Засоби для створення програм мовою асемблера

1. Введення тексту програми (Текстові редактори)

Для набору тексту асемблер них програм можна використовувати будь-який текстовий редактор. При виборі редактора слід враховувати, що він не повинен вставляти в текст «сторонніх» символів (символів форматування або інших керуючих символів)[1]. У Windows таким редактором може бути Блокнот (Notepad), хоча це не найкращий вибір. Значно кращим є безкоштовний редактор МегаБлокнот, який може використовувати різні кодування символів (для DOS та Windows), а також відображає номер рядка, що полегшує пошук помилок.

Більш функціональним є також редактор Notepad++, який має кольорове підсвічування синтаксису різних мов програмування, підтримку різних кодувань та багато інших можливостей.

Досить цікавими є редактори PSPad, ASM Editor for Windows, Quick Editor з комплекту Masm32 та багато інших.

У програмах для DOS використовують редактори типу EDIT або вбудовані редактори програм Norton Commander, Far та ін. Створений файл із текстом програми на асемблері зазвичай має розширення «.asm».

2. Трансляція програм (Програми-асемблери)

Для одержання об’єктного модуля програми використовують транслятор, який перетворює текст програми на асемблері в об'єктний модуль. Часто транслятор із мови асемблера називають просто асемблером. Найбільш популярними є асемблери TASM (Turbo Assembler) від Borland та MASM від Microsoft. Заслуговує на увагу безкоштовний Netwide Assembler або NASM, який також дуже поширений, причому він використовується як в операційних системах DOS та Windows, так і в ОС Linux та ін. Крім цього можна використовувати багато інших асемблерів (наприклад, FASM, GoAsm та ін.). У цьому збірнику ми будемо орієнтуватися на використання асемблера TASM або MASM, які мають дуже схожий синтаксис.

Об'єктний модуль – це програма на машинній мові, придатна для виконання процесором. Він містить інформацію, необхідну для компонування з іншими об’єктними модулями і може включати відомості, необхідні для відлагодження. Об'єктний модуль не має ознак початкової мови, тому можливо створювати програму із об’єктних модулів, створених на різних мовах програмування, якщо вони мають однаковий формат. Об'єктний мо-

[1] Текстові процесори Microsoft Word або WordPad для написання асемблерних програм не годяться.

дуль ще не можна використовувати як програму, у ньому не налаштовані адреси завантаження програми в пам'ять та зовнішні посилання на інші модулі.

При використанні пакета TASM об'єктний модуль одержують програмою tasm.exe. Формат командного рядка для запуску TASM.EXE наступний:

```
tasm [ключі] source_file [,object_file] [,list_file] [,refer_file]
```

де:

`source_file` – файл із текстом програми;

`object_file` – об'єктний файл;

`list_file` – файл лістингу;

`refer_file` – файл перехресних посилань.

Наприклад:

```
tasm [ключі] prog1.asm, prog1.obj, prog1.lst, prog1.crf
```

Якщо ім'я файлів, які створюються, співпадають з іменем текстового файлу, то можна використати команду.

```
tasm prog1,,,
```

Непотрібні файли можна замінити параметром `nul`. За умовчанням файл з текстом програми на асемблері має розширення .ASM, тому його можна опустити. Обов'язковим є лише такий фрагмент команди:

```
tasm prog1
```

Перелік ключів програми можна одержати командою `tasm`. Найчастіше використовують такі ключі:

`/l`	створити файл лістингу, якщо він не вказаний у командному рядку;
`/z`	при виникненні помилок виводити разом з повідомленням про них рядки тексту

`/zi`
`/zd`
`/zn`
	включити в об'єктний файл інформацію для відлагодження;
	включити в об'єктний файл інформацію про номери рядків програми;
	заборонити включення в об'єктний файл відлагоджувальної інформації.

Наприклад: `tasm /l prog1` – компілює програму PROG1.ASM і створює файл лістингу.

3. Створення завантажувального модуля (Редактори зв'язків або компонувальники)

Після виправлення помилок і одержання об'єктного модуля приступають до наступного етапу – створення завантажувального модуля (компоновки програми). Головна мета цього етапу — перетворення коду і даних в об'єктних файлах в формат, який може виконуватись як програма. Узгодження зовнішніх посилань і створення програми, яка виконується, забезпечується Редактором зв'язків (Компонувальником). Редактор зв'язків з'єднує усі об'єктні модулі в єдине ціле – програму. Оскільки Редактор зв'язків "бачить" усі компоненти програми, він має можливість обробити ті місця в об'єктних модулях, які містять зовнішні посилання. Результатом роботи Редактора зв'язків є завантажувальний модуль,

який має розширення «.exe» або «.com». Після цього операційна система може завантажити такий файл у пам'ять і виконати його.

Розглянемо формат командного рядка компонувальника TLINK.EXE (параметри в квадратних дужках є необов'язковими):

```
tlink [ключі] obj_files    [, exe_file] [, map_file] [,lib_file]
                           [, def_file] [, res_file]
```

Параметри командного рядка для запуску компонувальника:

`ключі` — необов'язкові параметри, які керують роботою компонувальника. Наприклад:

`/x`	не створювати файл карти (map)
`/m`	створити файл карти
`/s`	включити до файлу карти інформацію про сегменти (адреса, довжина в байтах, клас, ім'я сегменту і т. д.)
`/l`	створити у файлі карти розділ з номерами рядків
`/c`	розрізняти малі та великі букви в ідентифікаторах (у тому числі зовнішніх)
`/v`	включити відлагоджувальну інформацію у виконуваний файл
`/3`	підтримка 32-бітного коду у 16-бітних програмах
`/t`	створити файл типу .COM (за умовчанням .EXE)

Кожному ключу повинен передувати символ "-" (дефіс) або "/" (слеш). Регістр символів має значення.

`obj_files` – обов'язковий параметр, містить список компонованих файлів з розширенням .OBJ. Файли повинні бути розділені пропусками або знаком "+" (плюс), наприклад:

```
tlink /v prog + mdf + fdr
```

При необхідності імена файлів супроводжують вказанням шляху до них.

`exe_file` – необов'язковий параметр з іменем завантажувального модуля. Якщо параметр не вказується, то ім'я співпадає з іменем першого об'єктного модуля і має розширення «.EXE».

`map_file` – при наявності параметра компонувальник створює спеціальний файл з картою завантаження. В ній перераховуються імена, адреси завантаження і розміри всіх сегментів, які входять в програму.

`lib_file` – вказує на файл бібліотеки (з розширенням «.LIB»). Цей файл створюється утилітою TLIB.EXE пакета TASM (або LIB.EXE пакета MASM). Утиліта дозволяє об'єднувати підпрограми у вигляді об'єктних модулів, які часто використовуються, в один файл.

`def_file` – вказує на файл визначень (.DEF). Цей файл використовується для зберігання значної кількості ключів і параметрів.

`res_file` – вказує на файл ресурсів Windows-додатка (.RES).

Розглянутий формат командного рядка використовується також для 32-розрядного варіанту компонувальника TLINK32.EXE. При наявності багатьох параметрів командного рядка їх можна оформити у вигляді файлу конфігурації TLINK.CFG (TLINK32.CFG) і викликати компонувальник командою `tlink tlink.cfg`.

Список ключів програми TLINK.EXE можна одержати командою TLINK без параметрів.

У більшості випадків достатньо команди `tlink prog` або `tlink /v prog`

В результаті двох етапів одержуємо виконуваний модуль типу PROG.EXE. На жаль, відсутність синтаксичних помилок не означає, що програма буде працювати правильно. Тому обов'язковим етапом розробки програми є її відлагодження.

4. Запуск і відлагодження програми

На етапі відлагодження перевіряється правильність функціонування програми у відповідності із заданим алгоритмом. Завершальним етапом є тестування програми на правильність роботи при різноманітних (у тому числі граничних і некоректних) вхідних даних. Для цього складаються спеціальні тести. За результатами тестування вносяться виправлення в текст програми на асемблері і описані етапи повторюються.

Для відлагодження DOS-програм на асемблері можна використовувати 16-розрядний відлагоджувальник Turbo Debugger (TD), розроблений фірмою Borland. Це найбільш вдала програма для асемблерних програм реального режиму. Можна використовувати і інші відлагоджувальники, наприклад CodeView (CV.EXE) із пакету MASM. Одним із найкращих відлагоджувальників вважається програма SoftICE. Значної популярності набув також відлагоджувальник OllyDbg.

Відлагоджувальник TD являє собою віконне середовище відлагодження програм на рівні асемблерного тексту. Він дозволяє вирішувати дві головні задачі:

• визначити місце логічної помилки;

• визначити причину логічної помилки.

Можливості програми TD:

• трасування програми в прямому напрямі, тобто послідовне виконання програми, при якому за один крок виконується одна машинна інструкція;

• трасування програми в зворотному напрямі, тобто виконання програми по одній команді за один крок, але в зворотному напрямі;

• перегляд і зміна стану апаратних ресурсів процесора під час трасування.

Такі дії дозволяють визначити місце та джерело помилок у програмі. Слід відзначити, що TD не дозволяє вносити виправлення в початковий текст програми. Правда можна внести виправлення прямо в машинний код програми, але на практиці такий підхід не ви-

користовують. Виправлення вносять в текст програми на асемблері і заново створюють завантажувальний модуль.

Для спрощення одержання завантажувального модуля із асемблерного тексту можна використовувати пакетні файли (типу *.BAT). Для полегшення процесу відлагодження програм (можливість працювати з текстом програми) трансляцію слід проводити із ключами, які включають відлагоджувальну інформацію до об'єктного та виконуваного файлів, наприклад:

```
tasm /zi prog
tlink /v prog
```

Це означає, що до об'єктного та завантажувального модулів включається інформація про імена змінних, які використовуються в програмі (символічні імена) і які будуть доступними в процесі відлагодження.

Запуск відлагоджувальника здійснюється командою:

`td exe_file`,

де `exe_file` – виконуваний модуль програми.