

Связь ассемблера с языками высокого уровня



План темы:

1. Формы комбинирования программ на языках высокого уровня с ассемблером.
2. Варианты взаимодействия.
3. Взаимодействие Паскаль-Ассемблер.
- 4.4. Пример.



1. Формы комбинирования программ на языках высокого уровня с ассемблером:

- **Использование в языках высокого уровня специальных операторов (`inline`, `asm` в Паскале), позволяющих вставить в текст программы фрагменты на ассемблере;**
- **Использование внешних процедур и функций (этот способ более универсальный и предпочтительней).**

2. Варианты взаимодействия:

- **Основная программа на ассемблере вызывает подпрограммы, написанные на языке высокого уровня и хранящиеся в объектных модулях (obj-файлах);**
- **Основная программа на языке высокого уровня вызывает подпрограммы, написанные на ассемблере и хранящиеся в объектных модулях (obj-файлах) (этот способ более удобный, его мы и будем использовать).**

3. Взаимодействие Паскаль – Ассемблер.

- **Параметры в подпрограмму передаются через стек по принципу «слева направо»;**
- **Подпрограмма-функция (function) - это процедура ассемблера, которая возвращает результат в регистрах;**
- **В процедуре ассемблера возможно использование внешних переменных модуля Паскаля после их объявления в директиве **extrn <имя>:тип:количество****

3. Схема pas-файла, использующего asm-процедуры:

```
Program PasAsm;  
  Procedure ProcAsm(a,b:integer);external;  
  Function FunAsm(n:integer):integer;external;  
  Var x,k:integer;  
  {$L asmmmod.obj} {подключение файла}  
  Begin  
    ...  
    ProcAsm(x,5); {вызов asm-процедуры}  
    ...  
    k:=FunAsm(x); {вызов asm-функции}  
    ...  
  End.
```

3. Схема asm-файла, используемого в pas-программе:

.model small

.data

...

.code

begin:

ProcAsm proc

Public ProcAsm

push bp

mov bp,sp

...;[bp+6]-доступ к первому параметру (a),

;[bp+4]-доступ к второму параметру (b).

pop bp

ret 4

ProcAsm endp

3. Схема asm-файла, используемого в pas-программе (продолжение):

```
FunAsm proc  
Public FunAsm
```

```
    push bp
```

```
    mov bp,sp
```

```
    ...;[bp+4]- доступ к параметру n
```

```
    ...;в ax должен находиться результат
```

```
    pop bp
```

```
    ret 2
```

```
FunAsm endp
```

```
end begin
```


5. Пример.

Составить программу вычисления:

$$y = \sum_{i=1}^n 2 \cdot i - \sum_{i=1}^m 3 \cdot i$$

- Основную программу разработать на языке Паскаль;
- Для вычисления суммы создать функцию **Summa(n,k:integer):integer** на Ассемблере;
- Функцию **Summa** разместить во внешнем obj- файле;

Файл primer.pas:

```
Program PasAsm;  
Function Summa(n,k:integer):integer;external;  
Var n,m,y:integer;  
{ $L asmmod.obj } {подключение файла}  
Begin  
  Write('Введите n,m: ');  
  Readln(n,m);  
  y:=Summa(n,2)-Summa(m,3);  
  Writeln('y=',y);  
  Readln;  
End.
```

Файл asmmod.asm:

.model small

.code

begin:

Summa proc

Public Summa

push bp

mov bp,sp

mov cx,[bp+6] ;cx:=n

mov ax,0 ;ax:=0

for: add ax,cx ;ax:=ax+cx

loop for ;cx:=cx-1

imul [bp+4] ;ax:=ax*k

pop bp

ret 4

Summa endp

end begin