

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Лабораторная работа № 3

по Ассемблеру на тему:

"Создание интерфейса
пользователя для работа с файлами"

*Выполнил студент группы ВТ-21: Потеренко А.Г.
Проверил преподаватель: Коноплянов А.В.*

Димитровград 2004г.

Порядок работы.

1. Анализ индивидуального задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке ассемблер.
3. Разработка контрольных примеров.
4. Отладка программ.
5. Составление отчета.

Содержание отчета.

	Стр.
1. Текст постановки задачи.....	3
2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.....	3
3. Алгоритм.....	4-8
4. Описание и обоснование контрольных примеров.....	9-10
5. Текст программы с комментариями (в виде приложения).....	11-20

Отчет о выполненной работе.

1. Текст постановки задачи.

Необходимо написать программу, которая обеспечивает 3 функции для работы с файлами: удаление файлов, открытие и чтение файлов в рабочее окно, и сохранения содержимого рабочего окна в файл. При этом необходимо также написать небольшой интерфейс пользователя.

2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.

Данная программа и ее модули имеют модель памяти SMALL, где абсолютно весь код размещается в одном физическом сегменте, поэтому метки уточняются атрибутом NEAR (переход в пределах текущего сегмента, то есть без изменения cs). Данные и стек размещаются в других физических сегментах.

Программа в нашем случае состоит из нескольких модулей. Исходные модули (M1.ASM,...,M30.ASM) транслируются по отдельности, затем полученные объектные модули объединяются компоновщиком в исполняемую программу. Некоторые модули нашей программы связаны друг с другом. Одни модули содержат набор подпрограмм, к которым обращаются другие модули. Главный модуль (MN.ASM) вызывает абсолютно все подпрограммы из других модулей. В данном случае Турбо Ассемблер для обеспечения таких программ предусматривает три директивы: PUBLIC, EXTRN, GLOBAL. Директива PUBLIC указывает Турбо Ассемблеру, что соответствующие переменные, процедуры, определенные в модуле А можно сделать доступными для других модулей (В). Далее возникает вопрос, каким образом другие модули (В) могут ссылаться на эти метки (из модуля А). Для того чтобы сделать метки из другого модуля (В) доступными в данном модуле (А), используется директива EXTRN. Директива GLOBAL делает доступными метки (например, процедуру модуля В) для модуля А (в котором данная процедура объявлена с помощью EXTRN).

Главный объект данной задачи – окно, объявляется с помощью структуры OKNO. Данная структура имеет 11 полей: координаты окон (4 поля), цвет окна (1 поле), номер окна (1 поле); седьмое и восьмое поле – положение указателя для ввода в при открытии рабочего окна. Девятое поле – содержит смещение относительно начала сегмента буфера для хранения текста каждого окна; предпоследнее поле имеет значение от 0 до 693 – показывает, с какого места производить дальнейший ввод текста; последнее поле имеет значения 0,1,2 (данное окно еще не существует, открыто, закрыто пользователем).

Рассмотрим теперь работу с файлами. Данная программа работает с файлами, размером ≤ 693 байта, то есть может сохранять содержимое окна, которое вмещает в себя текст размером не больше 693 байтов и также отображать файлы, размером не большие 693 байтов. Программа не сохраняет пустые файлы (размером 0 байт). Удаление файлов также является функцией данной программы. Если файл, который захочет удалить пользователь, не существует, выдается сообщение об ошибке.

Некоторые характеристики исполняемой программы (MN.EXE):

- Размер: 14,4 КБ (14817 байт).
- На диске: 32 КБ (32768 байт).
- Трансляция и компоновка модулей производилась программой Borland Turbo Assembler v5.0.

3. Алгоритм.

Алгоритм процедуры модуля MN.ASM

НАЧАЛО

настроить мышь
убрать курсор
нарисовать рабочую область программы

ПОВТОРИТЬ

показать курсор, когда это необходимо
ввод с клавиатуры (включая фильтр от исключительных ситуаций)

ЕСЛИ произошел выход из программы ТО

НАЧАЛО

удалить обработчик
выйти из программы

КОНЕЦ

установить обработчик прерывания от мыши

ПОКА бесконечно

КОНЕЦ

Обработчик прерывания от мыши (HANDLER):

НАЧАЛО

все проверки относительно меню:
(где вводить с клавиатуры, отображать ли поля меню FILE, отображать ли курсор)
обработка левой кнопки мыши клавиатуры
ЕСЛИ пользователь хочет переместить окно ТО перемещаем его
ЕСЛИ пользователь хочет выделить окно ТО выделяем его
отображаем текст в рабочем окне

КОНЕЦ

Алгоритм процедуры модуля M1.ASM (FILE_POLE)

Рисует элементы главного меню FILE – 4 поля.

Алгоритм процедуры модуля M2.ASM (POLE)

Задача данной процедуры – выделение полей меню FILE при перемещении мыши.

Алгоритм процедуры модуля M3.ASM (NASTROIKA_MOUSE)

Настраивает мышь (инициализация и установка курсора в заданную область) в самом начале работы программы.

Алгоритм процедуры модуля M4.ASM (RAMA)

Данная процедура рисует раму любого размеры. В эту процедуру передаются 3 параметра RAM1 (смещение в видеопамяти), RAM2 (количество символов по горизонтали), RAM3 (количество символов по вертикали).

Алгоритм процедуры модуля M5.ASM (WINDOW)

Процедура рисования рабочего окна. Рисует само окно и различные к нему дополнения. Параметры, передающиеся в процедуру равны: PAR1, PAR2, PAR3, PAR4, PAR5, PAR6.

Алгоритм процедуры модуля M6.ASM (CLOSE_WINDOW)

Процедура модуля для рисования кнопки закрытия рабочего окна.

Алгоритм процедуры модуля M7.ASM (WINDOW_ZAGOLOBOK)

Процедура рисования любого окна. По пяти значениям PAR1, PAR2, PAR3, PAR4 (декартовы координаты) рисует окно цвета PAR5.

Алгоритм процедуры модуля M8.ASM (FILE)

Полностью рисует окно и раму меню FILE и вызывает соответствующую процедуру рисования полей меню.

Алгоритм процедуры модуля M9.ASM (WINDOW_STIR)

Закрашивает старое рабочее окно черным цветом.

Алгоритм процедуры модуля M10.ASM (PROCEDURE_MAS)

MAS: массив слов.

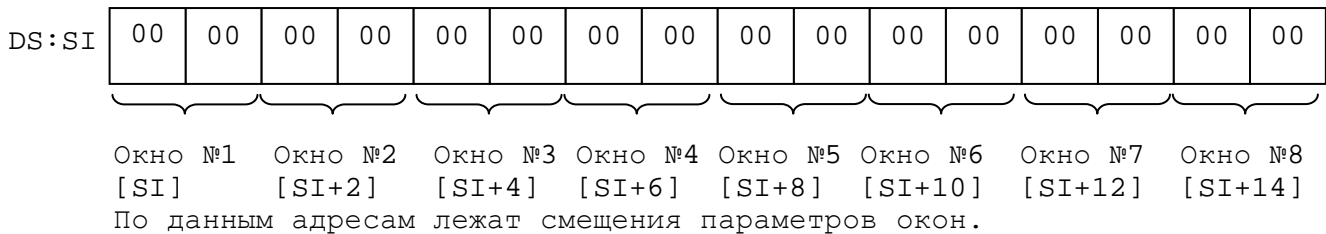
CD: количество открытых окон;

SI: смещение массива MAS относительно начала сегмента;

Выход:

DI: содержит смещение всех параметров окна, имеющего фокус ввода. В массив MAS значения

DI заносятся уже другой процедурой.



Алгоритм процедуры модуля M11.ASM (CLICK_MOUSE_FLAG_TRUE)

Основная процедура перемещения окон. Переменные ERC и ERD используются только в процедуре. Переменные CX_1 и DX_1 используются для того чтобы определить конечные и начальные координаты мыши при перемещении окон. Также здесь используются переменные, обозначенные как DELX и DELY – та разность, которую надо прибавить к координатам окон при перемещении.

НАЧАЛО

сохранить горизонтальную координату окна
вычесть из этой координаты предыдущее значение горизонтальной координаты окна
записать в DELX значение этой разности
сохранить вертикальную координату окна
вычесть из этой координаты предыдущее значение вертикальной координаты окна
записать в DELY значение этой разности
вызвать процедуру PROCEDURE_MAS
стереть старое окно, вызвав процедуру WINDOW_STIR
ЕСЛИ нет выхода из диапазона ТО
 преобразовать координаты, прибавив к старым DELX и DELY
восстановить горизонтальную координату
занести ее значение в переменную CX_1
восстановить вертикальную координату
занести ее значение в переменную DX_1

КОНЕЦ

Алгоритм процедуры модуля M12.ASM (ACTIV_POLE)

Процедура по данным значениям текущих координат мыши определяет и рисует то поле меню FILE, на котором находится указатель мыши в данный момент.

Алгоритм процедуры модуля M13.ASM (STIR_FILE)

Стирает окно меню FILE в то время, когда его необходимо закрыть.

Алгоритм процедуры модуля M14.ASM (TUTOR)

Процедура модуля отвечает за закрытие окна, имеющего фокус ввода. Если пользователь действительно нажал на кнопку закрытия, то величина поля [DI+20] данного активного окна устанавливается в 2.

Алгоритм процедуры модуля M15.ASM (USER)

Процедура данного модуля выводит на экран окна (с текстом) в порядке наложения одного окна на другое по правилу массива MAS, если это окно не является закрытым. Если окно закрыто, то его просто пропускаем не печатая.

Алгоритм процедуры модуля M16.ASM (PROBERKA_ZAKRITIE)

Процедура данного модуля сортирует массив MAS: смещение закрываемого окна записываем первым в массив MAS (становится последним по приоритету), а остальные сдвигаем вверх на 1. Причем, если создано всего одно окно (CD=1) и мы его закрываем, то вообще не сортируем массив.

Алгоритм процедуры модуля M17.ASM (OTOBRAGENIE_ACTIV_WINDOW)

НАЧАЛО

заполнить полями активного окна параметры PAR1, PAR2, PAR3, PAR4, PAR6
ЕСЛИ окно закрыто, то есть имеет значение 2 соответствующего поля ТО
 ПЕРЕХОД НА МЕТКУ NET_PROSMOTR
ИНАЧЕ вызвать процедуру WINDOW
МЕТКА NET_PROSMOTR

КОНЕЦ

Алгоритм процедуры модуля M18.ASM (USER_ACTIV_WINDOW)

Процедура модуля для выделения активного окна.

НАЧАЛО

ЕСЛИ количество открытых окон больше 1-цы ТО

НАЧАЛО

ищем то окно, которое необходимо выделить, перемещаясь по массиву MAS сверху вниз в порядке понижения приоритета

ЕСЛИ окно закрыто (поле [DI+20]=2) ТО не надо его выделять

ЕСЛИ окно открыто (поле [DI+20]=1) ТО

ЕСЛИ мы окно не закрываем ТО

НАЧАЛО

вызываем процедуру ОТОБРАЖЕНИЕ_ACTIV_WINDOW

ЕСЛИ окон открыто больше одного ТО вызываем процедуру SORTIROVKA

КОНЕЦ

ИНАЧЕ вызываем процедуру TUTOR

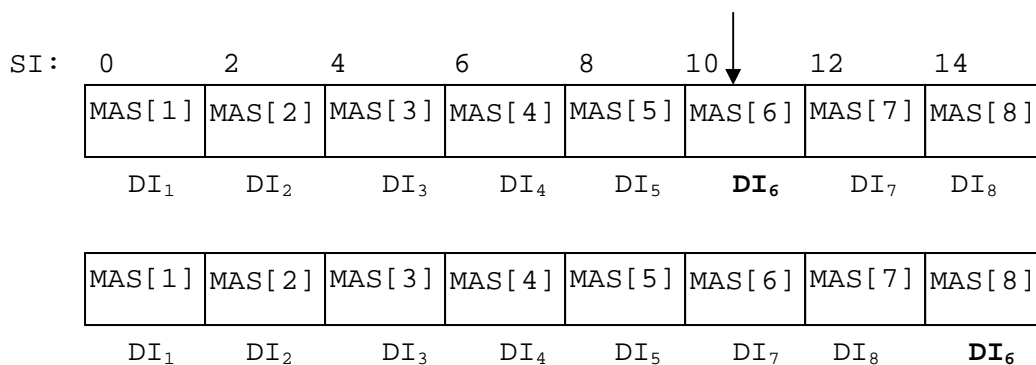
КОНЕЦ

КОНЕЦ

Алгоритм процедуры модуля M18.ASM (SORTIROVKA)

Выделенное окно, то есть его уникальное смещение DI в массиве MAS становится последним по порядку, а остальные DI других окон смещаются на 1 вниз, то есть массив сортируется своеобразным образом.

Хочет выделить пользователь



Алгоритм процедуры модуля M19.ASM (NEW_WINDOW)

Данная процедура модуля создает объект-окно, то есть заполняет соответствующую структуру координатами окна и полями, например, таким как номер окна.

Константы: размеры окна.

Xo1:3;

Xo2:67;

Yo1:3;

Yo2:15.

Константы: при открытии новых окон – разница между двумя окнами, выраженная в координатах.

DELTAX:1;

DELTAY:1.

Переменные: благодаря ним следующее окно при открытии смещается по X и Y на 1-цу. Не используются в основной программе.

CD1: имеет значения 0,1,2,3,4,5,6,7.

CD2: имеет значения 0,1,2,3,4,5,6,7.

XUMAS: имеет значения 0,2,4,6,8,10,12,14. Благодаря этому счетчику мы перемещаемся по массиву приоритетов, то есть выступает в роли смещения относительно массива MAS.

R: выступает счетчиком при открытии окон и имеет значения, равные 0, 22, 44, 66, 88, 110, 132, 154. То есть счетчик помогает продвигаться по структуре, прибавляя каждый раз к себе значение 22, так как в структуре 11 полей и каждое имеет размер слова.

Алгоритм процедуры модуля M20.ASM (RISOBANIE_NEW_WINDOW)

Процедура модуля, решающая создавать или открывать закрытые окна заново.

НАЧАЛО

загружаем в SI смещение массива MAS относительно начала сегмента

в DI загружаем [SI] – смещение самого неактивного окна (так как после закрытия окна смещение этого окна встает первым в массив MAS)

ЕСЛИ поле [DI+20]=2 (окно закрыто) ТО

НАЧАЛО

сортируем массив MAS: смещение данного окна ставим последним в MAS, а остальные смещения других окон сдвигаем вниз на 1
заносим 1 в поле [DI+20] данного окна
очищаем буфер этого окна

КОНЕЦ

ИНАЧЕ создаем новое окно

КОНЕЦ

Алгоритм процедуры модуля M21.ASM (TEXT)

Задача этой процедуры – вывести на экран в окно весь текст (не более 693 байт) из буфера, содержащего ASCII коды символов данного окна.

Алгоритм процедуры модуля M22.ASM (BBOD_WINDOW_SAVE)

Процедура ввода в меню SAVE и сохранения содержимого активного окна в файл. Мы создаем файл и открываем его, записываем в него буфер окна, закрываем. Причем, если окно пустое, то мы файл удаляем вообще.

Алгоритм процедуры модуля M23.ASM (BBOD_WINDOW_DELETE)

Процедура ввода в меню DELETE пути к файлу и удаление файла нажатием ОК. Если произошла ошибка, то происходит вывод соответствующего сообщения.

Алгоритм процедуры модуля M24.ASM (DELETE_WINDOW)

НАЧАЛО

устанавливаем флаг FLAG_DELETE в единицу, то есть данное меню открыто
очищаем рабочую область экрана
восстанавливаем содержимое рабочего окна
ЕСЛИ открыто хотя бы одно окно TO вызываем процедуру USER
рисуем меню DELETE

КОНЕЦ

Алгоритм процедуры модуля M25.ASM (SAVE_WINDOW)

НАЧАЛО

устанавливаем флаг FLAG_SAVE в единицу, то есть данное меню открыто
очищаем рабочую область экрана
восстанавливаем содержимое рабочего окна
ЕСЛИ открыто хотя бы одно окно TO вызываем процедуру USER
рисуем меню SAVE

КОНЕЦ

Алгоритм процедуры модуля M26.ASM (OPEN_WINDOW)

НАЧАЛО

устанавливаем флаг FLAG_OPEN в единицу, то есть данное меню открыто
очищаем рабочую область экрана
восстанавливаем содержимое рабочего окна
ЕСЛИ открыто хотя бы одно окно TO вызываем процедуру USER
рисуем меню OPEN

КОНЕЦ

Алгоритм процедуры модуля M27.ASM (PROBERKA_VSE_OKNA_CLOSE)

Возвращает значение FLAGG=1, если все окна закрыты.

НАЧАЛО

ПОВТОРИТЬ

ЕСЛИ нашли хотя бы одно открытое окно TO переход на метку NET_FLAGG

ПОКА не прошли все окна

значение флага FLAGG сделать равным 1-це

МЕТКА NET_FLAGG

КОНЕЦ

Алгоритм процедуры модуля M28.ASM (BBOD_WINDOW_OPEN)

Процедура ввода в меню OPEN и отображение содержимого файла в рабочем окне программы (не более 693 байтов). Программа открывает файл и производит из него чтение в буфер, затем записывает этот буфер в буфер рабочего окна при открытии.

Алгоритм процедуры модуля M29.ASM (BBOD_WINDOW_USER)

Данный алгоритм разработан для ввода текста пользователем в рабочее окно.

НАЧАЛО

вызвать процедуру PROCEDURE_MAS

узнать смещение в BUFER-e, так как мы записываем коды символов туда

ЕСЛИ пользователь нажал ESC ТО перейти на метку EXIT_USER
ЕСЛИ пользователь нажал клавишу "↑" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 отнять от текущего значения положения курсора значение 63
 перейти на метку EXIT_USER
 КОНЕЦ
ЕСЛИ пользователь нажал клавишу "←" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 отнять от текущего значения положения курсора значение 1
 перейти на метку EXIT_USER
 КОНЕЦ
ЕСЛИ пользователь нажал клавишу "→" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 прибавить к текущему значению положения курсора значение 1
 перейти на метку EXIT_USER
 КОНЕЦ
ЕСЛИ пользователь нажал клавишу "↓" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 прибавить к текущему значению положения курсора значение 63
 перейти на метку EXIT_USER
 КОНЕЦ
ЕСЛИ пользователь нажал клавишу "Enter" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 перемещаем курсор на начало следующей строки в рабочем окне
 переход на метку ES_BS
 КОНЕЦ
 ИНАЧЕ переход на метку NET_ENTER
МЕТКА **NET_ENTER**
ЕСЛИ пользователь нажал клавишу "BKSP" ТО
 ЕСЛИ нет выхода из диапазона ТО
 НАЧАЛО
 удаляем предыдущий символ
 переход на метку ES_BS
 КОНЕЦ
 ИНАЧЕ переход на метку NET_BS
МЕТКА **NET_BS**
 ЕСЛИ нет выхода из диапазона ТО
 записать по соответствующему адресу текущий код символа
 ИНАЧЕ переход на метку ES_BS
МЕТКА **ES_BS**
 вызвать процедуру TEXT
МЕТКА **EXIT_USER**
 вызвать процедуру PROCEDURE_CURSOR
КОНЕЦ

Алгоритм процедуры модуля M30.ASM (PROCEDURE_CURSOR)

По значению X1, Y1 и смещению указателя (место ввода текущего символа) относительно начала рабочего окна, имеющего фокус ввода, вычисляет положение курсора и устанавливает курсор в соответствующую точку.

4. Отладка программы.

Адекватные действия пользователя.

Действия пользователя	Реакция программы на эти действия
Пользователь открывает файл размером 0 байт.	Программа открывает пустое рабочее окно.
Пользователь открывает файл размером 1 байт.	Программа отображает содержимое файла в рабочем окне.
Пользователь нажал мышью на SAVE.	Открылось меню SAVE для ввода текста.
Пользователь нажал мышью на EXIT.	Программа завершает свою работу.
Пользователь нажал мышью на DELETE.	Открылось меню DELETE для ввода текста.
Пользователь нажал мышью на OPEN.	Открылось меню OPEN для ввода текста.
Пользователь сохраняет содержимое рабочего окна, в котором 6 символов.	В текущей директории (если пользователь не указал полный путь) появляется соответствующий файл размером 6 байт.
Пользователь открывает любое меню.	Курсор исчезает с экрана.
Пользователь закрывает рабочее окно, хотя открыты еще пару окон.	Курсор перемещается на другое рабочее окно, которое после закрытого имеет фокус ввода.
Пользователь закрывает рабочее окно, при этом нет больше открытых окон.	Курсор исчезает с экрана.

Неадекватные действия пользователя.

Действия пользователя	Реакция программы на эти действия
Нажатие мышью на кнопку SAVE, когда нет ни одного открытого окна.	Программа не реагирует на эту попытку.
Пользователь может попытаться сохранить пустое окно в файл.	Программа не сохраняет данные, размером <1 байта в файл.
Попытка записать содержимое рабочего окна в файл, который уже существует.	Сообщение об ошибке не выдается, но программа продолжает работать корректно.
Удаление несуществующих файлов на диске, либо попытка не ввести ничего и нажать ОК.	Программа выдает сообщение об ошибке и продолжает работать в нормальном режиме.
Открытие файлов с помощью кнопки OPEN размером, больше 693 байта.	Программа считывает первые 693 байта файла и закрывает его.
Попытка открыть несуществующий файл.	Программа открывает пустое окно и продолжает работать в нормальном режиме.
Попытка открыть больше 15 файлов.	Программа открывает больше 15 файлов, так как после каждого чтения она закрывает его – поэтому в момент работы программы число открытых файлов может быть максимум 1.
Попытка открыть более 8 окон.	Нет возможности открыть более 8 окон, поэтому программа не реагирует на кнопку OPEN.
Попытка переместить рабочее окно за границу рамы.	Программа не дает пользователю переместить окно за раму.
Попытка пользователя выйти за предел окна или меню курсором при нажатии клавиш "→", "←", "↑", "↓", "Bksp", "Enter", а также при вводе текста.	Программа проверяет границы перемещения курсора.
Нажатие клавиш расширенной клавиатуры, а также комбинации "CTRL-M" и "CTRL-[".	Программа не реагирует на эти клавиши и комбинации.
Попытка ввода с клавиатуры, когда ни одно меню и ни одно рабочее окно не открыто.	Программа не реагирует на нажатые клавиши.
Когда открыто любое меню (кроме FILE), пользователь может попытаться нажать кнопкой мыши на любую область экрана, кроме кнопок ESC или ОК данного меню.	Программа блокирует действия пользователя и не реагирует на эти попытки.

5. Текст программы с комментариями (в виде приложения).

Главный модуль программы (MN.ASM).

```
.MODEL SMALL
;-----ОКНО-----
OKNO STRUC
X1 DW 0 ;[DI]
Y1 DW 0 ;[DI+2]
X2 DW 0 ;[DI+4]
Y2 DW 0 ;[DI+6]
COLOR DW 30 ;[DI+8]
NUMBER DW 0 ;[DI+10]
CURX DW 1 ;[DI+12] - ПОЛОЖЕНИЕ КУРСОРА X В САМОМ НАЧАЛЕ
CURY DW 1 ;[DI+14] - ПОЛОЖЕНИЕ КУРСОРА Y В САМОМ НАЧАЛЕ
BUF DW 0 ;[DI+16] - СМЕЩЕНИЕ В БУФЕРЕ
POLOGENEI_KURSORA DW 0 ;[DI+18] - ПОЛОЖЕНИЕ КУРСОРА В ОКНЕ
OPEN_OR_CLOSE_WINDOW DW 0 ;[DI+20]
OKNO ENDS
;-----
.DATA
;-----
PUBLIC MAS1
MAS1 OKNO 8 DUP (<>) ;МАССИВ 8 РАБОЧИХ ОКОН - ЗАПИСЕЙ ТИПА ОКНО
;-----
PUBLIC MAS
MAS DW 8 DUP (0) ;МАССИВ ПРИОРИТЕТОВ ВСЕХ 8 ОКОН
;-----БУФЕРЫ ДЛЯ ВВОДА ТЕКСТА-----
PUBLIC BUFER
BUFER DB 5544 DUP (20H) ;ПО 693 БАЙТ НА КАЖДОЕ ОКНО
;-----ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С ОКНАМИ-----
PUBLIC CD ;СЧЕТЧИК ОКОН
CD DW 0
PUBLIC ANTICD ;УВЕЛИЧИВАЕТСЯ ПРИ ОТКРЫТИИ, УМЕНЬШАЕТСЯ ПРИ ЗАКРЫТИИ
ANTICD DW 0
PUBLIC CD1
CD1 DW 0
PUBLIC CD2
CD2 DW 0
;-----
PUBLIC XYMAS
XYMAS DW 0 ;ИНДЕКС УЖЕ В САМОМ MAS-СМЕЩЕНИЕ НАЧАЛА MAS
;-----
PUBLIC R
R DW 0 ;ПРИ СОЗДАНИИ НОВОГО ОКНА ЯВЛЯЕТСЯ СЧЕТЧИКОМ
PUBLIC CX_1
CX_1 DW 0 ;КООРДИНАТЫ МЫШИ ПО X
PUBLIC DX_1
DX_1 DW 0 ;КООРДИНАТЫ МЫШИ ПО Y
;-----
VIDEO DW 0 ;ВИДЕОРЕЖИМ ДО ВЫПОЛНЕНИЯ ПРОГРАММЫ
;-----ПЕРЕМЕННЫЕ КООРДИНАТЫ-----
PUBLIC PAR1
PAR1 DW 0
PUBLIC PAR2
PAR2 DW 0
PUBLIC PAR3
PAR3 DW 0
PUBLIC PAR4
PAR4 DW 0
PUBLIC PAR5
PAR5 DB 0
PUBLIC PAR6
PAR6 DW 0
;-----
PUBLIC RAM1
RAM1 DW 0
PUBLIC RAM2
RAM2 DW 0
PUBLIC RAM3
RAM3 DW 0
PUBLIC RAM4
RAM4 DW 0
;-----ПЕРЕМЕННАЯ ДЛЯ РАБОТЫ МЕЖДУ МОДУЛЯМИ-----
PUBLIC I
I DW 0
;-----ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С ОКНАМИ-----
P1X DW 0
P2X DW 0
P1Y DW 0
P2Y DW 0
;-----ФЛАГ - ЖМЕТ ЛИ ЮЗЕР КНОПКУ КЛАВИШИ МЫШИ ПРИ ПЕРЕМЕЩЕНИИ ОКНА-----
FLAG DW 0
;-----НЕКОТОРАЯ ПЕРЕМЕННАЯ БУФЕР - ДЛЯ РАБОТЫ С МОДУЛЯМИ-----
```

```

PUBLIC BUFQ
BUFQ DW 0
;-----В МОМЕНТ ОТКРЫТИЯ МЕНЮ ПЕРВОЕ ПОЛЕ ДОЛЖНО БЫТЬ ВЫДЕЛЕНО-----
OPEN@ DB 'OPEN.....'
;-----ASCII-КОД ВВОДИМОГО ПОЛЬЗОВАТЕЛЕМ СИМВОЛА-----
PUBLIC TEC_AL
TEC_AL DB 0
PUBLIC TEC_AH
TEC_AH DB 0
;-----ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С МЕНЮ FILE-----
FLAG_FILE DB 0
PUBLIC RW
RW DB 120
PUBLIC PW
PW DB 127
PUBLIC SW
SW DW 18
;-----ФЛАГ ВЫХОДА-----
FLAG_EXIT DB 0
;-----ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С МЕНЮ-----
PUBLIC POLE_K
POLE_K DW 0
PUBLIC POLE_AH
POLE_AH DB 0
PUBLIC POLE_DI
POLE_DI DW 0
;-----ФЛАГИ ОПРЕДЕЛЯЮЩИЕ, ОТКРЫТО ИЛИ ЗАКРЫТО ДАННОЕ МЕНЮ OPEN-----
PUBLIC FLAG_OPEN
FLAG_OPEN DB 0
;-----ФЛАГИ ОПРЕДЕЛЯЮЩИЕ, ОТКРЫТО ИЛИ ЗАКРЫТО ДАННОЕ МЕНЮ SAVE-----
PUBLIC FLAG_SAVE
FLAG_SAVE DB 0
;-----ФЛАГИ ОПРЕДЕЛЯЮЩИЕ, ОТКРЫТО ИЛИ ЗАКРЫТО ДАННОЕ МЕНЮ DELETE-----
PUBLIC FLAG_DELETE
FLAG_DELETE DB 0
;-----ФЛАГ ДЛЯ РАБОТЫ С МОДУЛЯМИ-----
PUBLIC FLEG
FLEG DW 0
;-----ФЛАГ ДЛЯ РАБОТЫ С МОДУЛЯМИ-----
PUBLIC FLAGG
FLAGG DW 0
;-----МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СИМВОЛОВ ВВОДИМЫХ В РАБОЧЕЕ ОКНО-----
COL_SYMBOLOV_STROKA_MAX EQU 63
;
;-----НЕОБХОДИМЫЕ ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С DELETE МЕНЮ-----
;
PUBLIC FLAG_DELETE_OK
FLAG_DELETE_OK DB 0
PUBLIC FLAG_DELETE_ESC
FLAG_DELETE_ESC DB 0
;
PUBLIC RI_TEC
RI_TEC DW 0
PUBLIC BUFER_DELETE
BUFER_DELETE DB 41 DUP (0)
;
;-----НЕОБХОДИМЫЕ ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С OPEN МЕНЮ-----
;
PUBLIC FLAG_OPEN_OK
FLAG_OPEN_OK DB 0
PUBLIC FLAG_OPEN_ESC
FLAG_OPEN_ESC DB 0
;
PUBLIC OI_TEC
OI_TEC DW 0
PUBLIC BUFER_OPEN
BUFER_OPEN DB 43 DUP (0)
;
;-----НЕОБХОДИМЫЕ ПЕРЕМЕННЫЕ ДЛЯ РАБОТЫ С SAVE МЕНЮ-----
;
PUBLIC FLAG_SAVE_OK
FLAG_SAVE_OK DB 0
PUBLIC FLAG_SAVE_ESC
FLAG_SAVE_ESC DB 0
;
PUBLIC SI_TEC
SI_TEC DW 0
PUBLIC BUFER_SAVE
BUFER_SAVE DB 43 DUP (0)
PUBLIC BUFER_SYMBOL_SAVE
BUFER_SYMBOL_SAVE DB 693 DUP (20H)
;
;---ЧИСЛО БАЙТ - МАКСИМАЛЬНЫЙ РАЗМЕР ФАЙЛА (ОТОБРАЖЕНИЕ ФАЙЛА В Р. ОКНЕ)---
;

```

```

CHISLO_BYTE EQU EQU 693
PUBLIC CHISLO_BYTE
CHISLO_BYTE DW 0 ;НА ОКНО МАКСИМАЛЬНОЕ ЧИСЛО БАЙТ - 639
;-----ПЕРЕМЕННАЯ ДЛЯ РАБОТЫ С РАБОЧИМ ОКНОМ-----
PUBLIC KAX
KAX DW 0
PUBLIC PRED_AL
PRED_AL DB 0
;-----ТЕКУЩЕЕ СОСТОЯНИЕ МЫШИ-----
BX_TEC DW 0
.386
.STACK 100H
.CODE
;-----ПРОЦЕДУРЫ МОДУЛЕЙ-----
EXTRN FILE_POLE:NEAR
EXTRN POLE:NEAR
EXTRN NASTROIKA_MOUSE:NEAR
EXTRN RAMA:NEAR
EXTRN WINDOW:NEAR
EXTRN WINDOW_ZAGOLOBOK:NEAR
EXTRN FILE:NEAR
EXTRN WHO_PROC:NEAR
EXTRN FILE_PROC:NEAR
EXTRN WINDOW_STIR:NEAR
EXTRN PROCEDURE_MAS:NEAR
EXTRN CLICK_MOUSE_FLAG_TRUE:NEAR
EXTRN ACTIV_POLE:NEAR
EXTRN STIR_FILE:NEAR
EXTRN TUTOR:NEAR
EXTRN USER:NEAR
EXTRN PROBERKA_ZAKRITIE:NEAR
EXTRN OTOBRAZENIE_ACTIV_WINDOW:NEAR
EXTRN USER_ACTIV_WINDOW:NEAR
EXTRN NEW_WINDOW:NEAR
EXTRN RISOBANIE_NEW_WINDOW:NEAR
EXTRN TEXT:NEAR
EXTRN BBOD_WINDOW_SAVE:NEAR
EXTRN BBOD_WINDOW_DELETE:NEAR
EXTRN DELETE_WINDOW:NEAR
EXTRN SAVE_WINDOW:NEAR
EXTRN OPEN_WINDOW:NEAR
EXTRN PROBERKA_BCE_OKNA_CLOSE:NEAR
EXTRN BBOD_WINDOW_OPEN:NEAR
EXTRN BBOD_WINDOW_USER:NEAR
EXTRN PROCEDURE_CURSOR:NEAR
;-----
;
; ТЕЛО ПРОГРАММЫ
;-----
MAIN:
    MOV AX,@DATA
    MOV DS,AX
    MOV AH,4FH
    MOV AL,03
    INT 10h
    MOV VIDEO,BX
    MOV AH,00h
    MOV AL,02h
    INT 10h
    MOV AX,0B800h
    MOV ES,AX
;-----
;-----УБРАТЬ КУРСОР-----
;-----
CALL DELETE_KURSOR
;-----
;-----НАСТРОЙКА МЫШИ-----
;-----
CALL NASTROIKA_MOUSE
MOV AX,@DATA
MOV DS,AX
XOR SI,SI
;-----
;-----РИСОВАНИЕ ЗАГОЛОВКА НА ГЛАВНОМ ОКНЕ-----
;-----
MOV PAR1,1
MOV PAR2,1
MOV PAR3,1
MOV PAR4,80
MOV PAR5,120
CALL WINDOW_ZAGOLOBOK
CALL WHO_PROC
CALL FILE_PROC
;-----
;-----РИСОВАНИЕ ОСНОВНОЙ РАМЫ-----

```

```

;-----
MOV RAM1,160
MOV RAM2,78
MOV RAM3,22
CALL RAMA
;-----
;-----ЗАТИРАНИЕ ВСЕГО ОКНА-----
;-----
MOV PAR1,3
MOV PAR2,24
MOV PAR3,2
MOV PAR4,79
MOV PAR5,0
CALL WINDOW_ZAGOLOBOK
;-----ЧИСЛО БАЙТ, СЧИТЫВАЮЩИХСЯ ИЗ ФАЙЛА-----
MOV AX,CHISLO_BYTE_EQU
MOV CHISLO_BYTE,AX
;-----
;-----ОБРАБОТЧИК ПРЕРЫВАНИЯ ОТ МЫШИ-----
;-----
BESKONECHN_BBOD:
;-----ПОКАЖЕМ КУРСОР, КОГДА ЭТО НАДО-----
CMP ANTICD,1
JL NR
;-----
CMP FLAG_SAVE,1
JE NR
;-----
CMP FLAG_OPEN,1
JE NR
;-----
CMP FLAG_DELETE,1
JE NR
;-----
CMP FLAG_FILE,1
JE NR
;-----ДА, ПОКАЗЫВАЕМ КУРСОР-----
CALL PROCEDURE_CURSOR
NR:
;-----ПРОВЕРЯЕМ, НАЖАТА ЛИ КЛАВИША-----
MOV AX,0B00H
INT 21h
CMP AL,0
JE NET_KLABA
;-----ЕСЛИ ВСЕ-ТАКИ НАЖАЛИ ЛЮБУЮ КЛАВИШУ-----
MOV AX,@DATA
MOV DS,AX
MOV AX,0B800h
MOV ES,AX
;-----СЧИТАТЬ СИМВОЛ БЕЗ ЭХА, С ОЖИДАНИЕМ-----
MOV AX,0
INT 16h
MOV TEC_AL,AL
MOV TEC_AH,AH
;-----CTRL-M И CTRL-[-----
CMP AX,320DH
JE BESKONECHN_BBOD
CMP AX,1A1BH
JE BESKONECHN_BBOD
;-----СТРЕЛКИ-----
CMP AH,72
JE ENTER_
CMP AH,75
JE ENTER_
CMP AH,77
JE ENTER_
CMP AH,80
JE ENTER_
;-----
CMP AL,20H
JE ENTER_ ;ДОЛЖНЫ ПРОПУСКАТЬ
CMP AL,13
JE ENTER_ ;ДОЛЖНЫ ПРОПУСКАТЬ
CMP AL,27
JE ENTER_ ;ДОЛЖНЫ ПРОПУСКАТЬ
CMP AL,8
JE ENTER_ ;ДОЛЖНЫ ПРОПУСКАТЬ
;-----ФИЛЬТРУЕМ РАСШИРЕННУЮ КЛАВИАТУРУ-----
CMP AL,32
JLE BESKONECHN_BBOD
CMP AH,3BH
JGE BESKONECHN_BBOD
;-----
;-----ЕСЛИ ОТКРЫТО МЕНЮ, ТО ВВОДИТЬ НЕЛЬЗЯ-----

```

```

;-----
ENTER_:
CMP FLAG_FILE,1
JE NET_KLABA
;-----
;-----НЕТ, МЕНЮ FILE ЗАКРЫТО-----
;-----
CMP FLAG_SAVE,1
JNE NET_FLAG_SAVE
;-----
;-----ДА, ВВОДИМ В ОКНЕ SAVE-----
;-----
CALL BBOD_WINDOW_SAVE
JMP NET_KLABA
NET_FLAG_SAVE:
;-----
;-----НЕТ, МЕНЮ FILE ЗАКРЫТО-----
;-----
CMP FLAG_DELETE,1
JNE NET_FLAG_DELETE
;-----
;-----ДА, ВВОДИМ В ОКНЕ DELETE-----
;-----
CALL BBOD_WINDOW_DELETE
JMP NET_KLABA
NET_FLAG_DELETE:
;-----
;-----НЕТ, МЕНЮ FILE ЗАКРЫТО-----
;-----
CMP FLAG_OPEN,1
JNE NET_FLAG_OPEN
;-----
;-----ДА, ВВОДИМ В ОКНЕ OPEN-----
;-----
CALL BBOD_WINDOW_OPEN
JMP NET_KLABA
NET_FLAG_OPEN:
;-----
;-----ВВОД В РАБОЧЕЕ ОКНО - ВСЕ МЕНЮ ЗАКРЫТЫ-----
;-----
CALL PROCEDURE_MAS
XOR AX,AX
CMP [DI+20],AX
JE NET_WINDOW_STOP
CALL BBOD_WINDOW_USER
NET_WINDOW_STOP:
;-----
;-----ВЫЗОВ ОБРАБОТЧИКА ПРЕРЫВАНИЯ-----
;-----
NET_KLABA:
MOV AX,@CODE
MOV ES,AX
MOV AX,0CH
MOV CX,11b
LEA DX,HANDLER
INT 33h
;-----ЕСЛИ НУЖНО ВЫЙТИ, ТО ВЫХОДИМ-----
CMP FLAG_EXIT,1
JE EXIT
NET_KLABA@:
JMP BESKONECHN_BBOD
;-----
EXIT:
MOV AX,0ch
MOV CX,0h
INT 33h
MOV BX,VIDEO
XOR BH,BH
MOV AH,00h
MOV AX,BX
INT 10h
MOV AH,4CH
INT 21h
;-----
;-----ОБРАБОТЧИК СОБЫТИЯ ОТ МЫШИ-----
;-----
HANDLER:
MOV AX,0B800h
MOV ES,AX
MOV AX,@DATA
MOV DS,AX
XOR SI,SI
;-----СПРЯТАЛИ МЫШЬ-----
MOV AX,2

```

```

INT 33h
;-----ОПРЕДЕЛИЛИ СОСТОЯНИЕ МЫШИ-----
MOV AX,3
INT 33h
;-----ПРЕОБРАЗОВАЛИ КООРДИНАТЫ-----
SHR CX,3
SHR DX,3
INC CX
INC DX
MOV BX_TEC,BX
;-----
;-----ПРОВЕРКА ФЛАГОВ ПРИ ВВОДЕ-----
;-----
CMP FLAG_SAVE,1
JE TRUE_SAVE
;-----
CMP FLAG_OPEN,1
JE TRUE_OPEN
;-----
CMP FLAG_DELETE,1
JE TRUE_DELETE
;-----ЕСЛИ НЕ ПОДХОДЯТ 3 УСЛОВИЯ ВЫШЕ-----
JMP ALL_WINDOW_MENU_BBOD_CLOSE
;-----
;-----ЕСЛИ МЕНЮ ВВОДА SAVE ОТКРЫТО-----
;-----
TRUE_SAVE:
    CMP BX_TEC,1
    JNE DRAIVER@
;-----
;-----ЕСЛИ НАЖАТА МЫШЬ И ОТКРЫТО ОКНО SAVE, ТО-----
;-----
CMP CX,1DH
JL DRAIVER
CMP CX,1FH
JGE DRAIVER
CMP DX,0BH
JL DRAIVER
CMP DX,0BH
JG DRAIVER
;-----ЕСЛИ НАЖАЛИ ОК-----
MOV FLAG_SAVE_OK,1
CALL BBOD_WINDOW_SAVE
JMP NEAR:NET_MOUSE
;-----
DRAIVER:
CMP CX,33H
JL DRAIVER@
CMP CX,39H
JGE DRAIVER@
CMP DX,0BH
JL DRAIVER@
CMP DX,0BH
JG DRAIVER@
;-----ДА, НАЖАТА КЛАВИША МЫШИ НА ESC-----
MOV FLAG_SAVE_ESC,1
CALL BBOD_WINDOW_SAVE
JMP NEAR:NET_MOUSE
;-----
DRAIVER@:
JMP NEAR:EXIT_PROC ;ЕСЛИ ОТКРЫТО МЕНЮ SAVE
;-----
;-----ЕСЛИ МЕНЮ ВВОДА DELETE ОТКРЫТО-----
;-----
TRUE_DELETE:
    CMP BX_TEC,1
    JNE DRAIVED@
;-----
;-----ЕСЛИ НАЖАТА МЫШЬ И ОТКРЫТО ОКНО DELETE, ТО-----
;-----
CMP CX,1DH
JL DRAIVED
CMP CX,1FH
JGE DRAIVED
CMP DX,0BH
JL DRAIVED
CMP DX,0BH
JG DRAIVED
;-----ЕСЛИ НАЖАЛИ ОК-----
MOV FLAG_DELETE_OK,1
CALL BBOD_WINDOW_DELETE
JMP NEAR:EXIT_PROC
;-----
DRAIVED:

```



```

CMP CX,33H
JL DRAIVED@
CMP CX,39H
JGE DRAIVED@
CMP DX,0BH
JL DRAIVED@
CMP DX,0BH
JG DRAIVED@
;-----ДА, НАЖАТА КЛАВИША МЫШИ НА ESC-----
MOV FLAG_DELETE_ESC,1
CALL BBOD_WINDOW_DELETE
JMP NEAR:EXIT_PROC
;-----
DRAIVED@:
JMP NEAR:EXIT_PROC ;ЕСЛИ ОТКРЫТО МЕНЮ DELETE
;-----
;-----ЕСЛИ МЕНЮ ВВОДА OPEN ОТКРЫТО-----
;-----
TRUE_OPEN:
CMP BX_TEC,1
JNE DRAIVEO@
;-----
;-----ЕСЛИ НАЖАТА МЫШЬ И ОТКРЫТО ОКНО OPEN, ТО-----
;-----
CMP CX,1DH
JL DRAIVEO
CMP CX,1FH
JGE DRAIVEO
CMP DX,0BH
JL DRAIVEO
CMP DX,0BH
JG DRAIVEO
;-----ЕСЛИ НАЖАЛИ ОК-----
MOV FLAG_OPEN_OK,1
CALL BBOD_WINDOW_OPEN
JMP NEAR:EXIT_PROC
;-----
DRAIVEO:
CMP CX,33H
JL DRAIVEO@
CMP CX,39H
JGE DRAIVEO@
CMP DX,0BH
JL DRAIVEO@
CMP DX,0BH
JG DRAIVEO@
;-----ДА, НАЖАТА КЛАВИША МЫШИ НА ESC-----
MOV FLAG_OPEN_ESC,1
CALL BBOD_WINDOW_OPEN
JMP NEAR:EXIT_PROC
;-----
DRAIVEO@:
JMP NEAR:EXIT_PROC ;ЕСЛИ ОТКРЫТО МЕНЮ OPEN
;-----
;-----ПРОВЕРКА НА НАЖАТИЕ КНОПКИ МЫШИ-----
ALL_WINDOW_MENU_BBOD_CLOSE:
CMP BX_TEC,1
JNE NET_MOUSE
;-----
;-----ЕСЛИ НАЖАЛИ FILE-----
;-----
CMP CX,2
JL NET_FILE
CMP CX,5
JG NET_FILE
CMP DX,1
JG NET_FILE
;-----ОТКРЫЛИ МЕНЮ ФАЙЛ-----
CALL DELETE_KURSОР
PUSH CX
PUSH DX
CALL FILE
CALL FILE_POLE
MOV AH,PW
MOV POLE_AH,AH
MOV POLE_DI,324
LEA SI,OPEN@
CALL POLE
POP DX
POP CX
MOV FLAG_FILE,1
JMP NET_MOUSE
;-----
NET_FILE:

```

```

CMP FLAG_FILE,1
JNE NET_MOUSE
;-----ДА, МЕНЮ ОТКРЫТО-----
CMP CX,2
JL NET_FLAG_FILE@
CMP CX,21
JG NET_FLAG_FILE@
CMP DX,2
JL NET_FLAG_FILE@
CMP DX,8
JG NET_FLAG_FILE@
;-----
;-----ЕСЛИ ПОЛЬЗОВАТЕЛЬ НАЖАЛ В ОБЛАСТИ МЕНЮ-----
;-----ОТРАБАТЫВАЕМ ПОДМЕНЮ-----
;-----
CMP CX,3
JL NET_OPEN
CMP CX,20
JG NET_OPEN
CMP DX,3
JL NET_OPEN
CMP DX,3
JG NET_OPEN
;-----
;-----ДА, КЮЗЕР НАЖАЛ OPEN-----
;-----
CMP ANTICD,8;CD-ВСЕГДА ВОЗРАСТАЮЩИЙ СЧЕТЧИК,А ANTICD-МОЖЕТ И УВ. И УМ.
JGE NET_NEW_WINDOW
MOV FLAG_FILE,0 ;СБРАСЫВАЕМ ФЛАГ МЕНЮ
CALL DELETE_KURSOR
CALL OPEN_WINDOW
NET_NEW_WINDOW:
JMP EXIT_PROC
;-----
NET_OPEN:
CMP CX,3
JL NET_DELETE
CMP CX,20
JG NET_DELETE
CMP DX,4
JL NET_DELETE
CMP DX,4
JG NET_DELETE
;-----
;-----ДА, КЮЗЕР НАЖАЛ DELETE-----
;-----
CALL DELETE_KURSOR
MOV FLAG_FILE,0
CALL DELETE_WINDOW
JMP EXIT_PROC
;-----
NET_DELETE:
CMP CX,3
JL NET_SAVE
CMP CX,20
JG NET_SAVE
CMP DX,5
JL NET_SAVE
CMP DX,5
JG NET_SAVE
;-----
;-----ДА, КЮЗЕР НАЖАЛ SAVE-----
;-----
CMP ANTICD,1
JL NET_CALL_SAVE_WINDOW
;-----ЕСЛИ ЕСТЬ ХОТЯ БЫ ОДНО ОТКРЫТОЕ ОКНО-----
CALL DELETE_KURSOR
MOV FLAG_FILE,0
CALL SAVE_WINDOW
NET_CALL_SAVE_WINDOW:JMP EXIT_PROC
;-----
;-----ПРОВЕРКА НАЖАТИЯ EXIT-----
;-----
NET_SAVE:
CMP CX,3
JL NET_EXIT
CMP CX,20
JG NET_EXIT
CMP DX,6
JL NET_EXIT
CMP DX,6
JG NET_EXIT
;-----

```

```

;-----ДА, ЮЗЕР НАЖАЛ EXIT-----
;
MOV FLAG_EXIT,1
JMP EXIT_PROC
;
;
NET_EXIT:
JMP NET_MOUSE
;
;-----ЮЗЕР НАЖАЛ ВНЕ ОБЛАСТИ МЕНЮ FILE-----
;
NET_FLAG_FILE@:
CALL STIR_FILE ;СТИРАЕМ ПОДМЕНЮ FILE
MOV FLAG_FILE,0 ;УСТАНАВЛИВАЕМ ФЛАГ-МЕНЮ FILE ЗАКРЫТО
;-----ПРОРИСОВКА ОКОН-----
CMP CD,1
JL NET_PRORISOVKA_MENU_FILE
CALL USER
NET_PRORISOVKA_MENU_FILE:
;
;-----ЕСЛИ МЕНЮ ЗАКРЫТО-----
;-----НЕ НАЖИМАЛИ FILE-----
;
NET_MOUSE:
;-----ЕСЛИ ОТКРЫТО МЕНЮ-----
CMP FLAG_FILE,1
JNE NET_ONE
;-----ДА, МЕНЮ ОТКРЫТО-ВЫДЕЛЯЕМ АКТИВНОЕ ПОЛЕ-----
CALL ACTIV_POLE
JMP EXIT_PROC
;-----РАБОТА С ОКНАМИ-----
;
NET_ONE:
CMP CD,1
JL EXIT_PROC
;
;-----РАБОТА С АКТИВНЫМ ОКНОМ ПРИ ПЕРЕМЕЩЕНИИ ОКНА-----
;-----НАЖАТА ЛИ ЛЕВАЯ КЛАВИША-----
;
CMP BX_TEC,1
JNE NET_FLAG
;-----ЕСЛИ НЕТ НИ ОДНОГО ОКНА НА ЭКРАНЕ-----
CALL PROBERKA_VSE_OKNA_CLOSE
CMP FLAGG,1
JE EXIT_PROC
;-----ЕСЛИ НАЖАЛИ ЛЕВУЮ КЛАВИШУ-----
MOV AX,FLAG
CMP AX,1
;-----ЕСЛИ ФЛАГ УСТАНОВЛЕН-----
JNE FLAG_USTANOVKA ;ЕСЛИ НЕТ,ТО ОТПРАВЛЯЕМСЯ УСТАНАВЛИВАТЬ ФЛАГ
;
;-----ЕСЛИ НАЖАТА КНОПКА И УСТАНОВЛЕН ФЛАГ-----
;
CALL CLICK_MOUSE_FLAG_TRUE
CALL USER ;ПРОРИСОВКА ВСЕХ ОКОН
JMP NET_NET_FLAG
;
;-----ЕСЛИ НАЖАТА МЫШЬ, НО ФЛАГ НЕ УСТАНОВЛЕН-----
;
FLAG_USTANOVKA:
;-----ОПРЕДЕЛИЛИ АКТИВНОЕ ОКНО-----
CALL PROCEDURE_MAS
;-----ЕСЛИ ВЗЯЛИ ЗА ЗАГОЛОВОК ИМЕННО ТЕКУЩЕГО ОКНА-----
CMP CX,[DI]
JL NET_NET_FLAG
CMP CX,[DI+4]
JG NET_NET_FLAG
CMP DX,[DI+2]
JL NET_NET_FLAG
CMP DX,[DI+2]
JG NET_NET_FLAG
;
MOV AX,1
MOV FLAG,AX
MOV CX_1,CX
MOV DX_1,DX
JMP NET_NET_FLAG
;-----ЕСЛИ НЕ НАЖАТА КНОПКА МЫШИ-----
NET_FLAG:XOR AX,AX
MOV FLAG,0
NET_NET_FLAG:
;
;-----ПРОВЕРЯЕМ НАЖАТИЕ КЛАВИШИ-----
;

```

```

CMP BX_TEC,1
JNE EXIT_PROC
;-----
;-----НОВОГО ОКНА НЕ ОТКРЫВАЛИ-----
;-----ВЫДЕЛЯЕМ АКТИВНОЕ ОКНО-----
;-----
CALL USER_ACTIV_WINDOW
;-----ПРОВЕРКА, ЕСЛИ ОКНО ЗАКРЫТО-----
CALL PROCEDURE_MAS
MOV AX,2
CMP [DI+20],AX
JE NET_WINDOW_STOP@
CALL TEXT
NET_WINDOW_STOP@:
;-----
EXIT_PROC:
        MOV AX,1
        INT 33h
RETF
;-----КОНЕЦ ОБРАБОТЧИКА-----
DELETE_KURSОР PROC
        MOV AX,0200H
        XOR BX,BX
        MOV DX,9999H
        INT 10H
        RET
DELETE_KURSОР ENDP
;-----
END MAIN

```