

# ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение высшего профессионального образования

Тихоокеанский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе

\_\_\_\_\_ С.В. Шалобанов

“ \_\_\_\_\_ ” \_\_\_\_\_ 2010 г.

ПРОГРАММА ДИСЦИПЛИНЫ  
по кафедре Вычислительной техники

## ОПЕРАЦИОННЫЕ СИСТЕМЫ

Утверждена научно-методическим советом университета  
для направлений подготовки (специальностей)  
в области **«Информатики и вычислительной техники»**

Направление  
210404.62 «Телекоммуникации»

Специальность 210404.65  
«Многоканальные телекоммуникационные системы»

Хабаровск 2010 г.



## 1. Цели и задачи дисциплины

Основной целью и задачей курса “Операционные системы” является получение студентами систематизированных сведений об архитектуре и принципах работы современных и перспективных операционных систем (ОС) различного назначения, особенностях программного обеспечения различного типа ОС. А также дополнительно, приобретение студентами знаний по современным методам построения, основам функционирования ОС различного назначения, получение первичных навыков системного подхода к их проектированию.

В результате изучения дисциплины студент должен знать основные понятия многозадачных ОС, их общую структуру, основные принципы организации и алгоритмы функционирования ОС и оболочек.

Изучение дисциплины базируется на знаниях, полученных при изучении дисциплин «Информатика» и «Основы алгоритмизации и программирования».

## 2. Требования к уровню освоения содержания дисциплины

В результате изучения дисциплины студент должен:

**- знать**

основные характеристики и принципы построения современных ОС;  
основные принципы функционирования многопроцессных ОС;  
основные принципы организации памяти в ОС;  
основные управления файлами в ОС;

**-уметь**

проводить первичную установку и настройку различных ОС;  
использовать системные утилиты для создания и отладки программного обеспечения;  
использовать системы с открытым исходным кодом;

**-иметь опыт**

работы с ОС с открытым исходным кодом ;

**-иметь представление**

возможностях современных ОС;  
о тенденциях и перспективах развития современных ОС.

### 3. Объём дисциплины и виды учебной работы.

Таблица 1.

Наименование	По учебным планам (УП)	
	С максимальной трудоёмкостью	С минимальной трудоёмкостью
<b>Общая трудоёмкость дисциплины</b>		
по ГОС	100	100
по УП	102	102
<b>Изучается в семестрах</b>	7	7
<b>Вид итогового контроля по семестрам</b>		
зачет		
экзамен	7	7
Курсовой проект (КП)		
Курсовая работа (КР)		
<i>Расчетно-графические работы (РГР)</i>		
<i>Реферат (РФ)</i>		
<i>Домашние задания (ДЗ)</i>		
<b>Аудиторные занятия:</b>		
всего	51	51
В том числе: лекции (Л)	34	34
Лабораторные работы (ЛР)	17	17
Практические занятия (ПЗ)		
<b>Самостоятельная работа</b>		
общий объем часов (С2)	51	51
В том числе на подготовку к лекциям	34	34
на подготовку к лабораторным работам	17	17
на подготовку к практическим занятиям		
на выполнение КР		
на выполнение РГР		
на написание РФ		
на выполнение ДЗ		

#### 4. Содержание дисциплины

Тема	Наименование тем лекционного курса
1. Введение. Современные ОС.	Цели и задачи дисциплины. Рекомендации по изучению. История, классификация ОС и основные определения. Основные характеристики ОС. Примеры. [Л.1-Л.7 Введение]
2. Структура ОС. Характеристики ОС.	Назначение и основные функциональные характеристики ОС. Детальная структура различных ОС. Особенности различных ОС. [Л.3 – с. 81 – 131]
3. Процессы. Состояния процессов. Модели процессов.	Описание процессов. Управление процессами. Модели процессов. [Л.3 – с. 139 – 190; Л.4 – с. 23 – 41; Л1. – 23 – 45;]
4. Потоки. Структура потоков.	Многопоточность. Структура потоков. Управление потоками и SMP. [Л.1 – с. 55 – 93; Л.3 – с. 195 – 242; Л.4 – с.45 – 55;]
5. Взаимные исключения.	Принципы параллельных вычислений. Требования к взаимным исключениям. Взаимоисключения: программный и аппаратный подходы. [Л. 3 – с. 245 – 264; Л. 2 – с. 112-155;]
6. Взаимные блокировки. Семафоры.	Семафоры. Мониторы. Взаимоблокировки. Алгоритмы предотвращения и устранения. [Л. 3 – с. 267 – 337; Л. 2 – с. 210 – 235;]
7. Планирование в системах с одним процессором.	Типы планирования. Алгоритмы планирования. Характеристики алгоритмов. [Л. 3 – с. 457 – 492; Л. 1 – с. 34 – 120;]
8. Планирование в многопроцессорных системах. ОС реального времени.	Многопроцессорное планирование. Вопросы разработки. Планирование реального времени. Планирование с предельными сроками. [Л. 3 – с. 505 – 539; Л. 1 – с. 47 – 127; Л. 5 – с. 149 – 164;]
9. Управление памятью.	Требования к управлению памятью. Логическая и физическая организации памяти. Распределение памяти. Страничная организация. Сегментация. [Л. 3 – с. 355 – 381; Л. 1 – с. 433 – 560; Л. 2 – с. 88 – 159]
10. Виртуальная память.	Управляющие структуры. Страничная организация. Сегментная организация. [Л. 3 – с. 391 – 446; Л. 1 – с. 561 – 615;]
11. Стратегии работы со страницами виртуальной памяти: выборка, размещение, замещение.	Стратегии выборки, замещения, размещения. Примеры систем управления памятью в различных ОС. [Л. 3 – с. 391 – 446; Л. 1 – с. 561 – 615;]
12. Операции ввода-вывода.	Устройства ввода-вывода. Буферизация. Сервисы ОС для операций ввода-вывода. [Л. 3 – с. 543 – 560;]
13. Дисковое планирование.	Дисковое планирование: параметры производительности, стратегии. Дисковый кеш. Примеры организации в различных ОС. [Л. 3 – с. 560 – 588; Л. 2 – с. 230 – 280;]
14. Файловые системы.	Системы управления файлами. Организация файлов. Организация файловых систем. Современные файловые системы. [Л. 3 – с. 603 – 643; Л. 1 – с. 780 – 860;]
15. Безопасность ОС.	Угрозы для безопасности. Защита. Вирусы. Безопасность различных ОС. [Л. 3 – с. 731 – 793; Л. 1 – с. 890 – 960; Л2. – 290 – 310;]

### Разделы дисциплины и виды занятий и работ

№	<i>Раздел дисциплины</i>	<i>Л</i>	<i>ЛР</i>	<i>ПЗ</i>	<b>КП (К Р)</b>	<b>РГ Р</b>	<i>ДЗ</i>	<b>Р Ф</b>	<b>С 2</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
1.	Введение. Современные ОС.	*	*					*	
2.	Структура ОС. Характеристики ОС.	*	*					*	
3.	Процессы. Состояния процессов. Модели процессов.	*	*						
4.	Потоки. Структура потоков.	*	*						
5.	Взаимные исключения.	*	*						*
6.	Взаимные блокировки. Семафоры.	*	*						*
7.	Планирование в системах с одним процессором.	*	*						
8.	Планирование в многопроцессорных системах. ОС реального времени.	*						*	*
9.	Управление памятью.	*	*						
10.	Виртуальная память.	*						*	*
11.	Стратегии работы со страницами виртуальной памяти: выборка, размещение, замещение.	*							*
12.	Операции ввода-вывода.	*	*						*
13.	Дисковое планирование.	*	*						*
14.	Файловые системы.	*	*						*
15.	Безопасность ОС.	*						*	*

## 5. Лабораторный практикум

### 1. Знакомство с ОС Linux.

*Цель работы:* знакомство с ОС Mandrake Linux.

*Исполнение:* Проверяется исполнение сетевых команд по их списку с различными значениями параметров.

*Обеспечение:* Персональный компьютер, подключенный к сети; Операционная система Linux.

*Оценка:* По результатам выполнения определяются основные характеристики ОС, структура ОС, изучаются команды СО Linux, структура каталогов. Обучающийся должен иметь представление об ОС Linux, уметь находить системную информацию в каталогах.

*Время выполнения работы:* 3 часа.

### 2. Разработка приложений на прикладном уровне с использованием процессов.

*Цель работы:* Изучение создания, контроля и завершения процессов в ОС Linux.

*Исполнение:* Знакомство с функциями ОС Linux для работы с процессами. Получение первоначальных навыков управления состояниями процессов. Написание программы для изучения функций создания, удаления процессов.

*Обеспечение:* Персональный компьютер, подключенный к сети; Операционная система Linux. Компилятор gcc.

*Оценка:* В результате выполнения должно быть представлено приложение, выполняющее заданные действия на ПК под управлением ОС Linux. Обучающийся должен знать основные системные вызовы создания и управления процессами и их параметры, а также последовательность системных вызовов при программировании.

*Время выполнения работы:* 4 часов.

### 3. Разработка приложений на прикладном уровне с использованием библиотеки поддержки потоков.

*Цель работы:* Изучение создания, контроля и завершения потоков в ОС Linux

*Исполнение:* Знакомство с библиотечными функциями из Pthreads, относящимися к работе с потоками. Получение первоначальных навыков разработки и анализа многопоточных приложений. Написать программу создания и управления потоками в ОС Linux.

*Обеспечение:* Персональный компьютер, подключенный к сети; Операционная система Linux. Компилятор gcc.

*Оценка:* В результате выполнения должно быть представлено приложение, выполняющее заданные действия на одном ПК под управлением ОС Linux. Обучающийся должен знать основные библиотечные функции из библиотеки Pthreads и их параметры.

*Время выполнения работы:* 4 часов.

### 4. Изучение методов синхронизации потоков.

*Цель работы:* Изучение функций, предназначенных для синхронизации потоков в ОС Linux с использованием библиотеки Pthreads.

*Исполнение:* Изучить основные методы синхронизации. Написать программу создания, управления и синхронизации потоков в ОС Linux.

*Обеспечение:* Персональный компьютер, подключенный к сети; Операционная система Linux. Компилятор gcc.

*Оценка:* В результате выполнения должно быть представлено приложение, выполняющее заданные действия на одном ПК под управлением ОС Linux. Обучающийся должен знать основные библиотечные функции синхронизации из библиотеки Pthreads и их параметры.

*Время выполнения работы:* 2 часов.

## **5. Изучение возможностей межпроцессных коммуникаций.**

*Цель работы:* Изучение различных методов межпроцессных коммуникаций в ОС Linux, знакомство и получение практических навыков использования средств межпроцессных коммуникаций – неименованных каналов и общей памяти.

*Исполнение:* Изучить основные методы межпроцессных коммуникаций в ОС Linux. Написать программу измерения скорости различных межпроцессных коммуникаций в ОС Linux.

*Обеспечение:* Персональный компьютер, подключенный к сети; Операционная система Linux. Компилятор gcc.

*Оценка:* В результате выполнения должно быть представлено приложение, выполняющее заданные действия на одном ПК под управлением ОС Linux, а также представлен графический материал, содержащий информацию о скоростях передачи данных по различным каналам межпроцессных коммуникаций. Обучающийся должен знать основные библиотечные функции создания каналов и их параметры.

*Время выполнения работы:* 4 часов.

### **Лабораторные занятия и их взаимосвязь с содержанием лекционного курса**

№ п/п	№ раздела по варианту содержания	Наименование лабораторной работы
1.	1, 2, 12, 13, 14	Знакомство с ОС Linux.
2.	3, 12, 13, 14	Разработка приложений на прикладном уровне с использованием процессов.
3.	4, 13, 7, 9, 14	Разработка приложений на прикладном уровне с использованием библиотеки поддержки потоков.
4.	5, 6, 7, 9, 14	Изучение методов синхронизации потоков.
5.	5, 6, 9, 12, 13, 14	Изучение возможностей межпроцессных коммуникаций.

## **6. Реферат**

Студентам дневной формы обучения во время самостоятельной работы может быть предложена подготовка реферата по современным актуальным направлениям в области сетей и телекоммуникаций. Подготовка реферата преследует цель ознакомления студентов с последними достижениями в рассматриваемой сфере, т.к. развитие информационных технологий происходит чрезвычайно быстро и последние разработки могут быть не включены в курс обучения.

Объем в страницах – до 20 стр. Время на разработку, включая поиск информации - 10 часов.

### **Примерные темы рефератов**

1. Вопросы распространения ОС с открытым исходным кодом.
2. Структура микроядра в ОС Linux.
3. Сравнение производительности различных ОС.
4. Современные файловые системы.
5. Безопасность ОС.



## 7. Контроль знаний студентов

### 1. Тематика вопросов входного контроля.

Студент должен знать:

- Теоретические основы построения ЭВМ (системы счисления, арифметические и логические операции, представление информации в ЭВМ, кодирование информации);
- Организацию ЭВМ и систем (состав и назначение технических средств ЭВМ, функционирование основных элементов ЭВМ, интерфейсы передачи данных, прием и передачу данных ЭВМ.)
- Алгоритмизацию и программирование (основы алгоритмизации, язык программирования С).

### 2. Текущий контроль знаний студентов.

Текущий контроль осуществляется на лабораторных занятиях путем решения задач, ответов на контрольные вопросы, защите лабораторных работ. Тематика лабораторных работ приведена выше.

### 3. Выходной контроль знаний студентов.

Дисциплина завершается зачетом и экзаменом. На экзамене проверяется степень усвоения студентами основных понятий дисциплины, понимание их взаимосвязи, знание основ построения современных ОС, умение использовать стандартные функции ОС.

### Примерный состав вопросов в билетах экзамена по дисциплине

1. Операционная система. Предназначение, эволюция операционных систем.
2. Классификация, функции и характеристики операционных систем.
3. Структура операционной системы. Архитектуры построения операционных систем.
4. Операционные системы реального времени и их характеристики. Планирование в системах реального времени.
5. Процессы. Описание процессов. Структуры управления процессами.
6. Состояния процессов. Модели процессов. Приостановленные процессы.
7. Создание и завершение процессов. Переключение процессов.
8. Потoki. Различия потоков и процессов. Пример использования потоков. Потoki на пользовательском уровне и на уровне ядра.
9. Параллельные вычисления. Принципы параллельных вычислений. Взаимодействие процессов.
10. Взаимные исключения. Требования к взаимным исключениям. Программный и аппаратный подход.
11. Взаимные исключения. Семафоры и мониторы. Определение функций `wait(semaphore s)`, `signal(semaphore s)`. Реализация семафоров.
12. Взаимоблокировки и голодание. Примеры. Условия возникновения взаимоблокировок.
13. Обнаружение и устранение взаимоблокировок.
14. Планирование в системах с одним процессором. Типы планирования.
15. Критерии и стратегии планирования.
16. Многопроцессорное планирование. Методы планирования процессов и потоков в многопроцессорной системе.
17. Управление памятью. Организация памяти. Распределение памяти. Фрагментация.
18. Виртуальная память. Страничная и сегментная организации.
19. Управление виртуальной памятью. Стратегии выборки, замещения, размещения страниц. Алгоритмы замещения страниц.

20. Виртуальная память. Управление резидентным множеством страниц.
21. Управление вводом-выводом. Организация функций ввода-вывода. Сервисы ядра операционной системы для операций ввода-вывода.
22. Управление вводом-выводом. Дисковое планирование.
23. Управление файлами. Файловые системы.

## **8. Контроль самостоятельной работы студентов-заочников**

Каждый студент должен выполнить одну контрольную и одну лабораторную работу.

Выполнение контрольной работы является важным звеном в обучении студентов-заочников и преследует следующие цели:

- оказать помощь студенту в овладении необходимыми навыками правильной организации самостоятельной работы в межсессионный период;
- привить навыки самостоятельного изучения материала по учебной дисциплине;
- указать правильную последовательность в изучении данной учебной дисциплины;
- закрепить знания основных положений учебной дисциплины;
- систематизировать знания по учебной дисциплине;
- выработать умение анализировать достоинства и недостатки отдельных технических решений;
- привить навыки применения теоретических знаний для решения практических вопросов;
- научить студента грамотно, лаконично излагать материал;
- проверить работу студента-заочника в межсессионный период по изучению данной дисциплины.

В контрольной работе студент должен продемонстрировать понимание предложенных в контрольной работе вопросов, показать знание теории ОС, знания программирования.

### **Примерный состав вопросов для контрольной работы**

1. Классификация и основные характеристики ОС
2. Классификация, функции и характеристики современных ОС.
3. Структура операционной системы. Архитектуры построения операционных систем.
4. Архитектура микроядра.
5. Операционные системы реального времени и их характеристики.
6. Применение ОС реального времени.
7. Процессы. Описание процессов.
8. Структуры управления процессами.
9. Модели процесса с 2я, 5ю состояниями.
10. Свопинг. Модели процесса с 7ю состояниями.
11. Приостановленные процессы.
12. Создание и завершение процессов.
13. Переключение процессов.
14. Состав блока управления процессом.
15. Потoki. Различия потоков и процессов.
16. Пример использования потоков.
17. Потoki на пользовательском уровне и на уровне ядра.
18. Параллельные вычисления. Принципы параллельных вычислений.
19. Взаимодействие процессов.
20. Взаимные исключения.
21. Требования к взаимным исключениям.

22. Взаимоисключения: программный подход.
23. Взаимоисключения: аппаратный подход.
24. Семафоры и мониторы.
25. Определение функций wait(semaphore s), signal(semaphore s).
26. Реализация семафоров.
27. Взаимоблокировки и голодание.
28. Условия возникновения взаимоблокировок.
29. Обнаружение взаимоблокировок.
30. Устранение взаимоблокировок.
31. Планирование в системах с одним процессором.
32. Типы планирования.
33. Критерии планирования.
34. Стратегии планирования
35. Многопроцессорное планирование.
36. Методы планирования процессов и потоков в многопроцессорной системе.
37. Управление памятью.
38. Организация памяти.
39. Распределение памяти.
40. Фрагментация.
41. Виртуальная память.
42. Страничная организация.
43. Сегментная организация.
44. Управление виртуальной памятью.
45. Стратегии выборки страниц.
46. Стратегии замещения страниц.
47. Стратегии размещения страниц.
48. Алгоритмы замещения страниц.
49. Управление резидентным множеством страниц.
50. Управление вводом-выводом.
51. Организация функций ввода-вывода.
52. Сервисы ядра операционной системы для операций ввода-вывода.
53. Дисковое планирование.
54. Управление файлами.
55. Обзор файловых систем.
56. Файловая система FAT.
57. Файловая система NTFS.

Студент должен выбрать 9 вопросов для выполнения контрольной работы в зависимости от последней цифры студенческого билета или зачетной книжки.

Приступая к выполнению контрольной работы, студент должен выписать из общего списка вопросы, которые включены в его вариант контрольной работы, уяснить какого ответа требуют предлагаемые вопросы. Затем в первом приближении изучить учебную дисциплину по рекомендованной литературе, руководствуясь учебной программой или списком вопросов для подготовки к экзамену с тем, чтобы иметь общее представление по всем вопросам учебной дисциплины и чувствовать взаимосвязь предложенных в контрольной работе вопросов с другими вопросами дисциплины. После этого можно приступить к более глубокому изучению материала по тем вопросам, которые заданы в контрольной работе и подготовке ответа на них.

Отрабатывать вопросы контрольной работы следует по нескольким рекомендованным пособиям, делая в тетради отдельные выписки и приводя необходимые рисунки (схемы). При отработке вопросов контрольной работы можно привлекать и другие источники, не приведён-

ные в списке рекомендованной литературы. После сбора необходимого материала для ответа на вопросы контрольной работы, разработки необходимых схем, следует написать черновой вариант контрольной работы, используя сделанные ранее выписки. После этого следует отредактировать текст контрольной работы и оформить работу начисто. Писать текст контрольной работы следует собственным языком. Не допускается компиляция и плагиат текста из используемой литературы.

Лабораторная работа выбирается из предложенного списка в разделе «Лабораторные работы». В зависимости от специальности выбирается или «Разработка сетевых приложений клиент-сервер на базе интерфейса сокетов», или «Проектирование корпоративной сети на базе моделирующей программы».

## **9. Учебно-методическое обеспечение дисциплины**

### **Основная литература**

1. Таненбаум Э. Современные операционные системы (Modern Operating Systems), 2002 г.
2. Гордеев А. В. Операционные системы, 2004 г.
3. Столлингс В. Операционные системы - М: Вильямс, 2004 г.
4. Стивенс У. UNIX: взаимодействие процессов, СПб: Питер, 2003 г.
5. Дансмур М., Дейвис Г. Операционная система UNIX и программирование на языке Си, 1989 г.
6. Робачевский А. Операционная система UNIX, 2000 г.
7. Келли-Бутл С. Введение в UNIX, 1995 г.

### **Дополнительная литература**

1. Эндриус Г.Р. Основы многопоточного, параллельного и распределенного программирования - М: Вильямс, 2003 г.
2. Керниган Б.В., Пайк Р. UNIX - универсальная среда программирования, 1992 г.
3. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы, СПб: Питер, 2006 г., 672 с.
4. Столингс В. Компьютерные системы передачи данных. - М: Вильямс, 2002 г., 920 с.
5. А.В. Горячев и др. Основы сетевых технологий. Учебное пособие. СПб. 2000.

### **Методические указания**

1. Сорокин Н.Ю. Процессы и потоки в операционной системе Linux. Методические указания к выполнению лабораторных работ. Хабаровск: Издательство ТОГУ, 2006 г., 19 с.
2. Сорокин Н.Ю. Организация межпроцессного взаимодействия в операционной системе Linux. Методические указания к выполнению лабораторных работ. Хабаровск: Издательство ТОГУ, 2006 г., 16с.
3. Сорокин Н.Ю. Создание и выполнение потоков. Библиотека Pthreads. Рукописный. Доступно на <http://evm.khstu.ru>
4. Сорокин Н.Ю. Синхронизация потоков. Библиотека Pthreads. Рукописный. Доступно на <http://evm.khstu.ru>
5. Сорокин Н.Ю. Изучение возможностей межпроцессных коммуникаций в ОС Linux. Рукописный. Доступно на <http://evm.khstu.ru>
6. Сорокин Н.Ю. Расширенные функции синхронизации потоков. Рукописный. Доступно на <http://evm.khstu.ru>

## **10. Материально-техническое обеспечение дисциплины.**

Персональный компьютер, подключенный к сети. Операционная система Linux. Текстовый редактор vi. Компилятор gcc.

## **11. Методические рекомендации по организации изучения дисциплины**

Курс рассматривает основы построения ОС, основные понятия и технологии, используемые на современном этапе науки и техники.

Рассмотрение ведется на базе теории ОС, детально отраженной в основных литературных источниках 1-3. Все разделы лекционного курса представляются студентам без привязки к какой-либо ОС, однако к концу каждого раздела приводится достаточное количество примеров конкретных реализаций функций/технологий/методов/стратегий/алгоритмов в современных ОС, что позволяет студентам усвоить теоретические разделы и наглядно продемонстрировать их применение.

На лабораторных работах значительное внимание уделяется использованию функций стандартных библиотек ОС с открытым исходным кодом (ОС Linux).

### **Организация самостоятельной работы**

Самостоятельная работа предполагает, что:

- 1) отдельные темы могут быть отнесены на самостоятельное изучение;
- 2) теоретическая подготовка к лабораторным работам с использованием МУ может осуществляться дома самостоятельно.

## Словарь терминов и персоналий

**Административная система** (management system) – система, обеспечивающая управление сетью либо ее частью.

**Адрес** (address) – закодированное обозначение пункта отправления либо назначения данных.

**Адрес IP** – адрес, однозначно определяющий компьютер в сети (адрес состоит из 32 двоичных разрядов и не может повторяться во всей сети TCP/IP). Адрес IP обычно разбивается на четыре октета по восемь двоичных разрядов (один байт); каждый октет преобразуется в десятичное число и отделяется точкой, например 102.54.94.97.

**Анонимные подключения** – эта функция, которая разрешает удаленный доступ к ресурсам компьютера по учетной записи компьютера без предъявления имени и пароля с правами, определяемыми этой учетной записью.

**Архитектура** – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов сети. Архитектура охватывает логическую, физическую и программную структуры и функционирование сети, а также элементы, характер и топологию взаимодействия элементов.

**База данных** (БД) – совокупность взаимосвязанных данных, организованная по определенным правилам в виде одного или группы файлов.

**Базовый порт ввода/вывода** (base I/O port) – адрес памяти, по которому центральный процессор и адаптер проверяют наличие сообщений, которые они могут оставлять друг для друга.

**Безопасность данных** (data security) – концепция защиты программ и данных от случайного либо умышленного изменения, уничтожения, разглашения, а также несанкционированного использования.

**Блок данных** (data unit) – последовательность символов фиксированной длины, используемая для представления данных или самостоятельно передаваемая в сети.

**Булева алгебра** – алгебраическая структура с тремя операциями И, ИЛИ, НЕ.

**Буфер** (buffer) – временная область, которую устройство использует для хранения входящих данных перед тем, как они смогут быть обработаны на входе, или для хранения исходящих данных до тех пор, пока не появится возможность их передачи.

**Буфер** (buffer) – запоминающее устройство, используемое между объектами при передаче данных для временного хранения данных с целью согласования скоростей.

**Гигабайт** (gigabyte) – обычно 1000 мегабайтов. Точно 1024 мегабайт, где 1 мегабайт равен 1 048 576 байтам ( $2^{20}$ ).

**Гипертекст** – текст, представленный в виде ассоциативно связанных друг с другом блоков.

**Гипертекстовый протокол HTTP** – протокол сети Internet, описывающий процедуры обмена блоками гипертекста.

**Главный контроллер домена** (Primary Domain Controller, PDC) – компьютер, на котором устанавливается Windows NT Server в режиме PDC для хранения главной копии базы данных учетных записей.

**Глобальная вычислительная сеть, ГВС** (Wide Area Network, WAN) – компьютерная сеть, использующая средства связи дальнего действия.

**Группа** (group) – совокупность пользователей, определяемая общим именем и правами доступа ресурсам.

**Данные** (data) – информация, представленная в формализованном виде, пригодном для автоматической обработки при возможном участии человека.

**Дейтаграммы** (datagrams) – сообщения, которые не требуют подтверждения о приеме от принимающей стороны. Термин, используемый в некоторых протоколах для обозначения пакета.

**Диагностическое программное обеспечение** (diagnostic software) – специализированные программы или специфические системные компоненты, которые позволяют исследовать и наблю-

дать систему с целью определения, работает она правильно или нет, и попробовать определить причину проблемы.

**Дистрибутив** – это форма распространения программного обеспечения. Дистрибутив обычно содержит программы для начальной инициализации системы (в случае дистрибутива операционной системы — инициализация аппаратной части, загрузка урезанной версии системы и запуск программы-установщика), программу-установщик (для выбора режимов и параметров установки) и набор специальных файлов, содержащих отдельные части системы (так называемые пакеты).

**Домен** (domain) – совокупность компьютеров, использующих операционную систему Windows NT Server, имеющих общую базу данных и систему защиты. Каждый домен имеет неповторяющееся имя.

**Доменная система имен** (DNS –Domain Name System) – система обозначений для сопоставления адресов IP и имен, понятных пользователю, используется в сети Internet. Система DNS иногда называется службой DNS.

**Доступ** (access) – операция, обеспечивающая запись, модификацию, чтение или передачу данных.

**Драйвер** (driver) – компонент операционной системы, взаимодействующий с внешним устройством или управляющий выполнением программ.

**Драйвер устройства** (device driver) – программа, которая обеспечивает взаимодействие между операционной системой и конкретными устройствами с целью ввода/вывода данных для этого устройства.

**Единообразный локатор ресурсов** (Uniform Resource Locator, URL) – идентификатор, или адрес ресурсов, в сети Internet. Обеспечивает гипертекстовые связи между документами WWW.

**Жесткий диск** (hard disk) – накопитель данных в вычислительных системах.

**Запрос прерывания** (IRQ – interrupt request) – сигнал, посылаемый центральному процессору от периферийного устройства. Сообщает о событии, обработка которого требует участие процессора.

**Зеркальные диски** (disk mirroring) – уровень 1 технологии RAID, при которой часть жесткого диска (или весь жесткий диск) дублируется на одном или нескольких жестких дисках. Позволяет создавать резервную копию данных.

**Изображение** (image) – графическая форма представления данных, предназначенная для зрительного восприятия.

**Интернет** – совокупность компьютеров, объединенных в глобальную сеть.

**Информационная сеть** (information network) – сеть, предназначенная для обработки, хранения и передачи данных.

**Информационная система** (information system) – объект, способный осуществлять хранение, обработку или передачу данных. К информационной системе относятся: компьютеры, программы, пользователи и другие составляющие, предназначенные для процесса обработки и передачи данных.

**Информационно-поисковая система** – (IRS – Information Retrieval System) – система, предназначенная для поиска информации в базе данных.

**Информация** (information) – совокупность фактов, явлений, событий, представляющих интерес, подлежащих регистрации и обработке.

**Информация** (information) – данные, обработанные адекватными им методами.

**Исходный код** – это текст программы для компьютера, написанный на языке, понятном человеку – языке программирования. Исходный код либо транслируется в исполняемый код при помощи компилятора, предназначенного для определенной компьютерной архитектуры, либо исполняется непосредственно по тексту при помощи интерпретатора.

**Канал** (link) – среда или путь передачи данных.

**Клиент (client)** – компьютер в сети, который запрашивает ресурсы или услуги от некоторых других компьютеров.

**Клиент (client)** – объект информационной сети, использующий сервис, предоставляемый другими объектами.

**Клиент-сервер (client-server)** – модель вычислений, при которой некоторые компьютеры запрашивают услуги (клиенты), а другие отвечают на такие запросы на услуги (сервер).

**Коммерческое программное обеспечение** – программное обеспечение, произведенное коммерческими компаниями и лицензированное для пользователей согласно сильно ограниченным лицензиям. Любое лицензированное программное обеспечение, кроме имеющего лицензии, одобренные Фондом свободного программного обеспечения и/или Комитетом OSI, рассматривается в рамках данного обзора как коммерческое. В большинстве случаев коммерческое программное обеспечение обычно не может распространяться пользователем; при этом ему не предоставлен доступ к программному тексту, хотя бывают исключения. Свободное программное обеспечение не рассматривается в качестве коммерческого.

**Коммуникационная сеть** – сеть, предназначенная для передачи данных, также она выполняет задачи, связанные с преобразованием данных.

**Корпоративная сеть (enterprise network)** – крупномасштабная сеть, обычно соединяющая многие локальные сети.

**Лазерный принтер (laser printer)** – принтер, в котором изображение символов печатаются лазерным лучом и переносятся на бумагу методом ксерографии.

**Логический диск (logical disk)** – часть физического диска, отформатированная под конкретную файловую систему и имеющая свое буквенное наименование.

**Логический канал (logical channel)** – путь, по которому данные передаются от одного порта к другому. Логический канал прокладывается в одном либо последовательности физических каналов и через уровни области взаимодействия.

**Локализация (англ. localization)** – перевод и адаптация элементов интерфейса, вспомогательных файлов и документации. Задача локализации не исчерпывается только переводом, более того, перевод как таковой обычно занимает скромное место в процессе локализации программного обеспечения. Типичными задачами адаптации являются использование национальных символов валюты, применение принятых форматов представления даты и времени, а также правил алфавитной сортировки текстов.

**Локальная группа (local group)** – В Windows NT Server – учетная запись, определенная на конкретном компьютере. Включает учетные записи пользователей данного компьютера.

**Локальная сеть (Local-Area Network)** – сеть, системы которой расположены на небольшом расстоянии друг от друга.

**Масштабируемость** – это возможность увеличить вычислительную мощность Web-сайта или компьютерной системы (в частности выполнение большего числа операций или транзакций за определенный период времени) за счет установки большего числа процессоров или их замены на более мощные.

**Мегабайт (megabyte)** – 1 048 576 байтов ( $2^{20}$ ).

**Микроядро (microkernel)** – центральная часть операционной системы, выполняющая основные функции управления системой.

**Модем (modem)** – сокращение от МОДулятор-ДЕМОдулятор. Устройство связи, позволяющее компьютеру передавать данные по обычной телефонной линии. При передаче преобразует цифровые сигналы в аналоговые. При приеме преобразует аналоговые сигналы в цифровые.

**Оболочка (shell)** – программное обеспечение, которое реализует взаимодействие пользователя с операционной системой (пользовательский интерфейс).

**Общий ресурс (shared resource)** – любое устройство, данные или программа.

**Октет** - байт.



**Оперативная память** (main memory) – память, предназначенная для хранения данных и команд, необходимых процессору для выполнения им операций.

**Операционная система** – основной вид системного программного обеспечения, комплекс программ, обеспечивающий управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит. Общими словами, операционная система — это первый и основной набор программ, загружающийся в компьютер. Помимо вышеуказанных функций ОС может осуществлять и другие, например предоставление пользовательского интерфейса, сетевое взаимодействие и т. п. С 1990-х наиболее распространенными операционными системами являются ОС семейства Microsoft Windows и UNIX-подобные системы.

**Операционная система реального времени** - операционная система с гарантированным временем реакции на событие. Такие ОС используются в системах технологического управления атомными станциями, химическими производствами и пр.

**Открытое программное обеспечение** – (англ. open source software) – это программное обеспечение с (открытым) исходным кодом, доступ к которому не закрыт. Это способ разработки программного обеспечения, при котором создаваемый исходный код программ открыт, то есть, общедоступен для просмотра и изменения. Это позволяет всем желающим использовать уже созданный код для своих нужд и, возможно, помочь в разработке открытой программы.

**Пакет** – это единица информации, передаваемый между станциями сети. Используется на сетевом уровне модели OSI.

**Пароль** (password) – признак, подтверждающий право пользователя или прикладной программы на использование какого-нибудь ресурса.

**Передача данных** (data communications) – процесс транспортирования данных из одной системы в другую.

**Пользователь** (user) – юридическое либо физическое лицо, использующее какие-либо ресурсы, возможности.

**Порт** (port) – точка доступа к устройству либо программе. Различают физические и логические порты.

**Программное обеспечение** – наряду с аппаратными средствами, важнейшая составляющая информационных технологий, включающая компьютерные программы и данные, предназначенные для решения определённого круга задач и хранящиеся на машинных носителях. Программное обеспечение представляет собой либо данные для использования в других программах, либо алгоритм, реализованный в виде последовательности инструкций для процессора. По назначению программное обеспечение делится на системное, прикладное и инструментальное.

**Проприетарное программное обеспечение** – иначе несвободное (иногда употребляется не совсем точный термин «коммерческое») программное обеспечение. Распространяется с условиями, запрещающими его свободное дальнейшее распространение, использование получателем в собственном программном обеспечении, изучение, декомпиляцию, внесение изменений, либо требующими для таких действий специального отдельного соглашения с поставщиком или производителем программного обеспечения. В настоящее время большинство широко распространённого программного обеспечения для массового применения является проприетарным. Из широко известного программного обеспечения к нему относится практически всё программное обеспечение фирмы Microsoft, в том числе операционная система Windows и офисный комплект Microsoft Office.

**Протокол** – набор правил, регламентирующих порядок сборки пакетов, содержащих данные и управляющую информацию, на рабочей станции-отправителе для передачи их по сети, а также порядок разборки пакетов по достижении ими рабочей станции-получателя.

**Реестр** (registry) – архив БД Windows NT для хранения информации о конфигурации компьютера, включая аппаратные средства, установленное программное обеспечение, установки окружения и др.

**Свободное программное обеспечение** – (англ. free software может пониматься как «бесплатное программное обеспечение» и «свободное программное обеспечение») – это программное обеспечение, которое доступно всем, без ограничений. Пользователь может свободно запускать, модифицировать и распространять такое программное обеспечение.

**Сегмент (segment)** – часть сети, ограниченная ретранслирующими устройствами (повторителями, мостами, маршрутизаторами и шлюзами).

**Сервер** – это компьютер сети, предоставляющий сервис другим объектам по их запросам.

**Сервис** – процесс обслуживания объектов.

**Сетевая операционная система** - операционная система, обеспечивающая обработку, хранение и передачу данных в информационной сети. Сетевая операционная система определяет взаимосвязанную группу протоколов верхних уровней, обеспечивающих основные функции сети: адресацию объектов, функционирование служб, обеспечение безопасности данных, управление сетью.

**Сеть (network)** – взаимодействующая совокупность сетевых узлов, связанных друг с другом каналами связи, предназначенная для передачи информации.

**Телекоммуникация (telecommunication)** – область деятельности, предметом которой являются методы и средства передачи информации.

**Терминал (terminal)** – устройство ввода/вывода данных и команд в систему или сеть.

**Тестирование (testing)** – процесс проверки правильности функционирования устройства либо программного обеспечения.

**Технология RAID** – используется для построения отказоустойчивости систем. Имеет пять уровней. 1 уровень – зеркализация дисков, 2 уровень – чередование дисков с записью кода коррекции ошибок, 3 уровень – код коррекции ошибок в виде четности, 4 уровень – чередование дисков блоками, 5 уровень – чередование с контролем четности.

**Транзакция** – короткий во времени цикл взаимодействия объектов, включающий *запрос - выполнение задания - ответ*.

**Трансивер** – устройство, предназначенное осуществлять передачу данных с сетевых интерфейсных плат в физическую среду.

**Трафик** – поток данных.

**Удаленный доступ (remote access)** – технология взаимодействия абонентских систем с локальными сетями через территориальные коммуникационные сети.

**Утилиты (utility)** – программа, выполняющая какую-либо функцию сервиса.

**Учетная запись (account)** – информация, хранящаяся в базе данных Windows NT (учетная запись пользователя, компьютера, группы).

**Фрагментация (fragmentation)** – процесс разделения длинного пакета данных с более высокого уровня на последовательность более коротких пакетов на нижнем уровне.

**Центральный процессор (central processing unit)** – управляющий и вычислительный модуль компьютера. Устройство, которое интерпретирует и выполняет команды.

**Циклический избыточный код (CRC – Cyclical Redundancy Check)** – число, получаемое в результате математических преобразований над пакетом данных и исходными данными. При доставке пакета вычисления повторяются. Если результат совпадает, то пакет принят без ошибок.

**Четность (parity)** – способ контроля за безошибочной передачей блоков данных с помощью добавления контрольных битов.

**Шина (bus)** – специализированный набор параллельных линий в персональном компьютере.

**Шина (bus)** – канал передачи данных, отдельные части которого называются сегментами.

**Шифрование (encryption)** – преобразование информации для ее защиты от несанкционированного доступа.

**Электронная цифровая подпись (ЭЦП)** – аналог собственноручной подписи физического лица, представленный как последовательность символов, полученная в результате криптографического преобразования электронных данных с использованием закрытого ключа ЭЦП, позво-

ляющая пользователю открытого ключа установить целостность и неизменность этой информации, а также владельца закрытого ключа ЭЦП.

**Электронная почта** (email) – компьютерная система обмена сообщениями, где текст и файлы могут быть посланы от одного пользователя к одному или многим другим пользователям в той же сети.

**Эталонная модель взаимодействия открытых систем** (OSI – Open System Interconnection) – семиуровневая модель, которая стандартизирует уровни услуг и виды взаимодействия между системами в информационной сети при передаче данных.

**Ядро операционной системы** - часть операционной системы: постоянно находящаяся в оперативной памяти; управляющая всей операционной системой; содержащая: драйверы устройств, подпрограммы управления памятью, планировщик заданий; реализующая системные вызовы и т.п.

**Язык программирования** – промежуточный язык для описания алгоритмов в виде, пригодном для перевода в машинный язык, понятный для компьютера.

**Язык структурированных запросов** (SQL – Structured Query Language) – язык управления базами данных, используемый для запроса, обновления и управления реляционными базами данных.

**GNU** – проект по созданию свободной операционной системы. Название рекурсивно расшифровывается как «GNU's Not UNIX!» («GNU – это не Юникс!»). Датой начала проекта считается 1984 г., когда Ричард Столлман уволился из Массачусетского Технологического Института с целью посвятить своё время написанию свободной операционной системы. Первой программой проекта стал текстовый редактор Emacs.

**GNU General Public License** (Стандартная общественная лицензия GNU) – «копилефт»-лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г. Её также сокращённо называют GNU GPL или даже просто GPL, если из контекста понятно, что речь идёт именно о данной лицензии (существует довольно много других лицензий, содержащих слова «general public license» в названии). Цель GNU GPL – предоставить пользователю права копировать, модифицировать и распространять программы (что по умолчанию запрещено законом об авторских правах), а также гарантировать, что и пользователи всех производных программ получают вышеперечисленные права. Принцип «наследования» прав называется «копилефт» (калька с английского «copyleft») и был придуман Ричардом Столлманом. По контрасту с GPL, лицензии собственного программного обеспечения очень редко дают пользователю такие права и обычно, наоборот, стремятся их ограничить, например, запрещая восстановление исходного кода.