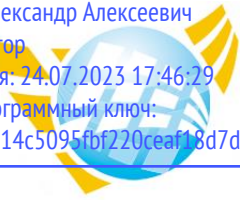


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Уваров Александр Алексеевич  
Должность: Ректор  
Дата подписания: 24.07.2023 17:46:29  
Уникальный программный ключ:  
711a9132de03714c5095fbf220ceaf18d7d7d5b5



**Частное образовательное учреждение  
высшего образования  
БАЛТИЙСКИЙ ГУМАНИТАРНЫЙ  
ИНСТИТУТ**

**КАФЕДРА ОБЩИХ МАТЕМАТИЧЕСКИХ И  
ЕСТЕСТВЕННО-НАУЧНЫХ ДИСЦИПЛИН**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
по учебной дисциплине (модулю)  
Системное программирование**

Программы подготовки специалистов среднего звена  
**09.02.07 Информационные системы и программирование**

Настоящая программа разработана в соответствии с Законом Российской Федерации «Об Образовании в Российской Федерации» от 29 декабря 2012 г. № 273-ФЗ, на основе требований ФГОС СПО специальности 09.02.07 Информационные системы и программирование. (приказ Минобрнауки России от 09.12.2016 г. № 1547).

**Составитель:** канд.экон.наук, доцент Амагаева Юлия Григорьевна

Рассмотрено и утверждено на заседании кафедры  
15.02.2023 протокол № 6.

Одобрено учебно-методическим советом вуза  
15.02.2023 протокол № 5.

## ПАСПОРТ ОЦЕНОЧНЫХ СРЕДСТВ

- 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, применять стандарты антикоррупционного поведения
- 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
- ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
- ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.
- ПК 1.4. Выполнять тестирование программных модулей.
- ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.
- ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

	Контролируемые этапы	Код контролируемой компетенции	Наименование оценочного средства
1	Раздел 1. Дисциплина системное программирование. Программы и программное обеспечение. Системное программирование. Этапы подготовки программы.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Устный опрос Тестирование
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Системное программирование Этапы подготовки программ.
2	Раздел 2. Предложения языка Ассемблера. Регистры. Биты и байты ASCII . .Двоичные числа и шестнадцатеричное представление. Сегменты. Расширение набора команд. Способы адресации. Директивы. Директивы	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Письменная работа Тестирование

	определения данных. Директива определения байта.(DB). Директива определения слова (DW). Директива определения двойного слова (DD).		
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Директива определения учетверенного слова (DQ). Директива определения десяти байт (DT) Непосредственные операнды. Директива EQU.
3	Раздел 3. Сегментные регистры: CS, DS, SS и ES. Регистры общего назначения: AX, BX, CX и DX. Регистровые указатели: SP и BP. Индексные регистры: SI и DI. Регистр командного указателя: IP.. Флаговый регистр.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Устный опрос Тестирование
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Регистр командного указателя: IP Флаговый регистр
4	Раздел 4. Арифметические операции. Обработка двоичных данных. Беззнаковые и знаковые данные. Умножение. Сдвиг регистровой пары DX:AX. Деление. Преобразование знака.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Письменная работа
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Обработка данных в форматах ASCII и BCD. Двоично-десятичный формат (BCD). Сдвиг и округление.
5	Раздел 5. Команды обработки строк. Свойства операций над строками. Операции над строками. REP: Префикс повторения цепочечной команды .MOVS: Пересылка строк LODS: Загрузка строки STOS:	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Проверочная работа Письменное тестирование
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Запись строки CMPS: Сравнение строк SCAS: Сканирование строк
6	Раздел 6. Свойства операторов работы с экраном. Команда прерывания INT. Установка курсора. Очистка экрана. Использование символов возврата каретки, конца строки и табуляции для вывода на экран .Расширенные возможности экранных операций.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10,	Расширенный ASCII код

		ПК 1.1. – ПК 1.6.	Другие операции ввода/вывода. Ввод с клавиатуры по команде BIOS INT 16H. Функциональные клавиши Цвет и графика
7	Раздел 7. Ввод программы и подготовка программы для выполнения Ассемблирование программы. Компоновка программы. Выполнение программы.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Файл перекрестных ссылок
8	Раздел 8. Команда JMP Команда LOOP Флаговый регистр Команды условного перехода Процедуры и оператор CALL. Сегмент стека Команды логических операций: AND, OR, XOR, TEST, NOT Изменение строчных букв на заглавные.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Команды сдвига и циклического сдвига Организация программ
9	Раздел 9. Атрибуты EXTRN и PUBLIC. Компоновка программ на языке. С ++ и Ассемблере. Выполнение COM программы. Выполнение EXE программы .	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Функции загрузки и выполнения программы
10	Раздел 10. Выполнение программ. Начинаем работать. Определение данных Машинная адресация . Определение размера памяти.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Специальные средства отладчика .
11	Раздел 11. <b>Макропроцессоры.</b> Основные понятия. Сравнение макросредств и подпрограмм. Некоторые возможности макроязыка. Локальные переменные макроопределения. Присваивание значений переменным макроопределения.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10,	Глобальные переменные

		ПК 1.1. – ПК 1.6.	макроопределения Уникальные метки
12	Раздел 12. Кросс системы. Вычислительные системы. Модель регистров. Модель оперативной памяти.	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Модель процессора
13	Раздел 13. Классификация ошибок программирования Ошибки при задании необходимых начальных условий для отдельных программ	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	
	Самостоятельная работа	ОК 01 - ОК 10, ПК 1.1. – ПК 1.6.	Распознавание ошибок Ассемблером Распространенные ошибки в драйверах ввода /вывода Распространенные ошибки в программах прерывания

**Знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные принципы отладки и тестирования программных продуктов;
- методы и средства разработки технической документации.

**Уметь:**

- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- оформлять документацию на программные средства;
- использовать инструментальные средства для автоматизации оформления документации.

**Иметь практический опыт:**

- разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;
- разработки кода программного продукта на основе готовой спецификации на уровне модуля;
- использования инструментальных средств на этапе отладки программного продукта;
- проведения тестирования программного модуля по определенному сценарию;

## 1 ТЕКУЩИЙ КОНТРОЛЬ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ (МОДУЛЮ)

### ➤ ОЦЕНОЧНОЕ СРЕДСТВО

#### ➤ ПР 1 Контрольный тест «Системное программирование»

1. Цель – проверить навыки по разработке кода программного модуля на языках программирования низкого уровня;



➤ Уметь: – осуществлять разработку кода программного модуля на современных языках программирования;  
– создавать программу по разработанному алгоритму как отдельный модуль;

2. Проверяемые компетенции (код): ОК 2 -3. ПК 1.1- 1.2.

➤ 3. Пример оценочного средства (*примерные тестовые задания, типовой вариант контрольной работы и др.*)

### Фонд тестовых заданий по дисциплине «Системное программирование»

Тестирование проводится по двум разделам курса на практических занятиях. Студент получает контрольный лист с вариантом теста, составленного по вопросам, которые приведены ниже. В каждом варианте 25 вопросов. Время на подготовку ответов 25 минут. Каждый студент заполняет контрольный лист теста, который возвращает вместе с заданием. Вся группа тестируется одновременно.

#### Тестирование «Системное программирование»

Вопрос 1 Системная программа – это

- a. программа, предназначенная для поддержания работоспособности СОИ или повышения эффективности ее использования.
- b. программа, предназначенная для решения задачи или класса задач в определенной области применения СОИ
- c. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 2 Исходный модуль – это

- a. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- b. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
- c. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 3 Автокод – это

- a. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
- b. язык программирования, понятия и структура которого удобны для восприятия человеком.
- c. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 4... - реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.

- a. Интерпретация
- b. Трансляция
- c. Компиляция

Вопрос 5 Программное обеспечение – это

- a. совокупность программ СОО и программных документов, необходимых для их эксплуатации
- b. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- c. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 6 Регистр - это

- a. Ячейка в оперативной памяти
- b. Ячейка памяти процессора
- c. Адресуемая ячейка памяти

Вопрос 7 Какие регистры относятся к сегментным:

- a. Es
- b. Dх
- c. Ip
- d. Cs
- e. Al

Вопрос 8 Команда mov –

- a. Команда помещения в стек
- b. Команда перемещения данных
- c. Команда пересылки байта

Вопрос 9 Какая команда относится к логическим командам:

- a. Sub
- b. Or
- c. Inc

Вопрос 10 Команда ... начинает выполнение с новой ветки в любом случаи

- a. Безусловного перехода
- b. Условного перехода
- c. Цикла

Вопрос 11 Объявляется непосредственно внутри ассемблера - ...

- a. Локальная метка
- b. Глобальная метка



с. Системная метка

Вопрос 12 Прерывание, которое возникает при изменении какого-либо устройства:

- a. Внутренние
- b. Аппаратное
- c. Программное

Вопрос 13

13. Какой командой вызывается процедура:

- a. Proc
- b. Call
- c. Ret

Вопрос 14 Директива – это

- a. Символическое имя для определения места в памяти, где размещены данные
- b. Блок команд, который может быть вызван из любого места основной программы
- c. Процесс приостановке программы

Вопрос 15 Какой командой вызывается прерывания:

- a. In
- b. Inc
- c. Int

Вопрос 16 Определить ошибки в программе:

1. Var X: integer;
2. Begin
3. X:=100;
4. asm
5. Mov bl, 25
6. Mov ax, X
7. Div ax,bl
8. Mov X,bl
9. end;
10. writeln ('X=',X);
11. end.

- a. Строка 5,7
- b. Строка 6,7
- c. Строка 7,8

Вопрос 17 Определить результат выполнения программы:

```
Begin  
asm  
Mov Ah,02h  
Mov Dl,'*'  
int 21h  
end;  
end.
```

Вопрос 18

Управляющая программа - это

- a. программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.
- b. программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ
- c. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 19 Объектный модуль – это

- a. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- b. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
- c. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 20 Язык высокого уровня – это

- a. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
- b. язык программирования, понятия и структура которого удобны для восприятия человеком.
- c. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 21 ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

- a. Интерпретация
- b. Трансляция
- c. Компиляция

Вопрос 22 Прикладное программное обеспечение – это

- a. совокупность программ СООИ и программных документов, необходимых для их эксплуатации
- b. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- c. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 23 Язык Ассемблер - это

- a. система записи программы с детализацией до отдельной машинной команды
- b. язык программирования, понятия и структура которого удобны для восприятия человеком
- c. система, образуемая языком программирования, компилятором или интерпретатором программ

Вопрос 24 Какие регистры относятся к регистрам данных:

- a. Es
- b. Dx
- c. Ip
- d. Cs

e. AI

Вопрос 25 Команда add –

- a. Команда сложения
- b. Команда умножения
- c. Команда вычитания

#### 4. Критерии оценивания:

Оценка «отлично»	ставится студенту, если студент из 25 вопросов ответил правильно от 23 до 25 вопросов.
Оценка «хорошо»	ставится студенту, если студент из 25 вопросов ответил правильно от 18 до 22 вопросов.
Оценка «удовлетворительно»	ставится, если студент из 25 вопросов ответил правильно от 13 до 17 вопросов.
Оценка «неудовлетворительно»	ставится, если студент из 25 вопросов ответил правильно на 12 вопросов и менее.

#### 5. Рекомендуемый перечень вопросов для самостоятельной подготовки:

1. Системное программирование. Этапы подготовки программы .
2. Программирование на языке Ассемблера. Предложения языка Ассемблера .
3. Регистры. Биты и байты ASCII . . . . .
4. Двоичные числа. Шестнадцатеричное представление.
5. Сегменты. Расширение набора команд.
6. Способы адресации. Директивы. Директивы определения данных.
7. Директива определения байта (DB). Директива определения слова (DW).
8. Директива определения двойного слова (DD). Директива определения учетверенного слова (DQ).
9. Директива определения десяти байт (DT). Непосредственные операнды Директива EQU.
10. Сегментные регистры: CS, DS, SS и ES.
11. Регистры общего назначения: AX, BX, CX и DX.
12. Регистровые указатели: SP и BP. Индексные регистры: SI и DI .
13. Регистр командного указателя IP. Флаговый регистр.
14. Обработка двоичных данных. Беззнаковые и знаковые данные.
15. Ассемблер. Умножение. Сдвиг регистровой пары DX:AX Деление.
16. Ассемблер. Преобразование знака Обработка данных в форматах ASCII и BCD.
17. Ассемблер. Двоично-десятичный формат (BCD)Сдвиг и округление.
18. Ассемблер. Команда прерывания INT. Установка курсора. Очистка экрана .
19. Ассемблер. Использование символов возврата каретки, конца строки и табуляции для вывода на экран
20. Ассемблер. Расширенные возможности экранных операций Расширенный ASCII код.
21. Ассемблер. Другие операции ввода/вывода Ввод с клавиатуры по команде BIOS INT 16H.
22. Ассемблер. Функциональные клавиши Цвет и графика.
23. Ассемблер. Межсегментные вызовы. Атрибуты EXTRN и PUBLIC.
24. Компоновка программ на языке C++ и Ассемблере.

## ОЦЕНОЧНОЕ СРЕДСТВО

### ➤ ПР2 Контрольный тест «Системное программирование»

1. Цель – проверить навыки по разработке кода программного модуля на языках программирования низкого уровня;

➤ Уметь: – – создавать программу по разработанному алгоритму как отдельный модуль;

– выполнять отладку и тестирование программы на уровне модуля;

➤ Знать:

➤ 2. Проверяемые компетенции (код): ОК 5 -9. ПК 1.5- 1.6.

➤ 3. Пример оценочного средства (*примерные тестовые задания, типовой вариант контрольной работы и др.*)

Вопрос 1 Управляющая программа - это

d. программа, предназначенная для поддержания работоспособности СОИ или повышения эффективности ее использования.

e. программа, предназначенная для решения задачи или класса задач в определенной области применения СОИ

f. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 2 Объектный модуль – это

d. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

e. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

f. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 3 Язык высокого уровня – это

d. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

e. язык программирования, понятия и структура которого удобны для восприятия человеком.

f. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 4 ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

d. Интерпретация

e. Трансляция

f. Компиляция

Вопрос 5 Прикладное программное обеспечение – это

d. совокупность программ СОИ и программных документов, необходимых для их эксплуатации

- e. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- f. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 6 Язык Ассемблер - это

- d. система записи программы с детализацией до отдельной машинной команды
- e. язык программирования, понятия и структура которого удобны для восприятия человеком
- f. система, образуемая языком программирования, компилятором или интерпретатором программ

Вопрос 7 Какие регистры относятся к регистрам данных:

- f. Es
- g. Dх
- h. Ip
- i. Cs
- j. Al

Вопрос 8 Команда add –

- d. Команда сложения
- e. Команда умножения
- f. Команда вычитания

Вопрос 9 Какая команда относится к арифметическим командам:

- a. In
- b. Or
- c. Inc

Вопрос 10 Команда ... повторяет действия несколько раз

- a. Безусловного перехода
- b. Условного перехода
- c. Цикла

Вопрос 11 Ссылка на определенное место в программе - ...

- a. Адресация
- b. Метка
- c. Переход

Вопрос 12 Прерывание, которое возникает при существовании ошибок в программе:

- a. Внутренние
- b. Аппаратное
- c. Программное

Вопрос 13 Какой командой осуществляется начало процедуры:

- a. Proc
- b. Call
- c. Ret

Вопрос 14 Прерывания – это

- a. Символическое имя для определения места в памяти, где размещены данные
- b. Блок команд, который может быть вызван из любого места основной программы

с. Процесс приостановке программы

Вопрос 15 Какие команды относятся к директивам:

- a. DD
- b. DE
- c. DB
- d. DC
- e. DW

Вопрос 16 Определить ошибки в программе:

- 1. Var a:boolean;
- 2. begin
- 3. asm
- 4. Mov AX,00111011b
- 5. Or Al,00001111b
- 6. Mov a,AL
- 7. end;
- 8. Writeln('A=',A);
- 9. end.

- a. Строка 1,4
- b. Строка 1,5
- c. Строка 4,6

Вопрос 17 Определить результат выполнения программы:

```
Var x:integer;  
Begin  
X:=50;  
asm  
mov ax,x  
sub ax,13  
sub ax,4  
sub ax,20  
mov x,ax  
end;  
writeln('x=',x);  
end.
```

- a. 13
- b. 14
- c. 15

Вопрос 18

Управляющая программа - это

- g. программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.
- h. программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ
- i. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 19 Объектный модуль – это

- g. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- h. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
- i. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 20 Язык высокого уровня – это

- g. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
- h. язык программирования, понятия и структура которого удобны для восприятия человеком.
- i. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 21 ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

- g. Интерпретация
- h. Трансляция
- i. Компиляция

Вопрос 22 Прикладное программное обеспечение – это

- g. совокупность программ СОО и программных документов, необходимых для их эксплуатации
- h. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- i. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 23 Язык Ассемблер - это

- g. система записи программы с детализацией до отдельной машинной команды
- h. язык программирования, понятия и структура которого удобны для восприятия человеком
- i. система, образуемая языком программирования, компилятором или интерпретатором программ

Вопрос 24 Какие регистры относятся к регистрам данных:

- k. Es
- l. Dх
- m. Ip
- n. Cs
- o. Al

Вопрос 25 Команда add –

- g. Команда сложения
- h. Команда умножения
- i. Команда вычитания

#### 4. Критерии оценивания:

Оценка «отлично»	ставится студенту, если студент из 25 вопросов ответил правильно от 23 до 25 вопросов.
Оценка «хорошо»	ставится студенту, если студент из 25 вопросов ответил правильно от 18 до 22 вопросов.
Оценка «удовлетворительно»	ставится, если студент из 25 вопросов ответил правильно от 13 до 17 вопросов.
Оценка «неудовлетворительно»	если студент из 25 вопросов ответил правильно на 12 вопросов и менее.

#### 5. Рекомендуемый перечень вопросов для самостоятельной подготовки:

1. Выполнение СОМ программы Выполнение EXE программы на языке С++ и Ассемблере.
2. Ассемблер. Функции загрузки и выполнения программы Основные понятия.
3. Сравнение макросредств и подпрограмм. Некоторые возможности Макроязыка.
4. Локальные переменные, макроопределения. Присваивание значений переменным макроопределения.
5. Ассемблер. Глобальные переменные макроопределения Уникальные метки.
6. Кросс системы. Вычислительные системы. Модель регистров.
7. Модель оперативной памяти Модель процессора.
8. Ошибки программирования. Классификация ошибок программирования.
9. Ошибки при задании необходимых начальных условий для отдельных программ.
10. Распознавание ошибок Ассемблером. Распространенные ошибки в драйверах ввода /вывода.
11. Распространенные ошибки Ассемблера в программах прерывания.



## ➤ ОЦЕНОЧНОЕ СРЕДСТВО

### ➤ ПР 3 Контрольный тест «Системное программирование»

1. Цель – проверить навыки по разработке кода программного модуля на языках программирования низкого уровня;



➤ Уметь: - создавать программу по разработанному алгоритму как отдельный модуль;

– выполнять отладку и тестирование программы на уровне модуля;

➤ Знать:

➤ 2. Проверяемые компетенции (код): ОК 6 -9. ПК 1.4- 1.6.

➤ 3. Пример оценочного средства (примерные тестовые задания, типовой вариант контрольной работы и др.)

Вопрос 1 Управляющая программа - это

a. программа, предназначенная для поддержания работоспособности СОИ или повышения эффективности ее использования.

b. программа, предназначенная для решения задачи или класса задач в определенной области применения СОИ

c. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 2 Программный модуль – это

a. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

b. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

c. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 3 Машинный язык – это

a. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

b. язык программирования, понятия и структура которого удобны для восприятия человеком.

c. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 4 ... - преобразование программы на машинный язык.

a. Интерпретация

b. Трансляция

c. Компиляция

Вопрос 5 Системное программное обеспечение– это

a. совокупность программ СОИ и программных документов, необходимых для их эксплуатации

- b. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- c. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 6 Адресация предназначена ...

- a. Для указания адреса ячеек
- b. Для обмена данными
- c. Для помещения данных в регистры

Вопрос 7 Назовите регистр указания команд:

- a. Es
- b. Dx
- c. Ip
- d. Cs
- e. Al

Вопрос 8 Команда pop –

- a. Команда помещения в стек
- b. Команда перемещения данных
- c. Команда извлечения из стека

Вопрос 9 Какая команда относится к битовым командам:

- a. Bt
- b. Or
- c. Inc

Вопрос 10 Команда ... начинает выполнение с новой ветки в случае, если условие ложно

- a. Безусловного перехода
- b. Условного перехода
- c. Цикла

Вопрос 11 Объявляется в Паскале - ...

- a. Локальная метка
- b. Глобальная метка
- c. Системная метка

Вопрос 12 Прерывание, которое возникает при обнаружении ошибок в работе процессора:

- a. Внутренние
  - b. Аппаратное
  - c. Программное
13. Какой командой осуществляется возврат из процедуры:
- a. Proc
  - b. Call
  - c. Ret

Вопрос 13 Подпрограмма – это

- a. Символическое имя для определения места в памяти, где размещены данные
- b. Блок команд, который может быть вызван из любого места основной программы
- c. Процесс приостановке программы

Вопрос 14 Какие функции относятся к прерыванию 21H:

- a. 02h
- b. 09h
- c. 08h
- d. 2ah
- e. 5fh

Вопрос 15 Определить ошибки в программе:

- 1. Var
- 2. str:='gruppa\$';
- 3. asm
- 4. Mov Cx,3
- 5. @Lab:
- 6. Mov Ah,9
- 7. Mab Dx,offset str
- 8. int 21h
- 9. Loop @Lab
- 10. end;
- 11. readln;
- 12. end.

- a. Строка 4,6
- b. Строка 4,7
- c. Строка 6,7

Вопрос 16 Определить результат выполнения программы:

```
Begin
asm
Mov ax,6;
Mov bx,8;
Mov dx, ax;
cmp ax, bx;
jae Li;
Mov dx, bx;
Li:
mov dx,ax
end;
end.
```

- a. 6
- b. 8
- c. 14

Вопрос 17 Определить результат выполнения программы:

```
Begin
asm
Mov Ah,02h
Mov Dl,'*'
int 21h
end;
end.
```

Вопрос 18

Управляющая программа - это

- j. программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.
- к. программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ
- l. системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

Вопрос 19 Объектный модуль – это

- j. программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.
- к. программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.
- l. программный модуль, получаемый в результате трансляции исходного модуля.

Вопрос 20 Язык высокого уровня – это

- j. символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.
- к. язык программирования, понятия и структура которого удобны для восприятия человеком.
- l. язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Вопрос 21 ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

- j. Интерпретация
- к. Трансляция
- l. Компиляция

Вопрос 22 Прикладное программное обеспечение – это

- j. совокупность программ СООИ и программных документов, необходимых для их эксплуатации
- к. комплекс программ, которые обеспечивают управление компонентами компьютерной системы
- l. компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы

Вопрос 23 Язык Ассемблер - это

- j. система записи программы с детализацией до отдельной машинной команды
- к. язык программирования, понятия и структура которого удобны для восприятия человеком
- l. система, образуемая языком программирования, компилятором или интерпретатором программ

Вопрос 24 Какие регистры относятся к регистрам данных:

- p. Es
- q. Dx

- r. Ip
- s. Cs
- t. Al

Вопрос 25 Команда add –

- j. Команда сложения
- k. Команда умножения
- l. Команда вычитания

#### 4. Критерии оценивания:

Оценка «отлично»	ставится студенту, если студент из 25 вопросов ответил правильно от 23 до 25 вопросов.
Оценка «хорошо»	ставится студенту, если студент из 25 вопросов ответил правильно от 18 до 22 вопросов.
Оценка «удовлетворительно»	ставится, если студент из 25 вопросов ответил правильно от 13 до 17 вопросов.
Оценка «неудовлетворительно»	если студент из 25 вопросов ответил правильно на 12 вопросов и менее.

#### 5. Рекомендуемый перечень вопросов для самостоятельной подготовки:

1. Выполнение СОМ программы Выполнение EXE программы на языке C++ и Ассемблере.
2. Ассемблер. Функции загрузки и выполнения программы Основные понятия.
3. Сравнение макросредств и подпрограмм. Некоторые возможности Макроязыка.
4. Локальные переменные, макроопределения. Присваивание значений переменным макроопределения.
5. Ассемблер. Глобальные переменные макроопределения Уникальные метки.
6. Кросс системы. Вычислительные системы. Модель регистров.
7. Модель оперативной памяти Модель процессора.
8. Ошибки программирования. Классификация ошибок программирования.
9. Ошибки при задании необходимых начальных условий для отдельных программ.
10. Распознавание ошибок Ассемблером. Распространенные ошибки в драйверах ввода /вывода.
11. Распространенные ошибки Ассемблера в программах прерывания.

## ➤ ОЦЕНОЧНОЕ СРЕДСТВО

### ПР 4 Деловая (ролевая) игра

по дисциплине «Системное программирование»

**1. Цель** - Закрепление и расширение следующих знаний студентов по знанию основных понятий в области системного программирования.

- Уметь: – выполнять отладку и тестирование программы на уровне модуля;  
– оформлять документацию на программные средства;  
– использовать инструментальные средства для автоматизации оформления документации.

Знать: – основные принципы отладки и тестирования программных продуктов;  
– методы и средства разработки технической документации.

- **2. Проверяемые компетенции (код): ОК 7 - 9. ПК 1.5-1.6.**
- **3. Пример оценочного средства (типовой вариант деловой игры.)**

### Системное программирование

Программа - это данные, предназначенные для управления конкретными компонентами системы обработки информации (СОИ) в целях реализации определенного алгоритма.

Определения даются по: ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения. - М.:Изд-во стандартов, 1990.

Обратить внимание: программа - это данные. Один из основных принципов машины фон Неймана - то, что и программы, и данные хранятся в одной и той же памяти. Сохраняемая в памяти программа представляет собой некоторые коды, которые могут рассматриваться как данные. Возможно, с точки зрения программиста программа - активный компонент, она выполняет некоторые действия. Но с точки зрения процессора команды программы - это данные, которые процессор читает и интерпретирует. С другой стороны программа - это данные с точки зрения обслуживающих программ, например, с точки зрения компилятора, который на входе получает одни данные - программу на языке высокого уровня (ЯВУ), а на выходе выдает другие данные - программу в машинных кодах.

***Программное обеспечение (ПО) - совокупность программ СОИ и программных документов, необходимых для их эксплуатации***

Существенно, что ПО - это программы, предназначенные для многократного использования и применения разными пользователями. В связи с этим следует обратить внимание на ряд необходимых свойств ПО.

1. Необходимость документирования. По определению программы становятся ПО только при наличии документации. Конечный пользователь не может работать, не имея документации. Документация делает возможным тиражирование ПО и продажу его без его разработчика. По Бруксу ошибкой в ПО является ситуация, когда программное изделие функционирует не в соответствии со своим описанием, следовательно, ошибка в документации также является ошибкой в программном изделии.
2. Эффективность. ПО, рассчитанное на многократное использование (например, ОС, текстовый редактор и т.п.) пишется и отлаживается один раз, а выполняется многократно. Таким образом, выгодно переносить затраты на этап производства ПО и освобождать от затрат этап выполнения, чтобы избежать тиражирования затрат.

### 3. Надежность. В том числе:

- Тестирование программы при всех допустимых спецификациях входных данных
- Защита от неправильных действий пользователя
- Защита от взлома - пользователи должны иметь возможность взаимодействия с ПО только через легальные интерфейсы.

"Ошибки в системе возможны из-за сбоев аппаратуры, ошибок ПО, неправильных действий пользователя. Первые - неизбежны, вторые - вероятны, третьи - гарантированы". Появление ошибок любого уровня не должно приводить к краху системы. Ошибки должны вылавливаться диагностироваться и (если их невозможно исправить) превращаться в корректные отказы.

Системные структуры данных должны сохраняться безусловно. Сохранение целостности пользовательских данных желательно.

### 4. Возможность сопровождения. Возможные цели сопровождения - адаптация ПО к конкретным условиям применения, устранение ошибок, модификация. Во всех случаях требуется тщательное структурирование ПО и носителем информации о структуре ПО должна быть программная документация. Адаптация во многих случаях м.б. передоверена пользователю - при тщательной отработке и описании сценариев инсталляции и настройки. Исправление ошибок требует развитой сервисной службы, собирающей информацию об ошибках и формирующей исправляющие пакеты. Модификация предполагает изменение спецификаций на ПО. При этом, как правило, должны поддерживаться и старые спецификации. Эволюционное развитие ПО экономит вложения пользователей.

## Системное программирование

*Системная программа - программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.*

*Прикладная программа - программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ.*

В соответствии с терминологией, системное программирование - это процесс разработки системных программ (в т.ч., управляющих и обслуживающих).

С другой стороны, по определению Гегеля система - *единое целое, состоящее из множества компонентов и множества связей между ними*. Тогда системное программирование - это разработка программ сложной структуры.

Эти два определения не противоречат друг другу, так как разработка программ сложной структуры ведется именно для обеспечения работоспособности или повышения эффективности СООИ.

Зафиксированное в ГОСТ подразделение ПО на системное и прикладное является до некоторой степени устаревшим. Сегодняшнее деление предусматривает по меньшей мере три градации ПО:

- Системное
- Промежуточное
- Прикладное

Промежуточное ПО (middleware) мы определяем как совокупность программ, осуществляющих управление вторичными (конструируемыми самим ПО) ресурсами, ориентированными на решение определенного (широкого) класса задач. К такому ПО относятся менеджеры транзакций, серверы БД, серверы коммуникаций и другие программные серверы. С точки зрения инструментальных средств разработки промежуточное ПО ближе к прикладному, так как не работает на прямую с первичными ресурсами, а использует для этого сервисы, предоставляемые системным ПО. С точки

зрения алгоритмов и технологий разработки промежуточное ПО ближе к системному, так как всегда является сложным программным изделием многократного и многоцелевого использования и в нем применяются те же или сходные алгоритмы, что и в системном ПО.

Современные тенденции развития ПО состоит в снижении объема как системного, так и прикладного программирования. Основная часть работы программистов выполняется в промежуточном ПО. Снижение объема системного программирования определено современными концепциями ОС, объектно-ориентированной архитектурой и архитектурой микроядра, в соответствии с которыми большая часть функций системы выносятся в утилиты, которые можно отнести и к промежуточному ПО. Снижение объема прикладного программирования обусловлено тем, что современные продукты промежуточного ПО предлагают все больший набор инструментальных средств и шаблонов для решения задач своего класса.

Значительная часть системного и практически все прикладное ПО пишется на языках высокого уровня, что обеспечивает сокращение расходов на их разработку/модификацию и переносимость.

*Системное ПО подразделяется на системные управляющие программы и системные обслуживающие программы.*

*Управляющая программа - системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОИ, восстановление работы системы после проявления неисправностей в технических средствах.*

*Программа обслуживания (утилита) - программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу СОИ.*

Управляющая программа совместно с набором необходимых для эксплуатации системы утилит составляют операционную систему (ОС).

Кроме входящих в состав ОС утилит могут существовать и другие утилиты (того же или стороннего производителя), выполняющие дополнительное (опционное) обслуживание. Как правило, это утилиты, обеспечивающие разработку программного обеспечения для операционной системы.

*Система программирования - система, образуемая языком программирования, компилятором или интерпретатором программ, представленных на этом языке, соответствующей документацией, а также вспомогательными средствами для подготовки программ к форме, пригодной для выполнения.*

## Этапы подготовки программы

При разработке программ, а тем более - сложных, используется принцип модульности, разбиения сложной программы на составные части, каждая из которых может подготавливаться отдельно. Модульность является основным инструментом структурирования программного изделия, облегчающим его разработку, отладку и сопровождение.

*Программный модуль - программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.*

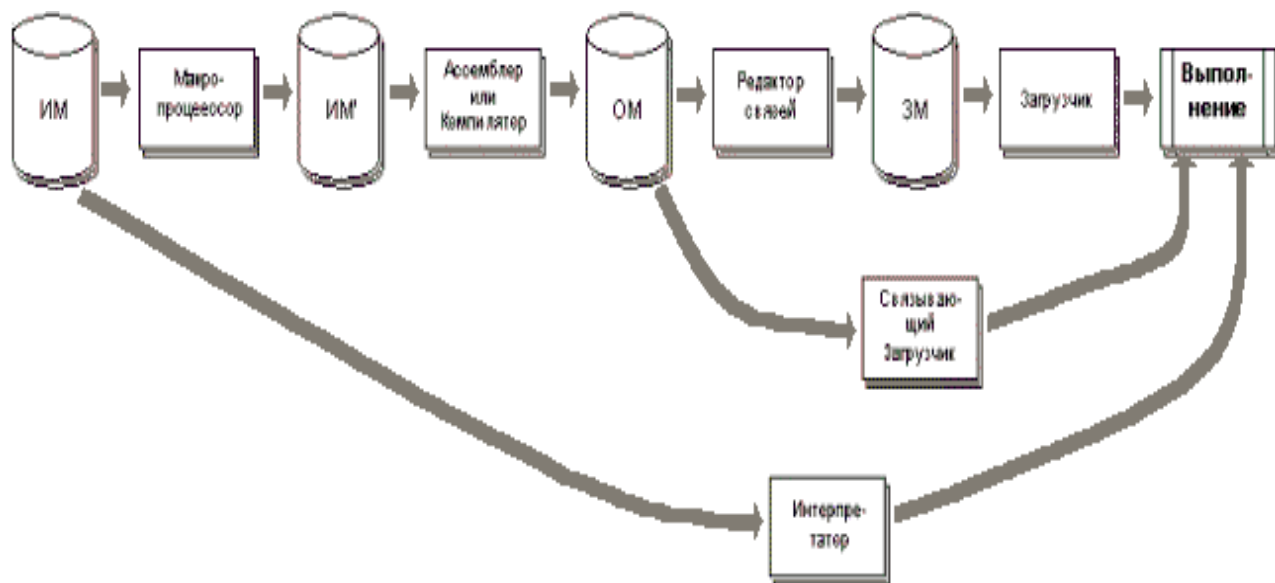
При выборе модульной структуры должны учитываться следующие основные соображения:

- Функциональность - модуль должен выполнять законченную функцию
- Несвязность - модуль должен иметь минимум связей с другими модулями, связь через глобальные переменные и области памяти нежелательна
- Специфицируемость - входные и выходные параметры модуля должны четко формулироваться

На рисунке показаны этапы, которые проходит программа от своего написания до



## ВЫПОЛНЕНИЯ



Программа пишется в виде исходного модуля, на рисунке - файл ИМ.

**Исходный модуль - программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.**

Первым (не для всех языков программирования обязательным) этапом подготовки программы является обработка ее Макропроцессором (или Препроцессором). Макропроцессор обрабатывает текст программы и на выходе его получается новая редакция текста (на рис. - ИМ'). В большинстве систем программирования Макропроцессор совмещен с транслятором, и для программиста его работа и промежуточный ИМ' "не видны". Следует иметь в виду, что Макропроцессор выполняет обработку текста, это означает, с одной стороны, что он "не понимает" операторов языка программирования и "не знает" переменных программы, с другой, что все операторы и переменные Макроязыка (тех выражений в программе, которые адресованы Макропроцессору) в промежуточном ИМ' уже отсутствуют и для дальнейших этапов обработки "не видны". Так, если Макропроцессор заменил в программе некоторый текст А на текст В, то транслятор уже видит только текст В, и не знает, был этот текст написан программистом "своей рукой" или подставлен Макропроцессором.

Следующим этапом является трансляция.

**Трансляция - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле (логическом, функциональном) равносильную первой.**

Как правило, выходным языком транслятора является машинный язык целевой вычислительной системы. (Целевая ВС - та ВС, на которой программа будет выполняться.)

**Машинный язык - язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.**

Трансляторы - общее название для программ, осуществляющих трансляцию. Они подразделяются на Ассемблеры и Компиляторы - в зависимости от исходного языка программы, которую они обрабатывают. Ассемблеры работают с Автокодами или языками Ассемблера, Компиляторы - с языками высокого уровня.

**Автокод - символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.**

**Язык Ассемблера - язык программирования, который представляет собой**

*символьную форму машинного языка с рядом возможностей, характерных для языка высокого уровня (обычно включает в себя макросредства).*

*Язык высокого уровня - язык программирования, понятия и структура которого удобны для восприятия человеком.*

***Объектный модуль - программный модуль, получаемый в результате трансляции исходного модуля.***

Поскольку результатом трансляции является модуль на языке, близком к машинному, в нем уже не остается признаков того, на каком исходном языке был написан программный модуль. Это создает принципиальную возможность создавать программы из модулей, написанных на разных языках. Специфика исходного языка, однако, может сказываться на физическом представлении базовых типов данных, способах обращения к процедурам/функциям и т.п. Для совместимости разноязыковых модулей должны выдерживаться общие соглашения.

Большая часть объектного модуля - команды и данные машинного языка именно в той форме, в какой они будут существовать во время выполнения программы. Однако, программа в общем случае состоит из многих модулей. Поскольку транслятор обрабатывает только один конкретный модуль, он не может должным образом обработать те части этого модуля, в которых запрограммированы обращения к данным или процедурам, определенным в другом модуле. Такие обращения называются внешними ссылками. Те места в объектном модуле, где содержатся внешние ссылки, транслируются в некоторую промежуточную форму, подлежащую дальнейшей обработке. Говорят, что объектный модуль представляет собой программу на машинном языке с неразрешенными внешними ссылками.

Разрешение внешних ссылок выполняется на следующем этапе подготовки, который обеспечивается Редактором Связей (Компоновщиком). Редактор Связей соединяет вместе все объектные модули, входящие в программу. Поскольку Редактор Связей "видит" уже все компоненты программы, он имеет возможность обработать те места в объектных модулях, которые содержат внешние ссылки. Результатом работы Редактора Связей является загрузочный модуль.

***Загрузочный модуль - программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения.***

Загрузочный модуль сохраняется в виде файла на внешней памяти. Для выполнения программа должна быть перенесена (загружена) в оперативную память. Иногда при этом требуется некоторая дополнительная обработка (например, настройка адресов в программе на ту область оперативной памяти, в которую программа загрузилась). Эта функция выполняется Загрузчиком, который обычно входит в состав операционной системы.

Возможен также вариант, в котором редактирование связей выполняется при каждом запуске программы на выполнение и совмещается с загрузкой. Это делает Связывающий Загрузчик. Вариант связывания при запуске более расходный, т.к. затраты на связывание тиражируются при каждом запуске. Но он обеспечивает:

- большую гибкость в сопровождении, так как позволяет менять отдельные объектные модули программы, не меня остальных модулей;
- экономию внешней памяти, т.к. объектные модули, используемые во многих программах не копируются в каждый загрузочный модуль, а хранятся в одном экземпляре.

Вариант интерпретации подразумевает прямое исполнение исходного модуля.

***Интерпретация - реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.***

Интерпретатор читает из исходного модуля очередное предложение программы, переводит его в машинный язык и выполняет. Все затраты на подготовку тиражируются

при каждом выполнении, следовательно, интерпретируемая программа принципиально менее эффективна, чем транслируемая. Однако, интерпретация обеспечивает удобство разработки, гибкость в сопровождении и переносимость.

Примеры интерпретаторов: языки процедур (sell, REXX), JVM.

Не обязательно подготовка программы должна вестись на той же вычислительной системе и в той же операционной среде, в которых программа будет выполняться. Системы, обеспечивающие подготовку программ в среде, отличной от целевой называются кросс-системами. В кросс-системе может выполняться вся подготовка или ее отдельные этапы:

- Макрообработка и трансляция
- Редактирование связей
- Отладка

Типовое применение кросс-систем - для тех случаев, когда целевая вычислительная среда просто не имеет ресурсов, необходимых для подготовки программ, например, встроенные системы.

Программные средства, обеспечивающие отладку программы на целевой системе можно также рассматривать как частный случай кросс-системы.

**Порядок деловой игры.** Все студенты разбиваются на группы по 3—4 человека. Одна группа будет представлять экспертную комиссию, а остальные — рабочие группы.

Предполагается, что экспертная комиссия — это представители высшего звена (директор по производству, директор по маркетингу, директор по логистике, коммерческий директор и др.). Рабочая группа — это представители среднего звена (маркетологи, логисты, сбытовики), которым после проведенного экстренного консилиума было выдано задание: разработать проект реорганизации подсистемы распределения с учетом сложившихся внутренних и внешних условий.

Таким образом, каждая рабочая группа должна самостоятельно изучить ситуацию и последовательно ответить на предлагаемые ниже вопросы. В результате у каждой группы получится свой вариант развития событий, который ей предстоит представить на суд экспертной комиссии.

Экспертная комиссия в свою очередь должна разработать систему оценки проектов на основе сформулированного ниже задания и оценить все проекты соответствующе.

По итогам работы экспертная комиссия выносит решение о победившем проекте.

**Примечание.** Для получения адекватного результата предлагается ввести в игру координатора экспертной комиссии, которым должен выступать преподаватель.

#### 4. Критерии оценивания

Оценка «отлично»	студент являлся активным участником игры и принимал участие в последующем ее анализе, выявляя и исправляя допущенные ошибки;
Оценка «хорошо»	студент являлся только активным участником игры и не принимал участие в ее анализе;
Оценка «удовлетворительно»	студент являлся пассивным участником игры, но принимал активное участие в ее анализе, выявляя и исправляя допущенные ошибки.
Оценка «неудовлетворительно»	выставляется студенту, если он являлся пассивным участником игры и не принимал участие в ее последующем анализе.

#### 5. Рекомендуемый перечень вопросов для самостоятельной подготовки:

1. Что представляет собой язык *Ассемблера*?
2. Назовите основные операторы *Ассемблера*?
3. В чем основные различия *Автокода* и *Ассемблера*?
4. Что представляет собой язык программный модуль, получаемый в результате трансляции исходного модуля?
5. Что представляет собой *Системная программа*?

## II ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

1. Форма проведения промежуточной аттестации: дифференцированный зачет в 3 семестре.
2. Перечень вопросов для аттестации

### Вопросы для зачета

1. Системное программирование. Этапы подготовки программы .
2. Программирование на языке Ассемблера. Предложения языка Ассемблера .
3. Регистры. Биты и байты ASCII . . . . .
4. Двоичные числа. Шестнадцатеричное представление.
5. Сегменты. Расширение набора команд.
6. Способы адресации. Директивы. Директивы определения данных.
7. Директива определения байта (DB). Директива определения слова (DW).
8. Директива определения двойного слова (DD). Директива определения учетверенного слова (DQ).
9. Директива определения десяти байт (DT). Непосредственные операнды Директива EQU.
10. Сегментные регистры: CS, DS, SS и ES.
11. Регистры общего назначения: AX, BX, CX и DX.
12. Регистровые указатели: SP и BP. Индексные регистры: SI и DI .
13. Регистр командного указателя IP. Флаговый регистр.
14. Обработка двоичных данных. Беззнаковые и знаковые данные.
15. Ассемблер. Умножение. Сдвиг регистровой пары DX:AX Деление.
16. Ассемблер. Преобразование знака Обработка данных в форматах ASCII и BCD.
17. Ассемблер. Двоично-десятичный формат (BCD)Сдвиг и округление.
18. Ассемблер. Команда прерывания INT. Установка курсора. Очистка экрана .
19. Ассемблер. Использование символов возврата каретки, конца строки и табуляции для вывода на экран
20. Ассемблер. Расширенные возможности экранных операций Расширенный ASCII код.
21. Ассемблер. Другие операции ввода/вывода Ввод с клавиатуры по команде BIOS INT 16H.
22. Ассемблер. Функциональные клавиши Цвет и графика.
23. Ассемблер. Межсегментные вызовы. Атрибуты EXTRN и PUBLIC.
24. Компоновка программ на языке C++ и Ассемблере.
25. Выполнение COM программы Выполнение EXE программы на языке C++ и Ассемблере.
26. Ассемблер. Функции загрузки и выполнения программы Основные понятия.
27. Сравнение макросредств и подпрограмм. Некоторые возможности Макроязыка.
28. Локальные переменные, макроопределения. Присваивание значений переменным макроопределения.
29. Ассемблер. Глобальные переменные макроопределения Уникальные метки.
30. Кросс системы. Вычислительные системы. Модель регистров.
31. Модель оперативной памяти Модель процессора.
32. Ошибки программирования. Классификация ошибок программирования.
33. Ошибки при задании необходимых начальных условий для отдельных программ.
34. Распознавание ошибок Ассемблером. Распространенные ошибки в драйверах ввода /вывода.
35. Распространенные ошибки Ассемблера в программах прерывания.

### 3. Критерии оценивания заданий к дифференцированному зачету

Оценка «отлично»	Представлен развернутый ответ на теоретический вопрос, а также поэтапное решение практического задания с пояснениями. Студент ориентируется в излагаемом материале, отвечает на дополнительные вопросы, связанные демонстрирует глубокие теоретические знания, знание первоисточников.
Оценка «хорошо»	Представлен достаточно развернутый ответ на теоретический вопрос, а также поэтапное решение практического задания с пояснениями. В решении практического задания могут быть допущены вычислительные ошибки, не искажающие лежащего в основе решения алгоритма. Студент уверенно отвечает на дополнительные вопросы, демонстрирует достаточно высокий уровень теоретических знаний, знание первоисточников.
Оценка «удовлетворительно»	Представлен не полный ответ на теоретический вопрос, В решении практического задания могут быть допущены вычислительные ошибки, не искажающие лежащего в основе решения алгоритма. Студент демонстрирует достаточный уровень теоретических знаний, однако затрудняется отвечать на отдельные вопросы.
Оценка «неудовлетворительно»	Теоретический вопрос не раскрыт, в решении практического задания допущены существенные ошибки, ввиду незнания алгоритмов решения. Либо дан ответ только на один из вопросов билета. Студент затрудняется отвечать на дополнительные вопросы, в том числе непосредственно относящиеся к сути теоретического и практического вопросов билета.