МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КОСТРОМСКАЯ ГОСУДАРСТВЕННАЯ СЕЛЬСКОХОЗЯЙСТВЕННАЯ АКАДЕМИЯ»

Утверждаю:
И.о. декана электроэнергетического факультета

Николай Александрович Климов
Подписано цифровой подписью: Николай Александрович Климов
Дата: 2024.09.11 16:11:10 +03'00'
/Климов Н.А./
11 сентября 2024 года

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Специальность 09.02.07 Информационные системы и программирование

Квалификация выпускника программист

Форма обучения очная

Срок освоения ППССЗ 3 года 10 месяцев

На базе основного общего образования

Караваево 2024

Фонд оценочных средств предназначен для оценивания сформированности компетенций по дисциплине «Системное программирование».

Разработчик:
преподаватель Д.Е. Кокорев Дмитрий Евгеньевич Кокорев Подписано цифровой подписью: Дмитрий Евгеньевич Кокорев

Утвержден на заседании кафедры информационных технологий в электроэнергетике, протокол № 1 от 05.09.2024

Заведующий кафедрой Н.А. Климов Николай Александрович Климов Подписано цифровой подписью: Николай Александрович Климов Дата: 2024.09.05 14:46:38 +03'00'

Согласовано:
Председатель методической комиссии электроэнергетического факультета

А.С. Яблоков Алексей Сергеевич Яблоков Подписано цифровой подписью: Алексей Сергеевич Яблоков Дата: 2024.09.10 15:13:03 +03'00'
протокол № 7 от 10.09.2024

Результаты освоения дисциплины

«Системное программирование»

ППССЗ (СПО) по специальности:

09.02.07 Информационные системы и программирование

Коды компетенций по ФГОС	Компетенции	Результат освоения
Общие компетенции		
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	**Знать** способы решения задач профессиональной деятельности применительно к различным контекстам. **Уметь** находить решения задач профессиональной деятельности применительно к различным контекстам проводить геометрические измерения, читать информацию, представленную в виде таблиц, графиков, схем. **Владеть** навыками выбора способа решения задач профессиональной деятельности и приемами геометрических измерений, чтения информации, представленной в виде таблиц, графиков, схем.
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	**Знать** формат оформления результатов поиска информации, порядок применения современных средств и устройств информатизации, как применять программное обеспечение в профессиональной деятельности, в том числе с использованием цифровых средств. **Уметь** оформлять результаты поиска, применять средства информационных технологий для решения профессиональных задач; планировать процесс поиска; структурировать получаемую информацию; оформлять результаты поиска информации, пользоваться современными средствами поиска информатизации. **Владеть** навыками оформления результатов поиска информации; навыками планирования процесса поиска и структурирования полученной информации.
Профессиональные компетенции		
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	**Знать** основные этапы разработки программного обеспечения. **Уметь** осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль. **Владеть** навыками разработки алгоритма решения поставленной задачи и реализации его средствами автоматизированного проектирования.
ПК 1.2	Разрабатывать программные модули в	**Знать** основные этапы разработки программного обеспечения основные принципы технологии структурного и объектно-ориентированного

	соответствии с техническим заданием	программирования **Уметь** осуществлять разработку кода программного модуля на языках низкого и высокого уровней, создавать программу по разработанному алгоритму как отдельный модуль. **Владеть** навыками разработки алгоритма решения поставленной задачи и реализации его средствами автоматизированного проектирования
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ	**Знать** основные этапы разработки программного обеспечения **Уметь** оформлять документацию на программные средства **Владеть** навыками разработки мобильных приложений

Паспорт
фонда оценочных средств

Таблица 1

№ п/п	Контролируемые дидактические единицы	Контролируемые компетенции (или их части)	Наименование оценочных средств		
			Тесты, кол-во заданий	Другие оценочные средства	
				вид	кол-во заданий
1	Тема 1. Архитектура реального режима работы микропроцессора 8086	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
2	Тема 2. Директивы и операторы языка ассемблера	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
3	Тема 3. Архитектура и система команд арифметического сопроцессора	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
4	Тема 4. Модульное программирование на ассемблере	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
5	Тема 5. Программирование Windows-приложений на ассемблере	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
6	Тема 6. Самостоятельная работа при изучении МДК 01.04 Системное программирование	ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6	20	Опрос	5
7	Темы 1-6	ОК 02 ПК 4.3 ПК 4.4		Собеседование	20
Всего:			120		50

Методика проведения контроля по проверке базовых знаний по дисциплине «Системное программирование»

Тема 1 Архитектура реального режима работы микропроцессора 8086

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Каковы основные отличия между адресацией в реальном режиме и защищенном режиме на архитектуре x86, и как эти отличия влияют на управление памятью?
2. Объясните принцип работы сегментации в микропроцессоре 8086. Как сегментные регистры (CS, DS, ES, SS) влияют на адресацию памяти?
3. Каково значение и функционирование режимов работы 8086 при выполнении операций с различными длинами данных (8-битные, 16-битные), и как это отражается на результатах арифметических операций?
4. Опишите процесс прерываний в архитектуре 8086. Каково значение векторных таблиц прерываний и как происходит обработка аппаратных и программных прерываний?
5. В чем состоит роль и функция модуля управления в архитектуре 8086, и как он взаимодействует с арифметико-логическим устройством (ALU) во время выполнения машинных команд?

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания

Выберите один правильный вариант ответа и нажмите кнопку «Далее»

Размер регистра общего назначения в 8086 составляет:

8 бит

+16 бит.

32 бит

64 бит

Для хранения данных в 8086 используется:
CS
+DS.
SS
ES

Команда MOV AX, [BX] в контексте адресации обозначает:
копирует значение из регистра AX в память
+загружает значение из памяти, адресованной регистром BX, в AX.
перемещает значение из AX в регистр BX
загружает адрес в регистр BX

На ноль результата операции в регистре флагов 8086 указывает флаг:
CF
+ ZF
SF
OF

Из перечисленных сегментов на адреса стека указывает:
DS
CS
+SS
ES

Если произойдет переполнение результата арифметической операции в 8086, то:
результат будет обрезан
+установится флаг OF
установится флаг CF
произойдет сбой системы

для перехода на адрес, заданный в регистре IP, нужно использовать:
CALL
JUMP
+ JMP
RET

Вид адресации, который используется в команде MOV AX, [1234h], называется:
+прямая адресация
косвенная адресация
регистровая адресация
индексная адресация

Размеры адреса сегмента и смещения в реальном режиме 8086 соответствует:
8 бит и 8 бит
16 бит и 16 бит
+20 бит и 16 бит
32 бит и 16 бит

Механизм сегментации в 8086 называется:
стековая адресация
плоская память
+сегментированная память
виртуальная память

Регистр, используемый для управления адресацией при работе со стеком в 8086, называется:
AX
+ SP
DI
SI

Флаг, указывающий на знаковый результат операции, называется:
CF
DF
+ SF
IF

Сегмент, который обычно используется для хранения данных в сегменте программ, называется:
ES
+ DS
CS
SS

Тип команд в 8086, который не имеет операндов, называется:
арифметические команды
команды управления
+ команды перехода
логические команды

Размер адреса, который может быть использован в команде MOV [SI], AX, соответствует:
8 бит
+ 16 бит
32 бит
64 бит

Частью архитектуры 8086 НЕ является:
ALU
блок управления
+ виртуальная память
регистры общего назначения

Флаг прерывания в регистре флагов 8086 отмечается:
+ IF
TF
DF
PF

Для взаимодействия с внешними устройствами используется сегмент:
DS
CS
+ ES
SS

Код прерывания, который чаще всего используется для завершения программы в 8086, называется:
+ INT 20h
INT 21h
INT 10h
INT 0Ah

Способ адресации, который позволяет использовать указатель на память и индекс, называется:
прямая адресация
косвенная адресация
+ индексная адресация
смешанная адресация

Методика проведения контроля

Параметры методики	Значение параметра
Предел длительности всего контроля	10 минут
Последовательность выбора вопросов	Случайная
Предлагаемое количество вопросов	10

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполнил 9-10 тестовых заданий.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполнил 7-8 тестовых заданий.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполнил 5-6 тестовых заданий.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполнил менее 4 тестовых заданий.

Тема 2. Директивы и операторы языка ассемблера

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Объясните разницу между директивой DB и директивой DW в языке ассемблера. Приведите примеры использования каждой из них. Обратите внимание на выделение различий в размере данных, назначении и способе определения.
2. Как работает директива EQU и для чего она используется в ассемблерном коде? Приведите пример, иллюстрирующий ее использование для определения констант и символьных имен. Важно подчеркнуть функциональность директивы EQU и ее влияние на процесс ассемблирования.
3. Опишите концепцию макросов в ассемблерном коде. Как создаются и используются макросы? Приведите пример макроса, который упрощает запись часто повторяющейся последовательности инструкций. В ответе необходимо объяснить механизм определения и вызова макросов, а также их преимущества в контексте кода.
4. Объясните различия между операторами JMP, CALL и RET в языке ассемблера. В каких случаях используется каждый из них и какие операции они выполняют? Необходимо подчеркнуть функциональность каждого оператора и ситуации, в которых они применяются.
5. Опишите, как работает директива ORG и для чего она используется в ассемблерном коде. Приведите пример ее использования для определения начального адреса кода или данных. Важно объяснить, как ORG влияет на адресацию и размещение данных в памяти.

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания

Выберите один правильный вариант ответа и нажмите кнопку «Далее»

Для определения сегмента данных в ассемблере используется следующий оператор:

PROC
DATA
+SEGMENT
CODE

Для резервирования памяти в сегменте данных используется директива:
RESERVE
+ DB
DW
EQU

Директива END в программе на ассемблере обозначает:
+Завершает работу программы
Обозначает конец блока данных
Указывает точку входа
Завершает обработку директив

Для определения постоянного значения используется оператор:
DEFINE
+EQU
CONST
ASSIGN

Для создания сегмента кода используется директива:
CODE
+SEG
START
TEXT

Директива ORG используется для:
+Установки начала сегмента
Определения размера переменной
Создания нового сегмента
Инициализации значения переменной

Для перехода к метке в программе на ассемблере используется оператор:
GOTO
+JMP
CALL
MOVE

Директива DB обозначает:
+Определение байта данных
пробел
Указание адреса
Определение блока данных

Для объявления подпрограммы используется оператор:
FUNCTION
+PROC
SUB
METHOD

Директива INCLUDE в ассемблере обозначает:
+Встраивает файлы в код
Удаляет лишние данные
Объявляет переменные
Определяет сегменты

Для определения метки в коде используется оператор:
+LABEL
NAME
MARK
DB

Для передачи управления в другую секцию кода используется оператор:
+CALL
JUMP
GOTO
TRANSFER

Для определения начального адреса сегмента в ассемблерном коде используется директива:
EQU
+ORG
SEGMENT
ENDS

Для возврата из подпрограммы в ассемблерном коде используется оператор:
JMP
CALL
+RET
LOOP

Для определения макроса в ассемблерном коде используется директива:
+MACRO
ENDM
LOCAL
ASSUME

Для сравнения двух значений в ассемблерном коде используется оператор:
ADD
SUB
+CMP
MOV

Для определения размера блока данных в ассемблерном коде используется директива:
+DUP
EQU
ORG
ENDS

Для условного перехода на другую строку кода в ассемблерном коде используется оператор:
JMP
CALL
+JE
RET

Директива “RESW” используется для:
+Выделения памяти для слова (2 байта)
Выделения памяти для байта
Выделения памяти для двойного слова
Установки адреса сегмента

Для завершения исходного кода программы используется оператор:
+END
FINISH
COMPLETE
STOP

Методика проведения контроля

Параметры методики	Значение параметра
Предел длительности всего контроля	10 минут
Последовательность выбора вопросов	Случайная
Предлагаемое количество вопросов	10

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполнил 9-10 тестовых заданий.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполнил 7-8 тестовых заданий.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполнил 5-6 тестовых заданий.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполнил менее 4 тестовых заданий.

Тема 3. Архитектура и система команд арифметического сопроцессора

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Объясните, как работает механизм округления в арифметическом сопроцессоре. Какие режимы округления доступны? (Важно описать типы округления (например, к нулю, к ближайшему числу, к +∞, к -∞) и их влияние на результат вычислений.)
2. Какие различия между форматами представления чисел с плавающей точкой IEEE-754 и форматами представления чисел в арифметическом сопроцессоре? (Ответ должен включать сравнение размеров, диапазонов представления чисел и особенностей реализации.)
3. Опишите, как работает команда FPU FADD и какие особенности её использования? (Необходимо объяснить назначение команды, порядок работы с операндами и влияние флагов состояния FPU.)
4. Как реализуется защита от переполнения в арифметическом сопроцессоре? (Ответ должен объяснить механизмы обнаружения переполнения, обработки ошибок и способы предотвращения ошибок.)

5. Объясните, как осуществляется обмен данными между арифметическим сопроцессором и основным процессором. Какие типы передач данных доступны? (В ответе нужно рассмотреть различные способы передачи данных, например, через регистры, через память, и их особенности.)

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания

Выберите один правильный вариант ответа и нажмите кнопку «Далее»

Флаги состояния операций с плавающей запятой регистра в x87 содержат:

+FPU Status Word (FSTSW)
FPU Control Word (FCTRL)
FPU Data Pointer (FDP)
FPU Operand Stack (FOS)

Тип данных, не поддерживаемый арифметическим сопроцессором x87, называется:

Дробные числа (реальные)
Целые числа
+Комплексные числа
Двойной точности (double precision)

Регистр отвечающий за управление округлением в арифметическом сопроцессоре, называется:

+FPU Control Word (FCTRL)
FPU Status Word (FSTSW)
FPU Data Register (FDR)
FPU Processor Status (FPS)

Для извлечения квадратного корня в x87 предназначен следующий оператор:

FSQR
+FSQRT
FSQ
FSROOT

Регистр в архитектуре x87 используемый для хранения временных результатов операций, называется:
XMM Регистры
FPU Data Register (FDR)
+Стековый регистр (ST)
Control Register (CR)

Механизм x87 позволяющий обрабатывать ошибки в вычислениях с плавающей запятой, называется:
Регистры состояния
+Обработка исключений
Даунгрейд команд
Применение кэширования

Типы команд, используемые для обработки логических операций в x87, называются:
+Арифметические команды
Логические команды
Управляющие команды
Параллельные команды

Команда FADD в x87 предназначена для:
Вычитание
Умножение
+Сложение
Деление

Подход к организации памяти, для оптимизации производительности в x87 , называется:
Оптимизация кэш-памяти
+Использование стека
Разделение данных
Обмен данных через шину

Формат данных, не поддерживаемый в инструментах арифметического сопроцессора, называется:
+HALF
SINGLE
DOUBLE
QUAD

Регистр, используемый для хранения адресов в x87, называется:
Pointer Register (PR)
FPU Control Word (FCTRL)
+FPU Operand Stack (FOS)
FPU Status Word (FSTSW)

Операция, выполняемая командой FDIV, является:
Сложение
+Деление
Умножение
Квадратный корень

Регистр в x87 отвечающий за хранение и управление результатами арифметических операций, называется:
FPU Data Register (FDR)
FPU Control Word (FCTRL)
+FPU Operand Stack (FOS)
FPU Status Word (FSTSW)

Встроенной командой для работы с плавающей запятой в x87 НЕ является:
FPTAN
FCOM
FADD
+FCLR

Инициализация стека в архитектуре сопроцессоров предназначена для:
Выбора режима работы
+Хранения данных и производимых значений
Сброс всех регистров
Оптимизации памяти

За обработку арифметических исключений в x87 отвечает:
Аппаратная маршрутизация
Векторизация команд
+Флаги состояния
Загруженные регистры

Команда, используемая для сравнения значений в x87, называется:
+FCOMP
FADD
FMUL
FSUB

Формат, являющийся форматом данных с плавающей точкой двойной точности, соответствует:
32-битный
16-битный
+64-битный
128-битный

При арифметическом переполнении с регистрами FPU происходит:
Переполнение игнорируется
+Возникает состояние ошибки
Происходит сброс всех данных
Защита от ошибок отключается

Основные компоненты архитектуры x87, лежащие в основе обработки операций с плавающей запятой, называются:
+Стек, регистры и контроллер
Процессор, кэш и шина
Регистры, кеш и память
Команды, фиксаторы и кэш

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполнил 9-10 тестовых заданий.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполнил 7-8 тестовых заданий.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполнил 5-6 тестовых заданий.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполнил менее 4 тестовых заданий.

Тема 4. Модульное программирование на ассемблере

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Объясните, как работает механизм передачи параметров в модули при модульном программировании на ассемблере. Какие способы передачи параметров существуют? (Необходимо описать различные методы передачи параметров (например, через регистры, через стек) и их преимущества и недостатки.)
2. Какие типы модулей существуют в ассемблерном программировании? В чем разница между процедурами, сегментами данных и сегментами кода в контексте модульного программирования? (В ответе необходимо описать типы модулей, их назначение и взаимодействие между ними.)
3. Как происходит связывание (линковка) модулей в ассемблерном программировании? Какую роль играет линковщик в этом процессе? (Ответ должен объяснить процесс линковки, роли линковщика в объединении модулей, решении внешних ссылок и создании исполняемого файла.)
4. Какие преимущества и недостатки имеет модульное программирование на ассемблере? В каких случаях оно особенно полезно? (В ответе нужно описать преимущества (например, повышение структурированности, упрощение отладки) и недостатки (например, сложность взаимодействия модулей) модульного программирования и ситуации, в которых оно наиболее эффективно.)
5. Объясните, как работает процесс компиляции модуля в ассемблерном программировании. Какие этапы проходят модули перед линковкой? (Ответ должен описать этапы компиляции (препроцессинг, ассемблирование), используемые инструменты и результаты компиляции.)

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания
Выберите несколько правильных вариантов ответа и нажмите кнопку «Далее»

Предложения языка ассемблера состоят из следующих компонент:
+метка или имя
+мнемоника
+операнды
+комментарии
константы
литералы

Результат при выполнении операции 96h AND 0Fh=будет равен:
A5h
10100101b
+110b
+06h
6
8CA
100011001010b

К регистрам общего назначения относят регистры:
+EAX
+EBX
+ECX
+EDX
EES
EDS
ESS
ECS

Из нижеперечисленных команд, правильно записаны:
mov ah,123h
mov bx,12345h
mov dl,100h
+mov cx,1234h
+mov al,56h
mov es,ds
+mov dx,0DEF0h

Из нижеперечисленных флагов, правильные трактовки имеют:
+флаг ZF – признак нуля
+флаг CF – признак переноса
+флаг SF – признак знака
флаг TF – признак полупереноса

После выполнения следующего фрагмента кода в регистрах сохраняться следующие значения:

```
1. .data
2. num equ 5
3. imd=num-2
4. .code
5. start:
6.       mov ax,@data
7.       mov ds,ax
8.       mov al,num
9.       sub al,2
10.      add si,imd
11.      mov ax,4C00h
12.      int 21h
13. end start
```

+imd=3
+регистр si будет содержать значение равное 3
+регистр al будет содержать значение равное 3
num=3
+num=5

При выполнении следующего фрагмента кода в регистрах сохраняться следующие значения:

```
1. .data
2. mask_b equ 10111011b
3. .code
4. mov ax,@data
5. mov ds,ax
6. mov al,mask_b shr 3
```

+в регистре al будет содержаться шестнадцатиричное значение 17
+регистр al будет содержать двоичное значение 00010111
регистр al будет содержать двоичное значение 11011000
регистр al будет содержать шестнадцатиричное значение D8

Выберите один правильный вариант ответа и нажмите кнопку «Далее»

Схема трансляции ассемблерного модуля состоит из следующих этапов:
+исходный модуль на языке ассемблера – объектный модуль – подключение библиотек и других объектных модулей – исполняемый модуль
исходный модуль на языке ассемблера - подключение библиотек и других объектных модулей – объектный модуль – исполняемый модуль
подключение библиотек и других объектных модулей - исходный модуль на языке ассемблера – объектный модуль – исполняемый модуль
нет правильного ответа

Для указания ассемблеру того, что в программе используются числа в двоичной системе исчисления необходимо:
+в конце каждого двоичного числа ставить букву «b»
в конце каждого двоичного числа ставить обозначение «bit»
в начале каждого двоичного числа ставить букву «b», а в конце 2
в начале каждого двоичного числа ставить цифру «2», а в конце букву «b»
в начале каждого двоичного числа ставить букву «b»
в конце каждого двоичного числа ставить цифру «2»

ничего не ставить

Шестнадцатеричное 96h в двоичной системе исчисления равно:
+10010110
01101001
0000011000001001
150
нет правильного варианта

Для представления отрицательного числа в компьютере выполняются следующие операции:
+инверсия положительного числа – прибавление 1 к результату инверсии = отрицательное число
прибавление 1 к положительному числу – инверсия результата = отрицательное число
побитовое сложение положительного числа с ним же самим – инверсия результата сложения плюс 1 = отрицательное число
инверсия положительного числа - побитовое сложение инвертированного результата с ним же самим плюс 1 = отрицательное число

Процессор – это:
+кремневая плата или подложка с логическими цепями, состоящими из транзисторов, скрытая в пластмассовом корпусе, снабжённом контактными ножками
кремневая плата, обеспечивающая механизм страничной организации памяти, которая необходима для любой многозадачной операционной системы
кремневая плата, хранящая инструкции и данные в виде двоичных сигналов в двоичной системе исчисления

Идентификатор регистра базы BH – это:
один из регистров общего назначения
верхние 16 разрядов регистра общего назначения
нижние 16 разрядов регистра общего назначения
один из сегментных регистров
часть сегментного регистра
+верхние 8 разрядов регистра общего назначения
нижние 8 разрядов регистра общего назначения

Сегментные регистры в архитектуре x86_32 имеют:
+16 разрядов
20 разрядов
8 разрядов
32 разряда
64 разряда
Если SA – адрес начала сегмента, OA – смещение искомого байта относительно этого начала, то физический адрес ячейки памяти можно получить по формуле:
+SA*16+OA
SA*4+OA
OA*16+ SA
OA*4+ SA

Сегментные регистры - это:
+хранят начальные адреса сегментов программы и обеспечивают возможность обращения к этим сегментам

используются для хранения данных. В эти регистры может быть записан адрес возврата в основную программу после завершения работы процедуры
хранят машинные коды команд после трансляции программы
хранят адрес инструкции, которая должна быть выполнена следующей

Имя метки – это:
+идентификатор, значением которого является адрес первого байта того предложения исходного текста программы, которое он обозначает
идентификатор, отличающий данную директиву от других одноимённых директив
мнемоническое обозначение соответствующей области памяти для хранения машинной команды или директивы транслятора
идентификатор, который обозначает поименованную область памяти для хранения адреса следующей выполняемой команды

КОП – это:
+мнемоническое обозначение соответствующей машинной команды, макрокоманды или директивы транслятора
часть команды, макрокоманды или директивы ассемблера, обозначающая объекты над которыми производятся командные операции
последовательность допустимых символов, обозначающих команду

После выполнения следующего фрагмента кода

+в AX запишется слово из области памяти по физическому адресу 0000:0000
в AX запишется двойное слово из области памяти по физическому адресу 0000:0000
в AX запишется физический адрес 0000:0000
в AX запишется логический адрес 0000:0000

Когда ассемблер встречает в программе команду jmp $+3 то:
прибавляет к переменной $ цифру 3
прибавляет к машинному коду операции цифру 3
+ к текущему смещению прибавляет 3 и переходит к команде, имеющей полученный адрес
прибавляет к содержимому регистра AX цифру 3 и переходит к команде, имеющей полученный адрес

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполнил 9-10 тестовых заданий.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполнил 7-8 тестовых заданий.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполнил 5-6 тестовых заданий.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполнил менее 4 тестовых заданий.

Тема 5. Программирование Windows-приложений на ассемблере

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Объясните, как работает механизм вызова API-функций Windows из ассемблерного кода. Какие типы вызовов API доступны в Windows? (Важно описать различные типы вызовов (например, стандартные вызовы, вызовы через таблицы импорта), их особенности, а также механизм передачи параметров и получения результата.)
2. Какие преимущества и недостатки имеет программирование Windows-приложений на ассемблере по сравнению с использованием языков высокого уровня? (Ответ должен сопоставить особенности ассемблера и языков высокого уровня, указать на преимущества (например, контроль над аппаратными ресурсами, эффективность) и недостатки (например, сложность, отсутствие средств абстракции) программирования на ассемблере.)
3. Опишите, как работает обработка сообщений в Windows-приложениях, написанных на ассемблере. Как организуется цикл обработки сообщений в ассемблерном коде? (Важно объяснить роль процедуры WndProc, механизм перехвата сообщений, и особенности их обработки в ассемблерном коде.)
4. Объясните, как осуществляется взаимодействие с графическим интерфейсом Windows из ассемблерного кода. Какие API-функции используются для создания окон, элементов управления и рисования графики? (Ответ должен описать API-функции (например, CreateWindow, DrawText, GetDC) и их назначение в контексте создания и управления графическим интерфейсом Windows.)
5. Какие сложности возникают при программировании Windows-приложений на ассемблере в сравнении с языками высокого уровня? (Ответ должен указать на сложности, связанные с управлением памятью, обработкой исключений, отладкой, а также на необходимость ручного управления ресурсами операционной системы.)

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания
Выберите один правильный вариант ответа и нажмите кнопку «Далее»

Основной компонент, отвечающий за управление окнами в Windows, называется:
API
+WNDCLASS
ORG
MODULE

Событие, вызываемое при создании окна в Windows, называется:
WM_CLOSE
WM_DESTROY
+WM_CREATE
WM_PAINT

Вызов функции, используемый для регистрации класса окна:
+RegisterClassEx
CreateWindow
GetMessage
ShowWindow

HWND в контексте Windows API – это:
+дескриптор окна
идентификатор приложения
состояние окна
цвет окна

Для создания нового окна используется:
+ CreateWindowEx
ShowWindow
MsgBox
SetWindowText

Основной поток выполнения для Windows-приложений называется:
System
+Main
Thread
Window

WM_PAINT – это:
сообщение о закрытии окна
+сообщение о рисовании окна
сообщение об управлении фокусом
сообщение о создании окна

Структура MSG относится к типу данных:
+структура сообщений
структура указателей
структура событий
структура параметров

Функция GetMessage в коде Windows-приложения используется для:
+обработки входящих сообщений
создания нового потока
изменения заголовка окна
обновления содержимого экрана

Параметр, указывающий на количество параметров для функции окна, называется:
+wParam
lParam
hWnd
nCmdShow

Выберите верное сообщение, когда окно получает фокус:
+WM_SETFOCUS
WM_KILLFOCUS
WM_ACTIVATE
WM_WINDOWPOSCHANGED

Windows 32-битная архитектура предоставляет следующие уровни защиты:
полный доступ
виртуальная память
+защита памяти
доступ к аппаратным средствам

Для отображения окна на экране используется вызов:
CreateWindow
+ShowWindow
UpdateWindow
RedrawWindow

Директива ".MODEL" в ассемблере используется для определения**:**
структуры данных
+ модели памяти
типа интерфейса
размера кода

За обработку сообщений в приложениях Windows отвечает:
MessageBox
+Window Procedure (WndProc)
ClassFactory
API Broker

Код, возвращающий функция WinMain при успешном завершении программы, соответствует:
+ 0
1
-1
255

Деривация класса окна – это _____:
+создание нового окна на базе существующего класса
установка размеров окна
определение обработчиков для событий окна
изменение внешнего вида окна

Необходимо обновить окно вручную при:
создании окна
+ изменении данных в окне
закрытии окна
минимизации окна

Взаимодействие между окнами в Windows осуществляется через:
глобальные переменные
+сообщения
прямой вызов процедур
потоки

Перед завершением работы приложения Windows необходимо:
закрыть все окна
+освободить все используемые ресурсы
сохранить все изменения
изменить статус выполнения на "завершено"

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполнил 9-10 тестовых заданий.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполнил 7-8 тестовых заданий.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполнил 5-6 тестовых заданий.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполнил менее 4 тестовых заданий.

Тема 6. Самостоятельная работа

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

Вопросы для устного опроса

1. Изучение принципа работы дизассемблеров
2. Изучение организации оперативной памяти
3. Изучение схемы ЭВМ и работы системных устройств
4. Изучение поведения регистров флагов при арифметических операциях с числами без знака
5. Изучение поведения регистров флагов при арифметических операциях с числами со знаком Модели памяти.
6. Выполнить практические задания с использованием разных моделей памяти.
7. Перехват прерываний в MSDOS
8. Команды помещения значений в стек и извлечения из него
9. Организация сопроцессора i8087
10. Обработка исключений в сопроцессоре
11. Дополнительные арифметические команды сопроцессора.
12. Выполнить предыдущие программы ассемблерными вставками языка высокого уровня C++

13. Сегментная модель Использование библиотеки WinAPI32 в ассемблере.
14. Вызов окна с надписью.
15. Создание простейшей математической библиотеки и его подключение к языку C++.

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполняет все задания, успешно применяет теоретические знания к решению практических задач с использованием алгоритма как способа автоматизации деятельности.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполняет не менее 80% заданий, применяет теоретические знания к решению практических задач.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполняет не менее 60% заданий, не совсем твердо владеет материалом, испытывает затруднения при решении достаточно сложных задач.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполняет менее 60% заданий, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при выполнении предусмотренных контрольных заданий

Вопросы собеседования к экзамену по темам 1-6

Контролируемые компетенции (знания, умения) ОК 01 ОК 02 ПК 1.1 ПК 1.2 ПК 1.6

1. Объясните разницу между физическими и логическими адресами в архитектуре компьютера. Как работает механизм преобразования адресов?
2. Как реализуется виртуальная память в современных операционных системах?
3. Какие преимущества и недостатки у виртуальной памяти?
4. Опишите архитектуру процессора с конвейерной обработкой. Как работает конвейерная обработка и какие преимущества она дает?
5. Как работает кэш-память и какие алгоритмы кэширования применяются?
6. Какую роль играет кэш-память в повышении производительности системы?
7. Объясните концепцию прерываний в архитектуре компьютера. Какие типы прерываний существуют и как они обрабатываются?
8. Какие основные задачи решает операционная система?
9. Как происходит взаимодействие между операционной системой и приложениями?
10. Объясните, как работает механизм планирования задач в операционной системе. Какие алгоритмы планирования существуют?
11. Как работает механизм синхронизации процессов в операционной системе?
12. Какие проблемы возникают при синхронизации процессов и как их решают?
13. Опишите структуру файловой системы. Как организована иерархическая структура файлов и папок?
14. Как работает управление памятью в операционной системе?
15. Какие методы управления памятью используются?
16. Объясните, как работают системные вызовы и как они используются приложениями для взаимодействия с операционной системой.
17. Что такое драйвер устройства?
18. Как драйверы устройств взаимодействуют с операционной системой и аппаратными компонентами?
19. Опишите основные этапы разработки операционной системы. Какие инструменты и технологии используются при разработке?
20. Какие трудности возникают при разработке системного ПО?

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно ответил на 5 случайно выбранных вопросов, показав достаточный уровень знаний. В случае если студент ответил на 4 вопроса правильно, но рассчитывает получить оценку «отлично» ему задаётся дополнительный вопрос ответить.

Оценка «хорошо» выставляется обучающемуся, который: правильно ответил на 4 случайно выбранных вопросов, показав достаточный уровень знаний. В случае если студент ответил на 3 вопроса правильно, но рассчитывает получить оценку «хорошо» ему задаётся дополнительный вопрос.

Оценка «удовлетворительно» выставляется обучающемуся, который: правильно ответил на 3 случайно выбранных вопросов, показав достаточный уровень знаний. В случае если студент ответил на 2 вопроса правильно, но рассчитывает получить оценку «удовлетворительно» ему задаётся дополнительный вопрос.

Оценка «неудовлетворительно» выставляется обучающемуся, который не ответил ни на один вопрос или ответил на 1 или 2 вопроса верно, но не ответил на дополнительный вопрос

Форма промежуточной аттестации по дисциплине *зачет с оценкой*.

Окончательные результаты обучения (формирования компетенций) определяются посредством перевода баллов, набранных студентом в процессе освоения дисциплины, в оценки:

– базовый уровень сформированности компетенции считается достигнутым, если результат обучения соответствует оценке «удовлетворительно» (50-64 рейтинговых баллов);

– повышенный уровень сформированности компетенции считается достигнутым, если результат обучения соответствует оценкам «хорошо» (65-85 рейтинговых баллов) и «отлично» (86-100 рейтинговых баллов).

Дополнительные контрольные испытания

для студентов, набравших менее 50 баллов (в соответствии с Положением «О модульно-рейтинговой системе»), формируются из числа оценочных средств по темам, которые не освоены студентом.