

Организация подпрограмм

Подпрограмма-некоторый поименованный участок кода программы синтаксически оформленный так, что имеется возможность его вызова из любого места программы.

Назначение подпрограмм:

- 1. Программирование на более высоком уровне.**
- 2. Уменьшение размеров программы.**

Организация подпрограмм

С целью создания возможности организации подпрограмм в архитектуру УМ включены специальные средства:

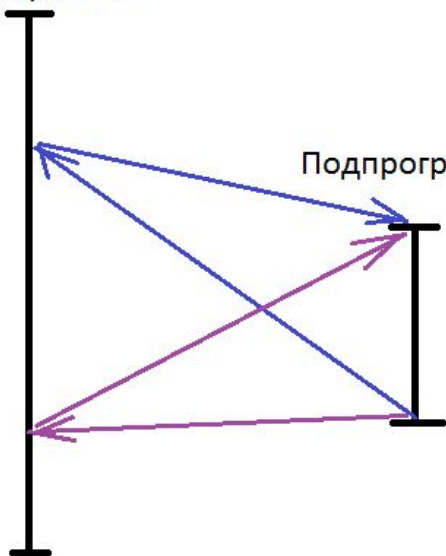
- регистр связи L;**
- команда " Переход к подпрограмме " (jsub);**
- команда " Возврат из подпрограммы" (rsub);**
- посылочные команды ldl и stl.**

Организация подпрограмм

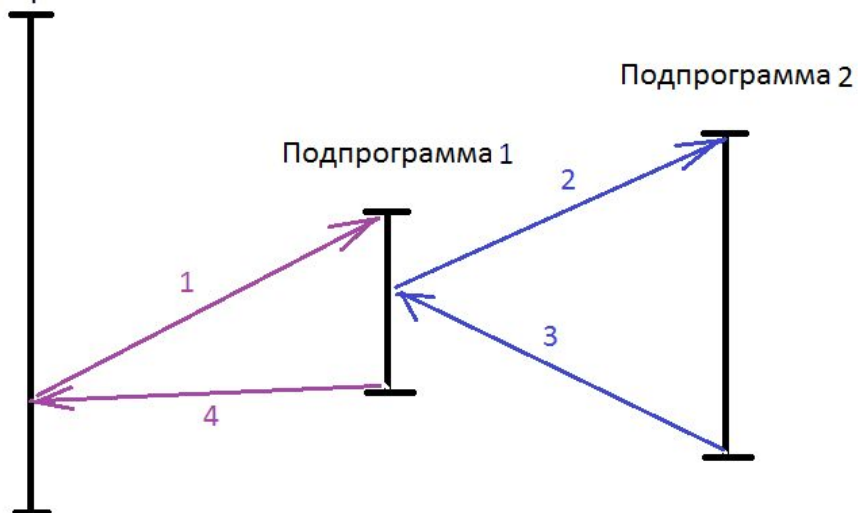
Проблемы при работе с подпрограммами:

1. Связь по управлению

Основная программа



Основная программа



2. Связь по данным

Связь по управлению

Пример иллюстрирует организацию обращения к подпрограмме

```
// Ex41.cpp
#include "stdafx.h"
int v;
void inc()
{
    v++;
}
int main()
{
    v = 1;
    inc();
    inc();
    return 0;
}
```

| | | | |
|--------|--------|--------------------|--------|
| H Ex41 | Ex41 | start | 0 |
| T 000 | 00000D | lda | c1 |
| T 003 | 0C0010 | sta | v |
| T 006 | 480013 | jsub | inc |
| T 009 | 480013 | jsub | inc |
| T 00C | FF | hlt | |
| | | ; данные | |
| T 00D | 000001 | c1 | word 1 |
| T 010 | | v | resb 3 |
| | | ; подпрограмма inc | |
| T 013 | 000010 | inc | lda v |
| T 016 | 18000D | | add c1 |
| T 019 | 0C0010 | | sta v |
| T 01C | 4C | | rsub |
| E 000 | | end | Ex41 |

Связь по данным

Существуют следующие основные способы передачи данных между программными единицами:

- передача данных через регистры (A и X);**
- использование общих переменных;**
- использование механизма формальных и фактических параметров.**

Связь по данным

Пример передачи параметров к подпрограмме по адресу

```
H Ex42
T 000 480019
T 003 000013
T 006 480019
T 009 000016
T 00C 480019
T 00F 000013
T 012 FF
T 013 000005
T 016 000007
T 019 AC21
T 01B 048000
T 01E 008000
T 021 18002F
T 024 0C8000
T 027 AC20
T 029 180032
T 02C AC02
T 02E 4C
T 02F 000001
T 032 000003
E 000
```

```
Ex42 start 0
jsub inc
word v ;адрес v
jsub inc
word w ;адрес w
jsub inc
word v ;адрес v
hlt

; данные
v word 5
w word 7

; подпрограмма inc
inc rmo 1,x ;X="адрес адреса параметра"
ldx 0,x ;X="адрес параметра"
lda 0,x ;A="значение параметра"
add c1 ;A=A+1
sta 0,x ;"параметр" = A
rmo 1,a ; Корректировка
add c3 ;адреса возврата
rmo a,1 ;на длину ТАЗ.
rsub

; данные п/п inc
c1 word 1
c3 word 3
end Ex42
```

```
// Ex42.cpp :
#include "stdafx.h"
int v = 5, w = 7;
void inc(int *i)
{ (*i)++; }
int main()
{
    inc(&v);
    inc(&w);
    inc(&v);
    return 0;
}
```

Связь по данным

Пример передачи параметров к подпрограмме по адресу и по значению

```
H Ex43
T 000 00001F
T 003 0C0009
T 006 480028
T 009
T 00C 000022
T 00F 000022
T 012 0C0018
T 015 480028
T 018
T 01B 000025
T 01E FF
; данные
T 01F 00000F
T 022
T 025
; подпрограмма copy
T 028 AC21
T 02A 008000
T 02D 048003
T 030 0C8000
T 033 AC20
T 035 18003B
T 038 AC02
T 03A 4C
; данные п/п copy
T 03B 000006
E 000
```

```
Ex43 start 0
lda u;ВЫЗОВ copy(u,v)
sta p11
jsub copy
p11 resb 3
p12 word v
lda v ;ВЫЗОВ copy(v,w)
sta p21
jsub copy
p21 resb 3
p22 word w
hlt
```

```
copy rmo 1,x ;X = "адрес ТА3"
lda 0,x ;A = p1
ldx 3,x ;X = p2
sta 0,x ;p2 = A
rmo 1,a ; Корректировка
add c6 ;адреса возврата
rmo a,1 ;на длину ТА3.
rsub
```

```
c6 word 6
end Ex43
```

```
// Ex43.cpp
#include "stdafx.h"
const int u = 15;
int v, w;
void copy(int x, int *y)
{ *y = x; }
int main()
{
  copy(u, &v);
  copy(v, &w);
  return 0;
}
```

Связь по данным

Пример организации подпрограммы, возвращающих значение

```
H Ex44
T 000 00001F
T 003 0C0009
T 006 480028
T 009
T 00C 0C0022
T 00F 000022
T 012 0C0018
T 015 480028
T 018
T 01B 0C0025
T 01E FF

; данные
T 01F 000011
T 022
T 025
; подпрограмма-функция inc
T 028 AC21
T 02A AC20
T 02C 18003B
T 02F AC02
T 031 008000
T 034 180038
T 037 4C

; данные inc
T 038 000001
T 03B 000003
E 000
```

| | | |
|------|-------------|---------------------|
| Ex44 | start | 0 |
| | lda | u ;v=inc(u) |
| | sta | p11 |
| | jsub | inc |
| p11 | resb | 3 |
| | sta | v |
| | lda | v ;w=inc(v) |
| | sta | p21 |
| | jsub | inc |
| p21 | resb | 3 |
| | sta | w |
| | hlt | |
| u | word | 17 |
| v | resb | 3 |
| w | resb | 3 |
| inc | rmo | l,x ;X="адрес ТАЗ" |
| | rmo | l,a ; Корректировка |
| | add | c3 ;адреса возврата |
| | rmo | a,l ;на длину ТАЗ. |
| | lda | 0,x ;A=p1 |
| | add | c1 |
| | rsub | |
| c1 | word | 1 |
| c3 | word | 3 |
| | end | Ex44 |

```
// Ex44.cpp
#include "stdafx.h"
const int u = 10;
int v, w;
int inc(int x)
{
    return x + 1;
}
int main()
{
    v = inc(u);
    w = inc(v);
    return 0;
}
```


Пример

Обработка массива подпрограммой

```
#include "stdafx.h"

void mas_max(int l, int *d, int *max)
{
    int max_l=d[0];
    for (int i =1; i < l; i++)
        if (d[i] > max_l) max_l = d[i];
    *max = max_l;
}

int main()
{
    int d1[5] = { 1,2,1,2,1 };
    int d2[3] = { 1,2,13};
    int max1, max2;
    mas_max(5, d1, &max1);
    mas_max(3, d2, &max2);
    return 0;
}
```

Пример

Обработка массива подпрограммой

```
H ExMas | ExMas start 0
T 000 480037 | jsub MasMax
T 003 00000F | 11 word 15 ;значение 1*3
T 006 000019 | word d1 ;адрес d1
T 009 000031 | word max1 ;адрес max1
T 00C 480037 | jsub MasMax
T 00F 000009 | 12 word 9 ;значение 1*3
T 012 000028 | word d2 ;адрес d2
T 015 000034 | word max2 ;адрес max2
T 018 FF | hlt
| ; данные
T 019 | d1 resb 15
T 028 | d2 resb 9
T 031 | max1 resb 3
T 034 | max2 resb 3
```

Пример

Обработка массива подпрограммой

```
T 037 AC21 | MasMax rmo    1,x ;X="адрес параметра 1 (адрес ТАЗ)"
T 039 008000 |         lda    0,x ;A="значение параметра 1"
T 03C 188003 |         add    3,x ;A=«адрес массива»+«длина массива(A)»
T 03F 0C008A |         sta    ll  ;ll = A(адрес ячейки памяти идущей
                    ; следующей после последнего элемента
массива)
T 042 048003 |         ldx    3,x ;X="адрес параметра d"(адрес начала массива)
T 045 008000 |         lda    0,x ;A="значение параметра d[0]"
T 048 0C008D |         sta    max ;сохранение певрого элемента массива
                    ; в лок. перем.
T 04B AC10   |         rmo    x,a ;i=i+1(X=X+3)
T 04D 180084 |         add    c3  ;X-адрес элемента d[1]
T 050 AC01   |         rmo    a,x
T 052 008000 | rpt     lda    0,x ;A=d[i]
T 055 28008D |         comp   max  ;A > max ?
T 058 380061 |         jlt    m    ;Переход, если "меньше".
T 05B 300061 |         jeq    m    ;Переход, если "равно".
T 05E 0C008D |         sta    max  ;max=A
T 061 AC10   | m       rmo    x,a ;i=i+1(X=X+3)
T 063 180084 |         add    c3
T 066 AC01   |         rmo    a,x
T 068 28008A |         comp   ll   ;X < ll ?
T 06B 380052 |         jlt    rpt  ;Переход, если "да".(Если проверели не все
                    ;элементы массива, то возвращаемся в начало цикла)
```

Пример

Обработка массива подпрограммой

```
T 06E AC21 | rmo    1,x  ;X="адрес (адреса параметра max(глобальной)-6)"
T 070 048006 | ldx    6,x  ;X="адрес параметра max(глобальной)"
T 073 00008D | lda    max  ;Сохранение максимального значения в глобальную
T 076 0C8000 | sta    0,x  ;переменную

T 079 AC20 | rmo    1,a  ;Корректировка
T 07B 180087 | add    c9   ;адреса возврата
T 07E AC02 | rmo    a,1  ;на длину ТАЗ.
T 080 4C    | rsub

; данные п/п MasMax
T 081 000001 | c1     word  1
T 084 000003 | c3     word  3
T 087 000009 | c9     word  9
T 08A      | l1     resb  3
T 08D      | max    resb  3
E 000      | end    ExMas
```

Пример

Организация вложенных подпрограмм

```
#include "stdafx.h"
```

```
void inc(int *i)
```

```
{
```

```
(*i)++;
```

```
}
```

```
void inc2(int *i)
```

```
{
```

```
inc(i);
```

```
inc(i);
```

```
}
```

```
int main()
```

```
{
```

```
int v = 5;
```

```
inc2(&v);
```

```
return 0;
```

```
}
```

Пример

Организация вложенных подпрограмм

| | | | |
|--------------|---------------------|-------------|---|
| H VPod | VPod | start | 0 |
| T 000 48000A | jsub | inc2 | |
| T 003 000007 | word | v | ; адрес v |
| T 006 FF | hlt | | |
| | ; данные | | |
| T 007 000005 | v | word | 5 |
| | ; подпрограмма inc2 | | |
| T 00A AC21 | inc2 | rmo | l,x ;X="адрес адреса параметра" |
| T 00C 008000 | | lda | 0,x ;A="адрес параметра" |
| T 00F 0C0020 | | sta | v1 |
| T 012 0C0026 | | sta | v2 |
| T 015 AC20 | | rmo | l,a ; Корректировка |
| T 017 18004B | | add | c3 ;адреса возврата |
| T 01A 0C002F | | sta | l1 ;на длину ТАЗ. Сохранение регистра L |
| T 01D 480032 | jsub | inc | |
| T 020 | v1 | resb | 3 ;адрес параметра |
| T 023 480032 | jsub | inc | |
| T 026 | v2 | resb | 3 ;адрес параметра |
| T 029 00002F | | lda | l1 |
| T 02C AC02 | | rmo | a,l ; Восстановление регистра L |
| T 02E 4C | rsub | | |
| | ; данные п/п inc2 | | |
| T 02F | l1 | resb | 3 |

Пример

Организация вложенных подпрограмм

```
T 032 AC21 | ; подпрограмма inc
T 034 048000 | inc rmo 1,x ;X="адрес адреса параметра"
T 037 008000 | ldx 0,x ;X="адрес параметра"
T 03A 180048 | lda 0,x ;A="значение параметра"
T 03D 0C8000 | add c1 ;A=A+1
T 040 AC20 | sta 0,x ;"параметр" = A
T 042 18004B | rmo 1,a ; Корректировка
T 045 AC02 | add c3 ;адреса возврата
T 047 4C | rmo a,1 ;на длину ТАЗ.
| rsub
| ; данные п/п inc
T 048 000001 | c1 word 1
T 04B 000003 | c3 word 3
E 000 | end VPod
```