

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
ИНГУШЕТИЯ
ЧАСТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«КОЛЛЕДЖ ЭКОНОМИКИ И ПРАВА»**

УТВЕРЖДАЮ
Директор ЧПОУ «КЭиП»
Х.Э.Холохоева
«_____» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

МДК.01.04 Системное программирование

по программе подготовки специалистов среднего звена (ППССЗ)

Специальность:	09.02.03 « Информационные системы и программирование »
Обучение:	по программе базовой подготовки
Уровень образования, на базе которого осваивается ППССЗ:	Основное общее, среднее общее
Квалификация:	«программист»
Форма обучения:	Очная,очно- заочная

Назрань

202__

Рабочая программа учебной дисциплины «Системное программирование» разработана на основе требований ФГОС, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины в соответствии с рекомендациями по организации получения образовательных программ среднего профессионального образования для ЧПОУ «КЭиП» по специальности 09.02.03 «Программирование компьютерных системах».

Организация-разработчик: ЧПОУ «Колледж экономики и права»

Разработчик: _____ преподаватель ЧПОУ «КЭиП»

Рассмотрена, одобрена и утверждена на заседании кафедры естественно-научных дисциплин

№ ___ от «___» _____ 202 г.

© _____
© КЭиП, 202

СОДЕРЖАНИЕ

	стр.
1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	5
2. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	13
3. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ	

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина "Системное программирование" относится к вариативной части образовательной программы

Дисциплина «Системное программирование» базируется на дисциплинах: Информатика.

Дисциплина «Системное программирование» предшествует изучению дисциплин: Научно-исследовательская работа.

Освоение дисциплины направлено на формирование следующих общекультурных, общепрофессиональных и профессиональных компетенций:

способность применять теорию и методы математики для построения качественных и количественных моделей объектов и процессов в естественнонаучной сфере деятельности (ОПК-2);

способность применять полученные знания для анализа систем, процессов и методов (ОПК-4);

способность выбирать и применять подходящее оборудование, инструменты и методы исследований для решения задач в избранной предметной области (ПК-3);

способность критически оценивать применимость применяемых методик и методов (ПК-4).

Целями преподавания дисциплины являются:

- освоение студентами системного программирования;
- приобретение навыков самостоятельного изучения отдельных тем дисциплины и решения типовых задач;
- приобретение навыков работы в современных интегрированных системах программирования для реализации программных продуктов;
- усвоение полученных знаний студентами, а также формирование у них мотивации к самообразованию за счет активизации самостоятельной познавательной деятельности.

Поставленные цели полностью соответствуют целям (Ц1-Ц5) ООП.

2. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебного кабинета, в котором имеется возможность обеспечить свободный доступ в Интернет во время учебного занятия и в период внеучебной деятельности обучающегося.

В состав учебно-методического и материально-технического обеспечения программы учебной дисциплины входят:

- многофункциональный комплекс преподавателя;
- наглядные пособия (комплекты учебных таблиц, плакатов, портретов выдающихся ученых, поэтов, писателей и др.);
- информационно-коммуникативные средства;

- экранно-звуковые пособия;
- комплект технической документации, в том числе паспорта на средства обучения, инструкции по их использованию и технике безопасности;
- библиотечный фонд.

Технические средства обучения: компьютер с лицензионным программным обеспечением и мультимедийный проектор.

Цели и задачи дисциплины – требования к результатам освоения

дисциплины:

Изучение дисциплины направлено на формирование следующих компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем учебной дисциплины и виды учебной работы (очная форма)

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	116
Обязательная аудиторная учебная нагрузка (всего)	78
в том числе:	
практические занятия	30
Лекции	48
Самостоятельная работа обучающегося (всего)	38
<i>Промежуточная аттестация в форме зачета</i>	
<i>Итоговая аттестация в форме экзамена</i>	

3.2. Объем учебной дисциплины и виды учебной работы (заочная форма)

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	
Обязательная аудиторная учебная нагрузка (всего)	
в том числе:	
практические занятия	
Лекции	
Самостоятельная работа обучающегося (всего)	
<i>Промежуточная аттестация в форме зачета</i>	
<i>Итоговая аттестация в форме экзамена</i>	

4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины студент должен:

Знать:

- основы построения и архитектуру ЭВМ;
- принципы построения современных операционных систем и особенности их применения;
- технологии разработки алгоритмов и программ, методов отладки и решения задач на ЭВМ в различных режимах, основы объектно-ориентированного подхода к программированию;

уметь:

- настраивать конкретные конфигурации операционных систем;
- ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные документы, работать с современными системами программирования, включая объектно-ориентированные;

владеть:

- навыками работы с различными операционными системами и их администрирование;
- языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее чем на одном из алгоритмических процедурных языков программирования высокого уровня.

В результате освоения дисциплины выпускник обладает следующими **общекультурными и профессиональными компетенциями:**

1. Универсальные (общекультурные):

- владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК-1);
- умеет логически верно, аргументировано и ясно строить устную и письменную речь (ОК-2);
- готов к кооперации с коллегами, работе в коллективе (ОК-3);
- стремится к саморазвитию, повышению своей квалификации и мастерства (ОК-6);
- умеет критически оценивать свои достоинства и недостатки, наметить пути и выбрать средства развития достоинств и устранения недостатков (ОК-7);
- осознает социальную значимость своей будущей профессии, обладает высокой мотивацией к выполнению профессиональной деятельности (ОК- 8);
- владение навыками работы с компьютером как средством управления информацией (ОК-12 ФГОС);
- способность работать с информацией в глобальных компьютерных сетях (ОК-13 ФГОС).

2. Профессиональные:

- осваивать методики использования программных средств для решения практических задач (ПК-2 ФГОС);

- разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования (ПК-5 ФГОС);
- обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности (ПК-6 ФГОС).

4.1 Аннотированное содержание разделов дисциплины:

1. Введение
 - a) Вводная информация: предмет курса; содержание теоретического материала курса; содержание цикла лабораторных работ, список литературы для изучения.
 - b) Системное программное обеспечение: основные понятия и их определения; расположение СПО в общей структуре ЭВМ, классификация и структура СПО; организация взаимодействия между аппаратурой ЭВМ, СПО и прикладным ПО.
 - c) Классификация системных программ: операционная система, загрузчики, трансляторы, компиляторы и интерпретаторы, отладчики, утилиты.
 - d) Интерфейс операционной системы: основные принципы и стандарты; системные вызовы; интерфейсы WinAPI, POSIX API; 32 и 64 разрядные интерфейсы; проблема локализации, стандарты ANSI и UNICODE.
 - e) Средства разработки Windows-программ, используемые при изучении дисциплины: Visual Studio
2. Особенности выполнения программ
 - a) Объекты ядра: создание, уничтожение, таблица описателей, учет пользователей объектов ядра, наследование.
 - b) Процесс выполнения программ: создание, завершение процессов и потоков.
 - c) Синхронизация потоков: механизмы синхронизации (семафоры, мониторы, сообщения, барьеры).
 - d) Решение классических проблем синхронизации: проблема обедающих философов, проблема читателей и писателей, проблема спящего бравобрея.
 - e) Реализация синхронизации: синхронизация потоков в пользовательском режиме; синхронизация потоков с использованием объектов ядра
 - f) Межпроцессные взаимодействия (IPC): механизмы, каналы, очереди сообщений, разделяемые сегменты памяти, сокеты, вызов удаленных процедур (RPC).
3. Ввод-вывод
 - a) Принципы аппаратуры ввода-вывода: устройства, контроллеры устройств; ввод-вывод, отображаемый на адресное пространство памяти; прямой доступ к памяти (DMA); настройка адресов и защита.
 - b) Принципы программного обеспечения ввода-вывода: задачи ПО; управляемый прерываниями ввод-вывод; ввод-вывод с использованием DMA.
 - c) Программные уровни ввода-вывода: обработчики прерываний, драйверы устройств, независимое от устройств ПО ввода-вывода; ПО ввода-вывода пространства пользователя.
 - d) Подсистема ввода-вывода в MS Windows XP: компоненты ввода-вывода и их взаимодействие; объекты, осуществляющие взаимодействие; драйвера.

4. Файловые системы

- a) Файлы: структура и типы файлов; доступ к файлу; атрибуты файла; операции с файлами; файлы, проецируемые в память. Синхронный и асинхронный ввод/вывод.
- b) Каталоги: одноуровневые, двухуровневые и иерархические системы каталогов; операции с каталогами
- c) Реализация файловой системы: структура файловой системы, реализация файлов и каталогов; совместно используемые файлы; надежность и производительность файловой системы
- d) Взаимоблокировки, их обнаружение. Избежание взаимоблокировок; безопасные и небезопасные состояния.

5. Драйвера устройств

- a) Драйвера: задачи, классификация и особенности их функционирования, основные свойства и характеристики; драйвера в ОС MS Windows и Unix.
- b) Организация работы подсистемы управления внешними устройствами в MS Windows 2000: типы драйверов; стек драйверов; загрузка, инициализация и выгрузка драйверов; инсталляция драйверов; синхронный и асинхронный ввод-вывод; выполнение операций ввода-вывода; Plug and Play;
- c) Унифицированная модель разработки драйверов для Windows платформ (WDM): свойства; структура драйвера и принципы функционирования; интерфейс Native API методы и средства разработки; управление памятью, работа со строками, осуществление операций ввода-вывода; драйвер-фильтр;

6. Подсистема безопасности

- a) Подсистема безопасности: цели; защита объектов; аудит; права и привилегии; выполнение действий от другого имени; аутентификация
- b) Реализация подсистемы безопасности в MS Windows 2000 и Unix: компоненты, основные принципы и механизмы защиты.

7. Службы, особенности их создания и работы

- a) Назначение, особенности их работы.
- b) Создание, запуск, удаление службы.

Лабораторные занятия

1. Создание простого Windows приложения. Каркас Win32 программы. Изучение принципов работы с Win32 API. Изучение возможностей используемой системы программирования (MS Visual Studio): компилятора, транслятора, отладчика.
2. Работа со строками. Использование ANSI, UNICODE строк в Windows-программах, преобразование, вывод. Обработка ошибок в Win32.
3. Работа с объектами ядра Win32. Создание, удаление объектов; работа с описателем объекта, наследование. Изучение особенностей выполнения программ на Windows-платформе.
4. Работа с процессами и потоками. Создание процессов, потоков, их идентификация. Получение параметров процесса и его состояния.
5. Синхронизация потоков. Создание нескольких потоков и синхронизация их одним из предложенных методов. Использование критических состояний, мьютексов, семафоров, барьеров.

6. Взаимодействие процессов. Передача данных между выполняющимися процессами одним из предложенных методов: при помощи почтовых ящиков, каналов или сокетов.
7. Взаимоблокировки. Моделирование тупиковой ситуации и реализация метода избежания ее на примере работы с файлами.
8. Создание драйвера. Организация ввода-вывода в Microsoft Windows XP. Каркас WDM драйвера. Знакомство со средой разработки драйверов Driver Development Kit и Native API. Инсталляция драйвера. Стек драйверов. Организация взаимодействия между драйверами. Создание приложения для управления драйвером.

НАИМЕНОВАНИЕ ТЕМ	СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА	ОБЪЕМ ЧАСОВ		УРОВЕНЬ ОСВОЕНИЯ
		Очная форма	Заочная форма	
Тема 1. Системное программное обеспечение (СПО)	Лекция: Основные понятия и их определения; расположение СПО в общей структуре ЭВМ, классификация и структура СПО; организация взаимодействия между аппаратурой ЭВМ, СПО и прикладным ПО.	4	2	1, 2, 3
	Практическое занятие: Создание простого DOS приложения. Каркас MS-DOS приложения. Изучение принципов работы INT21h.	2		
	Самостоятельная работа: 1. Назначение и функции ОС 2. Классификация ОС.	4	20	
Тема 2. Классификация системных программ.	Лекция: Операционная система, загрузчики, трансляторы, компиляторы и интерпретаторы, отладчики, утилиты.	3	2	1, 2, 3
	Практические занятия: Работа со строками.	2	2	
	Самостоятельная работа: Особенности выполнения программ	2	10	
Тема 3. Интерфейс операционной системы.	Лекции: Основные принципы и стандарты; системные вызовы; интерфейсы WinAPI, POSIX API; 32 и 64 разрядные интерфейсы; проблема локализации, стандарты ANSI и UNICODE.	2		1, 2, 3
	Практическое занятие: Ввод с клавиатуры. Прерывание INT16h.	4		
	Самостоятельная работа: 1. Структура драйверов. 2. Функции ввода/вывода. 3. Ввод/вывод в UNIX.	6	10	
Тема 4. Средства разработки.	Лекции: Средства разработки DOS, Windows-программ, используемые при изучении дисциплины: TASM, MASM32, Visual Studio	2	10	1, 2, 3
	Практические занятия: Вывод символов на экран. Прямое отображение памяти.	6		

	Самостоятельная работа: 1. Характеристики файлов. 2. Способы размещения файлов. 3. Разделение доступа	6	30	
Тема 5. Особенности выполнения программ.	Лекции: Процесс выполнения программ: создание, завершение процессов и потоков	2	2	1, 2, 3
	Практические занятия: Организация циклов.	4	2	
	Самостоятельная работа: 1. Структура диска. 2. Атрибуты файлов.	4	20	
Тема 6. Межпроцессные взаимодействия (IPC).	Лекции: Механизмы, каналы, очереди сообщений, разделяемые сегменты памяти, сокеты, вызов удаленных процедур (RPC).	2		2, 3
	Практические занятия: Ветвление программы. Работа с отладчиком.	6		
	Самостоятельная работа: Взаимоблокировки, безопасные и небезопасные состояния	2	10	
Тема 7. Ввод-вывод.	Лекции: Принципы аппаратуры ввода-вывода: устройства, контроллеры устройств; ввод-вывод, отображаемый на адресное пространство памяти; прямой доступ к памяти (DMA); настройка адресов и защита	2		2, 3
	Практические занятия: Процедуры. Сокращение, структурирование исходного текста. Создание библиотек	4		
	Самостоятельная работа: Принципы аппаратуры ввода-вывода, программное обеспечение	4	10	
Тема 8. Программные уровни ввода-вывода.	Лекции: Обработчики прерываний, драйверы устройств, независимое от устройств ПО ввода-вывода; ПО ввода-вывода пространства пользователя.	4		2, 3
	Практические занятия: Прорисовка рамки. Используя циклы, массивы.	6	2	

Тема 9. Подсистема ввода вывода в MS Windows XP.	Лекции: Компоненты ввода-вывода и их взаимодействие; объекты, осуществляющие взаимодействие; драйвера.	2	2	2, 3
	Практические занятия: Открытие текстового файла, и вывод его на экран. (консоль).	4		
	Самостоятельная работа: Работа с процессами и потоками	2	10	
Тема 10. Использование файловой системы и символьного ввода-вывода Win32	Лекции: Файлы, каталоги. Реализация файловой системы	4		1, 2, 3
	Практические занятия: Работа с объектами ядра Win32. Обработка ошибок в Win32.	6	2	
	Самостоятельная работа: Реализация файловой системы	4	10	
Тема 11. Структурная обработка исключений	Лекции: Объекты ядра, процесс выполнения программ	4		1, 2, 3
	Практические занятия: Взаимоблокировки, безопасные и небезопасные состояния	4		
	Самостоятельная работа: Синхронизация потоков	4	10	
Тема 12. Управление памятью, динамические библиотеки	Лекции: Взаимоблокировки, безопасные и небезопасные состояния	4		1, 2, 3
	Практические занятия: Взаимодействие процессов	6		
	Самостоятельная работа: Взаимодействие процессов	4	10	
Тема 13. Управление процессами	Лекции: Работа с процессами и потоками.	4		1, 2, 3
	Практические занятия: Создание процессов, потоков, их идентификация.	4		
	Самостоятельная работа: Получение параметров процесса и его состояния.	4	10	
Тема 14. Потоки и планирование	Лекции: Подсистема безопасности	4		1, 2, 3
	Практические занятия: Назначение, особенности работы	6	2	

	Самостоятельная работа: Организация подсистемы управления.	4	10	
Тема 15. Синхронизация потоков	Лекции: Синхронизация потоков.	4	2	1, 2, 3
	Практические занятия: Создание нескольких потоков и синхронизация их одним из предложенных методов.	4		
	Самостоятельная работа: Использование критических состояний, мьютексов, семафоров, барьеров.	4	10	
Тема 16. Асинхронный ввод - вывод и порты завершения	Лекции: Синхронизация потоков. Взаимодействие процессов.	4		1, 2, 3
	Практические занятия: Взаимодействие процессов. Передача данных между выполняющимися процессами одним из предложенных методов: при помощи почтовых ящиков, каналов или сокетов.	6	2	
	Самостоятельная работа: Передача данных между выполняющимися процессами одним из предложенных методов: при помощи почтовых ящиков, каналов или сокетов.	4	10	
Тема 17. Межпроцессное взаимодействие	Лекции: Взаимоблокировки. Моделирование тупиковой ситуации и реализация метода избежания ее на примере работы с файлами.	4	2	1, 2, 3
	Практические занятия: Взаимоблокировки. Моделирование тупиковой ситуации и реализация метода избежания ее на примере работы с файлами.	4		
	Самостоятельная работа: Моделирование тупиковой ситуации и реализация метода избежания ее на примере работы с файлами.	4	10	
Тема 18. Программирование драйверов	Лекции: Драйвера устройств. Организация подсистемы управления	6		1, 2, 3
	Практические занятия: Унифицированная модель разработки драйверов	6		
	Самостоятельная работа: Создание, запуск, удаление службы	6	10	
Тема 19.	Лекции: Создание драйвера. Организация ввода-вывода в Microsoft	4		1, 2, 3

WinAPI	Windows XP. Каркас WDM драйвера.			
	Практические занятия: Знакомство со средой разработки драйверов Driver Development Kit и Native API. Инсталляция драйвера.	4		
	Самостоятельная работа: Стек драйверов. Организация взаимодействия между драйверами. Создание приложения для управления драйвером	4	10	
Тема 20. Программирование в Win64	Лекции: Создание простого Windows приложения. Каркас Win64 программы. Изучение принципов работы с Win64 API.	4		1, 2, 3
	Практические занятия: Изучение возможностей используемой системы программирования (MS Visual Studio): компилятора, транслятора, отладчика.	6		
	Самостоятельная работа: Каркас Win64 программы. Изучение принципов работы с Win64 API.	4	10	
Тема 21. Программирование пользовательского интерфейса в Visual C++	Лекции: Работа со строками. Использование ANSI, UNICODE строк в Windows-программах, преобразование, вывод.	12		1, 2, 3
	Практические занятия: Обработка ошибок в Win32	4	2	
	Самостоятельная работа: Работа со строками. Использование ANSI, UNICODE строк в Windows-программах, преобразование, вывод.	12	11	
Всего		116	116	

1. – ознакомительный (узнавание ранее изученных объектов, свойств);

2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)

3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

Банк заданий к лабораторным работам (примерный)

Лабораторная работа №1

Вывести количество времени, прошедшее с момента запуска операционной системы (формат: сутки, часы, минуты, секунды)

Написать программу, которая выводит текущее время в MessageBox.

Вывести на экран имя компьютера и текущего пользователя

Лабораторная работа №2

Функцию написать в двух вариантах: для ANSI-строк и строк UNICODE (UTF-8).

- Написать функцию addslashes (char * str), которая вставляет символ обратного слэша (\) перед одинарными и двойными кавычками, встречающимися в строке.
- Написать функцию ltrim (char * str), которая удаляет пробелы в начале строки.
- Написать функцию comparestrings (char * str1, char * str2), которая сравнивает количество символов в строках. Возвращаемые значения: 1 - если str1 короче, чем str2; 2 - если str2 короче, чем str1; 3 - если количество символов в строках одинаково.

Лабораторная работа №3

- Создание одного из объектов ядра и получение описателя(handle) на него (функции Create: - CreateProcess, CreateMutex etc)
- Проверка, что объект создан, то есть описатель не равен NULL, иначе получить код ошибки (функция GetLastError), форматировать сообщение об ошибке (FormatMessage) и вывести его (с помощью MessageBox или в консоль)
- Получение параметров описателя объекта и вывод их на экран
- Дублирование описателя этого же объекта (функция DuplicateHandle)
- Проверка равны ли первоначальный описатель и его клон, вывод на экран результата
- Закрытие первого описателя (CloseHandle)
- Проверка существует ли еще объект ядра, то есть вызов любой функции, связанной с этим объектом (например для объекта File - ReadFile), проверка результата выполнения функции и вывод на экран сообщения об удачном/неудачном выполнении
- Закрытие второго описателя

- Проверка существует ли еще объект ядра и вывести результат на экран (аналогично пункту 7)

Лабораторная работа №4

1. Создать поток, который выводит два сообщения через MessageBox с интервалом в 5 секунд. После окончания его работы основной поток процесса выводит сообщение об этом.
2. Создать поток и вывести в нем параметры текущего процесса: заголовок окна, расстояния по осям X и Y от левого верхнего угла, ширину и высоту окна, дескрипторы стандартных устройств ввода и вывода.
3. Вывести параметры текущего процесса: заголовок окна, расстояния по осям X и Y от левого верхнего угла, ширину и высоту окна, дескрипторы стандартных устройств ввода и вывода. Затем создать процесс с заданными параметрами заголовка окна, ширины, высоты и смещения от левого верхнего угла экрана. После чего вывести те же параметры, что и для первого процесса.
4. Создать процесс и вывести его переменные окружения.

Лабораторная работа №5

1. Создать две функции потоков, которые выполняют следующие расчеты:

Поток 1: $a=(b+3)$; $b = (b-1)$;

Поток 2: $b = (b+2)$;

Каждая функция потока вычисляет свою формулу в цикле из ста итераций. При вычислении значения переменных a и b выводить на экран.

2. Произвести те же вычисления при помощи критических секций. После этого вывести значения переменных a и b на экран.

3. Написать две программы:

Программа 1 создает объект ядра «событие» выводит на экран текущее время и делает задержку на пять секунд, после чего освобождает «событие»; Перед задержкой запустить Программу 2;

Программа 2 ждет, пока освободится объект ядра «событие», после чего выводит на экран текущее время.

4. Создать 3 функции потоков, каждая из которых выводит в цикле из 25 итераций сообщение «Это поток № номер_потока». Сообщение заносить в строку, на которую

указывает глобальная переменная `str`. Сделать синхронизацию потоков при помощи мьютексов.

Лабораторная работа №6

1. Создать два дочерних процесса и соединить их анонимным каналом. Для этого использовать переопределение стандартных устройств ввода/вывода. Основная программа создает канал, процессы и передает им дескрипторы чтения/записи канала. Вторая программа получает от родительской дескриптор записи в канал в качестве стандартного устройства вывода и передает по каналу данные из файла (определяется программистом). Третья программа получает дескриптор чтения из канала в качестве стандартного устройства ввода, считывает через него информацию из канала и выводит ее на экран.
2. Вести широковещательную рассылку сообщений по сети при помощи почтовых ящиков. Сервер создает почтовый ящик и периодически записывает туда сообщения. Клиенты подключаются к этому ящику, считывают полученные сообщения и выводят их на экран.
3. Реализовать механизм клиент/сервер при помощи именованных каналов. Сервер передает клиенту аргументы, при помощи которых клиент проводит определенные вычисления (например, $y=3*a+5*b-2*c$). Результат вычислений передается обратно по каналу серверу, который осуществляет вывод на экран.

Лабораторная работа №7

Решить задачу обедающих философов ($n=5$).

Три процесса записывают в один и тот же файл текстовую строку. Первый процесс создает файл, второй процесс может работать с файлом, если первый успешно завершил свою работу, третий процесс дописывает в конец файла свою строку только в том случае, если второй процесс успешно выполнил свою работу и освободил файл.

Два процесса записывают данные в файл, один считывает и выводит их на экран. Доступ к файлу возможен только в монопольном режиме.

Лабораторная работа №8

Разработать драйвер устройства, имитирующий последовательный ввод/вывод.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

Требования к минимальному материально-техническому обеспечению

1. Компьютерный класс с развернутой ЛВС на базе ПЭВМ типа IBM PC (процессор Intel Pentium (Celeron) не ниже 1500 МГц, ОЗУ не менее 512 Mb RAM, HDD не менее 30 Gb);
2. Наличие программного обеспечения TASM, GCC, Virtual Box, MASM, Visual Studio2010.
3. Принтер (плоттер) для печати на бумаге формата А4.
4. LCD-проектор.
5. Экран.

Информационное обеспечение обучения.

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Канцедал С. А. Алгоритмизация и программирование: Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с.
2. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с.
3. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с.:
4. Гвоздева В.А. Введение в специальность программиста: Учебник / В.А. Гвоздева. - 2-е изд., испр. и доп. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 208 с
5. Языки программирования: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 3-е изд., перераб. и доп. - М.: Форум: ИНФРА-М, 2015. - 400 с.
6. Гуриков С.Р. Программирование в среде Lazarus для школьников и студентов: Учебное пособие/С.Р.Гуриков - М.: Форум, НИЦ ИНФРА-М, 2016. - 336 с.:
7. Исаченко О.В. Программное обеспечение компьютерных сетей: Учебное пособие / О.В. Исаченко. - М.: НИЦ ИНФРА-М, 2014. - 117 с.

Дополнительная литература:

1. Солдатов В.П. Программирование драйверов Windows. Изд. 3-е, перераб. и доп. — М.: ООО "Бином-Пресс", 2008 г. — 480 с: ил.
2. Несвижский Всеволод. Программирование аппаратных средств в Windows.-СПб.: БХВ-Петербург, 2004. — 880 с.: ил.
3. Фельфман С.К. Системное программирование на персональном компьютере. Изд. 2-е, - М.:Бук-пресс, 2006.-512с.
4. Пирогов В.Ю. Ассемблер для Windows. — М.: Издатель Молгачева С.В., 2002. —552 с.,

Контроль и оценка результатов освоения дисциплины

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения

(освоенные умения, усвоенные знания)

Формы и методы контроля и оценки результатов обучения

Умения:

Осуществлять разработку кода программного модуля на современных языках программирования;

Экспертная оценка работ студентов с использованием интерактивных технологий

Создавать программу по разработанному алгоритму как отдельный модуль;

Экспертная оценка работ студентов с использованием интерактивных технологий

Выполнять отладку и тестирование программы на уровне модуля;

Экспертная оценка работ студентов с использованием интерактивных технологий

Оформлять документацию на программные средства;

Экспертная оценка работ студентов с использованием интерактивных технологий

Использовать инструментальные средства для автоматизации оформления документации.

Экспертная оценка работ студентов с использованием интерактивных технологий

Знания:

Основные этапы разработки программного обеспечения;

Экспертная оценка работ студентов с использованием интерактивных технологий

Основные принципы технологии структурного и объектноориентированного программирования; использованием интерактивных технологий

Основные принципы отладки и тестирования программных продуктов;

Экспертная оценка работ студентов с использованием интерактивных технологий

Методы и средства разработки технической документации

Экспертная оценка работ студентов с использованием интерактивных технологий

КОНКРЕТИЗАЦИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Уметь:

Осуществлять разработку кода программного модуля на современных языках программирования;

Средства разработки DOS, Windows-программ, используемые при изучении дисциплины: TASM, MASM32, Visual Studio

Уметь:

Создавать программу по разработанному алгоритму как отдельный модуль;

Процедуры. Сокращение, структурирование исходного текста. Создание библиотек.

Уметь:

Выполнять отладку и тестирование программы на уровне модуля;

Основные принципы и стандарты; системные вызовы; интерфейсы WinAPI, POSIX API; 32 и 64 разрядные интерфейсы; проблема локализации, стандарты ANSI и UNICODE.

Уметь:

Оформлять документацию на программные средства;

Средства разработки DOS, Windows-программ, используемые при изучении дисциплины: TASM, MASM32, Visual Studio

Уметь:

Использовать инструментальные средства для автоматизации оформления документации.

Процедуры. Сокращение, структурирование исходного текста. Создание библиотек.

Знать:

Основные этапы разработки программного обеспечения;

Основные принципы и стандарты; системные вызовы; интерфейсы WinAPI, POSIX API; 32 и 64 разрядные интерфейсы; проблема локализации, стандарты ANSI и UNICODE.

Знать:

Основные принципы технологии структурного и объектноориентированного программирования;

Средства разработки DOS, Windows-программ, используемые при изучении дисциплины: TASM, MASM32, Visual Studio

Знать:

Основные принципы отладки и тестирования программных продуктов;

Процедуры. Сокращение, структурирование исходного текста. Создание библиотек.

Методы и средства разработки технической документации

Основные принципы и стандарты; системные вызовы; интерфейсы WinAPI, POSIX API; 32 и 64

разрядные интерфейсы; проблема локализации, стандарты ANSI и UNICODE.