

**Министерство науки и высшего образования РФ
ФГБОУ ВО «Ульяновский государственный университет»
Факультет математики, информационных и авиационных технологий**

Кафедра телекоммуникационных технологий и сетей

Смагин А.А.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
для семинарских (практических) занятий, лабораторного практикума
и самостоятельной работы
по дисциплине

«Системное программирование»
для студентов направлений
09.03.02 «Информационные системы и технологии»,
11.03.02 «Инфокоммуникационные технологии и системы»



Ульяновск
2019

Методические рекомендации для семинарских (практических) занятий, лабораторного практикума и самостоятельной работы по дисциплине «Системное программирование»

Настоящие методические рекомендации предназначены для студентов направлений обучения 09.03.02 «Информационные системы и технологии», 11.03.02 «Инфокоммуникационные технологии и системы». В работе приведены литература по дисциплине, темы дисциплины и вопросы в рамках каждой темы, рекомендации по изучению теоретического материала, контрольные вопросы для самоконтроля, задания для самостоятельной работы, задачи и упражнения для самостоятельной подготовки к семинарам или полностью самостоятельного освоения практических навыков, задания для лабораторного практикума и рекомендации по их выполнению.

Студентам всех форм обучения рекомендуется использовать данные методические рекомендации при подготовке к семинарам, самостоятельной подготовке, а также промежуточной аттестации по дисциплине «Системное программирование» или дисциплин, близких по содержанию

Рекомендованы к введению в образовательный процесс

Учёным советом факультета математики, информационных и авиационных технологий
УлГУ

протокол № 2/19 от «19» марта 2019 г.

СОДЕРЖАНИЕ

1. Задачи освоения дисциплины:

- изучение организации и принципов построения современных операционных систем и системных программ
формирование представлений об общей методологии разработки системно-ориентированных программ с использованием современных алгоритмических языков и систем программирования.
- приобретение в рамках освоения предусмотренного курсом занятий следующих знаний, умений и навыков, характеризующих определённый уровень сформированности компетенций (см. подробнее п.3):
 - изучение способов хранения данных на физическом уровне, типы и способы организации файловых систем;
 - понимание проблем и основных способов их решения при коллективном доступе к данным;
 - получение представления о специализированных аппаратных и программных средствах поддержки функционирования
 - эволюция развития современных операционных систем

2.. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

В учебно-методическое обеспечение входят следующие материалы, организующие и обеспечивающие самостоятельную работу обучающихся: -

- рабочая программа учебной дисциплины; -
- учебные и учебно-методические пособия;
- рабочие и фондовые лекции;
- методические рекомендации для подготовки к практическим занятиям;
- методические рекомендации по подготовке и написанию контрольных работ;
- методические рекомендации по организации самостоятельной работы по изучению дисциплины;
- вопросы для проведения промежуточной аттестации;

- тесты для промежуточного контроля успеваемости и контроля остаточных знаний;

- раздаточный материал по учебной дисциплине;

- комплекты наглядных (демонстрационных) материалов, в том числе мультимедийного сопровождения занятий.

3.РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ОТДЕЛЬНЫХ ТЕМ РАЗДЕЛОВ ДИСЦИПЛИНЫ

СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ (МОДУЛЯ)

Основные вопросы раздела

Тема 1. Операционные системы

Программирование - как фундаментальный метод реализации моделей на компьютере. 2. Задачи теоретического программирования. 3. Понятие алгоритма и исполнителя. 4. Основные типы и структуры данных и операторы. 5. Возникновение объектно-ориентированного программирования.

Рекомендации по изучению темы - вопросы изложены на стр.278-311 учебного пособия(1)

Тема 2. Системы программирования

Объектно-ориентированный анализ (ООА). Цели ООА. 8. Основные этапы создания объектно-ориентированного программного продукта 9. Модель взаимодействия объектов. 10. Рабочие продукты объектно-ориентированного анализа.

Рекомендации по изучению темы - вопросы изложены на стр 361 -370 учебного пособия(1), а также на стр 15-45 пособия (2)

Тема 3. Определение конфигурации аппаратных средств.

Объектно-ориентированный анализ (ООА). Цели ООА. Основные этапы создания объектно-ориентированного программного продукта Модель взаимодействия объектов

Рекомендации по изучению темы - вопросы изложены на стр.101 -130 учебного пособия(1)

Тема 4. Использование отладчика при отладке программ

Инкапсуляция и полиморфизм. Процедуры и функции как способ

реализация методов . Наследование и иерархия объектов. Формы наследования. Следствия наследования

Сообщения, экземпляры и инициализация. Параметры и данные, переносимые сообщениями. . Разновидности полиморфизма

Рекомендации по изучению темы - вопросы изложены на стр.130-163 учебного пособия(1) а также на стр 207-230 учебного пособия (2)

Тема 5. Проектирование программ

Механизмы передачи и обработки сообщений в объектно-ориентированных средах.. Видимость и зависимость на уровне классов и объектов. Конструирование программ на основе иерархии объектов. Абстрактные типы и структуры данных.. Объявление объекта. Реализация объекта. Конструктор и деструктор.. Создание объекта.. Объекты и динамическая память. Связные списки. Стеки.. Очереди. Деревья.

Рекомендации по изучению темы - вопросы изложены на стр.209-247 пособия(1) а также на стр 275-300 учебного пособия (2)

Тема 6. Разработка структуры программы

Конструирование программ на основе иерархии объектов. Абстрактные типы и структуры данных. Объявление объекта. Реализация объекта. Конструктор и деструктор. Создание ПРОГРАММЫ.

Рекомендации по изучению темы - вопросы изложены на стр.361-390 учебного пособия(1) . а также на стр 275-300 учебного пособия (2)

4. ТЕМЫ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Тема 1. Установка и настройка ОС (Практика, лабораторная работа)

Тема 2 Работа командной строкой /(Практика, лабораторная работа)

Тема 3 Определение конфигурации аппаратных средств. (Практика, лабораторная работа)

Тема 4 Разработка структуры программы (Практика, лабораторная работа)

Подготовка к выполнению лабораторных работ проводится на учебных компьютерах в лаборатории программного обеспечения кафедры ТТС УлГУ (к.213) на основе их описаний, имеющих в ПО компьютеров.

Вопросы к темам разделов

Понятие системного программного продукта и системного программирования.

Понятие операционной системы, ее назначение и функции.

2. Основные принципы, заложенные в современное системное программное обеспечение.

3. Средства разработки системного программного обеспечения.

4. Понятие процесса, свойства и основы программирования процессов.

5. Концепция процесса.

Состояния процесса и диаграмма смены состояний. Операции над процессами.

6. Планирование процессов. Политики планирования.

7. Сигналы и их использование для управления процессами.

8. Понятие потоков, их свойства и основы программирования потоков.

9. Асинхронные параллельные потоки.

10.. Системные средства синхронизации потоков.

11. Мониторы Хоара.

12. Проектирование и модели многопоточных приложений.

13. Понятие нити, свойства, программирование нитей

. 14. Понятие дескриптора, получение дескрипторов процесса и потока.

15. Понятие синхронизации.

Методы синхронизации.

16. Программирование файлового ввода-вывода. Асинхронный ввод-вывод.

17. Управление разграничением доступа к файлам.

18. Особенности доступа к устройствам.

19. Использование программных потоков для обеспечения ввода-вывода.

20. Отображение файлов на оперативную память.

21. Обмен данными между потоками: потоки ИРС, обзор механизмов ИРС, Реализация памяти общего доступа.

22. Программирование механизмов управления памятью: страничная организация памяти, работа с атрибутами страниц.

23. Программирование механизмов управления памятью: выделение и освобождение памяти.

24. Организация выделения и освобождения памяти в куче.

5. Виды самостоятельной работы

Для подготовки изучения основных разделов курса необходимо выполнить следующие работы.

Теоретическая работа

Необходимо выполнить следующие задания. Изучить

Разработку кода программного продукта на уровне модуля.

1. Управление памятью. Способы выделения памяти в программах. Программно - доступные ресурсы процессора.

2. Использование пользовательских регистров для сохранения данных в памяти ЭВМ.

3. Управление видеоадаптером. Особенности функционирования видеосистемы.

4. Управление программами в объектно-ориентированной среде.

5. Использование пользовательских регистров для обработки данных.

6. Обработка числовых данных при вводе и выводе.

7. Проверка состава оборудования. Механизмы взаимодействия с аппаратными ресурсами

8. Реализация механизмов взаимодействия с аппаратными устройствами через порты ввода-вывода.

9. Использование системных ресурсов через обработку прерываний.

10. Создание программы по разработанному алгоритму как отдельного модуля.

При этом требуется

1. Конспектирование учебной литературы по теме

2. Оформление отчета по выполнению практической работы

3. Проработка теоретических вопросов по учебникам и учебно - методическим пособиям.

4. Написание реферата : «Сравнительный анализ различных архитектур ЭВМ»

5. Выполнение сравнительного анализа операционных систем

6. Выполнение сравнительного анализа машинно-ориентированных языков программирования

7. Создание собственной библиотеки стандартных подпрограмм

8. Углубленное изучение выбранного языка программирования

9. Подготовка к самопроверочной работе по темам

Тема 1. Отладка и тестирование программного продукта на уровне модулей
Термины и определения. Виды ошибок и способы их определения. Виды тестирования.
Порядок разработки тестов. Аксиомы тестирования. Методы тестирования.

Тема 2 Модульное тестирование. Цель модульного тестирования. Тестирование на основе потока управления. Тестирование на основе потока данных. Использование инструментальных средств на этапе отладки. Анализ результатов тестирования программы. 2(62)

Практическая работа

Для выполнения лабораторных и практических занятий уметь проводить:

1. Отладку и тестирование программного продукта на уровне модулей. 2. Разработку системы тестов. Тестирование на основе потока управления. 2. Тестирование на основе потока данных. 3. Тестирование программного модуля.

Порядок выполнения лабораторных работ может быть произвольным и определяется уровнем освоения компетенций обучающегося.

Отчет по выполнению лабораторной работы должен содержать:

- Фамилию и номер группы учащегося, задание
- Описание алгоритма реализации алгоритма обработки данных, программную среду, сравнительную оценку с классическими методами. -перечень тестовых данных -заключение

Курсовая работа

Курсовая работа - самостоятельная письменная аналитическая работа , сопряжена с изучением актуальных вопросов систем системного программирования

Основной целью и задачами работы является актуализация, формулирование проблемы или концепции или выводов по анализу или разработке конкретного вопроса в области системного программирования применительно к заданной предметной области.

Она должна содержать анализ состояния вопроса, о результатах и достижениях, недостатках и недоработках, мешающих дальнейшему развитию , применению и эксплуатации новых методов и средств кодирования и обработки информации, а также тех сдерживающих факторов , которые тормозят появлению инноваций в науке и технике.

Для выполнения курсовых работ необходимо воспользоваться учебно - методическими пособиями , приведенными в списке используемой литературы

ТЕМАТИКА КУРСОВЫХ РАБОТ

1. Электронный тренажер по теории кодирования
2. Разработка системы управления версиями исходного кода ПО.
3. Разработка системы управления IT-проектами.
4. Разработка системы управления рабочим процессом по методологии GTD.
5. Поиск кратчайших путей и связывающих деревьев в графах с переменными вершинами.

- 6 Решение транспортных задач с меняющимися емкостями и затратами.
7. Численные и графические модели простых аукционов для распределения ресурсов.
- 8 Применение аукционов для распределения полосы частот в телекоммуникационной сети.
9. Поиск максимального потока в сети с переменными пропускными способностями.
10. Транспортная задача с ограничениями на пропускную способность линий.
11. Разработка информационных систем на базе web-технологий (PHP, ASP.NET, Silverlight+ASP.NET, Ruby on Rails).
12. Разработка приложений для мобильных устройств (C#, Java) с использованием web-технологий
13. Создание учебных тренажеров по учебным дисциплинам.
14. Создание и интеграция систем электронного документооборота.
15. Приложения алгоритмов машинного обучения (random forests, neural networks);
16. Разработка и реализация алгоритмов компьютерного зрения (object detection, object tracking);
17. Разработка и реализация стегоаналитических алгоритмов;
18. Разработка и реализация криптоаналитических атак на реализацию криптографических алгоритмов;
19. Разработка и реализация алгоритмов скрытой передачи данных (стеганографическое встраивание информации в цифровое видео).
20. Географические информационные системы для конкретной предметной области
21. Система автоматической проверки решений задач по программированию
22. Демонстратор решений задач на графах
23. Программное обеспечение системы резервирования билетов
24. Разработка параллельных программ для многопроцессорных систем

6 Тесты для самоподготовки

(жирным шрифтом отмечены правильные ответы)

1. Операционная система –

это _____:

- + **Комплекс программ**
- Прикладная программа
- + **Администратор**
- Обеспечение
- Назначение
- Пользование
- Сервисная программа
- + **Менеджер ресурсов компьютера**

2. Операционные системы являются ресурсами _____ компьютера:

- + **управления**
- + **диспетчеризация**
- + **планирование**
- администратор
- менеджер
- обеспечение
- назначение
- пользование

3. Операционные системы для программирования приложений Win32API:

- + **Windows 98**
- Linux
- + **Windows 2000**
- Unix
- + **Windows XP**
- Windows CE
- Windows NT
- MS DOS

4. Категории объектов ОС Windows приложения:

- Windows 98
- User
- + **Graphics Device Interface**
- + **Unix**
- Windows Shell
- + **Kernel**
- Network Services
- Base Services

5. Ресурсы принадлежащие каждому потоку в ОС Windows:

- потоки интерфейса пользователя
- пользовательские потоки
- + **код исполняемой функции**
- потоки исполняемой функции
- + **набор регистров процессора**
- библиотека общих элементов
- интерфейс пользователя
- + **стек для работы приложения**

6. Действия менеджера потоков, во время переключения процессора на исполнение другого потока:

- обслуживает очередь запускаемого потока
- + **сохранить контекст прерываемого потока**
- + **восстановить контекст запускаемого потока**
- + **передать управление запускаемому потоку**

- управляет доступом для ОС
- выполняет функции ввода и вывода
- обеспечивает функции для вывода графики
- обеспечивает функции для взаимодействия

7. Параметры функции CreateThread:

- TerminateThread
- + lpThreadAttributes**
- AttachThreadInput
- CreateProcess
- CreateThread
- ExitThread
- + dwStackSize**
- + lpStrtAddress**

8. Функции работы с процессами и потоками в Win32 API:

- TerminateThread
- lpThreadAttributes
- + AttachThreadInput**
- + CreateProcess**
- ExitThread
- + CreateThread**
- dwStackSize
- lpStrtAddress

9. Функция Sleep(_____) :

- + задерживает выполнения потока**
- возвращает нулевое значение
- + удаляет поток из очереди**
- выполняет выход из потока
- освобождает память
- + делает короткие паузы**
- завершает выполнение функций
- освобождает ресурсы

10. Классификация программ в зависимости от количества определяемых ими параллельных потоков управления:

- дуплексный
- многозадачный
- + многопоточный**
- системный
- + с параллельным потоком**
- параллельный
- + однопоточный**
- с прямым потоком

11. Операции над потоком связанные с операционной системой:

- Create
- + Run**
- Exit
- + Interrupt**
- Open
- + Block**
- Read
- Write

12. Ресурсы каждого процесса ОС Windows:

+ виртуальное адресное пространство

- код исполняемой функции

+ маркер доступа

- набор регистров процессора

- стек для работы приложения

+ страницы в реальной памяти

- стек для работы ОС

- консольный ввод

13. Объектам синхронизации первого класса в Windows, которые служат только для решения задач синхронизации параллельных потоков:

+ мьютекс (mutex)

+ событие (event)

+ семафор (semaphore)

- ожидающий таймер

- работа (job)

- процесс (process)

- поток (thread)

- консольный ввод (console input).

14. К третьему классу синхронизации относятся объекты, которые переходят в сигнальное состояние по завершении своей работы:

- мьютекс (mutex)

- событие (event)

+ работа (job)

+ процесс (process)

+ поток (thread)

- семафор (semaphore)

- ожидающий таймер

- консольный ввод (console input)

15. Системные объекты, созданные менеджером объектов в Win32 API:

- класс

- поток

+ события

+ семафор

- процесс

+ критический раздел

- приоритет потока

- контекст

16. Логические комбинации флагов параметра dwDesiredAccess:

+ EVENT_ALL_ACCESS

+ EVENT_MODIFY_STATE

- ACTIONS_AFTER_EVENT

+ SYNCHRONIZE

- SEMAPHORE_ALL_ACCESS

- SEMAPHORE_MODIFY_STATE

- ACTIONS_BEFORE_EVENT

- CREATE_NEW_CONSOLE

17. Логические комбинации флагов параметра dwDesiredAccess, определяющий доступ к семафору:

- EVENT_ALL_ACCESS

- EVENT_MODIFY_STATE

+ SYNCHRONIZE

+ SEMAPHORE_ALL_ACCESS

- ACTIONS_AFTER_EVENT
- + SEMAPHORE_MODIFY_STATE**
- ACTIONS_BEFORE_EVENT
- CREATE_NEW_CONSOLE

18. Значение функции WaitForSingleObject, в случае успешного завершения:

- wait_object_p
- wait_for_single
- + wait_object_o**
- wait_object
- + wait_abandoned**
- + wait_timeout**
- wait_exit
- wait_finally

19. Прототип функции WaitForMultipleObject:

- + DWORD nCount**
- + CONST HANDLE *lpHandles**
- + BOOL bWaitAll**
- DWORD dwReserved
- BOOL bResume
- LPOVERLAPPED lpOverLapped
- HANDLE hThread
- DWORD IDThread

20. Значения функции WaitForMultipleObjects:

- + ОТ WAIT_OBJECT_0 ДО (WAIT_OBJECT_0 + nCount - 1);**
- ОТ WAIT_OBJECT_0 ДО (WAIT_OBJECT_0);
- ОТ WAIT_OBJECT_0 ДО (WAIT_OBJECT_0 + nCount);
- + ОТ WAIT_ABANDONED_0 ДО (WAIT_ABANDONED_0 + nCount - 1);**
- WAIT_COMPLETION
- + WAIT_TIMEOUT**
- ОТ WAIT_ABANDONED_0 ДО (WAIT_ABANDONED_0 + nCount + 1);
- ОТ WAIT_ABANDONED_0 ДО (WAIT_ABANDONED_0);

21. Состояние блока адресов в адресном пространстве:

- + выделен**
- защищен
- + зарезервирован**
- + свободен**
- смещен
- удален
- копирован
- прикреплен

22. Интерфейсы (API) для управления памятью:

- Base Services
- Common Control Library
- + Virtual Memory**
- Network Services
- + Memory Mapped File**
- + Heap Memory**
- Windows Shell
- Windows System Information

23. Куча – это _____:

- объекты памяти

- страница памяти
- + динамическая область памяти**
- указатель блока памяти
- + блок памяти**
- указатель строки
- + мелкие фрагменты памяти**
- указатель функции

24. Функции для управления памятью кучи:

- + HeapAlloc()**
- HeapDestroy()
- + HeapReAlloc()**
- HeapFree()
- HeapSize()
- + HeapFree()**
- HeapCreate()
- MapViewOfFile()

25. Форматы реального и виртуального адресов:

- + номер реальной страницы**
- номер фактической страницы
- номер оперативной памяти
- + номер виртуальной страницы**
- смещение в адресной строке
- + смещение в реальной и виртуальной странице**
- файлы страницы
- номер динамической страницы

26. Форматы реального и виртуального адресов:

- a
- + r**
- b
- + v**
- c
- + d**
- e
- k

27. Описание линейного адреса процесса в ОС Windows:

- + 32 бит**
- 16 бит
- + от 0x00000000 до 0xFFFFFFFF**
- 2 Гбайт логической памяти
- 4 Гбайт виртуальной памяти
- 2 Гбайт виртуальной памяти
- + 4 Гбайт логической памяти**
- от 0x00000000 до 0x0000FFFF

28. Состояние страницы процесса виртуальной памяти:

- (блокирован);
- (выделен)
- + free (свободный)**
- (не выделен)
- + committed (распределены)**
- (готов)
- (не готов)
- + reserved (зарезервированный)**

29. Функция файловой системы:

- совместно использовать объект файла
- + открытие доступа к существующему файлу**
- выделяют мелкие фрагменты файла
- + закрытие доступа к существующему файлу**
- закрепление виртуальной памяти
- + установка указателя файла на нужную запись**
- физическое или форматирование низкого уровня
- разбиение диска на разделы

30. Структура и описание каталога:

- + древовидная**
- кольцевая
- + корневой**
- табличная
- + \ (обратная косая)**
- прямая
- выпуклая
- плоская

31. Имена каталогов и файлов не должны содержать:

- +
- + <**
- *
- + :**
- + /**
- ?
- !
- .

32. Параметр dwDesiredAccess задает способ доступа к файлу и принимает значения:

- + 0**
- FILE_SHARE_WRITE
- FILE_SHARE_READ
- + GENERIC_READ**
- + GENERIC_WRITE**
- OPEN_EXISTING
- OPEN_ALWAYS
- TRUNCATE_EXISTING

33. Значения параметра dwCreationDisposition при открытии файла:

- 0
- FILE_SHARE_WRITE
- FILE_SHARE_READ
- GENERIC_READ
- GENERIC_WRITE
- + OPEN_EXISTING**
- + OPEN_ALWAYS**
- + TRUNCATE_EXISTING**

34. Правильная запись имени файла:

- + "C:\\demo_file.dat"**
- "C:\\demo_file.dot"
- "C:\\demo\\file.dat"
- "C\\demo_file.dat"
- + "C:\\new_file.dat"**

- "C:\\ new_file.dat"
- + "C:\\back_file.dat"
- "\\back_file.dat"

35. Значение функций GetFileType:

- FILE_SHARE_WRITE
- FILE_SHARE_READ
- GENERIC_READ
- + **FILE_TYPE_DISK**
- + **FILE_TYPE_CHAR**
- TRUNCATE_EXISTING
- + **FILE_TYPE_PIPE**
- OPEN_ALWAYS

36. Значения параметра функции CopyFile:

- + **IpExistingFileName**
- lpThreadAttributes
- AttachThreadInput
- + **IpNewFileName**
- dwStackSize
- lpStrtAddress
- + **bFailIfExists**
- TerminateThread

37. Верхние уровни, составляющие основу структуры реестра Windows:

- HKEY_LOCAL
- HKEY_LOCAL_WORD
- + **HKEY_LOCAL_MACHINE**
- + **HKEY_CURRENT_USER**
- + **HKEY_CLASSES_ROOT**
- HKEY_CURRENT_MACHINE
- HKEY_CLASSES_MACHINE
- HKEY_CLASSES_USER

38. API функции, применяемые для выполнения операций с реестром:

- RegReserverd
- + **RegCloseKey**
- RegClass
- RegSubKeys
- + **RegCreateKey**
- RegOpen
- RegQery
- + **RegDeleteKey**

39. API функции применяемые в среде Windows NT:

- RegQueryValue()
- + **RegSetKeySecurity()**
- RegCloseKey()
- RegFlushKey()
- + **RegGetKeySecurity()**
- RegLoadKey()
- + **параметры безопасности**
- RegOpenKey()

40. HKey идентифицирует текущий раздел или предопределенные дескрипторы:

- HKEY_LOCAL_MACHINE
- HKEY_LOCAL_WORD

- + **HKEY_CLASSES_ROOT**
- HKEY_CURRENT_MACHINE
- HKEY_CLASSES_USER
- + **HKEY_CURRENT_USER**
- HKEY_CLASSES_MACHINE
- + **HKEY_USERS**

41. Параметры функции WriteFile:

- nNumberOfBytes
- lpThreadAttributes
- + **hFile**
- + **lpBuffer**
- dwStackSize
- lpNewFileName
- bFailIfExists
- + **nNumberOfBytesToWrite**

42. Системный вызов для файла:

- + **chown()**
- + **lchown()**
- chmod
- fchmod
- stat()
- lstat()
- + **fchown()**
- read()

43. Метаданные файла связанные со временем:

- st_blocks
- st_dev
- + **st_ctime**
- st_uid
- + **st_atime**
- st_gid
- st_size
- + **st_mtime**

44. Функции для перемещения файла:

- lpNewFileName
- + **replaceFile**
- readFile
- + **copyFile**
- hFile
- + **moveFile**
- write_File
- delete_File

45. Механизм отображения файлов в память:

- + **отображение содержимого файла**
- установка времени таймера
- + **представление или вид файла**
- + **когерентность данных**
- указывать на функцию завершения
- создать файловое пространство
- открыть файл приложения
- создать файл загрузки

46. Параметры функции CreateFileMapping (создание объекта отображающий файл в память):

- lpBuffer
- numberOfBytes
- completionKey
- + **flProtect**
- readfile
- + **lpAttributes**
- waitCommEvent
- + **hFile**

47. Значения параметра flProtect :

- PAGE_READ
- + **PAGE_READONLY**
- PAGE_WRITEONLY
- + **PAGE_READWRITE**
- PAGE_READCOPY
- + **PAGE_WRITECOPY**
- PAGE_WRITE
- PAGE_COPYONLY

48. Прототип функции MapViewOfFile:

- + **HANDLE hFileMappingObject**
- + **DWORD dwFileOffsetHigh**
- + **DWORD dwFileOffsetLow**
- DWORD dwReserved
- BOOL bResume
- LPOVERLAPPED lpOverLapped
- HANDLE hThread
- DWORD IDThread

49. Прототип функции MapViewOfFileEx, отображающий файл в адресное пространства с некоторого заданного виртуального адреса:

- + **HANDLE hFileMappingObject**
- DWORD dwReserved
- + **DWORD DesireAccess**
- + **LPVOID lpBaseAddress**
- BOOL bResume
- LPOVERLAPPED lpOverLapped
- HANDLE hThread
- DWORD IDThread

50. Механизм отображения файлов в память:

- узнать какой поток ее вызывает
- + **файл отображен несколькими процессами**
- хранить указатели на захваченную память
- запускать приложение
- + **когерентность отображений**
- + **обмен данными между процессами**
- открыть файл приложения
- создать файл загрузки

Источник: <https://yznaika.com/notes/627-system-programming>

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

основная а) Список рекомендуемой литературы

основная 1. Гордеев, А. В. Операционные системы [Текст] : учебник / А. В. Гордеев.- 2-е изд. - Санкт Петербург : Питер, 2007. - 416 с. - (Учебник для вузов). - Библиогр.: с. 406- 408. - Алф. указ.: с. 409-415. - ISBN 978-5-94723-632-3.

2. Молчанов А. Ю. Системное программное обеспечение: учеб. для вузов / А. Ю. Молчанов.- 3-е изд. - СПб. : Питер, 2010. - 398 с.: ил. - (Учебник для вузов). - Указ. лит.: с. 387-390. - Алф. указ.: с. 391-397. - ISBN 978-5-807-153-4.

Дополнительная

1. Бек, Л. Введение в системное программирование / Л. Бек. - М.: Мир, 2016. - 448 с.

2. Лав, Р. Linux. Системное программирование / Р. Лав. - М.: Питер, 2016. - 456 с.

учебно-методическая

- 1) Рекомендации по разработке и оформлению рефератов, курсовых, выпускных квалификационных и дипломных работ : учеб. пособие для вузов по спец. 550400 (210400.62) / Смагин Алексей Аркадьевич, Ю. Д. Украинцев; УлГУ, ФМиИТ. - Ульяновск : УлГУ, 2019. - 73 с. : ил. - б/п.
Электронная версия на сайте кафедры ТТС УлГУ.

Дополнительная

1) Смагин А.А., Украинцев Ю.Д., Булаев А.А. Рекомендации по разработке и оформлению результатов курсовых ,выпускных квалификационных работ. Учебно –методическое пособие для студентов, обучающихся по направлениям для студентов направлений: 09.03.02 «Информационные системы и технологии»,11.03.02 «Инфокоммуникационные технологии и системы» .УлГУ .Электронное издание УлГУ - 2019 г.

2)

- 1.4. **Лань** [Электронный ресурс]: электронно-библиотечная система / ООО ЭБС Лань. - Электрон. дан. – С.-Петербург, [2019]. - Режим доступа: <https://e.lanbook.com>.
- 1.5. **Znanium.com** [Электронный ресурс]: электронно-библиотечная система / ООО Знаниум. - Электрон. дан. – Москва, [2019]. - Режим доступа: <http://znanium.com>.
2. **КонсультантПлюс** [Электронный ресурс]: справочная правовая система. /Компания «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2019].
3. **База данных периодических изданий** [Электронный ресурс] : электронные журналы / ООО ИВИС. - Электрон. дан. - Москва, [2019]. - Режим доступа: <https://dlib.eastview.com/browse/udb/12>.
4. **Национальная электронная библиотека** [Электронный ресурс]: электронная библиотека. - Электрон. дан. – Москва, [2019]. - Режим доступа: <https://нэб.рф>.
5. **Электронная библиотека диссертаций РГБ** [Электронный ресурс]: электронная библиотека / ФГБУ РГБ. - Электрон. дан. – Москва, [2019]. - Режим доступа: <https://dvs.rsl.ru>.
- 6. Федеральные информационно-образовательные порталы:**
- 6.1. Информационная система [Единое окно доступа к образовательным ресурсам](#).
Режим доступа: <http://window.edu.ru>
- 6.2. Федеральный портал [Российское образование](#). Режим доступа: <http://www.edu.ru>
- 7. Образовательные ресурсы УлГУ:**
- 7.1. Электронная библиотека УлГУ. Режим доступа : <http://lib.ulsu.ru/MegaPro/Web>
- 7.2. Образовательный портал УлГУ. Режим доступа : <http://edu.ulsu.ru>

Программное обеспечение

1. (open source),