которой требуется согласо­вание | Кафедра, обеспе­чивающая изуче­ние дисциплины | Предложения об изменениях в со­держании учеб­ной программы по изу­чаемой дисциплине | Решение, принятое кафедрой, разработав­шей учебную программу (с указанием даты и номера протокола) |
| Алгоритмы и структуры данных | Информатики | нет | согласовано |
| Компьютерные сети | Информатики | нет | согласовано |
| Архитектура компьютеров | Информатики | нет | согласовано |
| Операционные системы | Информатики | нет | согласовано |
| Системное программирование | Информатики | нет | согласовано |

СОГЛАСОВАНО:

Зав. кафедрой информатики Л.И. Минченко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**Цель преподавания дисциплины.** Целью изучения дисциплины «Операционные системы и среды» является основ построения и функционирования операционных систем (ОС), преимущественно на примере ОС семейства Unix/Linux, и их компонентов, технологий взаимодействия задач и многозадачного программирования, а также получение теоретических знаний и практических навыков работы в различных ОС, создания программного обеспечения (ПО) и основ администрирования систем.

**Задачи изучения дисциплины.** В результате изучения курса студенты должны освоить теоретически и практически различные аспекты архитектуры и функционирования различных ОС, подходы и методы решения задач параллельного программирования, особенности функционирования, разработки и сопровождения ПО в многозадачной среде.

**Знать:**

– архитектуры, концепции и принципы построения и функционирования ОС различного типа и назначения;

– структуру, особенности, средства пользовательского и программного интерфейсов ОС семейств Unix/Linux и Win 32 (Win NT);

– принципы, модели, технологии и механизмы многозадачного программирования и обеспечения взаимодействия вычислительных процессов (потоков), методы и приемы параллельного программирования;

– основы администрирования и управления ОС:

– особенности и принципы разработки системного ПО;

**Уметь:**

– выполнять настройку подсистем ОС и отдельных приложения, базовое администрирование системы;

– разрабатывать ПО с использованием как универсальных, так и командных (скриптовых) языков для решения широкого круга задач;

– проектировать и создавать ПО системного уровня, а также мультипрограммные комплексы, включая выявление и разрешение конфликтов.

**Иметь представление о:**

– областях и особенностях применения ОС и иного системного ПО;

– особенностях разработки и сопровождения ОС, их компонентов и другого системного ПО.

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

|  |  |  |
| --- | --- | --- |
| №  п./п. | Название дисциплины | Раздел, тема |
| 1. | Программирование | Все разделы дисциплины |
| 2. | Алгоритмы и структуры данных | Все разделы дисциплины |
| 3. | Операционные системы | Все разделы дисциплины |
| 4. | Системное программирование | Все разделы дисциплины |
| 5. | Архитектура компьютеров | Все разделы дисциплины |

Кроме того, она тесно связана с дисциплиной «Компьютерные сети».

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № темы | Название и содержание тем | Контрольная работа | Индивидуальная практическая работа | Оснащение индивидуальных практических работ | Литература  (номера) | Рекомендуемый объём для изучения  (в часах) | Форма контроля знаний |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Четвертый курс** | | | | | | | |
| **Часть1, семестр 8** | | | | | | | |
| 1 | **Тема 1. Операционные системы (ОС).** Структура, назначение, виды ОС. Архитектуры ОС и их характеристика. |  |  |  | 1, 4, 6, 7, 8, 9, 13, 14, 18 | 8 |  |
| 2 | **Тема 2. Вычислительные процессы и потоки.** Понятие вычислительного процесса, потока. Атрибуты процесса и потока. Жизненный цикл и состояния процессов (потоков). Реализация процессов и потоков. |  |  |  | 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 17 | 8 |  |
| 3 | **Тема 3. Основы многозадачности и мультипрограммирования.** Многозадачность и многопоточность. Параллелизм в вычислительных системах. Подсистема управления процессами. |  |  |  | 2, 3, 4, 6, 7, 8, 9, 13, 14, 15, 22, 24 | 8 |  |
| 4 | **Тема 4. Планирование выполнения программ.** Задачи планирования, планировщик процессов. Методы и модели планирования выполнения программ. Приоритеты, управление приоритетами. | 1 |  |  | 2, 3, 4, 6, 7, 8, 9, 13, 14, 15, 22, 24 | 6 | Зачёт по контрольной работе |
| 5 | **Тема 5. Особенности выполнения процессов в многозадачной среде.** Взаимодействие процессов в многозадачной среде. Задачи и проблемы взаимодействия. Модели взаимодействия. Конфликты, методы их разрешения. |  |  | 1-6, 9-11 | 2, 4, 6, 9, 13, 14, 17, 22, 24 | 8 |  |
| 6 | **Тема 6. Задача взаимного исключения.** Критический ресурс, критическая секция. Коллизии. Механизмы реализации защиты критических секций. |  |  |  | 2, 4, 6, 7, 8, 9, 15, 16, 17 | 8 |  |
| 7 | **Тема 7. Задача синхронизации процессов.** Ситуации ожидания. Объекты ожидания. Функции ожидания и синхронизация функциями ожидания. |  | 1 | 1, 3-7, 9-11 | 2, 3, 4, 6, 7, 8, 9, 15, 16, 17 | 8 | Зачёт по индивидуальной практической работе |
| 8 | **Тема 8. Событийное управление.** События и их роль в управлении процессами. Сигналы и сообщения. Обработка прерываний и исключительных ситуаций. |  | 2 | 1, 3, 5-6, 10 | 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 17 | 4 | Зачёт по индивидуальной практической работе |
| 9 | **Тема 9. Элементы многозадачности в однозадачной среде.** Прерывания, обработчики прерываний и резидентные программы в однозадачной среде. | 2 |  |  | 2, 4, 19, 20, 21, 24, 25 | 4 | Зачёт по контрольной работе |
| 10 | **Тема 10. Средства контроля времени выполнения.** Таймеры, функции времени, системные счетчики, другие средства контроля времени. |  |  |  | 2, 3, 6, 7, 8, 9, 15, 16, 17, 19 | 4 |  |
|  | | | | | | Зачет (20 ч) | |
| **Пятый курс** | | | | | | | |
| **Часть 2, семестр 9** | | | | | | | |
| 11 | **Тема 1. Задача обмена данными.** Обмен данными в многозадачной среде. Механизмы межпроцессного обмена. Сравнительная характеристика, критерии сравнения. |  |  |  | 2, 3, 4, 6, 7, 8, 9, 15, 16, 17 | 4 |  |
| 12 | **Тема 2. Разделяемая память.** Особенности организации и использования разделяемой памяти. |  |  |  | 3, 4, 6, 7, 8, 9, 15, 16, 17 | 4 |  |
| 13 | **Тема 3. Использование разделяемых объектов файловой системы.** Разделяемые файлы, каналы, «почтовые ящики», сокеты. |  | 3 | 1, 3-7, 9-11 | 4, 5, 6, 9, 15, 16, 17 | 4 | Зачёт по индивидуальной практической работе |
| 14 | **Тема 4. Асинхронный ввод-вывод.** Организация асинхронного ввода-вывода и его особенности. Средства контроля и управления асинхронными операциями. Параллельный доступ к файлам и коммуникационным ресурсам. |  |  |  | 2, 4, 6, 9, 17 | 4 |  |
| 15 | **Тема 5. Подсистема памяти.** Задачи и функции управления памятью, распределения памяти. Методы управления памятью. Виртуальное адресное пространство и его использование при распределении памяти. |  |  |  | 2, 3, 4, 6, 7, 8, 9, 17 | 8 |  |
| 16 | **Тема 6. Отображение файлов в память.** Использование файловых отображений: доступ к данным, разделяемые данные. |  |  |  | 4, 6, 7, 9, 17 | 4 |  |
| 17 | **Тема 7. Подсистема управления файлами.** Файловые системы – принципы организации, основные функции. Дополнительные требования и функции. Примеры реализаций. |  |  |  | 4, 6, 9, 13, 14, 26 | 8 |  |
| 18 | **Тема 8. Основы сетевого программирования.** Сокеты для доступа к сетевым интерфейсам. Использование сокетов в сетевых приложениях. |  | 4 | 1, 3-7, 9-11 | 3, 6, 9, 7, 10, 11, 12, 17, 26 | 8 |  |
| 19 | **Тема 9. Управление внешними устройствами.** Логические устройства, драйверы. Виды устройств. Организация доступа к внешним устройствам. |  |  |  | 2, 3, 4, 6, 7, 8, 9, 10, 13, 14, 15, 16, 24 | 4 |  |
| 20 | **Тема 10. Интеграция приложений, разнородные приложения.** Удаленный доступ, удаленное исполнение. Распределенные приложения. Технологии OLE, COM, ActiveX, .NET. «Облачные» технологии. | 3 |  |  | 6, 9, 17 | 8 | Зачёт по контрольной работе |
| 21 | **Тема 11. Средства безопасности и их поддержка в ОС.** Безопасность в вычислительных системах. Угрозы, защищаемые объекты, участники подсистемы безопасности. Задачи подсистемы безопасности. Примеры реализаций. |  |  |  | 4, 6, 7, 8, 9, 10, 12, 17, 26 | 4 |  |
| 22 | **Тема 12. Системы реального времени.** Управляющие и встраиваемые системы. Механизмы системного ПО для режимов реального времени. Примеры реализаций. |  |  |  | 2, 4, 13, 14, 22 | 8 |  |
|  | | | | | | Курсовой проект (24 ч) | |
|  | | | | | | Экзамен (30 ч) | |

1. Наименование тем, их содержание

Тема 1. Задача обмена данными.

Обмен данными в многозадачной среде. Механизмы межпроцессного обмена. Сравнительная характеристика, критерии сравнения.

**Рекомендуемая литература:** 2, 3, 4, 6, 7, 8, 9, 15, 16, 17

Тема 2. Разделяемая память.

Особенности организации и использования разделяемой памяти.

**Рекомендуемая литература:** 3, 4, 6, 7, 8, 9, 15, 16, 17

Тема 3. Использование разделяемых объектов файловой системы.

Разделяемые файлы, каналы, «почтовые ящики», сокеты.

**Рекомендуемая литература:** 4, 5, 6, 9, 15, 16, 17

Тема 4. Асинхронный ввод-вывод.

Организация асинхронного ввода-вывода и его особенности. Средства контроля и управления асинхронными операциями. Параллельный доступ к файлам и коммуникационным ресурсам.

**Рекомендуемая литература:** 2, 4, 6, 9, 17

Тема 5. Подсистема памяти.

Задачи и функции управления памятью, распределения памяти. Методы управления памятью. Виртуальное адресное пространство и его использование при распределении памяти.

**Рекомендуемая литература:** 2, 3, 4, 6, 7, 8, 9, 17

Тема 6. Отображение файлов в память.

Использование файловых отображений: доступ к данным, разделяемые данные.

**Рекомендуемая литература:** 4, 6, 7, 9, 17

Тема 7. Подсистема управления файлами.

Файловые системы – принципы организации, основные функции. Дополнительные требования и функции. Примеры реализаций.

**Рекомендуемая литература:** 4, 6, 9, 13, 14, 26

Тема 8. Основы сетевого программирования.

Сокеты для доступа к сетевым интерфейсам. Использование сокетов в сетевых приложениях.

**Рекомендуемая литература:** 3, 6, 9, 7, 10, 11, 12, 17, 26

Тема 9. Управление внешними устройствами.

Логические устройства, драйверы. Виды устройств. Организация доступа к внешним устройствам.

**Рекомендуемая литература:** 2, 3, 4, 6, 7, 8, 9, 10, 13, 14, 15, 16, 24

Тема 10. Интеграция приложений, разнородные приложения.

Удаленный доступ, удаленное исполнение. Распределенные приложения. Технологии OLE, COM, ActiveX, .NET. «Облачные» технологии.

**Рекомендуемая литература:** 6, 9, 17

Тема 11. Средства безопасности и их поддержка в ОС.

Безопасность в вычислительных системах. Угрозы, защищаемые объекты, участники подсистемы безопасности. Задачи подсистемы безопасности. Примеры реализаций.

**Рекомендуемая литература:** 4, 6, 7, 8, 9, 10, 12, 17, 26

Тема 12. Системы реального времени.

Особенности режима реального времени. Управляющие и встраиваемые системы. Механизмы системного ПО для режимов реального времени. Примеры реализаций.

**Рекомендуемая литература:** 2, 4, 13, 14, 22

2. Индивидуальные практические работы, их характеристика

|  |  |  |  |
| --- | --- | --- | --- |
| № п./п. | Тема | Содержание | Объём в часах |
| 1. | Обмен данными между процессами | Разработка многопроцессного приложения (комплекса), предусматривающего взаимодействие процессов и обмен данными между ними | 8 |
| 2. | Программирование с использованием сокетов | Разработка программ, реализующих доступ к сетевым ресурсам и/или службам, серверных и клиентских приложений | 8 |
| **Итого** | | | **16** |

3. Контрольные работы, их характеристика

|  |  |  |  |
| --- | --- | --- | --- |
| № п./п. | Тема | Характеристика | Объём в часах |
| 1. | Распределенные и разнородные приложения | Цель работы – ознакомление с принципами и технологиями распределенных вычислений и обработки данных, а также особенностями создания и функционирования распределенных приложений, включая практические примеры | 8 |
| **Итого** | | | **8** |

4. ЛИТЕРАТУРА

ОСНОВНАЯ

1. Бах, М. Дж. Архитектура операционной системы Unix. [Электронный ресурс]. – Режим доступа : readr.ru/moris-bah-arhitektura-operacionnoy-sistemi-unix.html

2. Бек, Л. Введение в системное программирование / Л.Бек ; пер. с англ. – М.: Мир, 1988. – 448 с., ил.

3. Глас, Г. Unix для программистов и пользователей / Г. Глас, К. Эйблс. – 3-е изд. – СПб.: БХВ-Петербург, 2004.

4. Гордеев, А.В. Системное программное обеспечение / А.В. Гордеев, А.Ю. Молча­нов – СПб.: Питер, 2001. – 736 с.: ил.

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

6. Рихтер, Дж. Windows для профессионалов / Дж. Рихтер. – СПб.: Питер, 2000. – 752 с.

7. Робачевский, А. М. Операционная система UNIX / А.М. Робачевский, С. Немнюгин, О. Стесик. – 2-е изд. – БХВ-Петербург, 2007.

8. Рочкинд, М. Дж. Программирование для UNIX / М.Дж. Рочкинд. – 2-е изд. – СПб.: БХВ-Петербург, 2005.

9. Руссинович, М. Внутреннее устройство Microsoft Windows. / М. Рус­си­нович, Д. Соломон. – 4-е изд. – СПб.: Питер, Русская Редакция, 2005. – 992 с.

10. Сорокина, С.И. Программирование драйверов и систем безопасности: Учеб. пособие. / С.И. Сорокина, А.Ю. Тихонов, А.Ю. Щербаков. – СПб.: БХВ-Петербург, М.: издатель Молчанов С.В. – 2002. – 256 с.: ил.

11. Стивенс, У.Р. Протоколы TCP/IP. Практическое руководство. – BHV, 2003. – 672с.

12. Стивенс, У.Р., Феннер Б., Рудофф Э.М. UNIX. Разработка сетевых приложений / У.Р. Стивенс, Б. Феннер, Э.М. Рудофф – 3-е изд. – СПб.: Питер, 2007. – 1040 с.

13. Танненбаум, Э. Современные операционные системы / Э. Таннен­ба­ум. – 2-е изд. – СПб.: Питер, 2002.

14. Таненбаум, Э., Вудхалл, А. Операционные системы. Разработка и реализация. 3-е изд. – СПб.: Питер, 2007. – 704 с.

ДОПОЛНИТЕЛЬНАЯ

15. Беляков, М. И. Мобильная операционная система / М. И. Беляков, Ю.И. Рабовер, А. А. Фридман. – М.: Радио и связь, 1991.

16. Богатырев, А. Хрестоматия по программированию на Си в UNIX. [Электронный ресурс]. – Режим доступа : opennet.ru/docs/RUS/bogatyrev; lib.ru/CTOTOR/book.txt.

17. Вильямс, А. Системное программирование в Windows 2000 для профессионалов / А. Вильямс ; пер. с англ. – СПб.: Питер, 2001. – 624 с., ил.

18. Готье, Р. Руководство по ОС UNIX / Р. Готье. – М.: Финансы и статистика, 1985.

19. Джордейн, Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT / Р. Джордейн, ; пер. с англ. – М.: Финансы и статистика, 1992. – 544 с.

20. Зубков С.В. Assembler для DOS, Windows и UNIX. / С.В. Зубков.  – 3-е изд. – М.: ДМК пресс, СПб.: Питер, 2006. – 608 с.

21. Касаткин, А.И. Профессиональное программирование на языке Си. Управление ресурсами: Справ. пособие / А.И. Касаткин. – Минск: Выш. шк., 1992. – 432 с.

22. Программирование на параллельных вычислительных системах: Пер. с англ. / Р. Бэбб, Дж. Мак-Гроу, Т. Акселрод и др.; под ред. Р.Бэбба II. – М.:Мир, 1991. – 376 с.: ил.

23. Сейдж, Р. Приемы профессиональной работы в UNIX / Рассел С. Сэйдж // Онлайн библиотека rulibrary.com [Электронный ресурс] – Режим доступа : www.rulibrary.com/book-99.html.

24. Скляров, В.А. Программное и лингвистическое обеспечение персональных ЭВМ. Системы общего назначения: Справ. пособие / В.А. Скляров. – Минск: Выш. шк., 1992. – 462 с.: ил.

25. Скляров, В.А. Программное и лингвистическое обеспечение персональных ЭВМ. Новые системы: Справ. пособие / В.А. Скляров. – Минск: Выш. шк., 1992. – 334 с.: ил.

26. UNIX – руководство системного администратора / Э. Немеет [и др.] ; пер. с англ. – Киев : BHV, 2000.

5. ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ, НАГЛЯДНЫХ И ДРУГИХ   
ПОСОБИЙ, МЕТОДИЧЕСКИХ УКАЗАНИЙ И МАТЕРИАЛОВ   
И ТЕХНИЧЕСКИХ СРЕДСТВ ОБУЧЕНИЯ

1. Операционные системы и среды : электронный учеб.-метод. комплекс [Электронный ресурс]. – Минск : БГУИР, 2007. – Режим доступа : www.bsuir.by.

2. Система тестирования "Инфотест" кафедры информатики: тесты "GNU/Linux", "GNU/Linux Advanced", "GNU/Linux Kernel"

3. ПЭВМ с установленной ОС семейства Unix/Linux, либо средства для удаленного доступа к ЭВМ с указанными ОС.

4. ПЭВМ с установленной ОС семейства Win 32.

5. Стандартный пакет утилит Unix/Linux [Электронный ресурс]. – Режим доступа : стандартные компоненты операционных систем семейства Unix

6. Компилятор gcc и другие компоненты среды программирования Unix/Linux [Электронный ресурс]. – Режим доступа : стандартные компоненты операционных систем семейства Unix

7. Среда разработки Microsoft Visual Studio [Электронный ресурс]. – Режим доступа : Компонент ПО учебных классов

8. Среда разработки Borland IDE 3.1: C/C++ compiler, assembler [Электронный ресурс]. – Режим доступа : Компонент ПО учебных классов

9. Среда разработки Microsoft Driver Developer Kit (DDK) [Электронный ресурс]. – Режим доступа : http://www.microsoft.com/whdc/devtools/ddk/.

10. Среда разработки Microsoft Software Developer Kit (SDK) [Электронный ресурс]. – Режим доступа : http://www.microsoft.com/downloads/.

10. Справочная система man Unix [Электронный ресурс]. – Режим доступа : стандартный компонент операционных систем семейства Unix

11. Информационно-справочная система Microsoft Developer Network (MSDN) [Электронный ресурс]. – Режим доступа : http://www.microsoft.com/downloads.

12. Справочная система TeachHelp [Электронный ресурс]. – Режим доступа : Компонент ПО учебных классов.

# Теоретический раздел

## Лекции

### Тема 1. Задача обмена данными.

#### 2.1. Задача обмена данными в многозадачной среде

Важнейшими задачами при разработке многозадачных и многопоточных систем, а также организации их функционирования являются:

– общее управление задачами и потоками;

– обмен данными между ними;

– обеспечение их бесконфликтного согласованного взаимодействия.

Возникающие сложности вытекают из таких отличий многозадачных систем, в частности, Win 32, как:

– асинхронное выполнение отдельных задач (процессов) и потоков, причем механизм управления им скрыт, как правило, от прикладной программы;

– разделенные и изолированные адресные пространства отдельных задач (процессов), что исключает простой совместный доступ к элементам данных.

Как уже отмечалось в предыдущих темах, важнейшими проблемами при взаимодействии задач (процессов) и потоков в многозадачной системе асинхронность их выполнения и возможность одновременного доступа к объектам[[1]](#footnote-1) (ресурсам) в то время как их адресные пространства и, соответственно, множеств собственных объектов различных задач (процессов) изолированы друг от друга.

Могут быть выделены следующие основные ситуации:

– один и тот же объект должен использоваться несколькими программами, для чего необходим доступ к нему со стороны всех их;

– один и тот же объект используется несколькими программами, но требуется устранить возможные коллизии и некорректности одновременного (и заранее не предсказуемого) доступа.

Обеспечение совместного доступа нескольких процессов к разделяемым объектам вместе с решением возникающих сопутствующих проблем принято определять как задачу межпроцессного обмена.

Как и средства синхронизации, средства взаимодействия программ опираются на использование глобальных системных объектов. Отличие состоит в том, что обычно предполагается передача значительных объемов данных, поэтому среди требований к механизмам поддержки обмена на первый план выходит производительность и «пропускная способность». Также приходится учитывать совместимость форматов данных между участниками обмена, особенно в разнородных системах.

#### 2.2. Механизмы межпроцессного обмена (краткая характеристика)

#### 2.3. Критерии сравнения, сводная сравнительная характеристика

### Тема 2. Разделяемая память.

Особенности организации и использования разделяемой памяти.

### Тема 3. Использование разделяемых объектов файловой системы.

Разделяемые файлы. Монопольный и разделяемый доступ.

Каналы

«Почтовые ящики» (mailslot)

Сокеты.

### Тема 4. Асинхронный ввод-вывод.

Организация асинхронного ввода-вывода и его особенности.

Средства контроля и управления асинхронными операциями.

Параллельный доступ к файлам и коммуникационным ресурсам.

Порты завершения ввода-вывода (I/O Completion Ports).

### Тема 5. Подсистема памяти.

Задачи и функции управления памятью, распределения памяти.

Методы управления памятью.

Виртуальное адресное пространство и его использование при распределении памяти.

### Тема 6. Отображение файлов в память.

Механизм отображения файлов.

Использование файловых отображений: файлы прямого доступа

Использование файловых отображений: разделяемые данные (разделяемая область памяти).

Роль файловых отображений в системных задачах.

### Тема 7. Подсистема управления файлами.

Файловые системы – принципы организации, основные функции.

Дополнительные требования и функции. Примеры реализаций.

### Тема 8. Основы сетевого программирования.

Сокеты для доступа к сетевым интерфейсам.

Способы передачи – с установлением соединения и без него

Адресация, семейства адресов.

Типовая схема использования датаграмных сокетов.

Типовая схема использования потоковых сокетов.

Вспомогательные функции для подсистемы сокетов.

Использование сокетов в сетевых приложениях.

Особенности организации сетевых приложений.

### Тема 9. Управление внешними устройствами.

Логические устройства, драйверы.

Виды устройств.

Организация доступа к внешним устройствам.

### Тема 10. Интеграция приложений, разнородные приложения.

Удаленный доступ, удаленное исполнение.

Распределенные приложения.

Технологии OLE, COM, ActiveX, .NET.

«Облачные» технологии.

### Тема 11. Средства безопасности и их поддержка в ОС.

Безопасность в вычислительных системах.

Угрозы, защищаемые объекты, участники подсистемы безопасности.

Задачи подсистемы безопасности.

Примеры реализаций.

### Тема 12. Системы реального времени.

Особенности режима реального времени.

Управляющие и встраиваемые системы.

Механизмы системного ПО для режимов реального времени.

Примеры реализаций.

# Практический раздел

## Контрольные работы

### Контрольная работа №1

Распределенные и разнородные приложения

#### Указания по выбору варианта

Цель работы – ознакомление с принципами и технологиями распределенных вычислений и обработки данных, а также особенностями создания и функционирования распределенных приложений, включая практические примеры

Контрольная работа должна быть оформлена в соответствии с общеустановленными нормами и правилами, предъявляемыми к выполнению контрольных работ. При выполнении задания необходимо пользоваться несколькими источниками.

Контрольная работа не предполагает строгого разделения на теоретическую и практическую части. Материал, соответствующий заданию, должен содержать как описания методов, принципов, механизмов, объектов, функций и т.д., так и примеры, необходимые для иллюстрации излагаемого.

Выбор варианта задания контрольной работы осуществляется студентом самостоятельно на основании номера зачетной книжки (последние цифры, после номера группы). Если порядковый номер превышает количество вариантов, для выбора номера варианта используется остаток от деления порядкового номера на количество вариантов. В отдельных случаях задание может быть сформулировано преподавателем индивидуально.

Таблица 1 – Варианты заданий контрольной работы № 1

|  |  |
| --- | --- |
| № п/п | Задание |
| 1. | Основные концепции, методы, технологии распределенных систем |
| 2. | Архитектуры распределенных систем. Вычислительные сети, кластеры |
| 3. | Вопросы синхронизации и поддержания целостности данных в распределенных системах |
| 4. | Распределенные вычисления: цели, особенности, подходы и технологии реализации, типичные задачи |
| 5. | Распределенные вычисления: примеры проектов |
| 6. | Децентрализованное хранение информации: цели, особенности, примеры реализаций |
| 7. | Взаимодействие в распределенных базах данных |
| 8. | Методы поиска, обработки, интеграции информации |
| 9. | «Облачные» технологии, Cloud Computing |
| 10. | Вопросы надежности и безопасности в распределенных системах |

## Индивидуальные практические работы

### Индивидуальная практическая работа №1

Обмен данными между процессами.

#### Указания по выбору варианта

Разработка многопроцессного приложения (комплекса), предусматривающего взаимодействие процессов и обмен данными между ними.

Индивидуальная практическая работа предполагает защиту выполненного задания, а также ответы на теоретические вопросы, демонстрирующие владение тематикой.

Индивидуальная практическая работа оформляется в соответствии с общеустановленными нормами и правилами.

Выбор варианта задания индивидуальной практической работы осуществляется студентом самостоятельно на основании номера зачетной книжки (последние цифры, после номера группы). Если порядковый номер превышает количество вариантов, для выбора номера варианта используется остаток от деления порядкового номера на количество вариантов. В отдельных случаях задание может быть сформулировано преподавателем индивидуально. Задания «повышенной сложности» также обсуждаются индивидуально.

#### Теоретическая часть (вопросы)

1. Задачи межпроцессного взаимодействия: взаимное исключение

2. Задачи межпроцессного взаимодействия: синхронизация

3. Задачи межпроцессного взаимодействия: обмен данными

4. Понятие критического ресурса, критической секции.

5. Модели, описывающие ситуации межпроцессного взаимодействия.

6. Объекты IPC – файловая система, коммуникационные ресурсы.

7. Объекты IPC – сигналы, сообщения, очереди сообщений.

8. Объекты IPC – разделяемая память.

9. Объекты IPC – объекты синхронизации (семафоры и т.п.).

10. Особенности асинхронного ввода-вывода и его организация.

11. Особенности взаимодействия потоков.

#### Практическая часть (варианты заданий)

##### Вариант задания №1.

Обеспечить взаимодействие «головного» (родительского) процесса с 3 (или более) порожденными им процессами посредством отдельных (т.е. по одному для каждого «потомка») неименованных каналов (pipe). Каждый из порожденных процессов (клиентов) передает родительскому (серверу) текстовые строки, например вводимые с консоли. Родительский процесс отображает присылаемые строки в порядке их поступления. После завершения всех порожденных процессов родительский также завершается.

*Примечание.* Необходимо обеспечить передачу каждому порожденному процессу дескрипторов, соответствующих именно его «персональному» каналу, и одновременно возможность для родительского процесса контролировать все каналы.

##### Вариант задания №2.

Клиент-серверный «калькулятор» с взаимодействием посредством именованного канала (FIFO). Процессы изначально независимы, имя канала известно всем участникам. Сервер создает канал, затем в цикле принимает из него сообщения, выделяет из них аргументы – числа и знаки действий – и возвращает в канал ответное сообщение с результатом вычисления (предлагается ограничиться простейшим набором арифметических операций. «Клиент» вводит или генерирует необходимые числовые аргументы и знаки операций, формирует сообщение, передает его в канал, ожидает ответного сообщения, отображает результат. Также необходимо обеспечить обработку основных ошибок и особых ситуаций, возникающих в процессе функционирования комплекса.

##### Вариант задания №3.

Обмен сообщениями через общий именованный или неименованный канал между более чем двумя процессами. Структура сообщения может ограничиваться текстовым блоком, размер сообщения можно предполагать фиксированным, обработка сообщений может ограничиваться их генерацией и отображением. Участники обмена не равнозначны: сообщения могут иметь конкретного адресата.

*Примечание.* Необходимо обеспечить целостность сообщений в ходе их транспортировки и приема, а также идентификацию получателя сообщений.

##### Вариант задания №4.

Реализация модели «производитель – потребитель» с использованием разделяемой памяти. Распределение блоков внутри разделяемой памяти может быть выбрано произвольным образом, однако должна обеспечиваться буферизация нескольких «заявок». *Примечание.* Необходимо обеспечивать синхронизацию доступа к блокам и к служебным структурам, описывающим буфер.

##### Вариант задания №5.

Реализация конвейерной обработки (более двух «ступеней» – обрабатывающих процессов) с использованием разделяемой памяти.

##### Вариант задания №6.

Реализация конвейерной обработки с использованием очереди сообщений: обрабатываемые «объекты», помещаемые на конвейер, представляются в виде сообщений. Очередь сообщений общая для всех «ступеней» конвейера.

*Примечание.* Селективность выборки сообщений из очереди может быть обеспечена значениями кодов их «типов».

##### Вариант задания №7.

Реализация модели «читатели – писатели» с использованием разделяемой памяти. Комплекс должен быть устойчив при произвольном характере обращений к блокам.

*Примечание*. Необходимо предусмотреть учет принадлежности блоков памяти («объектов») различным процессам, а также характер занятия этих блоков (т.е. «чтение» или «запись»).

##### Вариант задания №8.

Реализация модели «читатели – писатели» с использованием очереди сообщений. Очередь сообщений общая для всех процессов, участвующих во взаимодействии.

##### Вариант задания №9.

(*Повышенной сложности*) Реализация модели системы массового обслуживания. Механизмы обеспечения взаимодействия могут выбираться самостоятельно в соответствии с уточненной постановкой задачи.

*Примечание*. Предполагается определенная гибкость модели за счет ее парметризации.

##### Вариант задания №10.

(*Повышенной сложности*) Реализация модели «обедающих философов. Механизмы обеспечения взаимодействия могут выбираться самостоятельно в соответствии с уточненной постановкой задачи.

*Примечание*. Предполагается демонстрация с различными сочетаниями параметров, позволяющими наблюдать различные ситуации в поведении модели, включая коллизии и их предотвращение.

### Индивидуальная практическая работа №2

Программирование с использованием сокетов

#### Указания по выбору варианта

Разработка программ, реализующих доступ к сетевым ресурсам и/или службам, серверных и клиентских приложений.

Индивидуальная практическая работа предполагает защиту выполненного задания, а также ответы на теоретические вопросы, демонстрирующие владение тематикой.

Индивидуальная практическая работа оформляется в соответствии с общеустановленными нормами и правилами.

Выбор варианта задания индивидуальной практической работы осуществляется студентом самостоятельно на основании номера зачетной книжки (последние цифры, после номера группы). Если порядковый номер превышает количество вариантов, для выбора номера варианта используется остаток от деления порядкового номера на количество вариантов. В отдельных случаях задание может быть сформулировано преподавателем индивидуально.

#### Теоретическая часть (вопросы)

1. Базовые сведения о поддержке сетевого взаимодействия. Стеки протоколов OSI, TCP/IP.

2. Программный интерфейс сокетов: назначение, основные концепции.

3. Адресация точек доступа к сети. Семейства адресов сокетов.

4. Способы организации передачи данных. Потоковые и датаграмные сокеты.

5. Общая характеристика схемы взаимодействия клиент-сервер. Примеры клиент-серверных систем.

6. Общий алгоритм взаимодействия без установления соединения.

7. Общий алгоритм взаимодействия с установлением соединения.

8. Контроль состояния соединения. Разрыв соединения.

9. Особенности представления данных в сети (разнородных). Форматы данных и их преобразование.

10. Обслуживание множественных запросов (многопользовательские серверы) – основные подходы и средства.

#### Практическая часть (варианты заданий)

##### Вариант задания №1.

##### Вариант задания №2.

1. Под объектом в данном случае имеется в виду любое образование, над которым программа («субъект») выполняет какие-либо действия: элементы данных, структуры, файлы и т.д. К понятию объекта как концепции в объектном и объектно-ориентированном программировании это отношения не имеет. [↑](#footnote-ref-1)