

```
void Increase(int a) {  
    a = a + 1;  
}
```

В этой программе присутствует функция **Increase**, параметр в которую передается по значению.

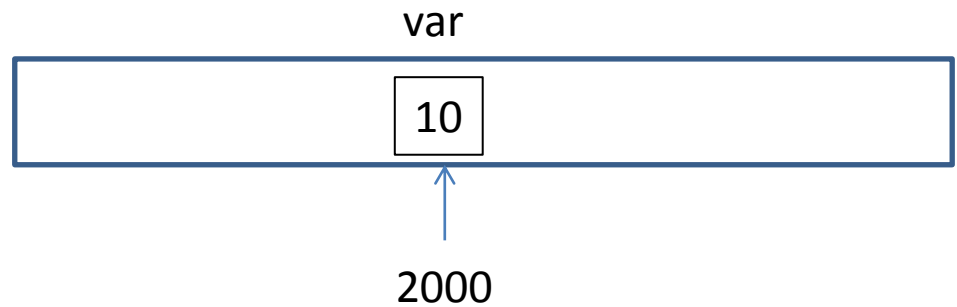
```
int main(void) {  
    int var = 10;  
  
    Increase(var);  
  
    return 0;  
}
```

Вызов данной функции никак не повлияет на значение переменной **var** в вызывающей программе.

Рассмотрим процесс вызова такой функции по шагам:

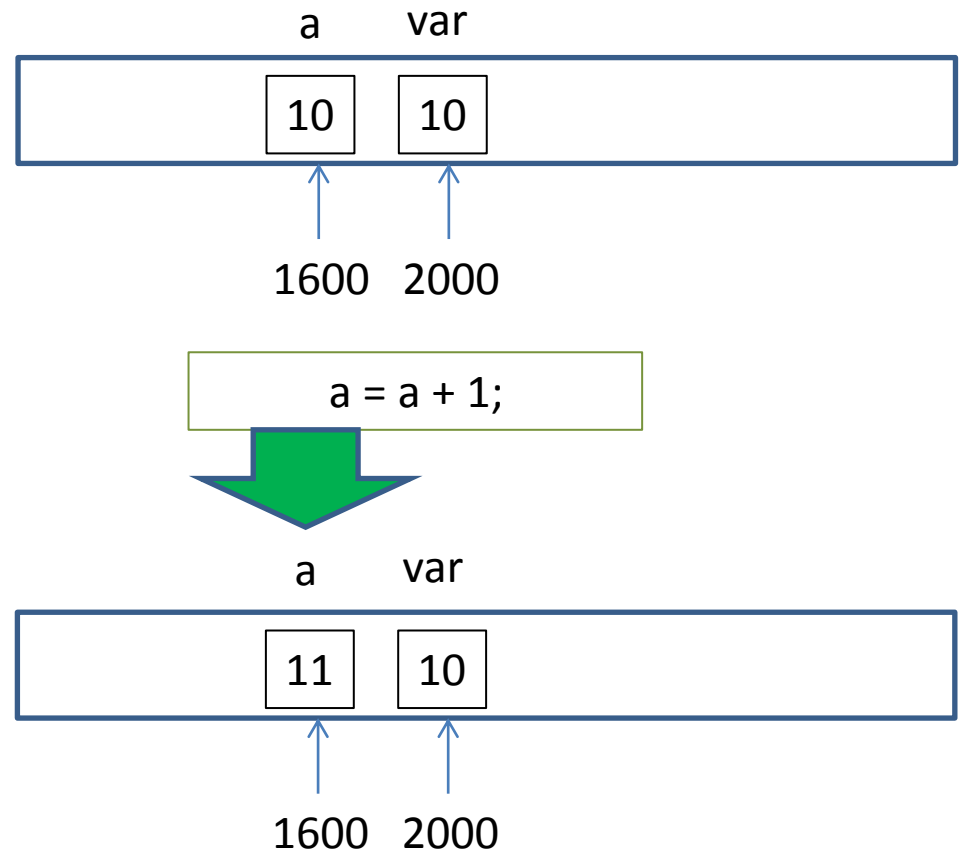
При объявлении локальной переменной **var** для нее выделяется место в стеке. Пусть это будет адрес **2000**.

```
int main(void) {  
    int var = 10;  
  
    Increase(var);  
  
    return 0;  
}
```



```
void Increase(int a) {  
    a = a + 1;  
}
```

```
int main(void) {  
    int var = 10;  
    Increase(var);  
    return 0;  
}
```



При вызове функции **Increase** для ее параметра **a** также выделяется место в стеке (пусть это будет адрес **1600**). Теперь в этот параметр **копируется** текущее значение переменной **var**. Таким образом, у нас получаются две независимые переменные, хранящиеся по разным адресам в памяти.

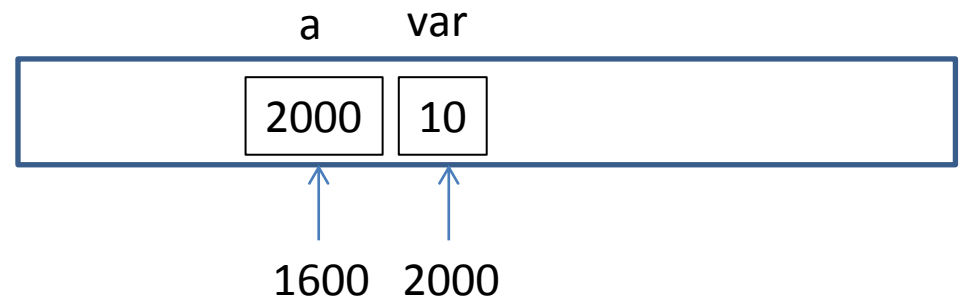
Строка **a = a + 1** изменяет только значение переменной **a**, но никак не влияет на переменную **var**.

В следующем примере параметр передается в функцию по ссылке.

Переменная **a** теперь является указателем и хранит в себе адрес некоторой переменной (в данном случае адрес переменной **var**, равный **2000**).

```
void Increase(int* a) {  
    *a = *a + 1;  
}
```

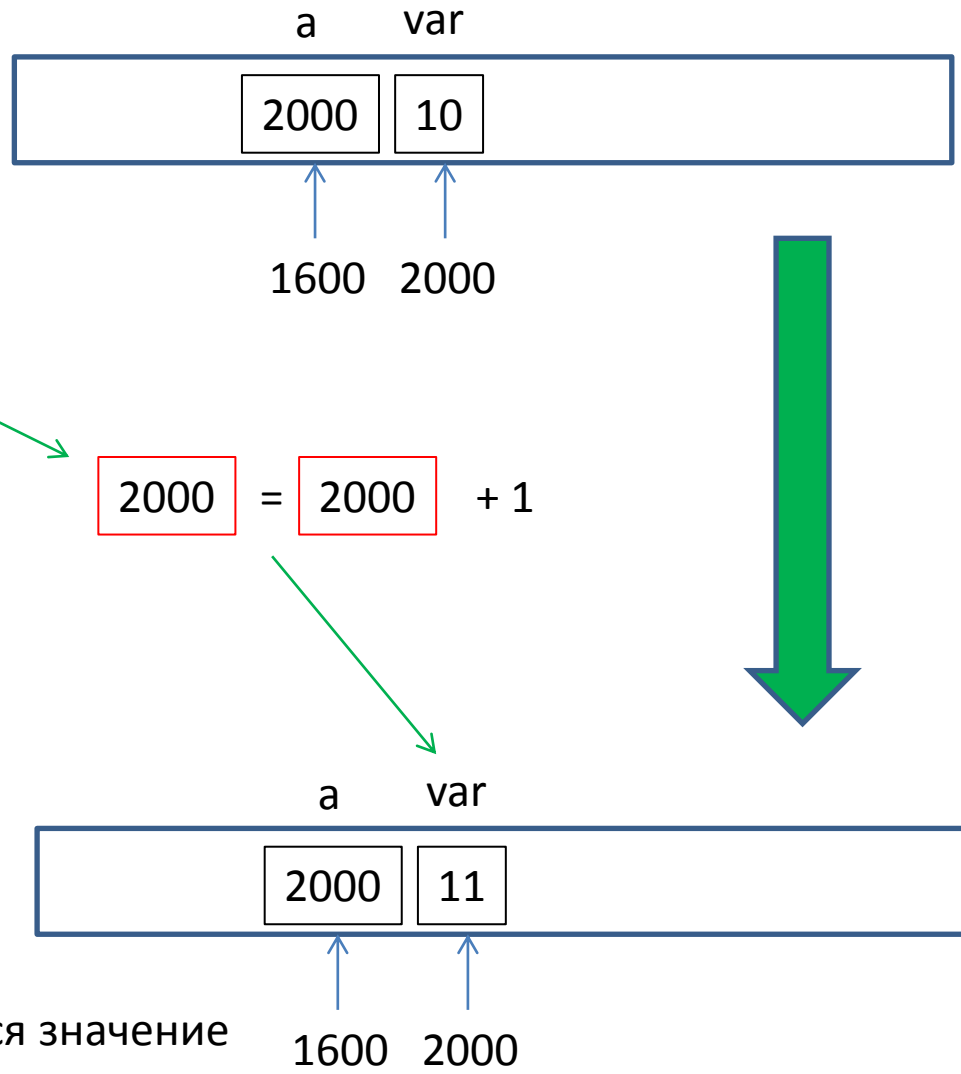
```
int main(void) {  
    int var = 10;  
    Increase(&var);  
    return 0;  
}
```



Строка ***a = *a + 1** берет значение по тому адресу, куда указывает **a**, увеличивает это значение на единицу и сохраняет полученное значение по тому адресу, куда указывает **a**.

```
void Increase(int* a) {  
    *a = *a + 1;  
}
```

```
int main(void) {  
    int var = 10;  
    Increase(&var);  
    return 0;  
}
```



Получается, что по адресу **2000** сохраняется значение **11**. Но по адресу **2000** хранится переменная **var** и, таким образом, ее значение также будет изменено.

Получается, что передавая параметр по значению, мы внутри функции изменяем его копию, но никак не влияем на этот параметр в вызывающей программе.

В то же время, передавая параметр по ссылке, мы из функции можем изменить его значение и вызывающая программа продолжит работу с уже новым значением переданного параметра.