

ГОУВПО  
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ**

к курсовому проекту по дисциплине

«Операционные системы»

(Для студентов направления подготовки

09.03.04 "ПРОГРАММНАЯ ИНЖЕНЕРИЯ")

Донецк-ДонНТУ-2016

ГОУВПО  
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ**

к курсовому проекту по дисциплине

«Операционные системы»

(Для студентов направления подготовки  
09.03.04 "ПРОГРАММНАЯ ИНЖЕНЕРИЯ")

Рассмотрено на заседании кафедры

Программной инженерии

Протокол № 1 от 30.08.2016

Утверждено на заседании

учебно-издательского Совета ДонНТУ

протокол № от

Донецк-ДонНТУ-2016

УДК

Методические указания и задания к курсовому проекту по дисциплине «Операционные системы» для студентов направления подготовки 09.03.04 "Программная инженерия" Сост.: Чернышова А.В. - Донецк, ДонНТУ, 2016 - 38 с.

Приведены методические указания и задания к выполнению курсового проекта по курсу «Операционные системы». Излагаются цели и задачи курсового проекта, порядок выполнения и содержание курсового проекта, дается определение основным терминам, связанными операционными системами, рассмотрены теоретические вопросы, связанные с проектированием операционных систем: структура файловой системы, организация процессов в операционных системах, определение общей структуры проектируемой ОС, формирование базы данных системы.

Методические указания предназначены для усвоения теоретических основ и формирования практических навыков по проектированию операционных систем и эмуляции работы отдельных модулей спроектированной операционной системы.

Составители:

ст. преп. А.В.Чернышова

# 1 ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА

*Операционная система (ОС)* - это совокупность средств, предназначенных для управления процессами, ресурсами и взаимодействия с пользователями.

Выделяют четыре основных типа ресурсов: процессоры, оперативная память, внешние устройства и информация. Основными операциями над ресурсами являются: отслеживание состояния, планирование, выделение и освобождение. Процесс - это работа, производимая при выполнении программы с её данными на ресурсной среде.

Операционные системы современных компьютеров относятся к числу наиболее сложных систем, созданных когда-либо человеком.

*Цель курсового проектирования:* выполнить обоснованный проект операционной системы, удовлетворяющей ряду специальных требований, приведенных в варианте задания, выполнить эмуляцию работы отдельных модулей спроектированной операционной системы.

*Задача проектирования:* приобретение профессиональных навыков практического применения знаний, полученных в процессе обучения.

Результаты проектирования представляются пояснительной запиской и документами, обеспечивающими непосредственную реализацию средств операционной системы в виде набора программ.

Полученные знания и навыки являются основой для принятия профессиональных решений при проектировании средств управления в различной операционной обстановке.

## 2 ПОРЯДОК ВЫПОЛНЕНИЯ И СОДЕРЖАНИЕ РАБОТЫ

Проектирование предполагает:

1. Изучение проектного задания.
2. Анализ требований к системе и выбор способов их реализации.
3. Изучение способов организации файловых систем, выбор и детальное изучение внутренней структуры файловой системы.
4. Изучение процессов в операционной системе. Понятие процесса, приоритета процесса, планирование и диспетчеризация процессов в операционных системах, принципы и алгоритмы межпроцессного взаимодействия.
5. Разработку структуры гипотетической операционной системы и алгоритмов ее функционирования.
6. Разработку программ эмуляции работы отдельных модулей проектируемой операционной системы: выбор и обоснование выбора программных средств для разработки программ эмуляции, эмуляция работы файловой системы, реализация командного языка для работы с элементами файловой системы и процессами, эмуляция работы планировщика процессов, реализация межпроцессного взаимодействия процессов. Подробное описание разработанных программ: структуры данных, классы, методы, алгоритмы.
7. Тестирование программы и анализ результатов проектирования и работы программ эмуляции.
8. Оформление пояснительной записки к курсовому проекту.
9. Защиту курсового проекта.

Проект представляется пояснительной запиской (30-40с.) и демонстрацией программ, которые отображают реализацию или эмуляцию некоторых основных функций проектируемой операционной системы.

Пояснительная записка должна наглядно отображать результаты проектирования и разработки программ эмуляции.

*Содержание пояснительной записки курсовой работы:*

Титульный лист

Техническое задание

Реферат

Введение

1 Структура проектируемой файловой системы

1.1 Общая организация файловой системы

1.2 Виртуальные страницы

1.3 Команды для работы с ФС

1.4 Системные вызовы для работы с ФС

1.5 Способы организации файлов

1.6 Алгоритмы работы некоторых системных вызовов ФС.

2 Процессы в ОС

2.1 Команды для работы с процессами

2.2 Системные вызовы управления процессами

2.3 Диаграмма состояний процесса

2.4 Приоритеты процессов

2.5 Межпроцессное взаимодействие.

2.6 Выбор дисциплины обслуживания планировщика процессов.

Алгоритм работы планировщика процессов в соответствии с выбранной дисциплиной обслуживания.

2.7 Свопинг процессов.

3 Режимы работы проектируемой ОС

3.1 Мультипрограммный режим работы ОС

3.2 Многопользовательская защита

3.3 Интерактивный режим работы ОС

3.4 Пакетный режим работы

## 4 Структура операционной системы

### 4.1 Общая структура проектируемой ОС

### 4.2 Структура ядра проектируемой ОС. Основные функции и назначение файловой подсистемы, подсистемы управления памятью и процессами, подсистемы управления устройствами.

### 4.3 Структура управляющих блоков базы данных ОС

## 5 Разработка программ эмуляции ОС

### 5.1 Описание программных средств

### 5.2 Разработка ФС

### 5.3 Разработка командного интерпретатора

### 5.4 Эмуляция планирования (или межпроцессного взаимодействия)

## 6 Тестирование программы. Анализ результатов

### Выводы

### Перечень ссылок

### Приложение А. Таблица трассировки

### Приложение Б. Экранные формы, отображающие результаты работы программ эмуляции

### Приложение В. Листинг программ

### 3 ЗАДАНИЕ К КУРСОВОМУ ПРОЕКТУ

Разработать технический проект операционной системы и оформить пояснительную записку.

Операционная система должна удовлетворять ряду специальных требований, указанных в варианте задания. В состав операционной системы не следует включать возможности, перечисленные в классификаторе вариантов, но отсутствующие в Вашем варианте.

Варианты заданий приведены в таблице 1. Описание элементов классификатора приводится ниже.

Столбец 29 «Вход и выход в систему» означает наличие в операционной системе команд, позволяющих осуществлять вход в систему и выход из нее.

Команда интерпретатора указанная в столбце 32 «Создание и удаление» предусматривает наличие команд, позволяющих создавать и удалять файлы, папки и пользователей (в зависимости от режима работы ОС и организации файловой системы)

Эмуляция межпроцессного взаимодействия указанная в столбце 37 должна осуществляется средствами отмеченными в столбцах 11 – 16.



Таблица 1 - Варианты задания к курсовому проекту по дисциплине «Операционные системы»

Вариант	Режим работы				Управление процессами						Средства взаимодействия процессов						Организация ОП			Внешние устройства		Организация файловой системы						Команды интерпретатора										
	Мультипрограммный	Многопользовательская защита	Интерактивный	Пакетный	Квантование времени	Абсолютные приоритеты	Относительные приоритеты	Статические приоритеты	Динамические приоритеты	Свопинг процессов	Каналы, именованные каналы	Семафоры	Сообщения	Сообщения и обмены	Разделяемая память	Отображаемая память	Динамические разделы	Страничная	Сегментно-страничная	Видеотерминал	НМД	Однуровневая	Многоуровневая	Битовая карта свободных/занятых кластеров	Списки свободных/занятых кластеров	Файлы с последовательным доступом	Файлы с прямым доступом	Копирование	Вход и выход в систему	Перемещение	Переименование	Создание и удаление*	Информация о процессах	Изменение приоритета процесса	Уничтожение процесса	Эмуляция планировщика	Эмуляция межпроцессного взаимодействия средствами 11-16 **	
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
12	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
17	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
18	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
19	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
21	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
22	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
23	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
24	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
25	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
26	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
27	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
28	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
29	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
31	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
32	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
33	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
34	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
35	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
36	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
37	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
38	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
39	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Примечание:

\*) Столбец 32 «Создание и удаление» предусматривает реализацию команд создания и удаления файлов, папок и пользователей.

\*\*\*) Столбец 37 «Эмуляция межпроцессного взаимодействия» предусматривает использование средств взаимодействия, указанных в столбцах 11 – 16.

Продолжение таблицы 1.

Вариант	Режим работы				Управление процессами						Средства взаимодействия процессов						Организация ОП			Внешние устройства		Организация файловой системы							Команды интерпретатора									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32*	33	34	35	36	37**	
41	X	X	X	X	X	X	X	X						X		X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
42	X	X	X	X	X	X	X	X						X		X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
43	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
44	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
45	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
46	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
47	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
48	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
49	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
50	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
51	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
52	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
53	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
54	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
55	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
56	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
57	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
58	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
59	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
60	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
61	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
62	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
63	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
64	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
65	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
66	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
67	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
68	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
69	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
70	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
71	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
72	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
73	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
74	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
75	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
76	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
77	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
78	X	X	X	X	X	X	X	X								X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
79	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
80	X	X	X	X	X	X	X	X	X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Примечание: \*) Столбец 32 «Создание и удаление» предусматривает реализацию команд создания и удаления файлов, папок и пользователей.

\*\*\*) Столбец 37 «Эмуляция межпроцессного взаимодействия» предусматривает использование средств взаимодействия, указанных в столбцах 11 – 16.

## 4 ОПИСАНИЕ ЭЛЕМЕНТОВ КЛАССИФИКАТОРА

Терминология операционных систем постоянно изменяется. Это связано с тем, что каждая новая промышленная ОС вносит свои терминологические особенности. Для однозначности толкования вариантов задания определим основные термины, приведенные в классификаторе.

*Мультипрограммный режим работы* - это режим функционирования ОС, при котором несколько программ (процессов) одновременно находятся на стадии выполнения. При наличии единственного центрального процессора параллелизм достигается за счет асинхронной работы устройств. В то время как одна из программ выполняет счет на процессоре, другие программы могут выполнять операции на внешних устройствах. Таким образом, выигрыш эффективности по сравнению с однопрограммным режимом достигается за счет параллельной работы устройств, входящих в состав компьютера.

*Многопользовательская защита* - средства ОС, обеспечивающие идентификацию пользователей и различные уровни их привилегий, а также установку и защиту прав собственности на информационные ресурсы, создаваемые в среде ОС. Надежная защита возможна только при наличии специальных аппаратных средств, обеспечивающих защиту оперативной памяти и различные режимы работы процессора.

*Режимы* различаются множеством доступных команд. Полный набор команд процессора доступен только в системном режиме, в котором исполняются модули ОС. Исполнение привилегированной команды в пользовательском режиме приводит к возникновению специального прерывания, обрабатываемого системой.

*Интерактивный режим* - режим работы ОС, при котором основными источниками команд являются пользователи, оперативно взаимодействующие с компьютером посредством терминалов. В интерактивном режиме необходимо обеспечить приемлемое для пользователя время исполнения команды (как правило, в пределах нескольких секунд).

*Пакетный режим* - режим работы ОС, при котором основными источниками команд являются специальные пакетные файлы, предварительно подготовленные пользователями. В пакетном режиме время исполнения конкретной команды не столь существенно как в интерактивном. Главным критерием работы пакетной ОС является, как правило, максимальная загрузка оборудования.

*Квантование времени* используется для обеспечения эффективной интерактивной работы с компьютером нескольких пользователей. Устанавливается интервал времени - квант, в течение которого процесс может исполняться на процессоре. По истечении кванта следующий процесс назначается на процессор. Готовые к исполнению процессы по очереди циклически получают свои кванты.

Таким образом, создается эффект одновременного параллельного исполнения процессов.

*Приоритет* - параметр, определяющий преимущества процесса в обладании ресурсами по сравнению с другими процессами.

*Статический приоритет* - приоритет, который устанавливается и изменяется пользователем. *Динамический приоритет* - приоритет, устанавливаемый и изменяемый операционной системой для обеспечения эффективной работы компьютера. Таким образом, основное отличие статических и динамических приоритетов в том, что статические приоритеты система только учитывает в своей работе, а динамические она формирует и модифицирует, исходя из общих критериев эффективности.

*Абсолютные и относительные приоритеты* различаются степенью срочности предоставления привилегий их обладателям. Относительные приоритеты учитываются при выделении свободных (освобожденных) ресурсов. Появление процесса с более высоким абсолютным приоритетом при отсутствии свободных ресурсов приводит к захвату ресурсов у менее приоритетных процессов. Например, замена менее приоритетного процесса на процессоре до истечения кванта либо вытеснение из оперативной памяти во внешнюю.

*Свопинг процессов* - вытеснение готового к исполнению процесса из оперативной памяти во внешнюю. При этом оперативная память, заполненная процессом, полностью освобождается. Свопинг процессов может использоваться как средство реализации абсолютных приоритетов при отсутствии виртуальной памяти либо инициироваться операцией на внешнем устройстве, требующей значительных временных затрат. Например, в некоторых системах выполняется свопинг процессов на время терминального ввода. Операция ввода с терминала может выполняться до нескольких десятков секунд и освобождение оперативной памяти на это время обеспечивает значительное повышение эффективности работы системы.

*Средства взаимодействия процессов* - средства ОС, обеспечивающие обмен данными между процессами и их взаимную координацию действий.

## 5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

Как было отмечено в начале, операционные системы современных компьютеров относят к числу наиболее сложных систем, созданных когда-либо человеком. Имеется обширная литература, содержащая теоретические основы проектирования ОС и детальные описания реальных ОС для наиболее распространенных семейств компьютеров.

Целью настоящих методических указаний является описание порядка выполнения проекта, а также рекомендация ряда проектных решений, позволяющих значительно снизить сложность проектируемой системы при сохранении наиболее существенных особенностей реальных ОС.

Процесс проектирования ОС, как и других сложных систем, имеет итерационный характер. Поэтому выполнение очередного этапа работ зачастую будет требовать внесения корректировок в результаты предыдущих этапов. Такую переработку уже полученных результатов следует воспринимать как неизбежный процесс, связанный со спецификой объекта проектирования. Перед оформлением пояснительной записки к защите необходимо выполнить окончательную взаимоувязку компонент системы, обеспечивающую их согласованную работу.

Рекомендуемые проектные решения, существенно снижающие сложность ОС:

1. Выбор гипотетической архитектуры аппаратных средств, отображающей лишь существенные черты реальной аппаратуры провести теоретически.
2. Выбор прототипа системы среди реальных работоспособных ОС (ОС семейства Windows или ОС семейства Linux).
3. Разделение командного языка ОС на множества основных и дополнительных команд. Основное множество команд сформировать

как минимальное достаточное для выполнения варианта задания. Прототипами команд могут являться команды ОС Linux. Основные команды файловой системы предполагают вход и выход из системы, работу с файлами и каталогами (создание файла, просмотр содержимого файла, запись в файл, копирование файлов, переименование файлов, удаление файлов и каталогов). Дополнительные команды предполагают реализацию средств защиты на уровне файлов между пользователями системы, создание новых пользователей системы, изменение пароля администратором учетной записи пользователя.

4. Не включать в состав ОС возможности, не предусмотренные в варианте задания.
5. Считать, что ядро системы функционирует в непрерываемом режиме и, таким образом, не рассматривать многоуровневые прерывания.
6. Считать, что терминал является алфавитно-цифровым монохромным, работающим в построчном режиме и, таким образом, не рассматривать специфические проблемы, связанные с организацией графических многооконных интерфейсов.
7. Считать, что система разрабатывается для автономного компьютера, и, таким образом, не рассматривать проблемы, связанные с работой в локальной или глобальной сети.
8. Не рассматривать проблемы, связанные с генерацией системы и ее настройкой на различные архитектуры аппаратных средств.
9. Считать, что операции ввода/вывода синхронны по отношению к запустившему их процессу. Т.е. процесс всегда ожидает завершения текущей операции и для каждого процесса в произвольный момент времени имеется не более одной запущенной операции.

Перечисленные решения носят характер рекомендаций и не являются обязательными для исполнения. Однако они позволяют снизить сложность работ

по проектированию ОС до уровня, выполняемого одним студентом в течение семестра.

При выборе прототипов проектируемой ОС следует рассматривать преимущественно системы, обеспечивающие мультипрограммный режим работы. Фактическим стандартом в области операционных систем стала система UNIX. Рекомендуется использовать в качестве прототипов современные ОС.



## 6 ВЫБОР АРХИТЕКТУРЫ КОМПЬЮТЕРА

С точки зрения программиста архитектура компьютера – это средства, доступные ему при работе на машинном языке (языке ассемблера).

При изучении теоретических вопросов, связанных с архитектурой, следует обратить внимание только на существенные черты, необходимые для понимания проекта. Основное требование к выбору архитектуры - это эффективная реализация механизмов управления ресурсами и процессами, указанных в варианте задания.

Из проектного задания вытекают следующие обязательные требования к элементам архитектуры, которые необходимо изучить:

- обеспечение асинхронного режима работы внешних устройств (ВУ);
- аппаратная поддержка указанного способа организации ОП;
- аппаратная поддержка средств защиты.

Асинхронный режим работы ВУ достигается за счет того, что устройство (множество однотипных устройств) является по существу специализированным процессором и может работать в течение некоторого времени без участия центрального процессора. Для взаимодействия с устройством служат специальные регистры, в которые центральный процессор помещает параметры текущей операции ввода/вывода, а устройство выводит информацию о своем состоянии. Операция на устройстве выполняется автономно, а по ее завершении устройство выставляет на процессор сигнал прерывания, обрабатываемый ОС. Таким образом, программный интерфейс устройства - это структура его регистров ввода/вывода и аппаратные прерывания, генерируемые устройством в различных условиях. Следует отметить, что в некоторых семействах компьютеров схема организации ввода/вывода может быть гораздо сложнее описанной.

Существенно отличается взаимодействие с байт-ориентированными (терминал, принтер) и блок-ориентированными (диск) устройствами. В первом случае каждый передаваемый байт непосредственно помещается в регистр

устройства. Во втором случае в регистре указывается адрес и длина последовательности байтов, передаваемой без участия центрального процессора. Для организации такой передачи необходим прямой доступ устройства к оперативной памяти.

Рассмотрим элементы архитектуры, необходимые для обеспечения различных способов управления оперативной памятью. Выделение памяти динамическими разделами вызывает ее фрагментацию.

Для устранения фрагментации в этом случае используют перемещение разделов. Поэтому необходима аппаратная поддержка относительных режимов адресации, при которых не требуется модификация адресов в перемещаемых разделах. Для поддержки страничной и сегментно-страничной организации памяти создаются таблицы сегментов и таблицы страниц. Если эти таблицы хранить в оперативной памяти, то работа системы может замедлиться в несколько раз. Поэтому необходимо иметь специальную быструю память (специальные регистры) для хранения фрагментов страничных и сегментных таблиц.

Основу описания системы прерываний составляет классификация прерываний. Как правило, различают аппаратные, программные и логические прерывания. Важно также указать способы управления прерываниями, позволяющие запрещать или разрешать все прерывания, либо отдельные их типы. Список всех прерываний, возникающих в компьютере необходим, поскольку основные модули ОС являются по существу обработчиками прерываний.

Средства защиты ОС базируются на аппаратном обеспечении нескольких режимов работы процессора и выделении привилегированного множества команд, недоступного в пользовательском режиме. При исполнении недоступной команды процессором генерируется соответствующее прерывание, обрабатываемое средствами ОС. Сегментные и страничные регистры в совокупности с привилегированным множеством команд обеспечивают надежную защищенность ОС и прикладных процессов. При организации памяти

динамическими разделами необходимы дополнительные ключи защиты памяти либо граничные регистры и средства управления ими.

## 7 СТРУКТУРА ФАЙЛОВОЙ СИСТЕМЫ

Данный раздел должен содержать описание основных принципов построения спроектированной файловой системы. Разработку раздела рекомендуется вести в следующей последовательности:

- выделить основные характеристики файловой системы, определенные в ТЗ;
- определить логическую структуру диска с разрабатываемой файловой системой. Для этого необходимо выделить, каким образом на диске будет храниться служебная информация и данные;
- определить перечень необходимых служебных структур. Служебные структуры должны содержать общую информацию о диске, информацию о распределении области данных между отдельными объектами (файлами, каталогами). В случае реализации многоуровневой системы разработать структуру каталогов и то, как информация об этой структуре будет храниться на диске;
- разработать состав всех служебных структур, определить назначение всех полей и их размерность.

Результатом, отображаемым в записке, является типовая структурная схема диска, пример которой показан на рисунке 1:



Рисунок 1 - Пример логической структуры диска

В текстовом пояснении к рисунку должен быть указан перечень всех областей, информация об их назначении и размещении.

Для каждой служебной области и области данных (если необходимо) следует подробно описать каждое поле, указав его размерность и назначение (рисунок 2):

Область 1		
Поле	Размерность	Назначение
BEGIN_CLUST	4 байта	Номер первого кластера объекта
.....	.....	.....

Рисунок 2 - Пример описания структуры

В текстовом пояснении необходимо подробно и четко изложить, как в разработанной файловой системе реализуется хранение файлов, как сохраняется информация об их размещении, наличии или отсутствии свободного места, как организовывается структура каталогов (если она присутствует), как реализуется вход в файловую систему (корневой каталог или др. способ) и т.д. В случае необходимости описание следует сопровождать иллюстрациями (пример размещения объекта на диске, пример содержимого файловой таблицы и т.д.)

Информации, изложенной в отчете должно быть достаточно для уяснения основных принципов построения разработанной ФС без дополнительных устных объяснений студента. Т.е. при написании раздела в качестве примера следует руководствоваться технической документацией на существующие ФС.

Разрешается брать за основу разрабатываемой ФС любую известную файловую систему: FAT, NTFS, HPFS, Ext2s, Ext3s и т.д, при этом разрешается упростить или доработать по своему усмотрению структуры данных, описывающие суперблок, структуру индексного дескриптора, структуру каталога. В случае использования в качестве базовой уже существующей файловой системы, не нужно приводить описание данной ФС. В записке должна присутствовать информация только о том, что разработано студентом. Т.е. то, как будет выглядеть ФС после удаления лишних компонентов и внесения изменений таким образом, чтобы она удовлетворяла ТЗ.

Обязательной является разработка в составе ФС средств для реализации многопользовательской защиты. Также должно быть приведено подробное описание того, как эти средства функционируют. Т.е. на основании каких признаков происходит разграничение доступа, где хранится эта служебная информация, и какой вид она имеет.

Студент должен также разработать перечень команд, необходимых для всесторонней и полной работы с файловой системой. За основу рекомендуется использовать системы команд известных ОС: Linux/Unix и т.д.

Для этого необходимо выделить перечень типовых операций, которые пользователь обычно совершает с объектами файловой системы (с учетом требований ТЗ):

- создание объекта;
- удаление объекта;
- копирование;
- перемещение;
- переименование;
- изменение прав доступа;
- просмотр информации об объекте;
- форматирование диска;
- и т.д.

Ни в коем случае не следует вставлять в курсовой проект перечень команд реальной ФС с описанием из справочной документации. В системе команд должны присутствовать только те команды, которые необходимы для разрабатываемой ОС. Описание команд приводится в виде таблицы (Таблица 1):

Таблица 1 – Пример организации таблицы для описания основных команд

Имя команды	Параметры	Описание
Copу	Src Dst	Копирует объект из места Src в место Dst
.....	.....	.....

Перечень параметров не следует делать слишком объемным. При использовании в качестве основы существующих команд, старайтесь оптимизировать список параметров до необходимого минимума.

Все команды, реализованные в программе, эмулирующей работу файловой системы, должны быть приведены в help программы (название, назначение команды, синтаксис).

Системные вызовы, разрабатываемые в данном разделе - это типовые подпрограммы, предоставляемые ОС для выполнения элементарных действий с объектами ФС.

Например, для организации команды копирования объекта, необходимо:

- проверить существование указанного объекта;
- проверить существование пути, куда производится копирование;
- получить размер объекта;
- определить, достаточно ли места для сохранения копии в заданном месте;
- поэлементно считать объект из одного места и записать в другое.

Для выполнения вот таких элементарных действий и следует использовать системные вызовы. Результатом разработки на данном этапе должна быть таблица, в которой указано наименование вызова, его параметры, возвращаемый результат и описание (Таблица 2):

Таблица 2 – Пример организации таблицы для описания системных вызовов

Вызов	Параметры	Результат	Описание
OPEN	PATH, MODE	DESC	Открывает объект по указанному пути PATH в режиме MODE (запись/чтение) и возвращает номер дескриптора файла в системной таблице
.....	.....	.....	.....

В подразделе «Способы организации файлов» должны быть рассмотрены особенности организации файлов с последовательным и произвольным доступом (см. вариант задания). В случае, если хранение файлов в разработанной файловой

системе имеет особенности по сохранения таких объектов, как длинные имена, расширенные атрибуты и т.д., их следует подробно описать в данном подразделе.

Обязательно необходимо привести примеры размещения тех или иных объектов и примеры работы с файлами с различным видом доступа.

Подраздел «Алгоритмы работы некоторых системных вызовов» должен содержать алгоритмы выполнения типовых системных вызовов (не менее), реализуемые в виде алгоритмов вызовы студент выбирает самостоятельно. Алгоритмы вызовов могут изображаться в виде блок-схем или описаний на псевдоязыке.

Обязательно наличие комментариев, поясняющих работу алгоритмов.

В подразделе «Разработка ФС» раздела «Разработка программ эмуляции ОС» должны описываться принцип эмуляции и описание разработанного ПО.

Разработка эмулятора файловой системы подразумевает, что студент не работает со служебными структурами данных реальной файловой системы. Наиболее простой способ эмуляции – это представление логической структуры диска в виде образа – файла, имеющего структуру, совпадающую со структурой диска в разработанной ФС.

Внутри данного файла должны содержаться все служебные области и области данных. Размер образа должен быть небольшим, чтобы его можно было легко переносить с одного компьютера на другой при демонстрации. Возможно также динамическое формирование образа (имитация форматирования) при запуске программы.

Все действия с объектами ФС должны осуществляться внутри вышеуказанного файла-образа. Т.е. при демонстрации студент должен продемонстрировать изменения, происходящие в образе при работе с объектами ФС.

На усмотрение студента при форматировании и работе со структурами ФС допускается создание ни одного файла-образа, а несколько служебных файлов (см. аналог файловой системы NTFS).

Для общения ОС – пользователь система должна обладать каким-либо интерфейсом (текстовая консоль, графическая оболочка на усмотрение студента). Наиболее простой способ - написание программы-эмулятора, которая создает текстовый интерфейс ввода-вывода, пример экранной формы работы эмулятора представлен на рисунке2.

```
MY_OS>>copy
MY_OS>>недостаточно параметров. Повторите ввод
MY_OS>>copy a1 a2
MY_OS>>OK
MY_OS>>
```

Рисунок 2 – Пример работы программы-эмулятора интерпретатора

Командный интерпретатор в данном случае – это программа, которая в непрерывном цикле ожидает ввода команд. При вводе команды должен реализовываться ее синтаксический разбор (существование имени команды, наличие всех обязательных параметров и т.д.). На экран должны выводиться подробные сообщения об ошибках ввода.

При разработке командного интерпретатора следует предусмотреть процедуру входа пользователя в систему (по схеме логин-пароль), выход из системы и другие вспомогательные действия.



## 8 ПРОЦЕССЫ В ОС

В пояснительной записке, в главе «Процессы в ОС» необходимо дать определение процесса в операционной системе, представить таблицу основных и дополнительных команд для работы с процессами, таких как просмотр запущенных процессов, запуск процессов в фоновом режиме, изменение приоритета процесса, запущенного пользователем, уничтожение процесса. Перечислить основные системные вызовы для работы с процессами в ОС (прототипами можно брать системные вызовы по управлению процессами в ОС Unix).

Процессы в операционной системе «за время жизни» могут находиться в различных состояниях – порождения, ожидания, выполнения и т.д. В ходе написания данной главы необходимо представить диаграмму состояний процесса и дать пояснение каждому из состояний и описать переходы между состояниями.

В соответствии с вариантом задания к курсовом проекте студенту предоставляется возможность изучить «поведение» процессов в зависимости от их приоритетов. При написании дальнейших пунктов курсового проекта (планирование и диспетчеризация процессов, описание структуры блока управления процессами, алгоритм планировщика процессов, перечень очередей процессов) необходимо обратить внимание на приоритеты процессов в соответствии с заданием.

Немаловажную роль при написании курсового проекта занимает вопрос теоретического изучения взаимодействия между процессами. Средствами межпроцессного взаимодействия являются: каналы, именованные каналы, сообщения, общие области оперативной памяти, разделяемая память, отображаемая память, семафоры. При изучении теоретического материала по взаимодействию процессов следует использовать теоретический материал обучающей системы, посвященной межпроцессному взаимодействию (доступной на сервере кафедры). В пояснительной записке обязательно представить понятие межпроцессного взаимодействия, кратное теоретическое описание средств

межпроцессного взаимодействия в соответствии с вариантом задания и алгоритм взаимодействия двух или более процессов по средствам вышеописанных средств. Студент, не усвоивший теоретические принципы межпроцессного взаимодействия, не сможет перейти к этапу программной реализации межпроцессного взаимодействия и синхронизации процессов.

Немаловажную роль при проектировании операционной системы играет решение задачи планирования и диспетчеризации процессов.

Планировщик процессов определяет, какой из процессов, когда и на сколько времени получит управление процессором.

Ими должны выполняться следующие функции:

1. Регистрация информации о состоянии всех процессов
2. Принятие решения о том, какой из готовых к выполнению процессов должен получить управление, какой из имеющихся в системе процессоров ему предоставить и на сколько времени.
3. Подключение процессора к процессу. При этом необходимо установить значения регистров процессора в соответствии с текущим состоянием процесса.
4. Отключение процессора от процесса, например, при истечении выделенного интервала времени или при переходе процесса в состояние ожидания ввода-вывода. При этом должны быть сохранены значения отражающих состояние процесса управляющих регистров, что необходимо для последующего его восстановления.

## 9 ДИСЦИПЛИНЫ ОБСЛУЖИВАНИЯ

Дисциплина обслуживания определяет порядок, в котором процессор обслуживает процессы, и длительность квантов времени обслуживания для каждого из них. Длительность интервала времени, в течение которого процессор обслуживает данный процесс, может зависеть от одного из следующих событий или их комбинации:

1. Выполнение процесса завершено.
2. Процесс перешел в состояние ожидания.
3. Процессор потребовался для обслуживания процесса, имеющего более высокий приоритет.
4. Завершился заранее выделенный процессу квант времени.
5. Произошла ошибка.

Решение о выборе процесса на обслуживание принимается путём просмотра очереди готовых РСВ в соответствии с принятой дисциплиной обслуживания.

Очередь готовых РСВ может быть организована одним из следующих способов:

1. Всякий раз, когда процесс переводится в состояние готовности (из состояний хранения, выполнения или ожидания), его РСВ помещается в соответствующую приоритету процесса позицию очереди. Когда процессор освобождается, он переключается на обслуживание процесса, стоящего в очереди первым.
2. Процессы располагаются в очереди произвольно. Когда необходимо выбрать процесс для обслуживания, производится полный просмотр списка.

Можно перечислить следующие широко распространённые дисциплины планирования:

1. *Равномерное циклическое квантование.* Длительность обслуживания каждого процесса ограничена величиной стандартного кванта времени, равного, например, 100 мс.
2. *Приоритет по характеру использования предшествующего кванта.* Если процесс полностью использовал свой предшествующий квант времени, он помещается в конец очереди. Если он использовал только половину (вследствие блокирования по вводу-выводу), он попадает в середину очереди. Помимо "справедливости", эта дисциплина хорошо обслуживает процессы с интенсивным вводом-выводом.
3. *Вариант равномерного циклического квантования с многоуровневой обратной связью.* В момент введения нового процесса в систему ему выделяется столько квантов, сколько в этот момент существует других параллельных процессов. После этого продолжается обычное равномерное квантование.
4. *Принудительный приоритет.* Выбирается процесс с наивысшим приоритетом из всех готовых. Приоритет назначается, исходя из важности соответствующей работы или с учётом дополнительной оплаты (например, задачам, обслуживающим ядерный реактор, назначается наивысший приоритет).
5. *Ограниченное равномерное циклическое квантование.* Процессы обслуживаются способом равномерного квантования некоторое определённое число раз, после чего они обслуживаются только при условии отсутствия в системе других процессов.
6. *Сбалансированная загрузка.* Для предотвращения простоя внешних устройств предпочтение отдаётся процессам, интенсивно использующим ввод-вывод.
7. *Предпочтительное обслуживание интерактивных процессов.* Если при решении задачи осуществляется оперативное взаимодействие с

пользователем (интерактивные задачи), обслуживание соответствующего процесса начинается сразу после запроса пользователя. С помощью такой дисциплины удаётся обеспечить малое время реакции системы.

8. *Автоматические приоритеты.* В некоторых случаях ответственность за назначение процессам приоритетов возлагают на саму ОС. Например, она сама может повышать приоритеты коротких заданий. В других системах производится автоматическое назначение приоритетов так, что достигается сбалансированная загрузка оборудования, как в случае совместного выполнения двух процессов, один из которых интенсивно использует ввод-вывод (*задача с преимущественным вводом-выводом*), а другой, наоборот, использует, его в очень малой степени (*задача с преимущественным счётом*). Если выполнять несколько процессов, имеющих сходные характеристики, близкие к одному из двух вышеназванных пределов, может возникнуть ситуация, когда каналы ввода-вывода окажутся перегруженными, а процессор будет простаивать или наоборот. Для достижения максимальной производительности система производит назначение приоритетов таким образом, что постоянно поддерживается состав смеси процессов, гарантирующий сбалансированную загрузку системы. Подобно этому, система может увеличивать приоритеты процессов, монополюно владеющих критическими ресурсами (например, накопителем на магнитной ленте), благодаря чему ресурсы освобождаются для использования другими процессами быстрее.

В зависимости от выбранной дисциплины обслуживания определяется алгоритм работы планировщика процессов. Студентам предлагается сначала рассмотреть структуры управляющих блоков базы данных ОС, а затем описать алгоритм работы планировщика процессов в соответствии с выбранной дисциплиной обслуживания. В соответствии с вариантом задания некоторые студенты выполняют эмуляцию работы планировщика процесса, поэтому

подробное изучение теоретических аспектов планирования и диспетчеризации процессов поможет в написании программы эмуляции работы планировщика.

## 10 ОПРЕДЕЛЕНИЕ СОСТАВА И ОБЩЕЙ СТРУКТУРЫ СИСТЕМЫ

Каждая операционная система создает собственную файловую и процессную среду. Один из главных вопросов при проектировании операционной системы является вопрос о том, что представляет собой операционная система с точки зрения собственной среды. С одной стороны, система располагается в определенном наборе файлов, из которых она запускается при загрузке. С другой стороны, для реализации функций ОС создается ряд специальных процессов, называемых системными.

Для общего структурирования ОС классическими являются понятия ядра и системных процессов. *Ядро* - это та часть операционной системы, которая с точки зрения самой системы не является процессом. Ядро является продолжением аппаратных средств и упрощенно представляет собой набор обработчиков основных прерываний. Системные вызовы, как правило, реализуются посредством программных прерываний и представляют собой косвенный вызов модулей ядра со сменой режима работы процессора. Ядро реализует основные системные вызовы по управлению процессами и оперативной памятью, а также, в большинстве систем, функции обмена с внешними устройствами на физическом уровне (драйверы устройств). Другие системные вызовы переадресуются ядром для исполнения системным процессам.

В форме специального системного процесса (файлового процессора) часто реализуются функции файловой системы. Для выполнения интерактивных команд и пакетных файлов также создаются системные процессы. При этом возможна реализация выполнения интерактивных команд всех терминалов одним системным процессом (командным процессором) либо запуск отдельного процесса для каждого подключенного терминала. Основные интерактивные команды выполняются командным процессором непосредственно, а для реализации дополнительных команд имеются специальные исполнимые файлы. Драйверы внешних устройств в некоторых ОС также представляются как системные процессы.

Первичное структурирование системы предполагает выделение ядра и системных процессов различных уровней и распределение их по файлам. Далее выполняется структурирование ядра и системных процессов путем выделения элементов данных и программ (модулей), имеющих определенное функциональное назначение.

Описание состава и общей структуры системы:

- состав ОС: перечень файлов (ядро и системные процессы), их назначение и порядок загрузки;

- структура ядра ОС: перечень модулей и их точек входа с кратким описанием назначения и указанием обрабатываемых прерываний.



## 11 ФОРМИРОВАНИЕ БАЗЫ ДАННЫХ ОС

Совокупность данных, используемую ОС для управления процессами и ресурсами, называют *базой данных ОС* (БД ОС). Основная часть базы данных размещается в ядре системы, что обеспечивает ее надежную защиту. Большая часть элементов базы данных создается и уничтожается динамически; также динамически создаются и модифицируются связи между элементами. Поэтому для реализации БД ОС, как правило, используют многосвязные списочные структуры данных.

Основу базы данных составляют управляющие блоки, объединяемые в списки сложной структуры. Заголовки основных списков и некоторые глобальные системные переменные образуют статическую часть базы данных. Динамически создаваемые и уничтожаемые блоки размещаются в куче ядра (kernel heap).

Управляющие блоки создаются для каждого типа ресурсов, а также для каждого процесса и для элементов, обеспечивающих взаимодействие процессов. Управляющий блок содержит общее описание параметров ресурса (процесса), а также подробную информацию о текущем состоянии. Управляющие блоки одного типа могут объединяться в общий список (очередь), либо группироваться в несколько списков для различных состояний (очередь готовых процессов, очередь заблокированных процессов).

Описание базы данных ОС:

- перечень и назначение управляющих блоков и очередей;
- общая схема организации БД - взаимосвязи управляющих блоков;
- описание структуры управляющих блоков.

Перед оформлением пояснительной записки по результатам перечисленных этапов необходимо выполнить окончательную взаимоувязку отдельных частей проекта, ручную верификацию работы ОС и оценку ее эффективности.

## ЛИТЕРАТУРА

1. Мэдник С., Донован Дж., «Операционные системы». – М.: Мир, 1978
2. Бах М.Д., «Архитектура операционной системы Unix»
3. Кейслер С., «Проектирование операционных систем для малых ЭВМ.» М.: Мир, 1986
4. Дейтел Х. М., «Введение в операционные системы». В 2-х томах – М.: Мир, 1987
5. Робачевский А.М., «Операционная система UNIX». – СПб.: BHV, 1997
6. Таненбаум Э., «Современные операционные системы» , - «Питер», 2002 г.

## СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА .....	4
2. ПОРЯДОК ВЫПОЛНЕНИЯ И СОДЕРЖАНИЕ ПРОЕКТА. ....	5
3. ЗАДАНИЕ К КУРСОВОМУ ПРОЕКТУ .....	8
4. ОПИСАНИЕ ЭЛЕМЕНТОВ КЛАССИФИКАТОРА .....	11
5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ.....	14
6. ВЫБОР АРХИТЕКТУРЫ КОМПЬЮТЕРА.....	16
7. СТРУКТУРА ФАЙЛОВОЙ СИСТЕМЫ .....	18
8. ПРОЦЕССЫ В ОС .....	24
9. ДИСЦИПЛИНЫ ОБСЛУЖИВАНИЯ .....	26
10. ОПРЕДЕЛЕНИЕ СОСТАВА И ОБЩЕЙ СТРУКТУРЫ СИСТЕМЫ .....	31
11. ФОРМИРОВАНИЕ БАЗЫ ДАННЫХ ОС .....	33
ЛИТЕРАТУРА .....	34

Приложение А  
Техническое задание к курсовому проекту

Утверждено  
зав. кафедрой ПИ  
\_\_\_\_\_ Федяев О.И.  
« \_\_\_\_ » \_\_\_\_\_ 2016 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**  
на курсовому проекту по дисциплине  
«Операционные системы»

студенту группы ИПО14 \_\_\_\_\_

Тема: “Проектирование гипотетической операционной системы и программная  
эмуляция отдельных модулей”

Вариант №

Задание:

1. Режим работы ОС: \_\_\_\_\_
2. Управление процессами: \_\_\_\_\_
3. Средства взаимодействия процессов: \_\_\_\_\_
4. Организация оперативной памяти: \_\_\_\_\_
5. Перечень внешних устройств: \_\_\_\_\_
6. Организация файловой системы: \_\_\_\_\_
7. Команды интерпретатора \_\_\_\_\_
8. Эмуляция работы: \_\_\_\_\_

График выполнения курсового проекта:

Неделя	Работа
1-2	Выдача задания и изучение задания
3	Анализ требований к системе и способов их реализации
4-5	Проектирование файловой системы
6-7	Организация процессов
8	Поддержка режимов работы, проектирование общей структуры
9-13	Разработка программ эмуляции
14	Оформление пояснительной записки
15-17	Защита курсового проекта

Дата выдачи задания

1.09.2016

Задание принял

Руководитель проекта

\_\_\_\_\_   
Чернышова А.В.

# **МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ**

к курсовому проекту по дисциплине

**«Операционные системы»**

(Для студентов направления подготовки

09.03.04 "ПРОГРАММНАЯ ИНЖЕНЕРИЯ")

Составители:

Алла Викторовна Чернышова