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

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 NxM=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 Debuggera** 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 течение10 сек. Вывод производится способом прямой записи в видеопамять.