

# Состав пакета

Пакет макроассемблера включает в себя основные программы, необходимые для создания, отладки и сопровождения программ на языке Ассемблера.

В состав пакета макроассемблера входят следующие программные компоненты:

- и **MASM** — макроассемблер;
- и **LINK** — объектный линкер;
- и **SYMDEB** — символьный отладчик программ;
- и **MAPSYM** — генератор символьного файла;
- и **CREF** — утилита перекрестных ссылок;
- и **LIB** — утилита обслуживания библиотек;
- и **MAKE** — утилита сопровождения программ.

Линкер **LINK** обрабатывает выработанную **MASM** объектную программу с целью разрешения ссылок к другим модулям и приведения программы к виду, пригодному для загрузки в память.

Утилита **LIB** обеспечивает формирование и обслуживание библиотек объектных модулей, которые могут быть использованы **LINK** для разрешения внешних ссылок.

Отладчик **SYMDEB** реализует отладку сформированной программы на двух уровнях: на уровне символьческих имен и на уровне абсолютных адресов.

Программа **MAPSYM** предназначена для создания символьного файла для **SYMDEB**. Символьный файл формируется на основе информации, полученной от **MASM**, и необходим для символьной отладки.

Утилита **CREF** может быть использована для формирования листинга перекрестных ссылок программы, наличие которого облегчает отладку.

При помощи утилиты **MAKE** процесс разработки программ может быть автоматизирован. В файле описаний **MAKE** могут быть заданы различные алгоритмы вызовов и взаимодействия программ пакета (и не только их).

# Утилита символьной отладки

SYMDEV совместим со следующими компиляторами:

- ❖ MICROSOFT FORTRAN версии 3.0 и выше
- ❖ MICROSOFT PASCAL версии 3.0 и выше
- ❖ MICROSOFT C версии 2.0 и выше
- ❖ MICROSOFT макроассемблер версии 1.0 и выше
- ❖ MICROSOFT BASIC COMPILER версии 1.0 и выше
- ❖ MICROSOFT BUSINESS BASIC COMPILER версии 1.0 и выше
- ❖ IBM PC FORTRAN версии 2.0 и выше
- ❖ IBM PC PASCAL версии 2.0 и выше
- ❖ IBM PC макроассемблер версии 1.0 и выше
- ❖ IBM PC BASIC COMPILER версии 1.0 и выше

Конечной целью подготовки символьной отладки является создание символьного файла. В общем случае это осуществляется путем обработки исходной программы соответствующим компилятором и программой LINK и формирования на основе полученной информации собственно символьного файла при помощи программы **MAPSYM**.

# Пример работы с отладчиком

Этапы ассемблирования и  
загрузки файла `primer.exe`

# Primer.asm

- text segment
- assume Cs:text,Ds:data
- begin:
  
- mov ax,data;
- mov ds,ax;
- int 21h ;
- text ends
  
- data segment
- hello db "Hello world!\$";
- data ends
  
- stack segment stack
- db 256 dup (0)
- stack ends
- end begin;
- ;

# В примере файлы на одном и том же накопителе

1. Masm primer
2. Link primer /map(Если задана опция /MAP, в файл включаются имена общих символов и их загрузочные адреса)
3. Mapsym primer (создание символьного файла)
4. Symdeb /"r;x8?" primer.sym primer.exe

- В первых строках после сообщения запуска показан статус регистров. Эти строки сформированы первой командой
- (R) , специфицированной опцией команды запуска.
- Вторая команда (x?\*)– воспроизводит все символы, загруженные из соответствующего символьного файла.

- Можно использовать после приглашения
- -R – просмотр содержимого регистров и флагов
- -T – команда трассировки
- И т.п.
- См. <http://nick-uk.narod.ru/doc/system.htm>



- В примерах **Примеры небольших подпрограмм на Ассемблере**
- <http://nick-yk.narod.ru/doc.html>
- Найдите программу **Pixel\$**, доработайте её, составьте отчёт

- org 100h
- start: mov ax,13h; устанавливаем режим 320x240x256 цветов
- int 10h
- push 0A000h; ES=0A000h начало видеопамати в графических режимах
- pop es
- mov di,50\*320+50; начало линий в точке X=50 Y=50
- mov al,0Fh; цвет линий
- mov cx,110
- rep stosb; выводим горизонтальную линию
- mov di,50\*320+50
- mov cx,100
- a1: stosb; рисуем вертикальную линию
- add di,319
- loop a1
- mov di,50\*320+50
- mov cx,75
- a2: stosb; линия под углом 45 градусов
- add di,320
- loop a2
- mov di,50\*320+50
- mov cx,50
- a3: stosb; линия под углом 30 градусов
- stosb
- add di,320
- loop a3
- mov ah,0; даем полюбоваться на дела рук своих пока не нажмут на клавиатуру
- int 16h
- mov ax,3; устанавливаем текстовый режим
- int 10h
- ret; выходим из программы

# Программа

В

Fasm.

Задание:

Нарисовать  
осмысленную  
картинку из  
линий