

# PASS By VALUE vs PASS By REFERENCE

By Mustafa Onur Parlak

- **Temelde** ikisi de fonksiyon çağrıldığında, bu işlevdeki parametreler **value** ve **reference** iletebilir.
- **Pass By Value**
  - **İki parametre** için farklı memory adresleri kullanılır.
  - **Parametreleri** fonksiyona taşıırken, **stack'e** parametrenin değerini taşıyan adres yerine direkt değer ataması yapmasıdır.
  - **Bu sayede**, fonksiyon parametreyle oynayarak değeri değiştirse bile adresten çekilen bilgi değişmeyeceği için sorun yaratmayacaktır.
  - **Bu durumun** sağlanması için orijinal değişkeni etkilemeyen geçici bir değişken yaratılır.
  - **Buradaki asıl sorun** ise **struct** içerikli parametreleri bu şekilde **stack'lediğimizde**, bellek alanı bakımında büyük yer kaplaması sebebiyle; **const kullanmak daha uygun olacaktır.**

## CODE Örneği (Pass by Value)

```
// Pass by Value
#include <stdio.h>
#include <stdlib.h>

void swap(int x, int y);

int main()
{
    int a = 10, b = 20;

    swap(a, b);

    printf("a=%d b=%d\n", a, b);

    return 0;
}

void swap(int x, int y)
{
    int t;

    t = x;
    x = y;
    y = t;

    printf("x=%d y=%d\n", x, y);
}
```

- **Pass By Reference**

- Fonksiyona değişkenin **adresini** gönderir.
- Fonksiyon içinde yapılan değişiklikler değişkeni etkiler.
- İki değişken aynı memory adresini kullanır.
- Birden fazla değer döndürme işlemlerinde tekrar tekrar kullanılabilir.
- Büyük veri yapılarının **stack** ile taşınmasına kıyasla (**Pass By Value**) çok daha verimli ve hızlıdır.
- **Pointer** taşınımı sağlanır.
- **Buradaki asıl sorun** ise; çağrılan fonksiyon, iletilen parametreyi bozar ise çağırıcı da etkilenmektedir.

### CODE Örneği (Pass by Reference)

```
// Pass by Reference

#include <stdio.h>

#include <stdint.h>

void swapx(int*, int*);

int main()
{
    int a = 10, b = 20;

    swapx(&a, &b);

    printf("a=%d b=%d\n", a, b);

    return 0;
}

void swapx(int* x, int* y)
{
    int t;

    t = *x;
    *x = *y;
    *y = t;

    printf("x=%d y=%d\n", *x, *y);
}
```

// İki farklı değeri birbirine atama durumu temsil edilmektedir.

```
#include <stdio.h>
```

```
#include <stdint.h>
```

```
void swap(int *pFirst, int *pSecond)
```

```
{
```

```
    int temp;
```

```
    temp = *pFirst;
```

```
    *pFirst = *pSecond;
```

```
    *pSecond = temp;
```

```
    return;
```

```
}
```

```
int main(void)
```

```
{
```

```
    int a = 100;
```

```
    int b = 200;
```

```
    printf("swap oncesi a degiskeni: %d \n", a);
```

```
    printf("swap oncesi b degiskeni: %d \n", b);
```

```
    swap(&a, &b);
```

```
    printf("swap sonrasi a degiskeni: %d \n", a);
```

```
    printf("swap sonrasi b degiskeni: %d \n", b);
```

```
    return 0;
```

```
}
```