Зовнішня модель реалізації асемблера

Приклад програми

Нижче записаний приклад асемблерної програми, яку можна використати для тестування і перевірки свого власного асемблера. Програма складена для трансляції в COM-програму. Звернемо увагу, що програму потрібно складати з директивою `.MODEL SMALL`, або `.MODEL TINY`. Крім того, має бути визначено для директиви SEGMENT параметр USE16, щоб всі команди і адреси були 16-бітними. Директива ORG забезпечує початок адресування від 256, а не від нуля. При таких умовах стає можливим виконати відтрансльовану COM-програму в консольному режимі для Windows, при цьому правильно виконуються більшість переривань INT 21H, які були початково розраховані на операційну систему DOS.

```
      .386P
      .MODEL      SMALL ; єдиний сегмент (модель SMALL)
CODESG      SEGMENT  DWORD PUBLIC USE16 'CODE'  ;;;USE16 !
      ASSUME      CS:CODESG,DS:CODESG,SS:CODESG,ES:CODESG
      ORG   100H  ;   резерв для PSP-префікса прогр. сегмента
BEGIN:      JMP   MAINPROG    ;   обхід даних
MESS1 DB    'Testing: print a line:'
      DB    '$'
MAINPROG    PROC  NEAR  ;  процедура ближнього типу
      MOV   AH,09 ;  вивести на екран рядок до '$'
      LEA   DX,MESS1    ;  адреса початку рядка запрошення
      INT   21H
      CALL  READLINE
      CALL  WRITELN
      JMP   EXIT  ;  вихід по 4c/Int 21h
MAINPROG    ENDP
; процедура читання рядка
READLINE    PROC  NEAR
      MOV   AH,0AH      ; прочитати з клавіатури рядок до Enter
      LEA   DX,MAXLEN   ; адреса списку параметрів
      INT   21H
      RET   ;
READLINE    ENDP
;  пам'ять та дані  для процедури читання рядка
MAXLEN      DB    30    ;  максимальна довжина введеного рядка
REALLEN     DB    ?     ;  кількість фактично введених символів
POLE  DB    30  DUP ('_')     ;  поле ( 30 літер ) для введених символів
;  процедура виведення на екран
WRITELN     PROC  NEAR
      MOV   AH,09
      LEA   DX,MESS2    ;  рядок-повідомлення
      INT   21H
; друкуємо рядок, який прочитали
      MOV   AH,40H      ;  вивести рядок вказаної довжини
      MOV   BX,1  ;  на екран
; CX = довжина рядка
      MOV   CH,0  ;  старший байт CX =0
      MOV   CL,REALLEN  ;  кількість введених символів
      LEA   DX,POLE     ;  адреса початку рядка
      INT   21H
; тепер ще надрукуємо літери для переходу до нового рядка
      MOV   AH,09
      LEA   DX,NEWLINE
      INT   21H
      RET
MESS2 DB    10,13,'This line:'
NEWLINE     DB    10,13 ; (10,13) - новий рядок
      DB    '$'
WRITELN     ENDP
EXIT:   MOV     AX,4C00h  ;  функція 4C (76)
```

```
        INT     21h     ;  "припинити програму"
CODESG      ENDS
      END   BEGIN
```

Трансляція і виконання програми

По-перше, для трансляції подібних COM-програм потрібно мати версії асемблера і завантажувача, які можуть будувати 16-розрядні програми. По-друге, потрібна додатково програма EXE2BIN, яка перетворює EXE-програми до формату COM-програми. Для компіляції наведеної програми були використані такі версії системних програм:

Macro Assembler Version 5.10

Microsoft (R) Segmented-Executable Linker Version 5.10. Copyright (C) Microsoft Corp 1984-1990.

EXE to BINary converter. Copyright (c) RaMax 1995,96 Part of FreeDOS.

Для трансляції асемблерної програми потрібно приготувати bat-файл з відповідними командами. Файл Compile.bat може виглядати так:

```
@echo off
CLS
if  exist List1.obj  del List1.obj
if  exist List1.lst  del List1.lst
MASM /N /T List1.asm,,;
PAUSE
EXIT
```

В результаті компіляції отримуємо об'єктний obj-файл і lst-файл протоколу трансляції. Протокол трансляції наведеної програми:

```
                              .386P
                              .MODEL      SMALL ; єдиний сегмент (модель SMALL)
0000                    CODESG      SEGMENT  DWORD PUBLIC USE16 'CODE';;;USE16 !
                              ASSUME      CS:CODESG,DS:CODESG,SS:CODESG,ES:CODESG
0100                          ORG   100H  ;   резерв для PSP-префікса прогр. сег.
0100  EB 18 90                BEGIN:      JMP   MAINPROG    ;   обхід даних
0103  54 65 73 74 69 6E       MESS1 DB    'Testing: print a line:'
      67 3A 20 70 72 69
      6E 74 20 61 20 6C
      69 6E 65 3A
0119  24                            DB    '$'
011A                    MAINPROG    PROC  NEAR  ;  процедура ближнього типу
011A  B4 09                         MOV   AH,09 ;  вивести на екран рядок до '$'
011C  8D 16 0103 R                  LEA   DX,MESS1    ;  адреса рядка запрошення
0120  CD 21                         INT   21H
0122  E8 012B R                     CALL  READLINE
0125  E8 0154 R                     CALL  WRITELN
0128  EB 5B 90                      JMP   EXIT  ;  вихід по 4c/Int 21h
012B                    MAINPROG    ENDP
                        ; процедура читання рядка
012B                    READLINE    PROC  NEAR
012B  B4 0A                         MOV   AH,0AH      ; прочитати з клавіатури
                                                      рядок до Enter
012D  8D 16 0134 R                  LEA   DX,MAXLEN   ; адреса списку параметрів
0131  CD 21                         INT   21H
0133  C3                           RET   ;
0134                    READLINE    ENDP
                        ;  пам'ять та дані  для процедури читання рядка
0134  1E                 MAXLEN      DB    30    ;  максимальна довжина
                        введеного рядка
0135  00                 REALLEN     DB    ?     ;  кількість фактично в
                        ведених символів
0136  001E[              POLE        DB    30 DUP ('_')      ;  поле ( 30 літер )
                                               для введених символів
        5F
```

```
                ]

                        ;  процедура виведення на екран
 0154                   WRITELN     PROC  NEAR
 0154  B4 09                        MOV   AH,09
 0156  8D 16 0176 R                 LEA   DX,MESS2    ;  рядок-повідомлення
 015A  CD 21                        INT   21H
                        ; друкуємо рядок, який прочитали
 015C  B4 40                        MOV   AH,40H      ;  вивести рядок вказаної
                                                         довжини
 015E  BB 0001                      MOV   BX,1  ; на екран  ; CX = довжина рядка
 0161  B5 00                        MOV   CH,0  ;  старший байт CX =0
 0163  8A 0E 0135 R                 MOV   CL,REALLEN  ;  кількість введених
                                                         символів
 0167  8D 16 0136 R                 LEA   DX,POLE     ;  адреса початку рядка
 016B  CD 21                        INT   21H
                        ; тепер ще надрук. літери для переходу до нового рядка
 016D  B4 09                        MOV   AH,09
 016F  8D 16 0182 R                 LEA   DX,NEWLINE
 0173  CD 21                        INT   21H
 0175  C3                           RET
 0176  0A 0D 54 68 69 73      MESS2 DB    10,13,'This line:'
       20 6C 69 6E 65 3A
 0182  0A 0D                  NEWLINE     DB    10,13 ; (10,13) - новий рядок
 0184  24                           DB    '$'
 0185                         WRITELN     ENDP
 0185  B8 4C00                EXIT: MOV   AX,4C00h  ;  функція 4C (76)
 0188  CD 21                        INT   21h    ;  "припинити програму"
 018A                   CODESG      ENDS
                                    END   BEGIN

     53 Source  Lines
     53 Total   Lines
     28 Symbols

  46976 + 442445 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

Для завантаження і побудови COM-програми потрібно приготувати bat-файл з відповідними командами. Файл Linking.bat може виглядати так:

```
@echo off
CLS
if exist  List1.exe  del  List1.exe
if exist  List1.com  del  List1.com
if exist  List1.map  del  List1.map
LINK /CP:1 /MAP List1.obj,,;
EXE2BIN List1.exe List1.com
PAUSE
EXIT
```

В результаті завантаження отримуємо протокол у файлі List1.map приблизно такого змісту:

```
LINK : warning L4021: no stack segment

 Start  Stop   Length Name                   Class
 00000H 00000H 00000H _TEXT                 CODE
 00000H 00189H 0018AH CODESG                CODE
 0018CH 0018CH 00000H _DATA                 DATA

 Origin   Group
 0018:0   DGROUP
  Address         Publics by Name
```

```
  Address         Publics by Value
Program entry point at 0000:0100
```

Якщо не було ніяких помилок, то повинні отримати програму у файлі List1.com. Для запуску на виконання такої програми потрібно мати ще один командний bat-файл GO.bat:

```
@echo off
CLS
D:\V_V\TestASM\SysProgEx\List1.com
PAUSE
EXIT
```

При запуску цього командного файла вікно консолі буде затримане на екрані до натиснення будь-якої клавіші (команда PAUSE), отже, можна переглянути результати, надруковані у вікні консолі.

Перевірка моделі виконання COM-програми

1. Складіть bat-файл, подібний до GO.bat, лише замість рядка з викликом com-програми запишіть яку-небудь системну програму, наприклад, notepad.exe. Це є блокнот. Натисність Enter на такому файлі у вікні провідника Windows. Повинна запуститись програма блокнота. Закрийте вікно блокнота, після чого – вікно консолі, бо обидва вікна працюють в одному потоці. Щоб мати два незалежні потоки – для блокнота і для консолі – команду запуску програми в bat-файлі записуємо з словом Start:

```
Start notepad.exe
```

2. Змоделюйте виклик такого bat-файла з будь-якої програми, написаної на C#. Це може бути звичайна консольна програма системи .NET. У програмі потрібно запустити на виконання паралельний потік з вказаним bat-файлом, наприклад, так:

```
System.Diagnostics.Process.Start(bat);
```

де параметр bat визначає ім'я bat-файла.

3. Запишіть в окремий файл двійкові коди відтрансльованих команд, взяті з початкового фрагмента наведеної вище асемблерної програми:

```
                                ORG   100H  ;   резерв для PSP-префікса прогр. сег.
EB 18 90                BEGIN:  JMP   MAINPROG    ;   обхід даних
54 65 73 74 69 6E       MESS1 DB    'Testing: print a line:'
67 3A 20 70 72 69
6E 74 20 61 20 6C
69 6E 65 3A
24                              DB    '$'
                        MAINPROG  PROC  NEAR  ;  процедура ближнього типу
B4 09                           MOV   AH,09 ;  вивести на екран рядок до '$'
8D 16 03 01 R                   LEA   DX,MESS1    ;  адреса рядка, який виводимо
CD 21                           INT   21H
B8 00 4C                EXIT:   MOV   AX,4C00h  ;  функція 4C (76)
CD 21                           INT   21h     ;  "припинити програму"
                        MAINPROG  ENDP
```

Обведені рамкою числа є шістнадцятковим позначенням окремих байтів відтрансльованої програми. Можна зробити, наприклад, так. Створити в C#-програмі масив типу byte, та ініціалізувати його шістнадцятковими константами, записаними вище. Шістнадцяткові константи позначають в C# префіксом 0x: 0xEB, 0x18, 0x90 і т.д.

Файл, у який записують байти, має бати двійковим (Binary), і записувати до нього потрібно через клас BinaryWriter. Назвіть файл з розширенням .com.

4. Поміняйте у bat-файлі ім'я програми на файл з розширенням .com, побудований в попередньому пункті. Слово Start перед іменем програми записувати не потрібно.

5. Ще раз змоделюйте виклик такого bat-файла, подібно до п.2. Якщо все зроблено точно, у вікні консолі має з'явитись повідомлення: `Testing: print a line:`.

6. Поміняйте вручну необхідні двійкові коди п.3 так, щоб отримати інший текст повідомлення.