

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Машинные языки и программирование

: 09.03.04

, :

: 3, : 6

,

		6
1	()	5
2		180
3	, .	84
4	, .	36
5	, .	0
6	, .	36
7	, .	18
8	, .	2
9	, .	10
10	, .	96
11	(, ,)	
12		

(): 09.03.04

229 12.03.2015 ., : 01.04.2015 .

: 1, ,

(): 09.03.04

, 6 20.06.2017

, 6 21.06.2017

:

, . .

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.1 владение основными концепциями, принципами, теориями и фактами, связанными с информатикой; в части следующих результатов обучения:	
13.	
Компетенция ФГОС: ПК.12 способность к формализации в своей предметной области с учетом ограничений используемых методов исследования; в части следующих результатов обучения:	
5.	

2.

2.1

(
---	--

.1. 13	
1.знать основные концепции, принципы, связанные с информатикой	; ;
.12. 5	
2.знать методы и приемы формализации задач	; ;

3.

3.1

	,			
: 6				
:				
1.				
IBM PC	3	6	1, 2	IBM PC.
IBM PC.				
: IBM PC				

2.	.EXE .COM).				C/C++.
3.	Include				.
4.					.
Borland C++ Turbo Accembler					

2.		0	4	1, 2	<div><div></div><div>:</div><div>-</div><div>,</div><div></div><div>;</div><div>-</div><div>,</div><div></div><div>;</div><div>-</div><div></div><div>TD.</div><div>-</div><div>;</div><div>-</div><div>,</div><div></div><div>.</div></div>
3.	MS DOS	0	8	1, 2	<div><div></div><div>:</div><div>-</div><div>DOS</div><div>-</div><div>(</div><div>,</div><div>DosInt21);</div><div>-</div><div>ASCII-</div><div></div><div>(</div><div>,</div><div>ScanAscii);</div><div>-</div><div>(</div><div>,</div><div>);</div><div>-</div><div>,</div><div></div><div>,</div><div></div><div>;</div><div>-</div><div>TD</div></div>

4.	BIOS	0	4	1, 2	<div><div></div><div>BIOS;</div><div></div><div>,</div><div>BIOS</div><div>;</div><div></div><div>,</div><div></div><div>,</div></div>
5.	-	0	8	1, 2	<div><div></div><div>:</div><div></div><div>-</div><div>;</div><div></div><div>;</div><div></div><div>;</div><div></div><div>,</div><div></div><div>(</div><div></div><div>),</div><div></div><div>(</div><div></div><div>);</div><div></div><div>;</div><div></div><div>;</div></div>

6.	++	0	8	1, 2	: Turbo assembler Borland C++ ; (,); ,
----	----	---	---	------	---

4.

: 6				
1		1, 2	20	4
: . . Assembler : ["]/ . . . - . [.], 2010. - 636 . : . , . . . "Assembler IBM PC" [] : - / . ; . . . - . , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000156300 . - .				
2		1, 2	46	4
: . . Assembler : ["]/ . . . - . [.], 2010. - 636 . : . , . . . "Assembler IBM PC" [] : - / . ; . . . - . , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000156300 . - .				
3		1, 2	30	2
: . . Assembler : ["]/ . . . - . [.], 2010. - 636 . : . , . . . "Assembler IBM PC" [] : - / . ; . . . - . , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000156300 . - .				

5.

- , (. 5.1).

5.1

	-
	e-mail;
	e-mail;
	e-mail;
	; ;

6.

(),

15-

ECTS.

. 6.1.

6.1

	.	
: 6		
Лабораторная:	24	60
"Assembler IBM PC" [] : http://elibrary.nstu.ru/source?bib_id=vtls000156300. -		
Курсовая работа:	50	100 (в состав баллов за КР)
"Assembler IBM PC" [] : http://elibrary.nstu.ru/source?bib_id=vtls000156300. -		
Экзамен:	20	40
"Assembler IBM PC" [] : http://elibrary.nstu.ru/source?bib_id=vtls000156300. -		

6.2

6.2

		/	/	
.1	13.	+	+	+
.12	5.	+	+	+

1

7.

1. Афанасьев В. А. Программирование. Раздел "Assembler IBM PC" [Электронный ресурс] : электронный учебно-методический комплекс / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156300. - Загл. с экрана.

2. Афанасьев В. А. Assembler IBM PC. Ч. 1. лабораторный практикум : учебное пособие [для 2 курса АВТФ по специальности 2201 "Вычислительные машины, комплексы, системы и сети"] / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2003. - 115 с. : табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000023724. - На обл. загл.: Assembler IBM PC.

1. Юров В. И. Assembler : [учебник] / В. И. Юров. - СПб. [и др.], 2008. - 636 с. : ил., табл.

2. Афанасьев В. А. Assembler IBM PC : конспект лекций / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 257, [1] с. : ил.

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>

2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>

3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>

4. ЭБС "Znaniium.com" : <http://znaniium.com/>

5. :

8.

8.1

1. Юров В. И. Assembler : [учебное пособие по направлению " Информатика и вычислительная техника"] / В. И. Юров. - СПб. [и др.], 2010. - 636 с. : ил., табл.

8.2

1 Microsoft Windows

9.

1	(Internet)	Internet

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Машинные языки и программирование

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии
разработки программного обеспечения

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине **Машинные языки и программирование** приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.1 владение основными концепциями, принципами, теориями и фактами, связанными с информатикой	з13. знать основные концепции, принципы, связанные с информатикой	Ассемблирование и отладка готовых программ. Изучение системы команд, их кодирование, способы адресации операндов Введение в язык ассемблера. Архитектурное своеобразие процессоров ПК. Характеристика типов данных в ПК IBM PC Представление чисел в формате с плавающей точкой. Программная модель IBM PC. Работа со стеком. Режимы адресации и форматы команд. Концепция модульного программирования. Процедуры, передача параметров между программой и процедурой. Многомодульные программы и их ассемблирование. Макросы. Блоки повторения. Библиотеки макросов. Директива Include Настройка программной среды. Принципы взаимодействия между Turbo Assembler и Borland C++. Практические приёмы построения двуязычных программ. Процедура вызова из C++ ассемблерных функций. Передача параметров ассемблерным модулям. Соглашения о сохранении регистров. Возврат значений из ассемблерного модуля в программу C++. Вызов функций C++ из программы на ассемблере. Демонстрационная программа с использованием взаимных вызовов. Преобразование форматов числовых данных в операциях ввода-вывода Программирование алгоритмов сложных арифметических операций Разработка программы на языке C++ с ассемблерным модулем Система прерываний. Программные и	Курсовая работа Отчет по лабораторной работе	Экзамен, вопросы 1-27

		<p>аппаратные прерывания и процедуры их обработки. Прикладные обработчики прерываний Системные функции MS DOS ввода - вывода символьной информации. Обработка строковых переменных Стандартные директивы управления сегментами. Упрощенные директивы описания сегментов. Образы программ на диске и в памяти (формат .EXE и .COM). Использование сегментов данных дальнего типа. Синтаксис ассемблера. Основные этапы разработки и отладки программы на ассемблере. Объектный код программы. Структуры данных и их инициализация в программе. Директивы резервирования и инициализации данных. Макроопределения или директивы присваивания. Выражения и операторы. Команды передачи данных. Способы адресации операндов. Арифметические команды и машинная арифметика. Команды логических операций и сдвига. Операции над битами и байтами. Команды передачи управления и организации циклов. Команды обработки строк. Программирование типовых управляющих структур C/C++ Функции BIOS для работы с экраном и клавиатурой</p>		
ПК.12/НИ способность к формализации в своей предметной области с учетом ограничений используемых методов исследования	35. знать методы и приемы формализации задач	<p>Ассемблирование и отладка готовых программ. Изучение системы команд, их кодирование, способы адресации операндов Введение в язык ассемблера. Архитектурное своеобразие процессоров ПК. Характеристика типов данных в ПК IBM PC Представление чисел в формате с плавающей точкой. Программная модель IBM PC. Работа со стеком. Режимы адресации и форматы команд. Концепция модульного программирования. Процедуры, передача параметров между программой и процедурой. Многомодульные программы и их ассемблирование. Макросы. Блоки повторения. Библиотеки макросов. Директива Include Настройка</p>	Курсовая работа Отчет по лабораторной работе	Экзамен, вопросы 1-27

		<p>программной среды.</p> <p>Принципы взаимодействия между Turbo Assembler и Borland C++. Практические приёмы построения двуязычных программ.</p> <p>Процедура вызова из C++ ассемблерных функций.</p> <p>Передача параметров ассемблерным модулям.</p> <p>Соглашения о сохранении регистров. Возврат значений из ассемблерного модуля в программу C++. Вызов функций C++ из программы на ассемблере.</p> <p>Демонстрационная программа с использованием взаимных вызовов. Преобразование форматов числовых данных в операциях ввода-вывода</p> <p>Программирование алгоритмов сложных арифметических операций</p> <p>Разработка программы на языке C++ с ассемблерным модулем Система прерываний. Программные и аппаратные прерывания и процедуры их обработки.</p> <p>Прикладные обработчики прерываний Системные функции MS DOS ввода - вывода символьной информации. Обработка строковых переменных</p> <p>Стандартные директивы управления сегментами.</p> <p>Упрощенные директивы описания сегментов. Образы программ на диске и в памяти (формат .EXE и .COM).</p> <p>Использование сегментов данных дальнего типа.</p> <p>Синтаксис ассемблера.</p> <p>Основные этапы разработки и отладки программы на ассемблере. Объектный код программы. Структуры данных и их инициализация в программе. Директивы резервирования и инициализации данных.</p> <p>Макроопределения или директивы присваивания.</p> <p>Выражения и операторы.</p> <p>Команды передачи данных.</p> <p>Способы адресации операндов. Арифметические команды и машинная арифметика. Команды логических операций и сдвига.</p> <p>Операции над битами и байтами. Команды передачи управления и организации циклов. Команды обработки строк. Программирование типовых управляющих</p>		
--	--	--	--	--

		структур C/C++ Функции BIOS для работы с экраном и клавиатурой		
--	--	--	--	--

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 6 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.1, ПК.12/НИ.

Экзамен проводится в устной форме

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 6 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.1, ПК.12/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Паспорт экзамена

по дисциплине «Машинные языки и программирование», 6 семестр

1. Методика оценки

Экзамен проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-18, второй вопрос из диапазона вопросов 19-26 (список вопросов приведен ниже). В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № 7

к экзамену по дисциплине «Машинные языки и программирование»

1. 1. Вопрос 1. Характеристика команд для выполнения логических операций и команд сдвига. Примеры типового использования команд.

2. 2. Вопрос 2. Макросы. Определения. Примеры. Локальные метки в макросе.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) (дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет 10 баллов.
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает не принципиальные ошибки, например, вычислительные, оценка составляет 20 баллов.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при

ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает ошибок при решении задачи, оценка составляет 30 баллов.

- Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет 35 баллов.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к экзамену по дисциплине «Машинные языки и программирование»

3. Представление числовых и символьных данных в ПК. Базовый одинарный формат для вещественных чисел.

4. Сегментированная модель памяти реального режима. Формирование исполнительного (физического) адреса.

5. Программная модель 32-разрядных процессоров. Общие и специальные функции регистров.

6. Функциональное назначение сегментных регистров в ассемблерных программах. Возможности переназначения. Организация стека.

7. Флаги условий и управления процессором. Характеристика команд с точки зрения их влияния на флаги.

8. Форматы команд базового процессора I8086 и основные принципы их двоичного кодирования. Виды адресации.

9. Структуры данных программы и их инициализация (размещение) в памяти.

10. Использование директив макроопределений EQU и =.

11. Операторы языка ассемблера и их использование в адресных и константных выражениях.

12. Стандартные директивы управления сегментами. Шаблоны программ .EXE и .COM.

13. Набор упрощенных директив управления сегментами. Шаблоны программ .EXE и .COM.

14. Организация программы с использованием сегментов дальнего типа.

15. Основные типы команд передачи данных. Примеры записи команд общего назначения для различных способов адресации операндов.

16. Команды загрузки эффективного адреса, табличной трансляции и преобразования форматов (CBW, CWD и CWDE).

17. Арифметические команды. Рассмотреть примеры выполнения команды сложения (вычитания) для конкретных операндов с целью определения состояния флагов результата. Сформулируйте понимание и реализацию принципа арифметики многократной точности.

18. Характеристика команд для выполнения логических операций и команд сдвига. Примеры типового использования команд.

19. Команды безусловной и условной передачи управления. Примеры ассемблерной записи команды JMP для различных типов переходов.

20. Перечень команд условных переходов с указанием условий перехода (состояний флагов).

21. Программирование на ассемблере управляющих структур языка C++:

- условный оператор if_else и его разновидности;
- логический переключатель switch-case.

22. Ассемблерная команда Loop организации цикла и цикл типа For в C++.

Дополнительные команды организации циклов в ассемблере. Пример применения.

23. Организация процедуры и её интерфейс с основной программой. Способы передачи параметров между основной программой и процедурой.

24. Макросы. Определения. Примеры. Локальные метки в макросе.

25. Концепция модульного программирования. Организация программных модулей. Директивы PUBLIC и EXTRN.

26. Система прерываний в компьютере. Аппаратные и программные средства, реализующие механизм прерываний. Процедура обработки прерываний в реальном режиме.

27. Резидентные программы и их организация.

Нереентерабельность MS DOS и пути её преодоления в обработчиках аппаратных прерываний.

Паспорт курсовой работы

по дисциплине «Машинные языки и программирование», 6 семестр

1. Методика оценки.

Комплект заданий для курсовой работы

по дисциплине *Машинные языки и программирование*
(наименование дисциплины)

Тема: "Разработка прикладных обработчиков прерываний и резидентных программ".
Объём часов самостоятельной работы для выполнения КР – 20 часов.

Цели:

1. Изучение вопросов, связанных с
 - процедурными особенностями программных и аппаратных прерываний;
 - организацией резидентных программ.
2. Разработка собственных прикладных обработчиков прерываний и резидентных программ в соответствии с индивидуальным заданием.

Методические материалы и индивидуальные задания на проектирование изложены в руководстве к выполнению КР

Требования к оформлению и содержанию КР.

1. КР должно иметь титульный лист и состоять из 12 - 15 листов печатного текста шрифтом, равный 12 пунктам.
2. Содержание КР включает следующие обязательные параграфы (разделы):
 - Задание.
 - Структуру программы и обоснование выбора её отдельных модулей (составных обработчиков прерываний). При разработке структуры должны учитываться следующие факторы:
 - вычислительные процедуры, используемые в программе, должны соответствовать заданному алгоритму реализации;
 - резидентная программы должна обязательно включать прикладной обработчик клавиатурного прерывания Int new_09h, быть защищенной от повторной установки, обладать способностью самовыгрузки из памяти;
 - прикладной аппаратный обработчик прерываний должны строиться с учётом свойства нереентерабельности DOS;
 - программа должна иметь хорошо продуманный экранный интерфейс.
3. Справочное описание используемых в программе системных обработчиков, сервисных функций DOS и BIOS.
4. Функциональное описание программы и результаты её демонстрации.
5. Листинги программ.
6. Необходимые компьютерные файлы на флэш-памяти для демонстрации программ.

2. Критерии оценки.

Каждое задание курсовой работы оценивается в соответствии с приведенными ниже критериями.

Курсовая работа считается **невыполненной**, если оценка составляет менее 50 баллов.

Курсовая работа считается выполненной на **пороговом** уровне, если дано теоретическое обоснование и предложен ход практического решения поставленной проблемы, оценка составляет 60 баллов.

Курсовая работа считается выполненной на **базовом** уровне, если дано теоретическое обоснование и выполнено практическое решение, оценка составляет 85 баллов.

Курсовая работа считается выполненной на **продвинутом** уровне, если дано теоретическое обоснование, выполнено практическое решение и предложены альтернативные варианты решения, оценка составляет 95 баллов.

3. Шкала оценки.

В общей оценке по дисциплине баллы за работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем курсового проекта (работы).

Задания на КР

1 – 10 варианты.

Индивидуальные особенности первых десяти вариантов определяются видом вычислительной процедуры Calc, которая представляет собой различные варианты реализации сложных арифметических операций, рассмотренных в лабораторной работе «Программирование алгоритмов сложных арифметических операций».

1. *Беззнаковое умножение в формате 16x16 (рис. 10.2).*
2. *Знаковое умножение по способу 1 в формате 16x16 (рис. 10.2).*
3. *Знаковое умножение по способу 2 в формате 16x16 (рис. 10.4).*
4. *Знаковое умножение по способу 3 в формате 16x16 (рис.10.5).*
5. *Двоичное знаковое умножение 32-битовых сомножителей на основе команды беззнакового умножения Mul для 16- битовых слов (рис.10.6).*
6. *Беззнаковое деление в формате 32:16 (рис 10.8).*
7. *Знаковое деление в формате 32:16 (рис.10.10).*
8. *Знаковое деление четырёхсловного делимого на однословный делитель на основе использования штатной команды процессора div src (формат 64:16=64).*
9. *Извлечение квадратного корня из 32- битового числа (рассмотреть два из наиболее распространённых алгоритмов).*
10. *Знаковое умножение многоразрядных неупакованных BCD-чисел в формате $N \times M = N + M - 1$, где N и M- число десятичных разрядов, включая знаковый разряд.*

Параграф §12 настоящего руководства по курсовому проекту как раз и посвящён реализации резидента с одной из указанных процедур.

Вычислительная процедура и алгоритмические особенности её реализации должна быть представлены в виде подпрограммы в специальном разделе проекта:

- Описание алгоритма;
- Программный модуль с использованием тех обозначений переменных и имён регистров, которые используются при рассмотрении модели алгоритма;
- Отладка модуля с использованием **Turbo Debugger** и результаты отладки (см. пункт 10.14 лабараторной работы «Программирование алгоритмов сложных арифметических операций»).

11. Резидент-замедлитель исполнения программ. Замедление реализуется процедурой задержки, вставленной в обработчик прерывания пользователя (продемонстрируйте различные способы реализации задержки на работу программы–замедлителя)

Инициализация резидента осуществляется через посредство комбинации клавиш **<Scroll Lock+Цифровая клавиша>**. Значение символа цифровой клавиши влияет на скорость замедления.

12. Разработать прикладной резидентный обработчик прерывания **new_04h**, вызывающий прерывание тестовой задачи (с выводом соответствующего сообщения) при возникновении переполнения в операциях сложения и вычитания. Резидентный обработчик **new_04h** переводится в активное состояние нажатием клавиш **Alt_A**.

13. Разработать тестовую задачу с прикладным обработчиком **new_09h**, осуществляющего вывод на экран (после каждого нажатия произвольной клавиши) содержимого порта **60h** контроллера клавиатуры, слово статуса флагов клавиатуры и первых 4-х байтов кольцевого буфера клавиатуры. Выход из программы по истечению обработки 20 клавиш.

14. Разработать резидентную программу с прикладным обработчиком клавиатуры **new_09h**. Программа осуществляет вывод на экран код нажатия и отпускания для произвольной клавиши. Выход из программы по истечению обработки 40 клавиш.

15. Разработать резидентную программу с прикладным обработчиком **new_09h** от клавиатуры. Программа осуществляет вывод на экран скан-код и **ASCII**- код для произвольной клавиши. Выход из программы по истечению обработки 40 клавиш.

16. Разработать резидентную программу дампа регистров общего назначения, активизируемую через вектор **new_03h** (прерывание в контрольной точке). Содержимое регистров выводится на экран в 16-теричном коде. Демонстрацию программы провести на основе тестовой задачи, в которой предусмотреть участки с занесением чисел в выводимые резидентом регистры (каждый такой участок оканчивается командой **Int 3**).а.

17. Разработать резидентную программу динамического дампа памяти данных тестовой задачи, активизируемую через вектор **new_03h** (прерывание в контрольной точке). Вывод данных осуществляется в формате **Turbo Debugger**, соответствующем команде внутреннего меню окна дампа **Alt+F10 > Display as > Byte**. Демонстрацию программы провести на основе тестовой задачи, в которой сегмент данных содержит текстовое сообщение в несколько строк.

18. Разработать прикладной обработчик **new_01h** пошагового выполнения (трассировки) программы. Содержательная часть обработчика должна выводить на экран содержимое всех регистров РОН после исполнения каждой команды для той части программы, для которой флаг **TF=1**. Покомандное исполнение реализуется клавишей **F8**. Демонстрацию программы провести на основе специально разработанной программы. Желательно, чтобы участок программы, предназначенный для исполнения программы в пошаговом режиме, сопровождался выводом на экран (так же как и его окончание), выводом соответствующего сообщения.

При написании программы и её демонстрации обратите внимание на тот факт, что некоторые команды не вызывают прерывание **int 1h**. Такими командами являются **cli, pushf, int, into, int 3, iret и popf**, а также команды, следующие за командами загрузки сегментных регистров.

19. Некоторая тестовая задача выводит на экран текущее время из КМОП – часов (функция **02h int 1Ah**) и, добавляя к нему 20 секунд, устанавливает будильник (функция **06h int 1Ah**). Срабатывание будильника активизирует резидентную программу (обработчик **4Ah**), которая выводит на экран приветственное сообщение и короткий звонок (одноголосая мелодия).

20. Некоторая тестовая задача считывает текущее время из КМОП – часов(функция **02h int 1Ah**) и, добавляя к нему 1 секунду, устанавливает будильник. Срабатывание будильника активизирует резидентную программу (обработчик **4Ah**), выводящую на экран монитора (правый верхний угол) текущее время из КМОП – часов с интервалом в 1 сек в течение 10 сек. Вывод производится способом прямой записи в видеопамять.