

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Амурский государственный университет»

Кафедра информационных и управляющих систем

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

«ОПЕРАЦИОННЫЕ СИСТЕМЫ»

Основной образовательной программы по направлению подготовки

230100.62 – Информатика и вычислительная техника

Благовещенск 2011

УМКД разработан канд. техн. наук, доцентом Т.А. Галаган

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от «__» _____ 201_ г. № _____

Зав. кафедрой _____ / А.В. Бушманов /
(подпись) (И.О. Фамилия)

УТВЕРЖДЕН:

Протокол заседания УМСС _____
(указывается название специальности (направления подготовки))

от «_____» _____ 201__ г. № _____

Председатель УМСС _____ / _____ /
(подпись) (И.О. Фамилия)

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель дисциплины – изучение принципов построения, назначения, теоретических основ функционирования и практического использования операционных систем как эффективного средства управления процессами обработки данных в современных ЭВМ.

Задачи дисциплины:

подготовка к участию во всех фазах проектирования и разработки объектов профессиональной деятельности;

подготовка к участию в разработке всех видов документации на программные, аппаратные и программно-аппаратные комплексы;

способность использовать современные методы, средства и технологии разработки объектов профессиональной деятельности.

По завершению изучения курса «Операционные системы» студент должен:

знать

назначение и функции операционной системы, современное состояние теории операционных систем и методы, используемые при их разработке;

теоретические основы планирования процессов и потоков и управление памятью;

современные средства вычислительной техники, коммуникаций и связи;

уметь использовать практические навыки работы с операционными системами Windows XP/NT, Linux;

владеть такими понятиями, как вычислительный процесс, мультипрограммирование, виртуализация памяти, файловая система, и способы их реализации в современных операционных системах

Преподавание курса «Операционные системы» связано с изучением курса государственного образовательного стандарта «Информатика» и является основой для изучения дальнейших дисциплин, использующих ЭВМ и программирование.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Код УЦ ООП ОПД.Ф.08 – общепрофессиональные дисциплины направления. Федеральный компонент

Содержание дисциплины: назначение и функции операционных систем; мультипрограммирование; режим разделения времени; многопользовательский режим работы; режим работы и ОС реального времени; универсальные операционные системы и ОС специального назначения; классификация операционных систем; модульная структура построения ОС и их переносимость; управление процессором; понятие процесса и ядра; сегментация виртуального адресного пространства процесса; структура контекста процесса; идентификатор и дескриптор процесса; иерархия процессов; диспетчеризация и синхронизация процессов; понятия приоритета и очереди процессов; средства обработки сигналов; способы реализации мультипрограммирования; понятие прерывания; многопроцессорный режим работы; управление памятью; совместное использование памяти; защита памяти; механизм реализации виртуальной памяти; стратегия подкачки страниц; принципы построения и защита от сбоев и несанкционированного доступа.

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 140 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекц.	Лаб.	Сам.	
1	Назначение, функции, классификация ОС	4	1 – 3	6		4	
2	Архитектура операционной систем	4	4 – 6	6		6	
3	Процессы и потоки	4	7-9	6		8	Тест №1
4	Управление памятью	4	10 – 13	8		6	
5	Файловая система	4	14, 15	4		8	Тест №2
6	Методы защиты от сбоев и несанкционированного доступа	4	16	2		4	
7	Навыки работы с современными ОС	4	1 – 16		32	40	Отчеты по лаб. раб.
	ИТОГО			32	32	76	

4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

4.1 Лекции

4.1.1 Принципы построения современных операционных систем: назначение, основные функции операционной системы; классификация операционных систем: мультипрограммные, мультизадачные, однопользовательские и многопользовательские, режим разделения времени и режим реального времени, универсальные ОС и системы специального назначения, многопроцессорные ОС; современные требования, система прерываний.

4.1.2 Архитектура операционной системы:

Макроядерная архитектура – модульная структура построения ОС, ядро и вспомогательные модули; многослойная структура ядра; микроядерная архитектура операционной системы.

4.1.3 Процессы и потоки:

мультипрограммирование; понятие процесса и потока; идентификатор процесса; контекст и дескриптор процесса; алгоритмы планирования потоков: понятие очереди, приоритетное обслуживание, квантование; диспетчеризация и синхронизация процессов; мультипрограммирование на основе прерываний.

4.1.4 Управление памятью:

функции операционной системы по управлению памятью; алгоритмы распределения памяти; свопинг и виртуальная память; механизмы реализации виртуальной памяти; кэширование данных совместное использование памяти; защита памяти.

4.1.5 Файловая система:

понятие файловой системы, ее основные свойства; реализации файловой системы; дескриптор файла; FAT, NTFS.

4.1.6. Методы защиты от сбоев и несанкционированного доступа

- 4.2 Лабораторные работы
 - 4.2.1 Загрузка операционной системы WINDOWS XP. Структура оперативной памяти. Распределение ресурсов компьютера
 - 4.2.2. Служебные программы WINDOWS XP
 - 4.2.3. Вирусы и антивирусные программные средства
 - 4.2.4. Работа в ОС Linux (установка, настройка, команды консоли, установка ПО)

5. САМОСТОЯТЕЛЬНАЯ РАБОТА

№ раздела (темы) дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в часах
1 – 6	Изучение разделов теоретического материала, обзоров о совр. ОС	36
7	Подготовка к лаб. работе, подготовка отчетов по лаб. раб.	40

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

К образовательным технологиям, используемым в преподавании данной дисциплины, относятся лекции и лабораторные работы.

В изложении лекционного материала наряду с традиционной лекцией используются такие неимитационные методы обучения, как:

- проблемная лекция, начинающаяся с постановки проблемы, которую необходимо решить в ходе изложения материала,

- лекция-визуализация, учащая студента преобразовывать устную и письменную информацию к визуальной форме в виде схем, рисунков, чертежей,

- лекция с заранее запланированными ошибками, которые студенты должны обнаружить самостоятельно в конце лекции.

На лекциях используются информационные технологии – презентации. Лабораторные работы проводятся в компьютерных классах и предназначены для решения прикладных задач с использованием современных инструментальных средств.

При проведении лабораторных работ используются неигровые имитационные методы обучения:

- контекстное обучение, направленное на решение профессиональных задач,

- работа в команде – совместная деятельность студентов в группе, направленная на решение общей задачи с разделением ответственности и полномочий.

При оценивании результатов обучения используется балльно-рейтинговая технология.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

7.1. Вопросы к зачету

1. Основные понятия ОС
2. Назначение и функции ОС
3. Классификация ОС
4. Требования, предъявляемые к современным ОС
5. Прерывания (определение, виды, механизм обработки)
6. Диспетчеризация и приоритезация прерываний
7. Макроядерная архитектура ОС
8. Микроядерная архитектура ОС

9. Понятия «процесс» и «поток»
10. Состояния потока
11. Вытесняющие и невытесняющие алгоритмы планирования потоков
12. Алгоритмы планирования, основанные на квантовании
13. Алгоритмы планирования, основанные на приоритетах
14. Синхронизация процессов и потоков
15. Функции ОС по управлению памятью
16. Типы адресов, виртуальное адресное пространство
17. Классификация методов распределения памяти
18. Методы распределение оперативной памяти без использования внешней памяти
19. Свопинг и виртуальная память
20. Страничное распределение памяти
21. Сегментное распределение памяти
22. Сегментно-страничное распределение памяти
23. Разделяемые сегменты памяти
24. Иерархия запоминающих устройств
25. Кэш-память, принцип действия
26. Понятие файловой системы
27. Функции файловой системы и иерархия данных
28. Файловая системы FAT
29. Файловая система NTFS
30. Классификация вирусов
31. Виды и классификация антивирусных программ
32. Методы защиты от сбоев
33. Методы защиты от несанкционированного доступа

7.2. Примерные вопросы теста №1

1. Выберите верное утверждение:

- 1) В мультипрограммных системах программы всегда выполняются в нескольких процессорах параллельно.
- 2) В мультипрограммных системах несколько программ выполняются в процессоре попеременно, совместно используя ресурсы компьютера.
- 3) Мультипроцессирование исключает мультипрограммирование.
- 4) В мультипрограммных системах несколько программ выполняются процессором одновременно.

2. Сколько состояний потока различают?

- 1) 2
- 2) 4
- 3) 6
- 4) 5

3. Операционная система выполняет планирование потоков в зависимости от их состояния. Состояние «ожидаемый» есть

- 1) активное состояние потока, в котором, обладая всеми ресурсами, поток ожидает завершения его выполнения процессором
- 2) пассивное состояние потока, при котором он заблокирован в связи с внешними обстоятельствами, например, выполнение более приоритетного потока
- 3) пассивное состояние потока, при котором он заблокирован по своим внутренним причинам – завершения ввода-вывода или освобождения некоторого ресурса
- 4) пассивное состояние вновь созданного потока

4. Утилиты - это

- 1) программы, реализующие специальный вариант пользовательского интерфейса

- 2) программы, решающие отдельные задачи управления и сопровождения компьютерной системы
- 3) библиотеки процедур различного назначения, упрощающие разработку приложений
- 4) модули-приложения, обеспечивающие расширяемость операционной системы

7.3. Примерные вопросы теста №2

1. Страничная виртуальная память организует перемещение данных между памятью и диском

- 1) частями виртуального адресного пространства произвольного размера, полученного с учетом смыслового значения данных
- 2) частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера
- 3) частями виртуального адресного пространства, фиксированного размера, которые в свою очередь, являются частью сегмента, полученного с учетом смыслового значения
- 4) частями, размеры которых соответствуют виртуальным адресным пространствам процессов

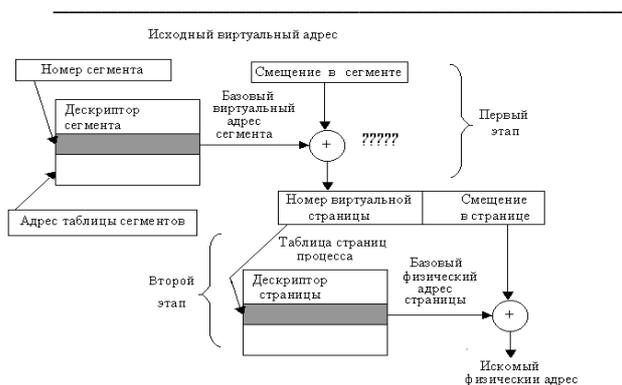
2. Восстановите иерархию запоминающих устройств в порядке убывания объема, если обозначить через a - регистры процессора, b - оперативная память, c - быстродействующая память, d - внешняя память

- 1) abcd 2) acbd 3) dbca 4) dcba

3. К методам распределения памяти без использования внешней памяти относятся

- 1) фиксированными и динамическими разделами
- 2) страничное и сегментное распределение
- 3) фиксированными разделами и страничное распределение
- 4) динамическими разделами и сегментное распределение

4. Запишите пропущенный в схеме адрес



7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОСНОВНАЯ ЛИТЕРАТУРА

1 Гордеев, А.В. Операционные системы. (Допущено Министерством образования РФ) – СПб:Питер, 2006, 2009. – 416с.

2 Олифер, В.Г. Сетевые операционные системы / В.Г. Олифер, Н.А. Олифер СПб.: Питер, 2005, 2010. – 539с.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1 Ивановский, С. Операционная система Linux. М: Познавательная книга плюс, 2001. – 512 с.

ИНТЕРНЕТ-РЕСУРСЫ

	Наименование ресурса	Характеристика
1	http://www.intuit.ru	ИНТУИТ - сайт, который предоставляет возможность дистанционного обучения по нескольким образовательным программам, касающимся, в основном, информационных технологий. Содержит несколько сотен открытых образовательных курсов.
2	http://ru.wikipedia.org	Википедия – свободная общедоступная мультязычная универсальная интернет-энциклопедия. Поиск по статьям, написанным на русском языке. Избранные статьи, ссылки на тематические порталы и родственные проекты.

ПЕРИОДИЧЕСКИЕ ИЗДАНИЯ

Журналы «Информационные технологии и вычислительные системы», «Компьютер-Пресс», «Программные продукты и системы»

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Мультимедийная лекционная аудитория (331)

В качестве программного обеспечения используются свободно распространяемая операционная система Linux и лицензионная версия Windows XP.

9. РЕЙТИНГОВАЯ ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

Балльная структура оценки за семестр

Семестровый модуль дисциплины						
Учебные модули	Виды контроля	Сроки выполнения (недели)	Макс. кол-во баллов	Посещение занятий, активность	Макс. кол-во баллов за уч. модуль	
1	Назначение, функции, классификация ОС	Тест №1	9	5	4.5	9.5

2	Архитектура операционной систем					
3	Процессы и потоки					
4	Управление памятью	Тест №2	15	5	3.5	8.5
5	Файловая система					
6	Методы защиты от сбоев и несанкционированного доступа					
7	Навыки работы с современными ОС	Отчеты по лаб. работам	1 – 16	34	8	42
	Сдача зачета					40
	Итого					100

Итоговая оценка

Сумма баллов, набранных в течение семестра (с возможностью проставления предварительной оценки за экзамен)	Общая сумма баллов (с учетом сдачи экзамена в период семестровой аттестации)	Итоговая оценка
56 – 60	91 – 100	отлично
51 – 55	75 – 90	хорошо
46 – 50	51 – 74	удовлетворительно
<46	< 51	неудовлетворительно

Краткое изложение программного материала

План-конспект лекций

Тема лекции 1 Основные понятия

План лекции

1. Введение в курс «Операционные системы и оболочки»
2. Понятие «Операционная система»
3. Назначение и функции операционной системы
4. Классификация операционных систем
5. Основные требования, предъявляемые к операционным системам.

Цель - формирование ориентировочной основы для последующего усвоения студентами учебного материала.

Задачи – знакомство студентов с целью и назначением курса, его ролью и местом в системе учебных дисциплин, а также основных терминов и определений, назначения и классификации ОС.

Ключевые вопросы – предназначение операционных систем, способы классификации операционных систем, требования к современным операционным системам.

Операционная система (ОС) – это комплекс специальных программных средств, предназначенных для управления загрузкой, запуском и выполнением других (пользовательских) программ, а также для планирования и управления вычислительными ресурсами ЭВМ.

ОС предназначена для решения двух основных задач:

1. Предоставление пользователю (программисту) вместо реальной аппаратуры компьютера расширенной виртуальной машины, с которой удобней работать и которую легче программировать;
2. Повышение эффективности использования компьютера путем рационального управления его ресурсами в соответствии с некоторыми критериями.

Основные функции операционной системы:

1. Прием от пользователя заданий или команд и их обработка.
2. Загрузка в оперативную память, подлежащих исполнению программ.
3. Распределение памяти и организация виртуальной памяти.
4. Запуск программ.
5. Идентификация всех программ и данных.
6. Прием и выполнение различных запросов от приложений.
7. Обслуживание всех операций ввода-вывода.
8. Обеспечение работы систем управления файлами.
9. Обеспечение режима мультипрограммирования.
10. Планирование и диспетчеризация задач в соответствии с заданными стратегией.
11. Организация механизма обмена сообщениями и данными между выполняющимися программами.
12. Обеспечение сохранности данных, защита приложений друг от друга, защита самой ОС от исполняемых приложений.
13. Аутентификация и авторизация пользователя.
14. Предоставление услуг на случай частичного сбоя системы.

Выводы

Современная ОС должна поддерживать мультипрограммную обработку, виртуальную память, многооконный графический интерфейс пользователя и др. Кроме того, к ним предъявляются эксплуатационные требования: расширяемость, переносимость, совместимость, надежность и отказоустойчивость, безопасность, производительность.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - основная литература пп.1,2.

Тема лекции 2 Прерывания

План лекции

1. Понятие «прерывание»
2. Механизм обработки прерывания
3. Классификация прерываний
4. Дисциплины обслуживания прерываний

Цель – организация целенаправленной познавательной деятельности студентов по овладению термина «прерывание».

Задачи – обеспечить формирование системы знаний по учебной дисциплине, формировать профессиональный кругозор

Ключевые вопросы – факт установления прерывания, обработка прерываний, важность прерываний в работе операционных систем.

Наступление того или иного события сигнализируется прерываниями – *Interrupt*. Источниками прерываний могут быть как аппаратура, так и программы. Каждое прерывание обрабатывается соответственно обработчиком прерываний, входящим в состав ОС.

Прерывания – механизм, позволяющий координировать параллельное функционирование отдельных устройств вычислительной системы и реагировать на особые состояния, возникающие при работе процессора. Прерывание обязательно влечет за собой изменение порядка выполнения команд процессором.

Механизм обработки прерываний независимо от архитектуры вычислительной системы подразумевает выполнение некоторой последовательности шагов, которые реализуются аппаратно и программно.

Классификация прерываний:

Внешние (асинхронные) прерывания происходят вне прерываемого процесса, например, прерывания внешних устройств, ввода-вывода, нарушение питания и др.

Внутренние прерывания называются событиями. Они связаны с работой процессора и синхронны с его операциями, например, нарушение адресации, деление на нуль.

Программные прерывания происходят по соответствующей команде прерывания, т.е. по этой команде процессор осуществляет практически те же действия, что и при внутренних прерываниях. Но этот механизм специально введен для переключения на системные программные модули.

Сигналы, формирующие прерывания могут возникать одновременно. Выбор одного из них осуществляется на основе их приоритетов.

Программное управление специальными регистрами маски (маскирование сигналов прерывания) позволяет реализовывать различные дисциплины обслуживания:

- с *относительными приоритетами*, т.е. обслуживание не прерывается даже при наличии запросов с более высоким приоритетом.

- с *абсолютными приоритетами*, т.е. всегда обслуживается прерывание с наивысшим приоритетом.

- по *принципу стека*, или как говорят, по дисциплине LCFS (Last Come First Served – последним пришел, первым обслужен), т.е. запросы с более низким приоритетом могут прервать обработку прерывания с более высоким приоритетом.

Выводы

После загрузки ОС работа компьютера осуществляется как реакция на события, происходящие в нем.

Механизм прерываний поддерживается аппаратными средствами компьютера и программными средствами операционной системы.

Для упорядочивания процесса обработки прерываний все они распределяются по нескольким приоритетным уровням.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2.

Тема лекции 3 Макроядерная архитектура операционной системы

План лекции

1. Понятие «архитектура операционной системы»
2. Многослойный подход к организации сложных систем
3. Макроядерная архитектура ОС
4. Структура ядра

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом, развитие кругозора по специальности.

Задачи – формирование знаний по организации архитектуры современных операционных систем.

Ключевые вопросы – сложность современных ОС, универсальность подходов к структурированию ОС, многослойный подход в организации сложных систем, ядро и вспомогательные модули, многослойный подход в структуре ядра.

Сложность современных ОС приводит к сложности ее архитектуры, под которой понимают структурную организацию ОС на основе различных программных модулей.

Какой-то единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС.

В макроядерных (или монолитных) ОС все модули разделены на две группы: 1) Ядро – модули, выполняющие основные функции ОС; 2) Модули, выполняющие вспомогательные функции ОС.

Модули ядра взаимодействуют с аппаратными средствами непосредственно и потому ядро ОС должно постоянно храниться в компьютере. В частности, программное обеспечение, входящее в состав ядра, отвечает за проверку работоспособности компьютера и выполнение элементарных (базовых) операций, связанных с работой монитора, клавиатуры, магнитных накопителей и т.п.

В состав ядра входят функции, решающие внутрисистемные задачи организации вычислительного процесса, такие как переключение контекстов, обработка прерываний. Эти функции недоступны для приложений. Другой класс функций ядра служит для поддержки приложений, создавая для них так называемую прикладную программную среду. Функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования – **API**.

Для обеспечения высокой скорости работы ОС все модули ядра или их большая часть постоянно находятся в оперативной памяти, то есть являются *резидентными*.

Остальные модули ОС выполняют полезные, но менее обязательные функции. Их обычно подразделяют на следующие группы: утилиты; системные обрабатываемые программы; программы представления пользователю дополнительных услуг; библиотеки процедур различного назначения, упрощающие разработку приложений.

Модули ОС, оформленные в виде утилит, системных обрабатываемых программ и библиотек, обычно загружаются в оперативную память только на выполнение своих функций, т.е. являются *транзитными*.

Поскольку ядро представляет собой сложный многофункциональный комплекс, то многослойный подход распространяется и на структуру ядра.

Ядро может состоять из следующих слоев: средства аппаратной поддержки ОС, машинно-зависимые компоненты ОС, базовые механизмы ядра, менеджеры ресурсов, интерфейс системных вызовов.

Выводы

Классическая структуризация ОС заключается в разбиении всех компонентов на слои: ядро и вспомогательные модули.

Модули ядра непосредственно взаимодействуют с аппаратурой компьютера и недоступны для приложений.

Многослойный подход распространяется и на структуру ядра. Взаимодействие слоев четко формализовано, а связи внутри слоя могут быть произвольными.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины.

Рекомендованная основная литература пп.1,2.

Тема лекции 4 Микроядерная архитектура операционной системы

План лекции

1. Микроядерная архитектура
2. Пример микроядерной операционной системы
3. Сравнение макроядерной и микроядерной архитектуры

Цель – изучение и первичное закрепление знаний. Актуализация ведущих знаний.

Задачи – формирование знаний по организации архитектуры современных операционных систем.

Ключевые вопросы – микроядро, серверы операционной системы, операционная система реального времени QNX.

В микроядерной архитектуре в привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром. В его состав входят машинно-зависимые модули, выполняющие базовые функции: управление виртуальной памятью; поддержка заданий и потоков; взаимодействие между процессами; управление вводом-выводом и прерываниями; сервисы хоста (главного компьютера или компьютера, имеющего IP-адрес) Все остальные высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме.

Менеджеры ресурсов, вынесенные в пользовательский режим, называются серверами ОС. Важнейшей задачей реализации микроядерной структуры является наличие удобного и эффективного механизма вызова процедур одного из процессов из другого.

Наиболее ярким представителем микроядерных ОС является ОС реального времени QNX. Здесь микроядро обеспечивает всего лишь пару десятков системных вызовов, но зато может быть целиком размещено во внутреннем кэше.

Микроядерные ОС удовлетворяют большинству требований, предъявляемым к современным ОС, обладая переносимостью, расширяемостью, надежностью, но за эти достоинства приходится платить снижением производительности, что является основным недостатком микроядерных ОС. Модель с микроядерной архитектурой подходит для поддержки распределенных вычислений, так как использует механизмы, аналогичные сетевым: взаимодействие клиентов и серверов путем обмена сообщениями. Серверы микроядерной ОС могут работать как на одном, так и на разных компьютерах. Однако микроядро за счет своей оптимизации позволяет обеспечить характеристики реального времени.

Недостатком монолитной ОС является сложность удаления одного из уровней; добавление новых функций требует хорошего знания всей архитектуры ОС. В монолитных ОС все компоненты ядра работают в едином адресном пространстве, что может привести к конфликтам между его частями и сложностью подключения новых драйверов. В микроядерной архитектуре каждый компонент представляет собой самостоятельный процесс.

Плодотворным является подход, основанный на модели клиент-сервер. Однако использование данной технологии не гарантия того, что ОС станет микроядерной (например, Windows NT).

Микроядерные ОС нынче разрабатываются чаще.

Выводы

Микроядерная архитектура является альтернативой классическому способу построения ОС.

Производительность микроядерной операционной системы всегда ниже макроядерной.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2.

Тема лекции 5 Стандарты и интерфейсы операционных систем

План лекции

1. Определение понятий «интерфейс»
2. Определение понятия «стандарт»
3. Обзор существующих стандартов ОС

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задачи – знакомство с основами стандартизации современных операционных систем.

Ключевые вопросы – проблемы и необходимость стандартизации интерфейсов операционных систем, примеры стандартов.

Под интерфейсами ОС понимают специальные интерфейсы системного и прикладного программирования (API).

Термин API делят на следующие направления:

- API как интерфейс высокого уровня, принадлежащий к библиотекам RTL;
- API прикладных и системных программ, входящий в состав в ОС;
- прочие интерфейсы API.

Программный интерфейс API включает не только сами функции, но и соглашения об их использовании, которые регламентируются самой ОС, архитектурой целевой вычислительной системы и системой программирования. Существует несколько вариантов реализации функций API: на уровне модулей ОС; на уровне систем программирования; на уровне внешней библиотеки процедур и функций.

Возможности API оценивают с позиций эффективности выполнения функций API, широты предоставляемых возможностей, зависимости прикладной программы от архитектуры целевой вычислительной системы.

Как правило, функции API не стандартизированы. В каждом конкретном случае набор вызовов API определяется архитектурой ОС. Принимаются попытки стандартизировать некоторый набор функций.

В последние годы большую популярность получили графические интерфейсы GUI (Graphical User Interface), которые являются частным случаем задачи управления вводом-выводом и не относятся к функциям ядра ОС, хотя в ряде случаев разработчики ОС относят GUI к основному системному интерфейсу API.

Частным случаем попытки стандартизировать API является внутренний корпоративный стандарт компании Microsoft – WinAPI. Он включает реализации Win16, Win32s, Win32, WinCE.

Примером стандартизации API служит один из самых распространенных стандартов – POSIX. POSIX – независимый от платформы системный интерфейс для компьютерного окружения – является стандартом IEEE. Он описывает системные интерфейсы для открытых ОС, в том числе оболочки, утилиты, инструментарию, задачи обеспечения безопасности, сетевые функции и обработку транзакций. Стандарт базируется на UNIX-системах, но допускает реализацию и в других ОС.

Этот стандарт подробно описывает систему виртуальной памяти, многозадачность и технологию переноса ОС. Он представляет собой множество стандартов POSIX.1 – POSIX.12.

Выводы

Стандартизация является важным этапом в развитие современного программного обеспечения.

Функции API, как правило, не стандартизированы.

Ярким примером стандартизации API служит один из самых распространенных стандартов – POSIX.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2, интернет-источники.

Тема лекции 6 Процессы и потоки

План лекции

1. Понятие «процесс», понятие «поток», их взаимосвязь
2. Дескриптор и контекст процесса
3. Планирование и диспетчеризация потоков

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение важных понятий, определяющих внутренние механизмы операционной системы.

Ключевые вопросы – реализация мультипрограммирования, внутренние единицы работы, задачи планирования и диспетчеризации потоков

Для реализации мультипрограммирования ОС должна определить и оформить для себя те внутренние единицы работы, между которыми будет разделяться процессор и другие ресурсы компьютера. В настоящее время в большинстве ОС определены два типа единиц работы. Более крупная единица работы, носящая название *процесса*, или *задачи*, требует для своего выполнения нескольких более мелких работ, для обозначения которых используют термины «*поток*» или «*нить*».

Процесс рассматривается ОС как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Процессорное время распределяется ОС между другими единицами работы – *потоками*, представляющими собой последовательности команд. Потоки возникли в ОС как средство распараллеливания вычислений, облегчающее работу программиста.

Создать процесс означает создать *описатель процесса*, в качестве которого выступает одна или несколько информационных структур, содержащих все сведения о процессе, необходимые операционной системе для управления им.

Дескриптор процесса содержит информацию, необходимую ядру в течение всего жизненного цикла процесса, независимо от того находится он в пассивном или активном состоянии, находится его образ (совокупность его кодов и данных) в оперативной памяти ли выгружен на диск. Дескрипторы отдельных процессов объединены в список, образующий таблицу процессов. Память для таблицы процессов отводится динамически в области ядра. В дескрипторе содержится информация о состоянии процесса, о расположении образа процесса в оперативной памяти и на диске, о значении отдельных составляющих приоритета, а также о его итоговом значении – глобальном приоритете, об идентификаторах пользователя, создавшего процесс, о родственных процессах и другая информация.

Контекст процесса содержит менее оперативную, но более объемную часть информации, необходимую для возобновления выполнения процесса с прерванного места: содержимое регистров процессора, информация о незавершенных операциях ввода-вывода и др.

Контекст процесса также доступен только программам ядра, однако он хранится не в области ядра, а непосредственно примыкает к образу процесса и перемещается вместе с ним при необходимости из оперативной памяти на диск.

Выводы

Важнейшей функцией ОС является организация рационального использования всех аппаратных и информационных ресурсов. Задачи управления ресурсами более сложны в мультипрограммных ОС.

Чтобы поддерживать мультипрограммирование, ОС определяет внутренние единицы работы, между которыми разделяются все ресурсы компьютера – процесс и поток.

Переход от выполнения одного потока к другому осуществляется в результате планирования и диспетчеризации.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2.

Тема лекции 7 Алгоритмы планирования потоков, основанные на квантовании

План лекции

1. Состояния потока
2. Классификация алгоритмов планирования потоков
3. Алгоритмы, основанные на квантовании

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

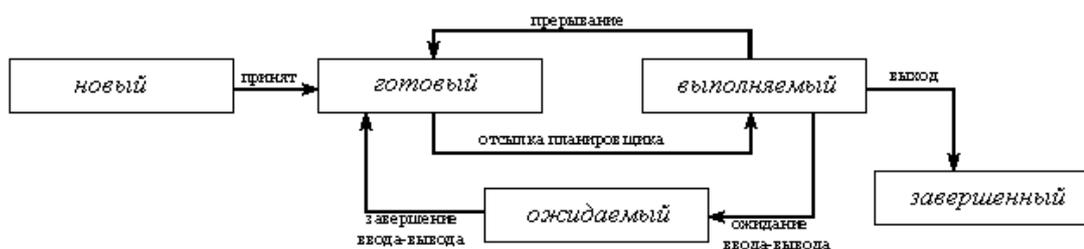
Задача – изучение важных понятий, реализующих внутренние механизмы операционной системы.

Ключевые вопросы – активные и пассивные состояния потоков, очереди потоков, вытесняющие и невытесняющие алгоритмы планирования.

Текстовый материал лекции

ОС выполняет планирование потоков, принимая во внимание их состояние. В мультипрограммной системе различают три состояния потока:

- выполнение – активное состояние потока, во время которого поток обладает всеми необходимыми ресурсами и непосредственно выполняется процессом;
- ожидание – пассивное состояние потока, находясь в котором, поток заблокирован по своим внутренним причинам;
- готовность – пассивное состояние процесса, но при этом поток заблокирован в связи с внешними обстоятельствами.



На рисунке приведена типовая диаграмма переходов для состояний процессора. В течение жизни поток переходит из одного состояния в другое в соответствии с алгоритмом планирования.

С самых общих позиций все множество алгоритмов планирования делят на классы: *невытесняющие* алгоритмы – в которых активному потоку позволяется выполнять, пока он по собственной инициативе не отдаст управление операционной системе для того, чтобы она выбрала другой готовый поток;

вытесняющие алгоритмы – в которых решение о переключении процессора с выполнения одного потока на другой принимается ОС.

В основе многих вытесняющих алгоритмов лежит концепция *квантования*, в которой каждому потоку поочередно для выполнения предоставляется ограниченный непрерывный период процессорного времени *квант*

Поток, который исчерпал свой квант времени, переводится в состояние готовности, а на выполнение в соответствии с определенным правилом выбирается новый поток из очереди готовых.

Выводы

Процессорное время распределяется ОС между потоками, а остальные ресурсы компьютера – между процессами.

Операционная система для реализации мультипрограммирования выполняет планирование и диспетчеризацию потоков.

При применении вытесняющих алгоритмов ОС получает полный контроль над вычислительным процессом.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2, интернет-источники.

Тема лекции 8 Алгоритмы приоритетного планирования потоков. Смешанные алгоритмы планирования

План лекции

4. Состояния потока
5. Классификация алгоритмов планирования потоков
6. Алгоритмы, основанные на квантовании

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение важных понятий, реализующих внутренние механизмы операционной системы.

Ключевые вопросы – привилегии и приоритеты, факторы, влияющие на назначение приоритетов, статические и динамические приоритеты, относительные и абсолютные приоритеты.

Другой важной концепцией, лежащей в основе многих вытесняющих алгоритмов, является *приоритетное обслуживание*. *Приоритет* – это число, характеризующее степень привилегированности потока при использовании ресурсов вычислительной машины, в частности процессорного времени: чем выше приоритет, тем выше привилегии, тем меньше времени будет проводить поток в очередях.

Приоритеты определяются исходя из совокупности внутренних и внешних по отношению к операционной системе факторов. Внутренние факторы: требования к памяти; количество открытых файлов; отношение среднего времени ввода-вывода к среднему времени CPU и др. Внешние факторы: важность процесса; тип и величина файлов, используемых для оплаты; отделение, выполняющее работы и др.

Существует две разновидности приоритетного планирования: обслуживание с относительными приоритетами и обслуживание с абсолютными приоритетами. В обоих случаях выбор потока на выполнение из очереди готовых процессов осуществляется одинаково: по наивысшему приоритету. Однако проблема определения момента смены активного потока решается по-разному. В системах с относительными приоритетами активный поток выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние ожидания или готовности.

В ОС также широко применяются алгоритмы планирования смешанной структуры. Например, в основе планирования лежит квантование, но величина кванта и/или порядок выбора потока из очереди готовых определяется приоритетами потоков. Примером служит ОС Windows NT, в которой квантование сочетается с динамическими абсолютными приоритетами. На выполнение выбирается готовый поток с наивысшим приоритетом. Ему выделяется квант времени. Если во время выполнения в очереди готовых процессов появляется поток с более высоким приоритетом, то он вытесняет выполняемый поток. Вытесненный поток возвращается в очередь готовых, причем он становится впереди всех потоков, имеющих такой же приоритет.

Выводы

Приоритетное планирование относится к вытесняющим алгоритмам планирования.

Выделяют относительные и абсолютные приоритеты.

Недостатком приоритетного планирования является возможность блокировки на долгое время низкоприоритетных потоков.

В современных ОС часто используются смешанные алгоритмы планирования потоков.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2, интернет-источники.

Тема лекции 9 Синхронизация процессов и потоков

План лекции

1. Средства межпроцессного взаимодействия
2. Понятие критической секции и блокирующей переменной
3. Семафоры Дийкстры
4. Тупики и методы борьбы с тупиками

Цель - организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение функций операционной системы по управлению памятью и особенностям их реализации.

Ключевые вопросы - методы синхронизации, гонки, тупики

В мультипрограммных ОС потребность в синхронизации процессов и потоков связана с совместным использованием аппаратных и информационных ресурсов. Средства ОС, обеспечивающие эту синхронизацию, часто называются средствами межпроцессного взаимодействия. – IPC (Inter Process Communication).

Выполнение потоков в мультипрограммной среде всегда происходит асинхронно, независимо друг от друга. Моменты прерывания потоков, время нахождения их в очередях, порядок выбора потоков для выполнения – случайные события, зависящие от стечения многих обстоятельств.

Важным понятием синхронизации потоков является понятие «*критической секции*» программы. Это часть программы, результат выполнения которой может непредсказуемо меняться, если переменные, относящиеся к этой части программы, изменяются другими потоками в то время, когда выполнение этой части еще не завершено.

Ситуации, в которых несколько потоков обрабатывают разделяемые данные и конечный результат зависит от соотношения скоростей потоков, называются *гонками*.

Чтобы исключить эффект гонок по отношению к критическим данным, необходимо обеспечить, чтобы в каждый момент времени в критической секции, связанной с этими данными, находился только один поток. При этом неважно, в активном состоянии он находится или в пассивном. Этот прием называют *взаимным исключением*. Для синхронизации потоков одного процесса прикладной программист может использовать глобальные *блокирующие переменные*. Каждому набору критических данных ставится в соответствие двоичная переменная, которой поток присваивает значение равное нулю, когда он входит в критическую секцию, и значение единица, если он ее покидает.

Обобщением блокирующих переменных являются так называемые *семафоры Дийкстры*, хранящие целые неотрицательные значения.

При параллельном выполнении двух процессов могут возникать также тупиковые ситуации, когда два или более процессов блокируют друг друга, вынуждая ожидать события, связанного с освобождением ресурса. Тупики чаще возникают из-за конкуренции несвязанных параллельных процессов за ресурсы, реже из-за ошибок программирования.

Существуют формальные, программно-реализованные методы распознавания тупиков, основанные на ведении таблиц распределения ресурсов и таблиц запросов к занятым ресурсам.

Выводы

Для синхронизации потоков прикладных программист может использовать либо собственные средства и приемы синхронизации, либо средства операционной системы.

К ним относятся блокирующие переменные, семафоры, мьютексы.

В случае возникновения тупика ОС должна предоставлять администратору средства распознавания тупика, и после его диагностики средства снятия взаимных блокировок и восстановления нормального функционирования системы.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1,2, интернет-источники.

Тема лекции 10 Подсистема управления памятью.

План лекции

1. Функции управления оперативной памятью в мультипрограммных операционных системах
2. Виды адресов
3. Виртуальное адресное пространство процессов
4. Виды структуризации виртуального адресного пространства

Цель - организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение функций операционной системы по управлению памятью и особенностей структуризации виртуального адресного пространства.

Ключевые вопросы – символьный, виртуальный и физический адреса, структуризация виртуального адресного пространства, свопинг и виртуальная память, страничный, сегментный и сегментно-страничный механизмы виртуальной памяти.

Функциями ОС по управлению памятью в мультипрограммной системе являются: отслеживание свободной и занятой памяти, выделение памяти процессам и ее освобождение по завершении процессов, вытеснение кодов и данных из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их обратно, когда в них освобождается место,

настройка адресов программы на конкретную область физической памяти.

защита памяти, которая состоит в том, чтобы не позволить выполняемому процессу записывать и читать данные другого процесса.

Для идентификации переменных и программ на разных этапах жизненного цикла программы используются различные имена: символьные, виртуальные и физические. Совокупность виртуальных адресов процесса называется виртуальным адресным пространством. Каждый процесс имеет свое адресное виртуальное пространство. Совпадение виртуальных адресов переменных и команд различных процессов не приводит к конфликтам, поскольку, когда переменные одновременно присутствуют в памяти, ОС отображает их на разные физические адреса.

Для обеспечения приемлемого уровня мультипрограммирования используется методы, в которых образы некоторых процессов целиком или полностью выгружаются на диск – свопинг и виртуальная память.

Ключевой проблемой виртуальной памяти является преобразование виртуальных адресов в физические. Ее решение зависит от способа структуризации виртуального адресного пространства, который реализуется методами: страничная виртуальная память; сегментная виртуальная память; сегментно-страничная организация памяти.

Для временного хранения сегментов и страниц на диске отводится специальная область – страничный файл.

Выводы

Оперативная память – важнейший ресурс вычислительной системы, требующий тщательного управления со стороны операционной системы.

Виртуальная память использует дисковую память для временного хранения не помещающихся в оперативную память данных и кодов выполняющихся процессов.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература пп.1. Интернет-источники.

Тема лекции 11 Классификация методов распределения памяти. Обзор методов без использования внешней памяти

План лекции

1. Классификация методов распределения памяти
2. Распределение памяти фиксированными разделами

3. Распределение памяти динамическими разделами
4. Фиксированными разделами для распределения памяти

Цель - организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение функций операционной системы по управлению памятью и особенностей их реализации.

Ключевые вопросы – фиксированные, динамические, перемещаемые разделы.

Текстовый материал лекции

Общая классификация методов распределения памяти приведена на рисунке. Некоторые из них сохранили актуальность и широко используются в современных ОС, другие представляют собой познавательный интерес, но их можно встретить до сих пор в специализированных системах.



Простейший способ управления оперативной памятью заключается в том, что память разбивается на несколько областей фиксированной величины, называемых разделами. Границы разделов не изменяются и могут быть сделаны вручную оператором во время старта.

При распределении памяти динамическими разделами память машины заранее не делится на части. Сначала вся память, отведенная для приложений, свободна. Каждому вновь поступающему на выполнение приложению на этапе создания процесса выделяется вся необходимая ему память. Если достаточный объем памяти отсутствует, то приложение не принимается, и процесс не создается. После завершения процесса память освобождается, и на это место может быть загружен другой процесс. Таким образом, в любой момент времени ОП представляет собой случайную последовательность занятых и свободных участков произвольного размера. Существенный недостаток – фрагментация памяти. Данный метод лежит в основе подсистем управления памятью многих мультипрограммных ОС 60-70-х годов, в частности OS/360.

Одним из способов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших или младших адресов так, чтобы вся свободная память образовывала единую свободную память. Эта процедура называется сжатием. Сжатие может выполняться либо при каждом завершении процесса, либо только тогда, когда вновь созданному процессу недостаточно места в свободном разделе. В первом случае требуется меньше вычислительной работы при корректировке таблиц, а во втором – реже выполняется процедура сжатия.

Выводы

Все множество алгоритмов распределения памяти делится на два класса без использования внешней памяти и с ее использованием.

Алгоритмы первого класса использовались в ранних мультипрограммных системах. Алгоритмы второго класса актуальны.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1, интернет-источники.

Тема лекции 12 Страничное распределение памяти

План лекции

1. Свопинг и виртуальная память
2. Структуризации виртуального адресного пространства в страничной организации памяти
3. Сегментное распределение
4. Достоинства и недостатки каждого сегментного распределения памяти

Цель - организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение функций операционной системы по управлению памятью и особенностям их реализации.

Ключевые вопросы – свопинг и виртуальная память, таблица страниц, страничное прерывание, критерий выгрузки страниц

Для обеспечения приемлемого уровня мультипрограммирования используется методы, в которых образы некоторых процессов целиком или полностью выгружаются на диск – свопинг и виртуальная память.

Ключевой проблемой виртуальной памяти является преобразование виртуальных адресов в физические. Ее решение зависит от способа структуризации виртуального адресного пространства, который реализуется методами: страничная виртуальная память; сегментная виртуальная память; сегментно-страничная организация памяти.

При страничном распределении виртуальное адресное пространство каждого процесса делится на части одинакового фиксированного для данной системы размера, называемые *виртуальными страницами*. Вся оперативная память машины также делится на части того же размера, называемые *физическими страницами*. Для каждого процесса ОС создает *таблицу страниц*. При каждом обращении к памяти выполняется поиск номера виртуальной страницы, содержащей требуемый адрес. Если нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое *страничное прерывание*. Выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди процессов, находящихся в состоянии готовности. Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее. Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, иначе на основании принятой в данной системе стратегии замещения страниц решается вопрос о том, какую страницу следует выгрузить из оперативной памяти.

Деление виртуального адресного пространства на сегменты осуществляется компилятором на основе указаний программиста или, по умолчанию, в соответствии с принятыми в системе соглашениями. Размер сегмента определяется с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п.

Выбор оптимального размера страницы является сложной оптимизационной задачей, требующей учета множества факторов. Типичный размер страницы в процессорах x86 и Pentium поддерживают страницы размером 4096 байт (4 Кбайт).

Выводы

Подмена (*виртуализация*) оперативной памяти дисковой памятью позволяет повысить уровень мультипрограммирования.

Она осуществляется совокупностью программных модулей ОС и аппаратных схем процессора.

Достоинства страничной виртуальной памяти – высокая скорость обмена, низкий уровень фрагментации, недостатки – сложность организации защиты данных, разделенных на части механически.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1, интернет-источники.

Тема лекции 13 Сегментная и сегментно-страничные способы организации памяти.

План лекции

1. Сегментное распределение памяти.
2. Сегментно-страничное распределение памяти.
3. Механизм преобразования виртуального адреса в физический при сегментно-страничной организации памяти.

Цель - организация целенаправленной познавательной деятельности студентов по овладению программным материалом.

Задача – изучение функций операционной системы по управлению памятью и особенностям их реализации.

Ключевые вопросы – сегменты, базовый линейный виртуальный адрес.

Деление виртуального адресного пространства на сегменты осуществляется компилятором на основе указаний программиста или, по умолчанию, в соответствии с принятыми в системе соглашениями. Размер сегмента определяется с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п.

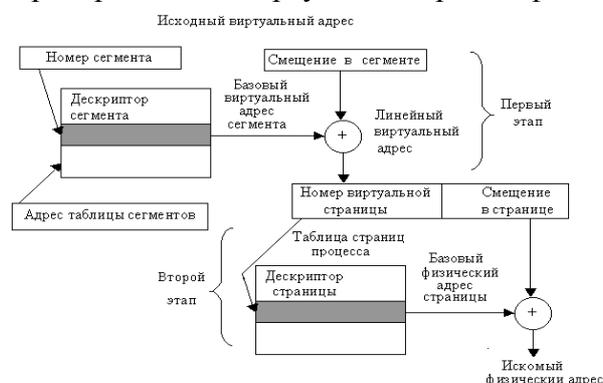
При загрузке процесса в ОП помещается только часть его сегментов, полная копия виртуального адресного пространства находится в дисковой памяти. Для каждого загружаемого сегмента ОС подыскивает непрерывный участок свободной памяти подходящего размера. Смежные в виртуальной памяти сегменты одного процесса могут занимать в ОП несмежные участки. Если во время выполнения процесса происходит обращение по виртуальному адресу, содержащемуся в сегменте, отсутствующему в памяти, происходит прерывание. ОС приостанавливает процесс, запускает на выполнение следующий процесс из очереди и параллельно организует загрузку нужного сегмента с диска. При отсутствии места ОС выбирает сегмент на выгрузку, используя критерии аналогичные критериям выбора страниц при страничном распределении.

Сегментно-страничное распределение представляет собой комбинацию страничного и сегментного механизмов управления памятью и направлен на реализацию достоинств обоих подходов.

Виртуальное адресное пространство процесса разделено на сегменты так же как при сегментной организации памяти, что позволяет определять разные права доступа к разным частям кодов и данных программы.

Перемещение же данных между памятью и диском осуществляется не сегментами, а страницами. Для этого каждый виртуальный сегмент и физическая память делятся на страницы равного размера, что позволяет более эффективно использовать память, сократив до минимума фрагментацию.

Преобразование виртуально адреса в физический происходит в два этапа:



Выводы

Недостатки сегментного распределения памяти – избыточность – перемещения между памятью и диском составляет сегмент, имеющий объем больше страницы и е фрагментация, возникающая из-за непредсказуемости размеров сегмента.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1. Интернет-источники.

Тема лекции 14. Файловая система

План лекции

1. Понятия «файловая система», «система управления файлами»
2. Функции файловой системы
3. Типы файлов

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом, расширение кругозора по специальности.

Задача – изучение особенностей построения файловых систем.

Ключевые вопросы - логическая и физическая организация файловой системы, современные файловые системы

ОС подменяет физическую структуру хранящихся данных некоторой удобной для пользователя логической моделью, базовым элементом которой является файл.

Файловая система включает:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих различные операции над файлами, такие как создание, удаление, чтение и т.д.

Специальное системное программное обеспечение, реализующее работу с файлами по принятым спецификациям файловой системы, часто называют *системой управления файлами*. Она отвечает за создание, уничтожение, организацию, чтение, запись, модификацию и перемещение файловой информации, а также за управление доступом к файлам и за управление ресурсами, которые используются файлами. Для работы с файлами, организованными в соответствии с некоторой файловой системой, для каждой операционной системы разрабатывается система управления файлами.

Для логической организации файлов используют каталоги - файлы, содержащие информацию о входящих в него файлах. Все каталоги, находящиеся на носителе, образуют иерархическую структуру, которая может быть древовидной или сетевой.

Специальные файлы – фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к файлам и внешним устройствам. Они позволяют выполнять операции ввода-вывода посредством обычных команд записи в файл или вывода из файла.

Выводы

Файловая система представляет собой комплекс системных программных средств, реализующих различные операции с файлами.

Физическая организация файловой системы подразумевает способы размещения и адресации отдельных частей файлов в разделах и секторах дисковой памяти.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1. Интернет-источники.

Тема лекции 15. Файловые системы FAT, NTFS

План лекции

1. Файловая система FAT
2. Файловая система NTFS

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом, расширение кругозора по специальности.

Задача – изучение особенностей построения современных файловых систем.

Ключевые вопросы - логическая и физическая организация файловой системы, современные файловые системы

Текстовый материал лекции

Файловая система FAT получила свое название благодаря простой таблице, в которой указываются:

- непосредственно адресуемые участки логического диска, отведенные для размещения в них файлов или их фрагментов;
- свободные области дискового пространства;
- дефектные области диска (эти области содержат дефектные участки и не гарантируют чтение и запись данных без ошибок).

Файловая система FAT имеет множество реализаций как система управления файлами.

В файловой системе FAT дисковое пространство любого логического диска делится на две области: *системную область* и *область данных*. Системная область состоит из компонентов (расположенных в логическом адресном пространстве друг за другом): загрузочной записи; зарезервированных секторов; таблицы размещения файлов; корневого каталога.

Вся область данных разбивается на *кластеры*. Кластер — это минимальная адресуемая единица дисковой памяти, выделяемая файлу (или некорневому каталогу).

В связи с чрезвычайной важностью таблицы FAT она обычно хранится в двух идентичных экземплярах, второй из которых непосредственно следует за первым. Обновляются копии FAT одновременно, используется же только первый экземпляр. Если он по каким-либо причинам окажется разрушенным, то произойдет обращение ко второму экземпляру.

NTFS (New Technology File System) — файловая система новой технологии. Действительно, файловая система NTFS по сравнению FAT16 (и даже FAT32) содержит ряд значительных усовершенствований и изменений. С точки зрения пользователей файлы по-прежнему хранятся в каталогах (называемых *папками* (folders)). Однако в ней появилось много новых особенностей и возможностей: надежность, ограничения доступа к файлам и каталогам, расширенная функциональность, поддержка дисков большого объема.

Одним из основных понятий, используемых при работе с NTFS, является понятие *тома* (volume). Том означает логическое дисковое пространство, которое может быть воспринято как логический диск, то есть том может иметь букву (буквенный идентификатор) диска. Частным случаем тома является логический диск. Файловая система NTFS поддерживает размеры кластеров от 512 байт до 64 Кбайт; неким стандартом же считается кластер размером 2 или 4 Кбайт.

Все дисковое пространство в NTFS делится на две неравные части. Первые 12 % диска отводятся под так называемую зону MFT (главная таблица файлов). Остальные 88 % тома представляют собой обычное пространство для хранения файлов.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1, интернет-источники.

Тема лекции 16 Средства защиты от сбоев и несанкционированного доступа

План

1. Основные концепции ввода-вывода
2. Режимы управления ввода-вывода
3. Закрепление устройств ввода-вывода
4. Основные системные таблицы ввода-вывода

Цель – организация целенаправленной познавательной деятельности студентов по овладению программным материалом, расширение кругозора по специальности.

Задача – изучение особенностей построения современных операционных систем.

Ключевые вопросы - сбои оборудования, искажение информации, нелегальный

доступ

Конструкторы и разработчики аппаратного и программного обеспечения прилагают немало усилий, чтобы обеспечить защиту информации:

от сбоев оборудования;

от случайной потери или искажения информации, хранящейся в компьютере;

от преднамеренного искажения, производимого, например, компьютерными вирусами;

от несанкционированного (нелегального) доступа к информации (её использования, изменения, распространения).

К многочисленным, далеко не безобидным ошибкам компьютеров добавилась и компьютерная преступность, грозящая перерасти в проблему, экономические, политические и военные последствия которой могут стать катастрофическими.

При защите информации от сбоев оборудования используются следующие основные методы:

периодическое архивирование программ и данных. Причем, под словом «архивирование» понимается как создание простой резервной копии, так и создание копии с предварительным сжатием (компрессией) информации.;

автоматическое резервирование файлов. Если об архивировании должен заботиться сам пользователь, то при использовании программ автоматического резервирования команда на сохранение любого файла автоматически дублируется и файл сохраняется на двух автономных носителях (например, на двух винчестерах). Выход из строя одного из них не приводит к потере информации. Резервирование файлов широко используется, в частности, в банковском деле.

Защита от случайной потери или искажения информации, хранящейся в компьютере, сводится к следующим методам:

автоматическому запросу на подтверждение команды, приводящей к изменению содержимого какого-либо файла;

установке специальных атрибутов документов;

возможности отменить последние действия;

разграничению доступа пользователей к ресурсам файловой системы, строгому разделению системного и пользовательского режимов работы вычислительной системы.

Ссылки на литературные источники, приведенные в рабочей программе дисциплины - рекомендованная основная литература п.1, интернет-источники.

Методические указания

Методические указания построения лекций

Содержание лекций должно:

обеспечивать постановку цели и задач курса и строиться с учетом его методологических основ;

включать характеристики основных понятий по изучаемой дисциплине;

учитывать основные направления в развитии представляемой научной дисциплины в решении актуальных проблем;

отражать системные законы и закономерности, принципы, лежащие в основе изучаемого предмета, актуальные вопросы данного курса;

строиться с позиций анализа конкретных сведений, фактов и явлений, иллюстрирующих основные теоретические положения в их взаимосвязи.

Лекция – ведущая форма учебного процесса. Слушая лекцию и конспектируя материал, студенты включаются в мыслительную деятельность, приобретают навыки делать выводы.

Для наибольшей эффективности усвоения материала студентам рекомендуется, ознакомившись с планом лекций, подготовиться к ней, прочитав соответствующий раздел учебника и проработать конспект после прослушивания лекции, выделив в нем наиболее важные моменты.

Методические указания по изучению дисциплины

В результате освоения дисциплины у студентов развивается способность оценивать и выбирать современные операционные среды и информационно-коммуникационные технологии для информатизации и автоматизации решения прикладных задач и создания ИС. Для успешного освоения данной дисциплины студентам предлагаются:

- содержание разделов и тем дисциплины, включающей лекционные и лабораторные занятия;
- контрольные вопросы для самостоятельной работы;
- список рекомендуемой основной и дополнительной литературы;
- вопросы к экзамену.

В течении семестра студент не только должен изучать материал лекций, но и готовить вопросы самостоятельной работы на основе рекомендуемой литературы.

Основными формами самостоятельной (внеаудиторной) работы студентов являются:

подготовка отдельных вопросов по темам программы;

участие в научных и научно-практических студенческих конференциях;

подготовке к лабораторной работе по определенной теме;

подготовка и написание отчетов к лабораторным работам;

подготовка к промежуточному и итоговому контролю.

Самостоятельная работа начинается до прихода студента на лекцию. Весьма эффективно использование «системы опережающего чтения», т.е. предварительного прочтения лекционного материала, содержащегося в учебниках, учебных пособиях, в результате чего закладывается база для более глубокого восприятия лекции.

В процессе организации самостоятельной работы большое значение имеют консультации преподавателя, в ходе которых решаются многие проблемы изучаемого курса, уясняются наиболее сложные вопросы.

Контроль самостоятельной работы осуществляется тестированием, которое может проводиться в письменной форме, либо с использованием компьютерной системы тестирования. Примерные тестовые задания приведены в рабочей программе. Тест включает в

себя вопросы с открытыми и закрытыми вариантами ответов. Результаты тестирования включены в балльно-рейтинговую систему оценки знаний (см. рабочую программу).

Зачет является завершающим этапом учебного процесса, на котором проводится подведение итогов всей самостоятельной работы студентов.

Подготовку к экзамену требуется начинать с просмотра перечня всех вопросов с целью оценки требуемого объема учебного материала, логики и структуры построения курса. С учетом накопленных за семестр знаний студент должен запланировать распределение времени на подготовку. Желательно зарезервировать время для повторения материала. Работа над каждым из вопросов рекомендуется прочитать конспект лекции, дополнительно прочитать рекомендованный учебник, если материал трудно усваивается. Завершается работа восстановлением в памяти прочитанного.

Экзаменационный билет включает в себя два теоретических вопроса из перечня, и практическое задание, предполагающее знание основ работы в операционных системах Windows XP и Linux.

При оценке на экзамене учитывается: полнота ответа на поставленный вопрос, точность формулировок, логичность ответа, умение делать выводы, выявлять закономерности, соблюдение норм литературной речи и использования специализированной терминологии. Высшего балла заслуживает ответ, удовлетворяющий всем этим требованиям.

Методические указания к лабораторным работам

Лабораторные работы проводятся по подгруппам в компьютерном классе. На первом занятии обязательно проводится инструктаж по выполнению техники безопасности.

Задания к лабораторным работам выполняются студентами парами на одном компьютере (работа в команде).

Желательно готовиться к лабораторным работам заранее по перечню тем, выданным преподавателем.

Выполняя задание, студенты пользуются материалом, изложенным в тексте лабораторной работы; готовят письменный отчет, включающий краткое изложение проделанных действий, ответы на контрольные вопросы, выводы.

Преподаватель, принимая лабораторную работу, проверяет навыки, полученные студентами при выполнении задания, отчет, задает дополнительные вопросы по отчету.

Технология выполнения лабораторных работ

Тема Загрузка операционной системы WINDOWS XP. Структура оперативной памяти. Распределение ресурсов компьютера

Аудиторные занятия 2 часа Самостоятельная работа 4 часа

Задание

1. Проследить и описать предшествующую загрузке операционной системы информацию.
2. Описать процедуру загрузки операционной системы WINDOWS XP.
3. Получить отчет об устройствах компьютера и распределении совместно используемых ими ресурсов
5. Составить полный отчет в тетради.

Для выполнения задания необходимо перевести и пояснить все термины, участвующие в загрузочной информации.

Тема Служебные программы WINDOWS XP

Аудиторные занятия 2 часа Самостоятельная работа 2 часа

Задание

Научитесь выполнять следующие виды работ:

1. отслеживание производительности системы;
2. улучшение производительности жесткого диска путем его дефрагментации;
3. очистка диска;
4. назначения расписания для автоматического запуска на компьютере;
5. восстановление системы;
6. архивирование данных;
7. получение сведений о системе.

Тема Вирусы и антивирусные программные средства

Аудиторные занятия 2 часа Самостоятельная работа 4 часа

Задание

1. Дайте определение вируса и приведите способы классификации вирусов.
2. Выполните установку с диска антивирусной программы.
3. Установите автоматическое сканирование загружаемых дисков антивирусной программы (по требованию, по расписанию)
4. Произведите обновление антивирусной базы одновременно, по расписанию.
5. Приведите классификацию антивирусных программ.
6. Подготовьте обзор современных антивирусных средств.

Тема Работа в консоли DOS

Аудиторные занятия 2 часа Самостоятельная работа 2 часа

Задание

1. Выполнить в командной строке DOS следующие команды:
 - Открытие для редактирования документа
 - Удаление файла
 - Копирование файлов
 - Переименование всех файлов с указанными параметрами.

Тема Командные файлы

Аудиторные занятия 2 часа Самостоятельная работа 4 часа

Задание

Операционная система позволяет записать последовательность команд, которую необходимо повторять, чтобы осуществить периодически выполняемые действия, в специальный файл, называемый командным файлом. Командный файл должен иметь расширение .bat. Последовательность команд, записанную в файл, можно выполнить набрав имя командного файла. Написать командный файл, который выполняет действия в соответствии с выданным вариантом.

Тема Определение режима доступа в Windows NT

Аудиторные занятия 2 часа Самостоятельная работа 2 часа

Задание

1. Проверьте правила доступа, которые определяют состояние разрешений доступа при перемещении и копировании объектов в NTFS. Сделайте выводы.
2. Создайте папку с программами и обеспечьте возможность их запуска, без возможности их копирования, перемещения и удаления.
3. Создайте папку для записи отчетов лабораторных работ студентов группы с возможностью записи в нее работ и невозможность чтения чужих работ.

Тема Утилиты ОС Windows

Аудиторные занятия 2 часа Самостоятельная работа 2 часа

Задание

1. Изучите работу утилиты системный монитор. Исследуйте загрузку центрального процессора и подсистемы управления памятью при запуске ресурсоемких приложений.

Тема Работа в VirtualBox

Аудиторные занятия 2 часа Самостоятельная работа 2 часа

Задание

VirtualBox позволяет запускать различные виды операционных непосредственно в рабочей операционной системе, в специальной среде, называемой "виртуальная машина". При этом физический компьютер, на котором проходит работа, обычно называют "главным компьютером" или "хостом", в то время как виртуальную машину часто называют "гостем" или "гостевой ОС". Изучите интерфейс VirtualBox, установите операционную систему по образцу в соответствии с вариантом.

Тема Работа в ОС Linux (установка, настройка, команды консоли, установка ПО)

Аудиторные занятия 6 часов Самостоятельная работа 24 часа

Задание

1. Выполните установку ОС Linux в качестве гостевой системы Virtual Box
2. Научитесь выполнять следующие команды в консоли:
 - Переключение на новую консоль,
 - Аутентификация и авторизация пользователя,
 - Монтирование и размонтирование привода CD-ROM,
 - Просмотр содержимого директории и содержимого выбранного файла,
 - Создание и удаление пользователя,
 - Поиск в файле,
 - Слияние файлов,
 - Запуск команд в фоновом режиме,
 - Принудительное завершение процессов.
3. Установите пакет в соответствии с вариантом (веб-сервер, редактор Tex, midnight commander, приложение для записи дисков, файловый сервер и т.д.)
4. Просмотреть в консоли состояние установленного пакета и всю доступную информацию
5. Выполнить те же действия, используя менеджер пакетов.
6. Установить SSH и подключиться к соседнему компьютеру
7. Выполнить через SSH следующие задания:
 - Изучить все конфигурации сетевого интерфейса ifcfg-ethN
 - Настроить доменное имя и resolv.conf (IP адрес не изменять)
 - Продемонстрировать перезагрузку сетевых настроек
 - Выполнить команду Ifconfig.
 - Вывести на экран все активные TCP соединения

Методические рекомендации по организации самостоятельной работы студентов

Общая структура самостоятельной работы представлена в рабочей программе и включает 82 часа, 46 из которых предназначены для подготовки к лабораторным работам, остальные для подготовки вопросов самостоятельного изучения. Вопросы даются в соответствии с лекционными занятиями.

Контрольные вопросы для самоподготовки студентов

1. Перечислите современные операционные системы?
2. Что обозначает термин «авторизация»? Что обозначает термин «аутентификация»?

Какая из этих операций выполняется раньше? Почему?

3. Что такое операционная среда? Перечислите известные операционные среды?
4. Изучите понятие «обработчик прерывания»
5. В чем заключаются векторный и опрашиваемый способы обработки прерываний?
6. Как реализуются дисциплины обслуживания прерываний?
7. С какой целью в ОС вводится специальный системный модуль, называемый супервизор прерываний?
8. Какие части ОС являются резидентными, какие нерезидентными?
9. Кто определяет количество слоев ядра?
10. Макроядерные или микроядерные ОС более производительны?
11. Изложите кратко содержимое стандартов POSIX.1 – POSIX.12.
12. Поясните термины «задача», «поток», «процесс». Как они соотносятся между собой?
13. Почему понятие процесс является одним из основных понятий ОС?
14. Как образуются очереди потоков?
15. Каковы возможные привилегии в обслуживании потоков, использующих свой квант неполностью?
16. Поясните термины «изменяемые и неизменяемые приоритеты»
17. Каковы недостатки приоритетного планирования?
18. Какие алгоритмы планирования смешанной структуры известны?
19. Сравните механизмы диспетчизации в системах Windows NT и Linux.
20. Каковы стратегии замещения страниц?
21. Изучите и сравните сквозную и обратную записи при использовании кэш.
22. Существует ли кэш для регистров процессора?
23. Сравните файловые системы FAT16 и FAT 32. В чем их достоинства и недостатки?
24. Каковы атрибуты файлов в системе NTFS?
25. Изучите основные метафайлы.

Контроль знаний

В качестве текущего контроля знаний выступают сданные лабораторные работы, а также тестирование.

Примерный вариант теста:

1. Выберите верное утверждение:

- 1) В мультипрограммных системах программы всегда выполняются в нескольких процессорах параллельно.
- 2) В мультипрограммных системах несколько программ выполняются в процессоре попеременно, совместно используя ресурсы компьютера.
- 3) Мультипроцессирование исключает мультипрограммирование.
- 4) В мультипрограммных системах несколько программ выполняются процессором одновременно.

2. Сколько состояний потока различают?

- 1) 2 2) 4 3) 6 4) 5

3. Операционная система выполняет планирование потоков в зависимости от их состояния. Состояние «ожидаемый» есть

- 1) активное состояние потока, в котором, обладая всеми ресурсами, поток ожидает завершения его выполнения процессором
- 2) пассивное состояние потока, при котором он заблокирован в связи с внешними обстоятельствами, например, выполнение более приоритетного потока
- 3) пассивное состояние потока, при котором он заблокирован по своим внутренним причинам – завершения ввода-вывода или освобождения некоторого ресурса
- 4) пассивное состояние вновь созданного потока

4. *Утилиты* - это
- 1) программы, реализующие специальный вариант пользовательского интерфейса
 - 2) программы, решающие отдельные задачи управления и сопровождения компьютерной системы
 - 3) библиотеки процедур различного назначения, упрощающие разработку приложений
 - 4) модули-приложения, обеспечивающие расширяемость операционной системы
5. В оперативной памяти хранятся постоянно
- 1) утилиты, системные обрабатывающие программы и ядро операционной системы
 - 2) только модули ядра операционной системы
 - 3) утилиты, системные обрабатывающие программы
 - 4) приложения пользователей
6. Внутренние прерывания возникают
- 1) при появлении аварийной ситуации в ходе исполнения некоторой инструкции программы, например, деление на нуль, обращение к несуществующему адресу
 - 2) при выполнении особой команды процессора для перехода на новую последовательность инструкций
 - 3) в результате действия пользователя или сигналов от аппаратных устройств
7. Совокупность виртуальных адресов процесса называется _____
8. Страничная виртуальная память организует перемещение данных между памятью и диском
- 1) частями виртуального адресного пространства произвольного размера, полученного с учетом смыслового значения данных
 - 2) частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера
 - 3) частями виртуального адресного пространства, фиксированного размера, которые в свою очередь, являются частью сегмента, полученного с учетом смыслового значения
 - 4) частями, размеры которых соответствуют виртуальным адресным пространствам процессов
9. Восстановите иерархию запоминающих устройств в порядке убывания объема, если обозначить через a - регистры процессора b - оперативная память c - быстродействующая память d - внешняя память
- 1) abcd
 - 2) acbd
 - 3) dbca
 - 4) dcba
10. *Операционная система* это
- 1) комплекс взаимосвязанных программ однородный по характеру и уровню
 - 2) это несвязный набор программных средств, предназначенных для управления загрузкой, запуском и выполнением других программ
 - 3) это комплекс взаимосвязанных программ, неоднородный по характеру и многоплановый по уровню
 - 4) несвязный набор программ, реализующий интерфейс между пользователем и аппаратурой компьютера
11. С самых общих позиций множество алгоритмов планирования потоков делится на 2 класса:
- 1) вытесняющие и невытесняющие
 - 2) статические и динамические
 - 3) приоритетные и беспriorитетные
 - 4) на квантовании и без квантования

12. Механизм прерываний поддерживается
- 1) только аппаратными средствами компьютера
 - 2) только программными средствами операционной системы
 - 3) программными средствами операционной системы и аппаратными средствами компьютера
13. Выберите неверное утверждение об ядре операционной системы:
- 1) модули ядра постоянно хранятся в оперативной памяти
 - 2) функции ядра, решающие внутрисистемные задачи, доступны для приложений
 - 3) в состав ядра входят функции, решающие внутрисистемные задачи организации вычислительного процесса
 - 4) функции ядра, вызываемые приложениями, образуют прикладную программную среду
14. Внутренние прерывания происходят _____ с выполнением программы (внутри тактов команды) при появлении аварийной ситуации в ходе выполнения некоторой инструкции.
15. Какая информационная структура содержит информацию, необходимую ядру в течение всего жизненного цикла процесса, независимо от того находится он в пассивном или активном состоянии, находится его образ (совокупность его кодов и данных) в оперативной памяти или выгружен на диск. _____
16. В системах с абсолютными приоритетами активный поток выполняется до тех пор, пока
- 1) он сам не покинет процессор, перейдя в состояние ожидания или готовности
 - 2) пока в очереди готовых потоков не появится поток с большим приоритетом
 - 3) пока в очереди готовых потоков не появится новый поток
 - 4) пока очередь готовых потоков не пуста
17. Для временного хранения сегментов и страниц на диске отводится специальный файл, который называют _____
18. К внутренним факторам, влияющим на значение приоритета относятся:
- 1) требования к памяти и количество открытых файлов
 - 2) отношение среднего времени ввода-вывода к среднему времени CPU и важность процесса;
 - 3) тип и величина файлов, используемых для оплаты и отделение, выполняющее работы
 - 4) количество открытых файлов и важность процесса

Интерактивные технологии и инновационные методы, используемые в образовательном процессе

К ним относится электронное тестирование знаний, проводимое в качестве текущего контроля знаний.

Содержание

Рабочая программа	3
План-конспект лекций	10
Методические указания построения лекций	26
Методические указания по изучению дисциплины	26
Методические указания к лабораторным работам	27
Технология выполнения лабораторных работ	27
Методические рекомендации для самоподготовки студентов	29
Контроль знаний	30
Интерактивные технологии и инновационные методы, используемые в качестве текущего контроля знаний	32