



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Кубанский государственный университет»  
в г. Славянске-на-Кубани

УТВЕРЖДАЮ



Проректор по работе с филиалами  
ФГБОУ ВО «Кубанский  
государственный университет»

А.А. Евдокимов

15 октября 2022 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ**

специальность 09.02.07 Информационные системы и программирование

Краснодар 2022

Рабочая программа учебной дисциплины МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) среднего профессионального образования (далее – СПО) по специальности 09.02.07 Информационные системы и программирование (технологический профиль), утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, (зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936) и примерной основной образовательной программы по специальности 09.02.07 Информационные системы и программирование (утвержденная протоколом Федерального учебно-методического объединения по УГПС 09.00.00 от 15 июля 2021 г. №3).

Дисциплина	МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ
Форма обучения	очная
Учебный год	2023-2024
3 курс	5 семестр
всего 235 часа, в том числе:	
лекции	128 ч.
практические занятия	96 ч.
курсовое проектирование	—
самостоятельные занятия	2 ч.
консультация	3 ч.
промежуточная аттестация	6 ч.
форма итогового контроля	экзамен

Составитель: преподаватель  Б.А. Шишкин

Утверждена на заседании предметной (цикловой) комиссии физико-математических дисциплин и специальных дисциплин УГС 09.00.00 Информатика и вычислительная техника протокол №4 от «24» ноября 2022 г.

Председатель предметной (цикловой) комиссии

 М.С. Бушуев  
«24» ноября 2022 г.

Рецензенты:

Инженер-программист 1 категории  
отдела АСУТП управления АСУТП,  
КИПиА, МОП Краснодарского РПУ  
филиала «Макрорегион ЮГ» ООО ИК  
«СИБИНТЕК»

ООО ИК «СИБИНТЕК»  
Филиал «Макрорегион ЮГ»  
352000, г. Туапсе, ул. Дачная, 40  
ИНН 7705115801 ОГРН 770501001

 М.В. Литус

Директор ООО «Бизнес ассистент»

 Д.С. Зима

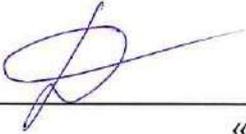
КРАСНОДАРСКИЙ ОБЛАСТНОЙ ЦЕНТР  
ОБРАЗОВАНИЯ И НАУКИ  
ИНН 7705000000 ОГРН 7705000000

ЛИСТ  
согласования рабочей программы по учебной дисциплине  
МДК.01.04 «Системное программирование»

Специальность среднего профессионального образования:  
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

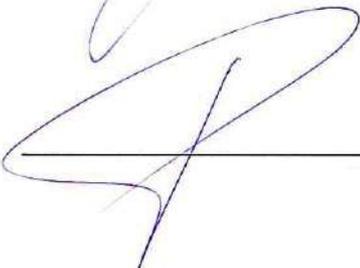
Нач. УМО филиала

  
\_\_\_\_\_ А.С. Демченко  
«25» ноября 2022 г.

Заведующая библиотекой филиала

  
\_\_\_\_\_ М.В. Фуфалько  
«25» ноября 2022 г.

Нач. ИВЦ (программно-  
информационное обеспечение  
образовательной программы)

  
\_\_\_\_\_ В.А. Ткаченко  
«25» ноября 2022 г.

## СОДЕРЖАНИЕ

1	ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
1.1	Область применения программы	5
1.2	Место учебной дисциплины в структуре программы подготовки специалистов среднего звена	5
1.3	Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины	5
1.4	Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)	5
2	СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	8
2.1	Объем учебной дисциплины и виды учебной работы	8
2.2	Структура дисциплины	8
2.3	Тематический план и содержание учебной дисциплины МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	8
2.4	Содержание разделов дисциплины	11
2.4.1	Занятия лекционного типа	11
2.4.2	Занятия семинарского типа	11
2.4.3	Практические занятия (Лабораторные занятия)	11
2.4.4	Содержание самостоятельной работы (Примерная тематика рефератов)	15
2.4.5	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	15
3	ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	14
3.1	Образовательные технологии при проведении лекций	14
3.2	Образовательные технологии при проведении практических занятий	14
4	УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	15
4.1	Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	15
4.2	Перечень необходимого программного обеспечения	15
5	ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	16
5.1	Основная литература	16
5.2	Дополнительная литература	16
5.3	Периодические издания	16
5.4	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	17
6	МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	19
7	КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	20
7.1	Паспорт фонда оценочных средств	20
7.2	Критерии оценки знаний	20
7.3	Оценочные средства для проведения текущей аттестации	21
7.4	Оценочные средства для проведения промежуточной аттестации	23
7.4.1	Примерные вопросы для проведения промежуточной аттестации	23
7.4.2	Примерные задачи для проведения промежуточной аттестации	25
8	ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	27

# **1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ**

## **1.1 Область применения программы**

Рабочая программа учебной дисциплины МДК.01.04 «Системное программирование» является частью основной профессиональной образовательной программой в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования (далее ФГОС СПО) и примерной основной образовательной программой для специальности 09.02.07 Информационные системы и программирование.

## **1.2 Место учебной дисциплины в структуре основной профессиональной образовательной программы**

Дисциплина «Системное программирование» относится к общепрофессиональным дисциплинам профессиональной подготовки.

## **1.3 Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины**

В результате освоения дисциплины обучающийся должен иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- использовании инструментальных средств на этапе отладки программного продукта;
- проведении тестирования программного модуля по определенному сценарию;
- использовании инструментальных средств на этапе отладки программного продукта;
- разработке мобильных приложений.

В результате освоения дисциплины обучающийся должен уметь:

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;
- оформлять документацию на программные средства.

В результате освоения дисциплины обучающийся должен *знать*:

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов.

Максимальная учебная нагрузка обучающегося 235 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 224 часа;
- самостоятельная работа обучающегося 2 часа;
- консультации 3 часа;
- промежуточная аттестация 6 часов.

#### 1.4. Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)

Освоение дисциплины «Системное программирование» способствует формированию у студентов следующих профессиональных компетенций:

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	иметь практический опыт
1	ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием	<p>Основные этапы разработки программного обеспечения.</p> <p>Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем.</p>	<p>Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства.</p> <p>Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.</p>	<p>Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.</p> <p>Разрабатывать мобильные приложения.</p>
2	ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.	<p>Основные принципы отладки и тестирования программных продуктов.</p> <p>Инструментарий отладки программных продуктов.</p>	<p>Выполнять отладку и тестирование программы на уровне модуля.</p> <p>Оформлять документацию на программные средства.</p> <p>Применять инструментальные средства отладки программного обеспечения.</p>	<p>Использовать инструментальные средства на этапе отладки программного продукта.</p> <p>Проводить тестирование программного модуля по определенному сценарию.</p>

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестр
		5
Учебная нагрузка (всего)	235	235
Аудиторная нагрузка (всего)	224	224
в том числе:		
лекционные занятия	128	128
практические занятия	96	96
Курсовое проектирование	–	–
Самостоятельная работа	5	5
в т.ч. консультации	2	2
Промежуточная аттестация – экзамен	6	6

### 2.2 Структура дисциплины

Освоение учебной дисциплины МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ включает изучение следующих разделов и тем:

№	Тема	Всего часов	Лекции	Практические	Самостоятельная работа
		226	128	96	2
1	<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	226	128	96	2

### 2.3 Тематический план и содержание учебной дисциплины

#### МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия	Объем в
		Программист
<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	<b>Содержание</b>	<b>224</b>
	1. Подсистемы управления ресурсами.	
	2. Управление процессами.	
	3. Управление потоками.	
	4. Параллельная обработка потоков.	
	5. Создание процессов и потоков.	
	6. Обмен данными между процессами. Передача сообщений.	
	7. Анонимные и именованные каналы.	
	8. Сетевое программирование сокетов.	
	9. Динамически подключаемые библиотеки DLL	
	10. Сервисы.	
	11. Виртуальная память. Выделение памяти процессам.	
	12. Работа с буфером экрана.	
<b>В том числе практических занятий и лабораторных</b>	<b>96</b>	
1. Использование потоков.		

	2. Обмен данными.	
	3. Сетевое программирование сокетов.	
	4. Работы с буфером экрана.	
	<b>Самостоятельная работа</b>	<b>2</b>
	<b>Консультация</b>	<b>3</b>
	<b>Итоговая аттестация</b>	<b>6</b>
<b>Всего</b>		<b>235</b>

## 2.4 Содержание разделов дисциплины

### 2.4.1 Занятия лекционного типа

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	<b>Тема 1.4.1 Программирование на языке низкого уровня</b>	1. Подсистемы управления ресурсами. 2. Управление процессами. 3. Управление потоками. 4. Параллельная обработка потоков. 5. Создание процессов и потоков. 6. Обмен данными между процессами. Передача сообщений. 7. Анонимные и именованные каналы. 8. Сетевое программирование сокетов. 9. Динамически подключаемые библиотеки DLL 10. Сервисы. 11. Виртуальная память. Выделение памяти процессам. 12. Работа с буфером экрана.	У, КР

Примечание: Р - написание реферата, У - устный опрос, КР - контрольная работа

### 2.4.2 Занятия семинарского типа

- не предусмотрены

### 2.4.3 Практические занятия (Лабораторные занятия)

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	<b>Тема 1.4.1 Программирование на языке низкого уровня</b>	1. Использование потоков. 2. Обмен данными. 3. Сетевое программирование сокетов. 4. Работы с буфером экрана.	ПР

Примечание: ПР- практическая работа

### 2.4.4 Содержание самостоятельной работы

- Подготовка к тестированию

## 2.4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа учащихся является важнейшей формой учебновоспитательного процесса.

Основная цель самостоятельной работы при изучении дисциплины - закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области архитектуры аппаратных средств.

НА САМОСТОЯТЕЛЬНУЮ РАБОТУ СТУДЕНТОВ ОТВОДИТСЯ 2 ЧАСОВ УЧЕБНОГО ВРЕМЕНИ.

Наименование раздела, темы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие / С. В. Белугина. — Санкт-Петербург : Лань, 2020. — 312 с. — ISBN 978-5-8114-4496-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/133920">https://e.lanbook.com/book/133920</a> .

Кроме перечисленных источников учащийся может воспользоваться поисковыми системами сети Интернет по теме самостоятельной работы.

Для освоения дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе может быть использовано следующее учебно-методическое обеспечение:

- методические рекомендации к выполнению лабораторных работ;
- методические рекомендации к самостоятельной работе.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Учащийся должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

### 3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для освоения курса «Системное программирование» предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе обучения применяются образовательные технологии личностно-деятельностного, развивающего и проблемного обучения. Обязателен лабораторный практикум по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

#### 3.1. Образовательные технологии при проведении лекций

Тема	Виды применяемых образовательных технологий	Кол. час
<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	Аудиовизуальная технология, технология развивающего обучения	128
	Итого	128

#### 3.2. Образовательные технологии при проведении практических занятий

Тема	Виды применяемых образовательных технологий	Кол. час
<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	Технология личностно-деятельностного обучения, технология проблемного обучения	96
	Итого	96

## **4 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ**

### **4.1 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатории Программного обеспечения и сопровождения компьютерных систем, оснащенные в соответствии с п. 6.2.1. Примерной программы по специальности:

Оснащенные базы практики, в соответствии с п 6.2.3 Примерной программы по специальности.

### **4.2 Перечень необходимого программного обеспечения**

1. 7-zip (лицензия на англ. <http://www.7-zip.org/license.txt>)
2. Adobe Acrobat Reader (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
3. Adobe Flash Player (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
4. Apache Open Office (лицензия - <http://www.openoffice.org/license.html>)
5. Free Commander (лицензия - <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>)
6. Google Chrome (лицензия - [https://www.google.ru/chrome/browser/privacy/eula\\_text.html](https://www.google.ru/chrome/browser/privacy/eula_text.html))
7. LibreOffice (в свободном доступе)
8. Mozilla Firefox (лицензия - <https://www.mozilla.org/en-US/MPL/2.0/>)
9. Oracle VM VirtualBox (лицензия - <https://www.virtualbox.org/wiki/GPL>)

## **5 ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

### **5.1 Основная литература**

1. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1172261>. – Режим доступа: по подписке.

2. Кузин, А. В. Программирование на языке Си : учебное пособие / А.В. Кузин, Е.В. Чумакова. — Москва : ФОРУМ : ИНФРА-М, 2020. — 143 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-556-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/961653>. – Режим доступа: по подписке.

### **5.2 Дополнительная литература**

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие / С. В. Белугина. — Санкт-Петербург : Лань, 2020. — 312 с. — ISBN 978-5-8114-4496-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/133920>. — Режим доступа: для авториз. пользователей.

2. Гуриков, С. Р. Программирование в среде Lazarus : учеб. пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2019. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-555-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/961652>. – Режим доступа: по подписке.

3. Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. - Москва : ИНФРА-М, 2021. - 317 с. - (Среднее профессиональное образование). - ISBN 978-5-16-015995-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1074164>. – Режим доступа: по подписке.

4. Основы программирования : учебник / Макарова Н.В., под ред., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. — Москва : КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL: <https://book.ru/book/936582>. — Текст : электронный.

5. Программирование: математическая логика : учебное пособие для среднего профессионального образования / М. В. Швецкий, М. В. Демидов, А. В. Голанова, И. А. Кудрявцева. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 675 с. — (Профессиональное образование). — ISBN 978-5-534-13248-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/457284>.

### **5.3 Периодические издания**

1. Computerworld Россия. – URL: <http://dlib.eastview.com/browse/publication/64081/udb/2071>.

2. Windows IT Pro / Re. – URL: <http://dlib.eastview.com/browse/publication/64079/udb/2071>.

3. БИТ. Бизнес & информационные технологии – URL : <http://dlib.eastview.com/browse/publication/66752/udb/2071>.
4. Виртуализация. Облачные структуры. Системы хранения данных. – URL : <https://dlib.eastview.com/browse/publication/84826/udb/2071>.
5. Информатика, вычислительная техника и инженерное образование. - URL: [https://www.elibrary.ru/title\\_about.asp?id=32586](https://www.elibrary.ru/title_about.asp?id=32586).
6. Информационно-управляющие системы. – URL: <http://dlib.eastview.com/browse/publication/71235>.
7. Мир больших данных. – URL : <https://dlib.eastview.com/browse/publication/90728/udb/2071>.
8. Мир ПК. – URL: <http://dlib.eastview.com/browse/publication/64067/udb/2071>.
9. Прикладная информатика. – URL: [https://e.lanbook.com/journal/2067#journal\\_name](https://e.lanbook.com/journal/2067#journal_name).
10. Программные продукты и системы. – URL: <https://dlib.eastview.com/browse/publication/64086/udb/2071>.
11. Программные продукты и системы. – URL: <http://dlib.eastview.com/browse/publication/64086/udb/2071>.
12. САПР и графика. - URL: <https://sapr.ru/list>,
13. Системный администратор. – URL: <https://dlib.eastview.com/browse/publication/66751/udb/2071>.
14. Системный анализ и прикладная информатика. – URL: [https://e.lanbook.com/journal/2420#journal\\_name](https://e.lanbook.com/journal/2420#journal_name).

#### **5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. ЭБС «BOOK.ru» [учебные издания – коллекция для СПО] : сайт. – URL: <https://www.book.ru/cat/576>.
2. ЭБС «Университетская библиотека ONLINE» [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы; мультимедийная коллекция, карты, онлайн-энциклопедии, словари] : сайт. – URL: [http://biblioclub.ru/index.php?page=main\\_ub\\_red](http://biblioclub.ru/index.php?page=main_ub_red).
3. ЭБС издательства «Лань» [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы] : сайт. – URL: <http://e.lanbook.com>.
4. ЭБС «Юрайт» [учебники и учебные пособия издательства «Юрайт»] : сайт. – URL: <https://urait.ru/>.
5. ЭБС «Znanium.com» [учебные, научные, научно-популярные материалы различных издательств, журналы] : сайт. – URL: <http://znanium.com/>.
6. Научная электронная библиотека. Монографии, изданные в издательстве Российской Академии Естествознания [полнотекстовый ресурс свободного доступа] : сайт. – URL: <https://www.monographies.ru/>.
7. Научная электронная библиотека статей и публикаций «eLibrary.ru» [российский информационно-аналитический портал в области науки, технологии, медицины, образования; большая часть изданий – свободного доступа] : сайт. – URL: <http://elibrary.ru>.
8. Базы данных компании «Ист Вью» [периодические издания (на русском языке)] : сайт. – URL: <http://dlib.eastview.com>.
9. Российская электронная школа : государственная образовательная платформа [полный школьный курс уроков] : сайт. – URL: <https://resh.edu.ru/>.
10. Единое окно доступа к образовательным ресурсам : федеральная информационная система свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для всех уровней образования: дошкольное, общее, среднее профессиональное, высшее, дополнительное : сайт. – URL: <http://window.edu.ru>.
11. Федеральный центр информационно-образовательных ресурсов [для общего,

среднего профессионального, дополнительного образования; полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://fcior.edu.ru>.

12. Единая коллекция цифровых образовательных ресурсов [для преподавания и изучения учебных дисциплин начального общего, основного общего и среднего (полного) общего образования; полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://school-collection.edu.ru>.

13. Официальный интернет-портал правовой информации. Государственная система правовой информации [полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://publication.pravo.gov.ru>.

14. Кодексы и законы РФ. Правовая справочно-консультационная система [полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://kodeks.systems.ru>.

15. ГРАМОТА.РУ : справочно-информационный интернет-портал : сайт. – URL: <http://www.gramota.ru>.

16. Энциклопедиум [Энциклопедии. Словари. Справочники : полнотекстовый ресурс свободного доступа] // ЭБС «Университетская библиотека ONLINE» : сайт. – URL: <http://enc.biblioclub.ru/>.

17. СЛОВАРИ.РУ. Лингвистика в Интернете : лингвистический портал : сайт. – URL: <http://slovari.ru/start.aspx?s=0&p=3050>.

18. Электронный каталог Кубанского государственного университета и филиалов. – URL: <http://212.192.134.46/MegaPro/Web/Home/About>.

## **6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Учащиеся для полноценного освоения курса «Системное программирование» должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа учащихся является важнейшей формой учебно- познавательного процесса. Цель заданий для самостоятельной работы - закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины «Системное программирование»; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Отчеты по практическим занятиям должны содержать полные ответы на поставленные задания, необходимые таблицы должны быть заполнены.

### *Общие правила выполнения письменных работ*

На первом занятии студенты должны быть проинформированы о необходимости соблюдения норм академической этики и авторских прав в ходе обучения. В частности, предоставляются сведения:

- общая информация об авторских правах;
- правила цитирования;
- правила оформления ссылок;

Все имеющиеся в тексте сноски тщательно выверяются и снабжаются «адресами».

Недопустимо включать в свою работу выдержки из работ других авторов без указания на это, пересказывать чужую работу близко к тексту без отсылки к ней, использовать чужие идеи без указания первоисточников (это касается и информации, найденной в Интернете). Все случаи плагиата должны быть исключены.

Список использованной литературы должен включать все источники информации, изученные и проработанные студентом в процессе выполнения работы, и должен быть составлен в соответствии с ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила».

## 7 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»

### 7.1 Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Компетенции	Наименование оценочного средства
1.	<i>Тема 1.4.1 Программирование на языке низкого уровня</i>	ПК 1.2, ПК 1.3	Проверка конспектов, устный опрос,

### 7.2 Критерии оценки знаний

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических работ, тестирования, собеседования по результатам выполнения лабораторных работ, а также решения задач, составления рабочих таблиц и подготовки сообщений к уроку. Знания студентов на практических занятиях оцениваются отметками «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки (<b>на указанном языке программирования</b>) методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки (<b>на указанном языке программирования</b>) методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки (<b>на указанном языке программирования</b>) методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/зачет в форме собеседования:</p> <p>практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
---	---	--

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств	<p>Оценка «<b>отлично</b>» - выполнена отладка модуля (с использованием инструментария среды проектирования); с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка «<b>хорошо</b>» - выполнена отладка модуля (с использованием инструментария среды проектирования); сохранены и представлены результаты отладки.</p> <p>Оценка «<b>удовлетворительно</b>» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/зачет в форме собеседования:</p> <p>практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
--	---	--

### 7.3 Оценочные средства для проведения текущей аттестации

- фронтальный опрос
- индивидуальный устный опрос
- письменный контроль
- тестирование по теоретическому материалу
- практическая (лабораторная) работа

Форма аттестации	Знания	Умения	Владения (навыки)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	оценка умения различать некоторые понятия	Оценка навыков логического анализа и синтеза при сопоставлении некоторых понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

*Контрольная работа.* Контрольная работа является набором практических заданий и задач по темам изучаемой дисциплины, позволяющих формировать знания, а также умения обучающихся в области архитектуры аппаратных средств.

## **Разработка и проектирование баз данных**

### **Задание 1. Разработка проекта «База данных»**

#### **Напишите программу работы с локальной базой данных.**

##### **Вариант 1. Учет авиапассажиров.**

Таблица авиапассажиров: код авиапредприятия, код города назначения, номер рейса, дата вылета, фамилия, адрес пассажира.

Справочники: авиапредприятия (код, наименование, адрес, телефон), города (код и наименование).

Отчет по авиапредприятиям и городам с числом пассажиров по авиапредприятиям и городам и в целом по отчету.

##### **Вариант 2. Учет выпуска изделий.**

Таблица выпущенных изделий: код изделия, код предприятия, количество выпущенных изделий (шт.), их себестоимость, отпускная цена, затраты на электроэнергию, затраты на зарплату, дата выпуска.

Справочники: изделия (код, наименование), предприятия (код, наименование, адрес, телефон).

Отчет по изделиям и предприятиям со средними затратами по каждому изделию.

##### **Вариант 3. Учет поставок товаров.**

Таблица поставленных товаров: код поставщика, код товара, количество, цена, единица измерения, дата поставки.

Справочники: поставщики (код, наименования, адрес, телефон), товары (код, наименование).

Отчет по товарам и поставщикам с итоговой стоимостью по товарам, поставщикам и по отчету.

##### **Вариант 4. Учет уволившихся с предприятия.**

Таблица уволенных: код подразделения, код причины увольнения, дата увольнения, фамилия, номер приказа.

Справочники: подразделения (код, наименование), причины увольнения (код и наименование).

Отчет по причинам увольнения и подразделениям с итоговым числом уволившихся по причинам увольнения, подразделениям и по отчету.

##### **Вариант 5. Учет рождаемости.**

Таблица новорожденных: код населенного пункта, код национальности; вес, рост новорожденного, дата рождения, фамилия, имя, отчество.

Справочники: населенные пункты (код, наименование), национальности (код, наименование).

Отчет по населенным пунктам и национальностям с итоговым числом новорожденных по населенным пунктам, национальностям и по отчету.

##### **Вариант 6. Учет участников Олимпиады.**

Таблица олимпийцев: код страны, код вида спорта, фамилия участника, дата рождения, рост, вес, пол.

Справочники: страны (код, наименование, начальник команды, телефон), виды спорта (код, наименование).

Отчет по странам и видам спорта с итогами по странам и видам спорта: числоучастников, средний вес, средний рост.

**Вариант 7. Учет проданных товаров.**

Таблица проданного товара: код торгового предприятия, код товара, стоимость проданного товара за последние 3 года (три поля).

Справочники: торгующие предприятия (код, наименование, директор, телефон), товары (код, наименование).

Отчет по товарам и торгующим предприятиям с итогами по товарам и предприятиям.

**Вариант 8. Учет малых предприятий.**

Таблица малых предприятий: код населенного пункта, код основного вида деятельности МП, численность работающих, адрес, объем выполняемых работ за год (всего) и по основному виду деятельности, директор, дата создания.

Справочники: населенные пункты (код, наименование), виды деятельности (код, наименование).

Отчет по населенным пунктам и видам деятельности с итогами по видам деятельности, населенным пунктам и по отчету в целом.

**Вариант 9. Учет больных в больнице.**

Таблица больных: фамилия больного, код диагноза, дата начала лечения в больнице, адрес, код подразделения, стоимость лечения за один день, дата выписки, дата рождения больного.

Справочники: диагнозы (код, наименования), подразделения (код наименования, заведующий).

Отчет стоимости лечения по подразделениям и диагнозам с итогами стоимости лечения по подразделениям, диагнозам и по больнице в целом.

**Вариант 10. Учет движения общественного транспорта.**

Таблица движения общественного транспорта: код типа транспорта, номер и длинамаршрута, табельный номер водителя, дата выхода.

Справочники: типы транспорта (код, наименования), водители (табельный номер, фамилия).

Отчет по видам транспорта и маршрутам с итогами (километраж) по видам транспорта, маршрутам и по отчету.

**Вариант 11. Учет платежных поручений в банке.**

Таблица платежных поручений: код клиента, код вида платежа, дата, сумма, номерпоручения.

Справочники: клиенты банка (код, наименование, адрес, директор, телефон), видыплатежей (код, наименование).

Отчет по клиентам и видам платежа с итогами по клиентам, видам платежа и в целомпо отчету.

**Вариант 12. Учет договоров займа.**

Таблица договоров займа: код организации, номер договора, дата заключения, дата окончания, сумма, процент, сумма возврата, возвращенная сумма, особые условия, процент штрафа за каждый просроченный день, код денежной единицы.

Справочники: организации (код, наименование, адрес, директор, телефон); денежныединицы (код, обозначение, наименование).

Отчет по видам валют и дням возврата с итогами по видам валют и дням возврата.

**Вариант 13. Учет проданных ценных бумаг.**

Таблица проданных ценных бумаг: код эмитента, код типа ценной бумаги, число, номинальная стоимость, курсовая стоимость, дата выпуска, номер регистрации выпуска ценных бумаг.

Справочники: эмитенты (код, наименование, адрес, директор, телефон), типы ценныхбумаг (код, наименование).

Отчет по типам ценных бумаг и эмитентам с итогами по типам ценных бумаг и эмитентам: суммарное число бумаг, средняя стоимость.

**14. Учет кадров.**

Таблица сотрудников предприятия: табельный номер работника, фамилия, дата рождения, код специальности, код подразделения, дата приема, дата начала трудовой деятельности, оклад, образование.

Справочники: специальности (код, наименование); подразделения (код, наименование, фамилия начальника, телефон).

Отчет по подразделениям и специальностям с итогами (число принятых) по подразделениям, специальностям и по предприятию.

**Вариант 15. Учет очередников на получение жилья.**

Таблица очередников: фамилия очередника, номер очереди, номер приказа, код организации, код основания включения в очередь, дата, паспортные данные, число членов семьи, адрес.

Справочники: организации (код, наименование, адрес, директор, телефон); основания включения в очередь (код, наименование).

Список очередников по предприятиям и основаниям включения в очередь и числом очередников по предприятиям и основаниям включения в очередь.

**Вариант 16. Учет покупки и продажи валюты.**

Таблица продажи и покупки валюты: дата, фамилия, паспортные данные, код валюты, количество валюты, проданной и купленной, количество рублей, код обменного пункта.

Справочники: валюта (код, наименование, текущий курс) и обменные пункты (код, наименование).

Отчет по видам валют и обменным пунктам с итогами по видам валют и обменным пунктам (сумма проданной и купленной валюты).

**Вариант 17. Учет объектов строительства.**

Таблица объектов строительства: код населенного пункта, сметная стоимость, стоимость выполненных работ, дата начала строительства, наименование объекта, код заказчика.

Справочники: населенные пункты (код, наименование), заказчики (код, наименование, адрес, телефон).

Отчет по заказчикам и населенным пунктам с итогами по заказчикам, населенным пунктам и по отчету в целом (сметная стоимость и объем выполненных работ).

**Вариант 18.** Учет книг в библиотеке.

Таблица книг: код темы, наименование книги, авторы, дата, число экземпляров, адресхранения, код издательства.

Справочники: темы (код, наименование), издательства (код, наименование, адрес, телефон).

Отчет по темам и издательствам с итоговым числом книг по темам, издательствам и по отчету в целом.

**Вариант 19.** Учет успеваемости студентов.

Таблица успеваемости студентов: код предмета, дата, оценка, номер зачетной книжки.

Справочники: предметы (код и наименование), студенты (группа, номер зачетной книжки, фамилия).

Отчет по группам и студентам со средней оценкой по группам, студентам и в целом по ВУЗу.

**Вариант 20.** Каталог почтовых марок. Концептуальные требования:

Номер Год издания Тематика Ф.И.О. художника Сюжет Номинал Стоимость Тираж Способ печати Размеры ФотоРежимы работы:

Сортировка по полям – Номер, Год издания, Тематика, Способ печати. Поиск по полям - ФИО художника, Номер, Номинал.

Фильтрация по полям на диапазон значений- ФИО художника, Год издания, Способ печати.

**Вариант 21.** Каталог монет России. Концептуальные требования:

Номер Номинал Материал Размер Вес Год выпуска Ф.И.О. гравера Монетный двор Тираж Редкость ФотоРежимы работы:

Сортировка по полям – Номер, Год выпуска, Материал, Монетный двор. Поиск по полям - ФИО гравёра, Номер, Номинал, Редкость.

Фильтрация по полям на диапазон значений - ФИО гравёра, Год выпуска, Материал, Номинал

**Вариант 22.** Каталог сыров. Концептуальные требования:

Наименование Категория Способ хранения Срок хранения Употребление Форма выпуска Завод изгот. Ф.И.О. директора Адрес завода Фото Режимы работы:

Сортировка по полям – Наименование, Завод изготовитель, Срок хранения. Поиск по полям - ФИО директора, Наименование.

Фильтрация по полям на диапазон значений - Наименование, Категория, Срок хранения, Завод изготовитель, Форма выпуска.

**Вариант 23.** Автомобили завода ГАЗ. Концептуальные требования:

Наименование Тип Кузов Двигатель Ходовые мосты Дата начала вып. Кол-во Ф.И.О. инженера Масса ФотоРежимы работы:

Сортировка по полям – Наименование, Двигатель, Дата начала выпуска

Поиск по полям - ФИО инженера, Наименование, Масса.

Фильтрация по полям на диапазон значений - Тип, Кузов, Двигатель, Дата начала выпуска.

**Вариант 24.** Исторические здания г. Москвы. Концептуальные требования:

Наименование Адрес Дата постройки Ф.И.О. архитект. Владелец Размеры  
Этажность Фото Дата ремонта

Режимы работы:

Сортировка по полям – Наименование, Дата постройки, Адрес. Поиск по полям - ФИО архитектора, Дата постройки, Наименование.

Фильтрация по полям на диапазон значений- ФИО архитектора, Дата постройки, Владелец.

**Вариант 25.** Каталог периодических изданий Концептуальные требования:

Наименование Ф.И.О. гл. редакт. Вид Цена 1 экз. Периодичность Объем Тираж  
Подп. Индекс Объем подписки

Режимы работы:

Сортировка по полям – Наименование, Объём, Тираж, Подписной индекс. Поиск по полям - Наименование, ФИО главного редактора, Цена.

Фильтрация по полям на диапазон значений - Вид, Периодичность, Тираж, Объём, Цена.

#### 7.4 Оценочные средства для проведения промежуточной аттестации

Форма аттестации	Знания	Умения	Владение (навыки)	Личные качества студента	Примеры оценочных средств
Итоговая аттестация					
Экзамен	Контроль знания базовых положений в области операционных систем	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов, работы и администрирования операционной системы	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области операционных систем	Оценка навыков логического мышления при решении задач в области операционных систем	Оценка способности грамотно и четко излагать ход решения задач в области архитектуры операционных систем и аргументировать результаты	Задачи прилагаются

##### 7.4.1 Примерные вопросы для проведения промежуточной аттестации

1. Определение массива переменных
2. Синтаксис массива и возможные формы его объявления
3. Размерность массива
4. Правила объявления многомерных массивов
5. Способы инициализации массива
6. Размещение массива в памяти
7. Определение вектора
8. Инициализация вектора
9. Использование вектора на практике

10. 10. Вектор как замена массива
11. 11. Указатели. Указатели на переменные
12. 12. Константные указатели
13. 13. Указатели на массивы
14. Указатели на указатели
15. Определение функции
16. Пример построения функции
17. Рекурсия
18. Объявление функций
19. Определение функции sum
20. Вызов функции
21. Определение прикладной программы
22. Определение веб-приложения
23. Основные отличия прикладной программы от веб-приложения
24. Язык программирования C++. Нововведения языка в сравнении с языком Си
25. 25. Язык программирования C++. Его основные особенности
26. Язык программирования C++. Технический обзор
27. Инкапсуляция
28. Наследование
29. Полиморфизм
30. Достоинства языка C++
31. Недостатки языка C++
32. Классификация программного обеспечения
33. Основные этапы разработки программ и их краткая характеристика
34. 34. Определение информационной системы
35. Определение автоматизированной информационной системы
36. Определение Многоуровневое представление ИС
37. Определение Аппаратное обеспечение ИС
38. Определение Программное обеспечение
39. Определение Программист
40. Определение Пользователь
41. Определение Прикладное ПО
42. Определение Пакеты прикладных программ
43. Основные особенности ППП и их характеристики
44. Структура и основные компоненты ППП
45. Определение Входной язык
46. Определение и основные характеристики предметного обеспечения

47. Определение и основные характеристики системного обеспечения
48. Этапы развития ППП
49. Первое поколение ППП
50. Второе поколение ППП
51. Третье поколение ППП
52. Четвертое поколение ППП
53. Алгоритмическая декомпозиция
54. Объектно-ориентированная декомпозиция
55. Понятие объектно-ориентированного программирования
56. Определение жизненного цикла
57. Основные этапы жизненного цикла
58. Компиляция программы
59. Сборка исполняемого модуля
60. Исходный код
61. Исполняемый модуль
62. Именованное пространство
63. Ссылки
64. Правила определения времени жизни переменной
65. Динамическое распределение данных в памяти
66. Одномерные массивы
67. Двумерные массивы
68. Структуры
69. Стек, очередь
70. Дерево
71. Граф
72. Выражения
73. Операции
74. Арифметические операции
75. Условные конструкции
76. Циклические конструкции
77. Селективные конструкции
78. Потоки ввода-вывода

#### **7.4.2 Примерные задачи для проведения промежуточной аттестации**

1. Задание: составить программу вычисления  $X$  в зависимости от значения символа  $ch$  по формуле, соответствующей варианту. Значение символа  $ch$  вводится с клавиатуры. Для контроля за правильностью работы программы использовать модуль

Ю.ASM. Формулы для расчета:

$(8g)/(5m)$ , если  $ch > 'c'$   $X = d+g-6$ , если  $ch = 'c'$

$3v-4d+100$ , если  $ch < 'c'$

2. Задание: составить программу вычисления  $X$  в зависимости от значения символа  $ch$  по формуле, соответствующей варианту. Значение символа  $ch$  вводится с клавиатуры. Для контроля за правильностью работы программы использовать модуль Ю.ASM. Формулы для расчета:

$3+vg-m$ , если  $ch > 't'$

$X = (10d)/(7m)$ , если  $ch = 't'$

$m-7+3d$ , если  $ch < 't'$

3. Составить программу вычисления  $y$  по формуле:  $y=2m-7g+4-d*v$

4. Составить программу вычисления  $y$  по формуле:  $y=2m+8-g*d+5v$

5. Найти в DSEG: 1) 3-й по порядку нулевой байт; 2) 4-й по порядку код CR (0Dh);

3) 4-й байт из числа тех, которые ниже 20h; 4) 3-й по порядку код '\$'(24h);

6. Найти в DSEG: 1) байт, следующий за 3-м кодом ';' (3Bh); 2) 4-й байт из числа больших, чем 29h; 3) байт, следующий за 3-м отрицательным байтом;

7. Найти в DSEG: 1) байт, являющийся 4-м нечетным; 2) байт, следующий за 3-м кодом пробела (20h); 3) 3-й байт из числа тех, которые выше 10h;

8. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Если символ в нижнем регистре, перевести его в верхний регистр; если в верхнем – в нижний

9. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Вывести строку в обратном порядке

10. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Вывести строку, в закодированном виде, от каждого кода символа строки отнимается число 10.

11. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Удалить все символы в верхнем регистре;

12. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Найти позицию символа (вводится с клавиатуры) в строке и вывести позицию (и) в шестнадцатеричном виде.

13. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления  $y$  по формуле:  $y=5v-d*m+6g+3$

14. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления у по формуле:  $y=5d-g*v+7m-2$

15. Вычислить:  $(A*B-C)/D$

16. Найдите ошибку в следующих командах: MOV DS, @DATA  
MOV AX, ES MOV DS, AX MOV CS, AX

MOV (ячейка памяти), (ячейка памяти)

17. Найдите ошибку в следующих командах: XCNG AX, BX  
XCHG CX, BL XCHG ES, DS XCHG DATA1, AX

18. Вычисление результата выполнения арифметического выражения, в котором некоторые числа постоянны, а другие переменные.

Формула вычислений:  $X = (A * 2 + B * C) / (D - 3)$

19. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления у по формуле:  $y=2v+6d-m*g+3$

20. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления у по формуле:  $y=g*m-7d+v-20$

21. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления у по формуле:  $y=g*m-4d+8v-7$

22. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)
- возраст студента ( количество полных лет ) – число v (байт) Задание:

Составить программу вычисления у по формуле:  $y=2+m*d-3g+7v$

## 8 ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### ЛЕКЦИЯ 3. ПОДСИСТЕМЫ УПРАВЛЕНИЯ РЕСУРСАМИ.

Сердцем операционной системы UNIX является подсистема управления процессами. Практически все действия ядра имеют отношение к процессам, будь то обслуживание системного вызова, генерация сигнала, размещение памяти, обработка особых ситуаций, вызванных выполнением процесса или обеспечением услуг ввода/вывода по запросу прикладного процесса.

Вся функциональность операционной системы в конечном счете определяется выполнением тех или иных процессов, как и уровни выполнения системы (run levels) представляют собой ни что иное, как удобную форму определения группы выполняющихся процессов. Возможность терминального или сетевого доступа к системе, различные сервисы, традиционные для UNIX, — система печати, удаленные архивы FTP, электронная почта и система телеконференций (news) — все это результат выполнения определенных процессов. Процессы в UNIX неотъемлемо связаны с двумя важнейшими ресурсами системы процессором и оперативной памятью. Как правило, этих ресурсов никогда не бывает "много", и в операционной системе происходит активная конкурентная борьба за право обладания процессором и памятью. И по скольку UNIX является многозадачной системой общего назначения, задача справедливого распределения этого ресурса между задачами различного класса и с различными требованиями является нетривиальной.

По скольку запущенная на выполнение программа порождает в системе один или более процессов (или задач). Подсистема управления процессами контролирует:

- Создание и удаление процессов
- Распределение системных ресурсов (памяти, вычислительных ресурсов)

между процессами

- Синхронизацию процессов
- Межпроцессное взаимодействие

Очевидно, что в общем случае число активных процессов превышает число процессоров компьютера, но в каждый конкретный момент времени на каждом процессоре может выполняться только один процесс. Операционная система управляет доступом процессов к вычислительным ресурсам, создавая ощущение одновременного выполнения нескольких задач. Специальная задача ядра, называемая планировщиком процессов (scheduler), разрешает конфликты между процессами в конкуренции за системные ресурсы (процессор, память, устройства ввода/вывода). Планировщик запускает процесс на выполнение, следя за тем, чтобы процесс монополично не захватил разделяемые системные ресурсы. Процесс освобождает процессор, ожидая длительной операции ввода/вывода, или по прошествии кванта времени. В этом случае планировщик выбирает следующий процесс с наивысшим приоритетом и запускает его на выполнение.

**Модуль управления памятью** обеспечивает размещение оперативной памяти для прикладных задач. В случае, если для всех процессов недостаточно памяти, ядро перемещает части процесса или нескольких процессов во вторичную память (как правило, в специальную область

жесткого диска), освобождая ресурсы для выполняющегося процесса.

Все современные системы реализуют так называемую виртуальную память: процесс выполняется в собственном логическом адресном пространстве, которое может значительно превышать доступную физическую память. Управление виртуальной памятью процесса также входит в задачи модуля управления памятью.

**Модуль межпроцессного взаимодействия** отвечает за уведомление процессов о событиях с помощью сигналов и обеспечивает возможность передачи данных между различными процессами.

Основы управления процессом

Процесс UNIX представляет собой исполняемый образ программы, включающий отображение в памяти - исполняемого файла, полученного в результате компиляции, то есть трансляции модулей программы, языков высокого уровня, в эквивалентные программные модули языка низкого уровня, как то: стек, код и данные библиотек, а также ряд структур данных ядра, необходимых для управления процессом, что есть весьма важный момент для дальнейшего понимания принципов распределения памяти, как одного из важнейших ресурсов в "конкуренции" процессов, то есть данные - структурированы!, условно на:

- Stack (стек) — область памяти, в которой программа хранит информацию о вызываемых функциях, их аргументах и каждой локальной переменной в функциях. Размер области может меняться по мере работы программы. При вызове функций стек увеличивается, а при завершении — уменьшается.

- Heap (куча) — это область памяти, в которой программа может делать всё, что угодно. Размер области может меняться. Программист имеет возможность воспользоваться частью памяти кучи с помощью функции malloc(), и тогда эта область памяти увеличивается. Возврат ресурсов осуществляется с помощью free(), после чего куча уменьшается.

- Code (кодовый сегмент) — это область памяти, в которой хранятся машинные инструкции скомпилированной программы. Они генерируются компилятором, но могут быть написаны и вручную. Обратите внимание, что эта область памяти также может быть разделена на три части (текст, данные и BSS). Эта область памяти имеет фиксированный размер, определяемый компилятором. **UNIX.**

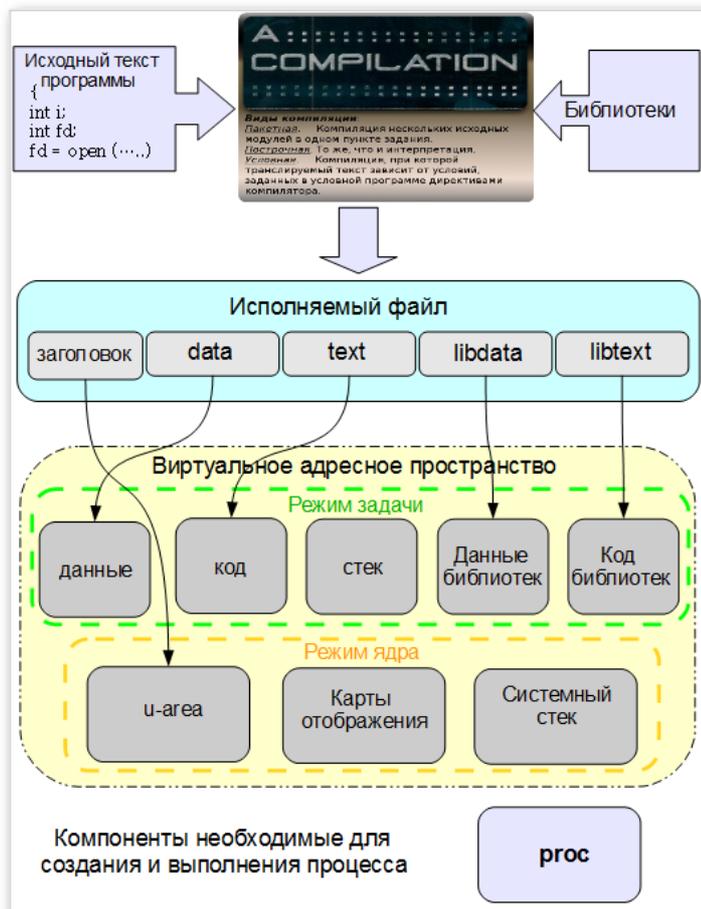
**Профессиональное программирование ст.259**

**Стек и Куча**

Это в дальнейшем(в том числе) и предопределяет возникновение(существование) сегментов и страниц. Процесс во время выполнения использует различные системные ресурсы - память, процессор, услуги файловой подсистемы и подсистемы ввода/вывода. Суровая правда любого современного вычислительного комплекса в том, что один процессор может обслуживать только один процесс в единицу времени, что в свою очередь предопределило возникновение(существование) "планировщика", по средствам которого операционная система UNIX обеспечивает иллюзию одновременного выполнения нескольких процессов, эффективно распределяя системные ресурсы между активными процессами и не позволяя в то же время ни одному из них монополизировать использование этих ресурсов.

Новорожденная операционная система UNIX обеспечивала выполнение всего двух процессов, по одному на каждый подключенный к PDP-7 терминал. Спустя год, на той же PDP-7 число процессов заметно увеличилось, появился системный вызов **fork**. В Первой редакции UNIX появился вызов **exec**, но операционная система по прежнему позволяла размещать в памяти только один процесс в каждый момент времени. После реализации аппаратной подсистемы управления памятью на PDP-11 операционная система была модифицирована, что позволило загружать в память сразу несколько процессов, уменьшая тем самым время на сохранение образа процесса во вторичной

памяти (на диске) и считывание его, когда процесс продолжал выполнение. Однако до 1972 года UNIX нельзя было назвать действительно многозадачной системой, т. к. операции ввода/вывода оставались синхронными, и другие процессы не могли выполняться, пока их "коллега" не завершал операцию ввода/вывода. Истинная многозадачность появилась только после того, как код UNIX был переписан на языке C в 1973 году. С тех пор основы управления процессами практически не изменились.



Выполнение процесса может происходить в двух режимах — в режиме ядра (**kernel mode**) или в режиме задачи (**user mode**). В режиме задачи процесс выполняет инструкции прикладной программы, допустимые на непривилегированном уровне защиты процессора. При этом процессу недоступны системные структуры данных. Когда процессу требуется получение каких-либо услуг ядра, он делает системный вызов, который выполняет инструкции ядра, находящиеся на привилегированном уровне.

Несмотря на то что выполняются инструкции ядра, это происходит от имени процесса, сделавшего системный вызов. Выполнение процесса при этом переходит в режим ядра. Таким образом ядро системы защищает собственное адресное пространство от доступа прикладного процесса, который может нарушить целостность структур данных ядра и привести к разрушению операционной системы.

Более того, часть процессорных инструкций, например, изменение регистров, связанных с управлением памятью могут быть выполнены только в режиме ядра.

Соответственно и образ процесса состоит из двух частей: данных режима ядра и режима задачи. Образ процесса в режиме задачи состоит из сегмента кода, данных, стека, библиотек и других структур данных, к которым он может получить непосредственный доступ. Образ процесса в режиме ядра состоит из структур данных, недоступных процессу в режиме задачи, которые используются ядром для управления процессом. Сюда относятся данные, диктуемые аппаратным уровнем, например состояния

регистров, таблицы для отображения памяти и т. д., а также структуры данных, необходимые ядру для обслуживания процесса. Вообще говоря, в режиме ядра процесс имеет доступ к любой области памяти.

### Структуры данных процесса

Каждый процесс представлен в системе двумя основными структурами данных **proc** и **user**,

описанными, соответственно, в файлах *sys/proc.h* и *sys/user.h*. Содержимое и формат этих структур различны для разных версий UNIX.

.....  
<https://ru.wikipedia.org/wiki/Procfs>

<https://www.opennet.ru/docs/RUS/lkmpg/node16.html>

[https://www.opennet.ru/docs/RUS/kernel2prog/linux\\_khg\\_03.html](https://www.opennet.ru/docs/RUS/kernel2prog/linux_khg_03.html)

<http://linux.yaroslavl.ru/docs/setup/mandrake/cl/proc-fs.html>

<http://www.ibm.com/developerworks/ru/library/l-proc/> <-----

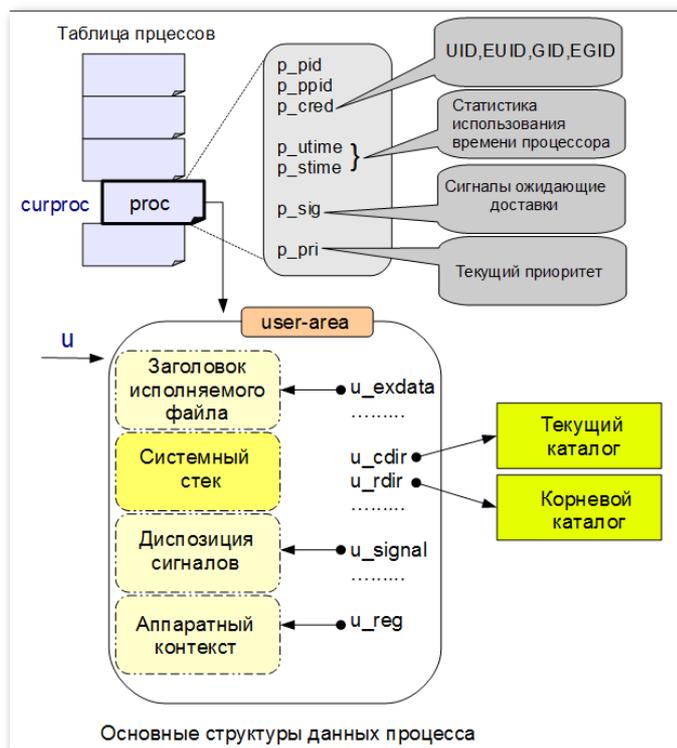
[https://m.habr.com/ru/company/\\*nix](https://m.habr.com/ru/company/*nix) <-----  
.....

В любой момент времени данные структур **proc** для всех процессов должны присутствовать в памяти, хотя остальные структуры данных, включая образ процесса, могут быть перемещены во вторичную память, область свопинга. Это позволяет ядру иметь под рукой минимальную информацию, необходимую для определения местонахождения остальных данных, относящихся к процессу, даже если они отсутствуют в памяти. Структура **proc** является записью системной таблицы процессов, которая, как мы только что заметили, всегда находится в оперативной памяти. Запись этой таблицы для выполняющегося в настоящий момент процесса адресуется системной переменной *curproc*. Каждый раз при переключении контекста, когда ресурсы процессора передаются другому

процессу, соответственно изменяется значение переменной *curproc*, которая теперь указывает на структуру **proc** активного процесса. Вторая упомянутая структура **user**, также называемая **u-area** или **u-block**, содержит дополнительные данные о процессе, которые требуются ядру только во время выполнения процесса (т. е. когда процессор выполняет инструкции процесса в режиме ядра или задачи). В отличие от структуры **proc**, адресованной указателем *curproc*, данные **user** размещаются

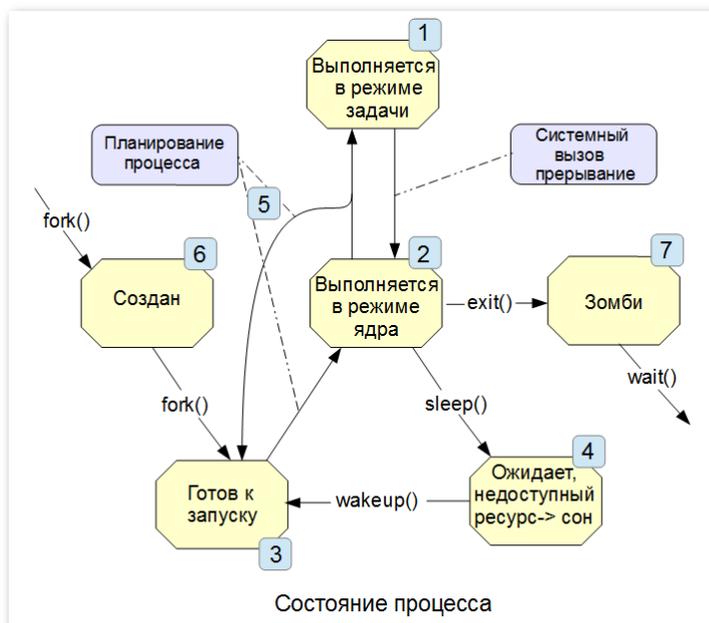
(точнее, отображаются) в определенном месте виртуальной памяти ядра и адресуется переменной **u**. Ниже показаны две основные структуры данных процесса и способы их адресации ядром UNIX.

В **u-area** хранятся данные, которые используются многими подсистемами ядра и не только для управления процессом. В частности, там содержится информация об открытых файловых дескрипторах, диспозиция сигналов, статистика выполнения процесса, а также сохраненные значения регистров, когда выполнение процесса приостановлено. Очевидно, что процесс не должен иметь возможности модифицировать эти данные произвольным образом, поэтому **u-area** защищена от доступа в режиме задачи. Как видно из рис., **u-area** также содержит стек фиксированного размера, системный стек или стек ядра (*kernel stack*). При выполнении процесса в режиме ядра операционная система использует этот стек, а не обычный стек процесса.



### Состояния процесса

Жизненный цикл процесса может быть разбит на несколько состояний. Переход процесса из одного состояния в другое происходит в зависимости от наступления тех или иных событий в системе.



- 1. Процесс выполняется в режиме задачи. При этом процессором выполняются прикладные инструкции данного процесса.
- 2. Процесс выполняется в режиме ядра. При этом процессором выполняются системные инструкции ядра операционной системы от имени процесса.
- 3. Процесс не выполняется, но готов к запуску, как только планировщик выберет его (состояние runnable). Процесс находится в очереди на выполнение и обладает всеми необходимыми ему ресурсами, кроме вычислительных.

- 4. Процесс находится в состоянии сна (asleep), ожидая недоступного в данный момент ресурса, например завершения операции ввода/вывода.
- 5. Процесс возвращается из режима ядра в режим задачи, но ядро прерывает его и производит переключение контекста для запуска более высокоприоритетного процесса.
- 6. Процесс только что создан вызовом fork и находится в переходном состоянии: он существует, но не готов к запуску и не находится в состоянии сна.
- 7. Процесс выполнил системный вызов exit и перешел в состояние зомби (zombie, defunct). Как такового процесса не существует, но остаются записи, содержащие код возврата и временную статистику его выполнения, доступную для родительского процесса. Это состояние является конечным в жизненном цикле процесса.

Необходимо отметить, что не все процессы проходят через все множество состояний, приведенных выше. Процесс начинает свой жизненный путь с состояния 6, когда родительский процесс выполняет системный вызов **fork()**. После того как создание процесса полностью завершено, процесс завершает "дочернюю часть" вызова и переходит в состояние 3 готовности к запуску, ожидая своей очереди на выполнение. Когда планировщик выбирает процесс для выполнения, он переходит в состояние 1 и выполняется в режиме задачи. Выполнение в режиме задачи завершается в результате системного вызова или прерывания, и процесс переходит режим ядра, в котором выполняется код системного вызова или прерывания. После этого процесс опять может вернуться в режим задачи. Однако во время выполнения системного вызова в режиме ядра процессу может понадобиться недоступный в данный момент ресурс. Для ожидания доступа к такому ресурсу, процесс вызывает функцию ядра **sleep()** и переходит в состояние сна 4. При этом процесс добровольно освобождает вычислительные ресурсы, которые предоставляются следующему наиболее приоритетному процессу. Когда ресурс становится доступным, ядро "пробуждает процесс", используя функцию **wakeup()** помещает его в очередь на выполнение, и процесс переходит в состояние "готов к запуску" 3.

При предоставлении процессу вычислительных ресурсов происходит переключение контекста (context switch), в результате которого сохраняется образ, или контекст, текущего процесса, и управление передается новому. Переключение контекста может произойти, например, если процесс перешел в состояние сна, или если в состоянии готовности к запуску находится процесс с более высоким приоритетом, чем текущий. В последнем случае ядро не может немедленно прервать текущий процесс и произвести переключение контекста. Дело в том, что переключению контекста при выполнении в режиме ядра может привести к нарушению целостности самой системы. Поэтому переключение контекста откладывается до момента перехода процесса из режима ядра в режим задачи, когда все системные операции завершены, и структуры данных ядра находятся в нормальном состоянии.

Таким образом, после того как планировщик выбрал процесс на запуск, последний начинает свое выполнение в режиме ядра, где завершает переключение контекста. Дальнейшее состояние

процесса зависит от его предыстории: если процесс был только что создан или был прерван, возвращаясь в режим задачи, он немедленно переходит в этот режим. Если процесс начинает выполнение после состояния сна, он продолжает выполняться в режиме ядра, завершая системный вызов. Заметим, что такой процесс может быть прерван после завершения системного вызова в момент перехода из режима ядра в режим задачи, если в очереди существует более высокоприоритетный процесс. В UNIX

4xBSD определены дополнительные состояния процесса, в первую очередь связанные с системой управления заданиями и взаимодействием процесса с терминалом. Процесс может быть переведен в состояние "остановлен" с помощью сигналов останова **SIGSTOP**, **SIGTTIN** или **SIGTTOU**. В отличие от других сигналов, которые обрабатываются только для выполняющегося процесса, отправление этих сигналов приводит к немедленному изменению состояния. В этом случае, если процесс выполняется или находится в очереди на запуск, его состояние изменяется на "остановлен". Если же процесс находился в состоянии сна, его состояние изменится на "остановлен в состоянии сна". Выход из этих состояний осуществляется сигналом продолжения **SIGCONT**, при этом из состояния "остановлен" процесс переходит в состояние "готов к запуску", а для процесса, остановленного в состоянии сна, следующим пунктом назначения является продолжение "сна".

Описанные возможности полностью реализованы и в SVR4.

*Существует исключение, касающееся процессов, находящихся в состоянии сна для*

*низкоприоритетного события, т. е. события, вероятность наступления которого относительно мала (например, ввода с клавиатуры, который может и не наступить).*

Наконец, процесс выполняет системный вызов **exit()** и заканчивает свое выполнение. Процесс может быть также завершён вследствие получения сигнала. В обоих случаях ядро освобождает ресурсы, принадлежавшие процессу, за исключением кода возврата и статистики его выполнения, и переводит процесс в состояние "зомби". В этом состоянии процесс находится до тех пор, пока родительский процесс не выполнит один из системных вызовов после чего вся информация о процессе будет уничтожена, а родитель получит код возврата завершившегося процесса.

## РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины  
МДК.01.04 Системное программирование  
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.01.04 «Системное программирование» соответствует ФГОС специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936.

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертизы можно сделать заключение, что рабочая программа учебной дисциплины МДК.01.04 «Системное программирование» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Директор ООО «Бизнес ассистент»

«    »

20 г.



Д.С. Зима

## РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины  
МДК.01.04 Системное программирование  
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.01.04 «Системное программирование» соответствует ФГОС специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936.

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

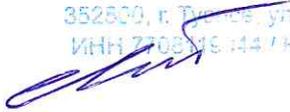
Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертизы можно сделать заключение, что рабочая программа учебной дисциплины МДК.01.04 «Системное программирование» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Инженер-программист 1 категории  
отдела АСУТП управления АСУТП,  
КИПиА, МОП Краснодарского РПУ  
филиала «Макрорегион ЮГ» ООО ИК  
«СИБИНТЕК»

«    »                    20    г.

ООО ИК «СИБИНТЕК»  
Филиал «Макрорегион ЮГ»  
352300, г. Туапсе, ул. Ставская, 4/  
ИНН 770318344 / КПП 177030101

  
М.В. Литус