

انتقال پارامتر به زیربرنامه

پارامترهای مجازی با ۳ مدل معنایی مشخص می شوند:

۱- داده ها را می توانند از پارامترهای واقعی متناظر دریافت کنند.

۲- داده ها را می توانند به پارامترهای واقعی انتقال دهند.

۳- هر دو مورد فوق را با هم انجام می دهند.

پارامترهای واقعی : a, b, c

ارسال با مقدار

مقدار پارامتر واقعی در پارامتر مجازی متناظر کپی می‌شود که به عنوان یک متغیر مجازی در زیر برنامه عمل می‌کند.

```
void f( int x){  
    x=5;  
    cout << x;  
}
```

```
main ( ){  
    int a=1;  
    f(a);  
    cout << a;  
}
```

خروجی:

5
1

مثال

Call by Value

```
int x = 1;
```

```
void f(int a){  
    x = 2;  
    a = 5;  
}
```

```
void main ( ) {  
    f(x);  
    cout<<x;  
}
```

خروجی: 2

ارسال با نتیجه

وقتی پارامتر با نتیجه ارسال می‌شود، هیچ مقداری به زیر برنامه انتقال نمی‌یابد.

پارامتر مجازی متناظر با آن به عنوان متغیر محلی عمل می‌کند، اما درست قبل از اینکه کنترل به فراخوان برگردد، مقدارش به پارامتر واقعی فراخوان منتقل می‌شود.

- یکی از مشکلات **ارسال با نتیجه**، استفاده از پارامترهای واقعی یکسان است.

- در روش **ارسال با نتیجه**، برنامه قابلیت حمل ندارد.

ارسال با مقدار و نتیجه

Call by Value-result

```
int x = 1;
```

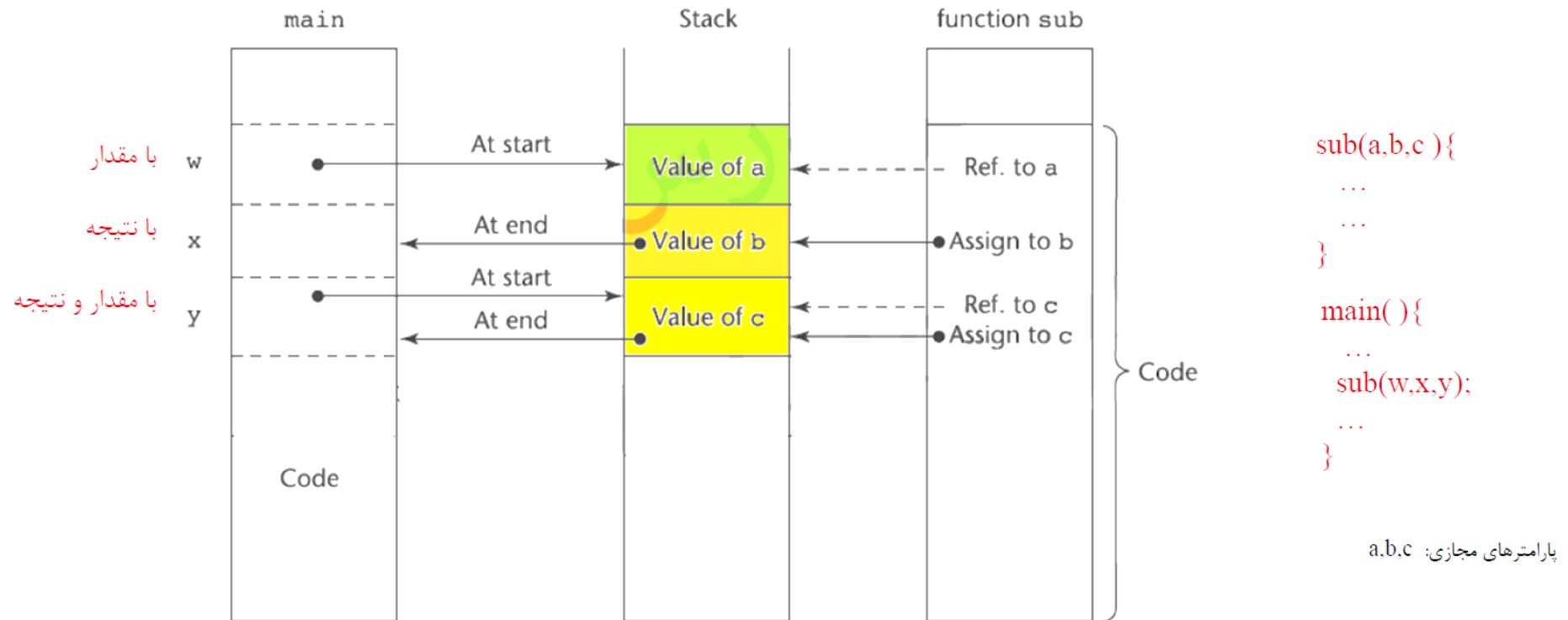
```
void f(int a)
{
    a = 3;
    x = 4;
}
```

```
main ( ) {
    f(x);
    cout<<x;
}
```

مقدار پارامتر واقعی موجب مقدار اولیه دادن به پارامتر مجازی می شود که به عنوان متغیر محلی عمل می کند.
پارامترهای مجازی، باید حافظه ای وابسته به زیر برنامه فراخوانی شده داشته باشند.
با خاتمه زیر برنامه، مقدار پارامتر مجازی به پارامتر واقعی برگردانده می شود.

پیاده سازی

ارتباط پارامترها به وسیله **پشته زمان اجرا** صورت می گیرد.



ارسال با ارجاع (call by refrence)

بر خلاف روش قبل که مقادیر را کپی می کرد، مسیر دستیابی (آدرس) به زیر برنامه ارسال می شود. زیر برنامه فراخوانی شده می تواند به پارامتر واقعی در زیر برنامه فراخوان دستیابی داشته باشد.

```
void f ( int x) {  
    x=5;  
    cout << x;    5  
}
```

```
main ( ){  
    int a=1;  
    f (&a);  
    cout << a;    5  
}
```

- ارسال با ارجاع از نظر زمان و فضا کارآمد است.
- از معایب ارسال با ارجاع، ایجاد نام مستعار است.
- دستیابی به پارمترهای مجازی در ارسال با ارجاع، کندتر از ارسال با مقدار است.

انتقال پارامتر

در C

```
swap(&p,&q);
```

```
void swap (int *a, int *b) {  
    int t;  
    t = *a;  
    *a = *b;  
    *b = t;  
}
```

All parameters passed **by value**.

در C++

```
swap (p,q);
```

```
void swap (int &a, int &b) {  
    int t;  
    t = a;  
    a = b;  
    b = t;  
}
```

C++ has **reference** parameters.

ارسال با نام

```
var i : integer;  
function f ( x : integer) : integer ;  
begin  
    i=2;      i=2;  
    x=8;      a[i]=8;  
end;  
begin  
    a : array [1..2] of integer = {7,3};  
    i=1;  
    f ( a [ i ] );  
    print( a[1] );    7  
    print( a[2] );    8  
end;
```

پارامترهای واقعی از نظر متنی، به جای پارامتر مجازی در داخل زیر برنامه قرار می‌گیرد.
پارامترهای مجازی، در زمان فراخوانی زیر برنامه به روش دستیابی بایند می‌شود، ولی بایند واقعی به مقدار یا آدرس، تا انتساب به پارامتر مجازی یا ارجاع به آن، به تاخیر می‌افتد.
در این روش، هنگام فراخوانی زیر برنامه، پارامترهای واقعی ارزیابی نمی‌شوند، تا زمانی که در زیر برنامه به آن‌ها مراجعه شود.

پارامترها بدون ارزیابی ارسال می‌شوند و زیر برنامه تصمیم می‌گیرد که کی آن‌ها را ارزیابی کند.

روش ارسال با نام در زبان‌های متداول کاربرد ندارد. (در ALGOL استفاده می‌شد).

مثال

```
P(x) {  
    x = x + x;  
}
```

```
F(m) {  
    m = m + 1;  
    return m;  
}
```

```
int A[10];  
m = 1;
```

```
P( A[ F(m) ] );
```

```
A[F(m)] = A[F(m)] + A[F(m)]  
A[++m] = A[++m] + A[++m]  
A[2] = A[3] + A[4]
```

