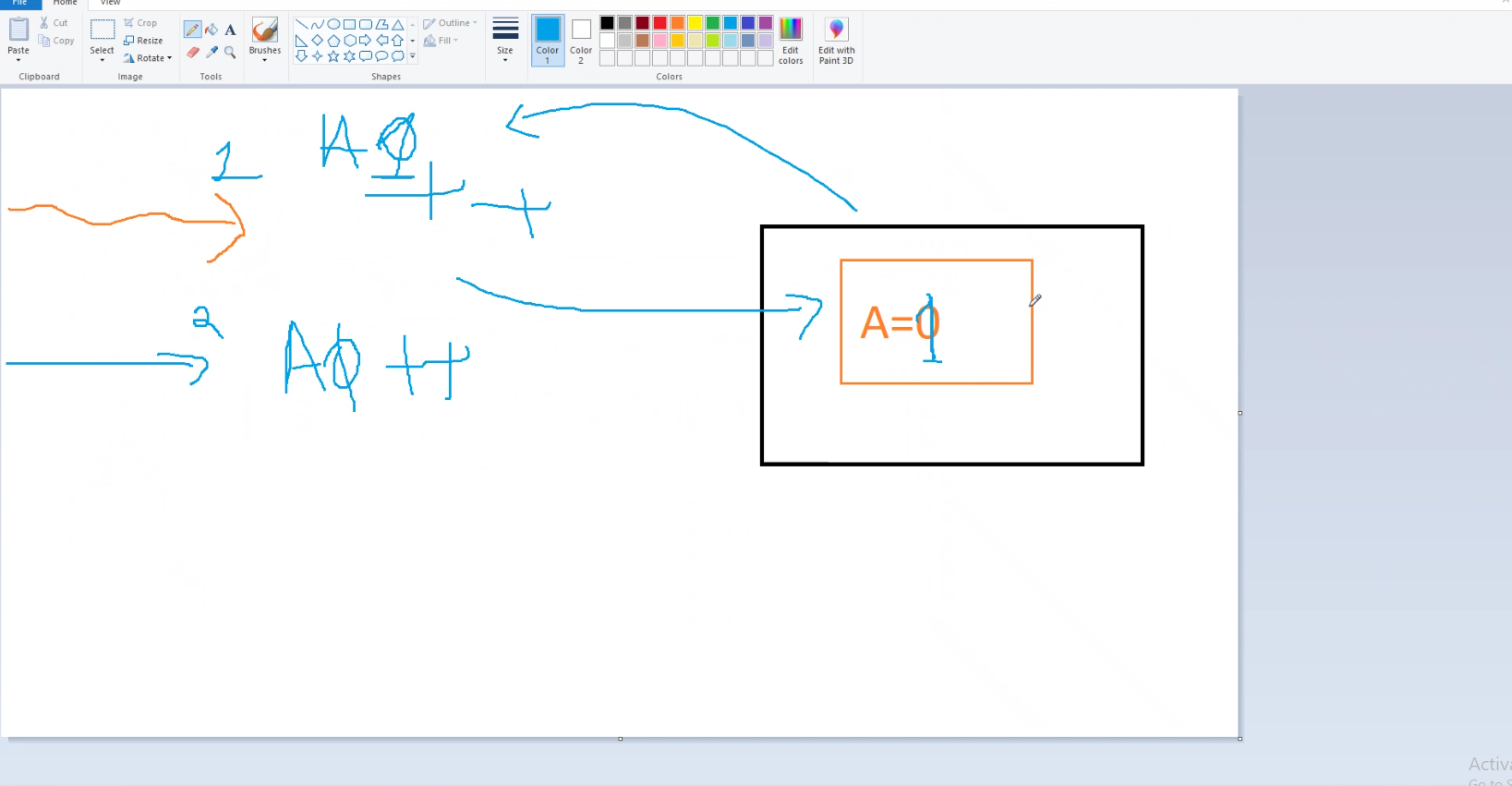
**Synchronization**

**Synchronization Problem**

Iki thread eyni zamanda girib datani deyisende problem olur



Helli : datani izolasiyaya almaq, belelikle gelen thread data izolasiyadadirsa gozleyir.

Hell yollari:  
1) **Interlocked**(az islenir)

2) **Monitor** (lock un arxasinda bu islenir)

3) **Lock** (yaxsidir)

4) **Mutex** (lockdan da yaxsidir, en yaxsisidir, arxada monitor isledir)

Artirmaq ve azaltmaq olan problemerin qarsisini bunlarla almaq olar:

**Interlocked**.Increment(ref Counter.Count);

**Interlocked**.Decrement(ref Counter.Count);

Bu zaman threadleri de join etmek lazimdir.

**Monitor** classi mueyyen bir bloku izolasiya etmek ucundur. Mes. aciq olan bir programi yeniden acmagin qarsini bununla almaq olar Mes. teams.

Monitor.Enter(obj);

// bu arada her sey izolasiya olunub, basqa bir thread gire bilmez

Monitor.Exit(obj);

**Lock**

Ozunde monitor ve try finally bloku var

try

{

Monitor.Enter(obj);

Counter.Count++;

}

finally

{

Monitor.Exit(obj);

}

**Syntax**

lock (obj)

{

Counter.Count++;

}

**Mutex**

Ad qoymadiqda application seviyyesinde, qoyduqda ise emeliyyat system seviyyesinde isleyir.

Emeliyyat sistem seviyyesinde locka salmaq ucundur.

[Mutex App is only One Instance - Google Sənəd](https://docs.google.com/document/d/1P8cL8u6rlJ3ZQ-RhycZUkTmA8a6ZC4P4-2Jp_YTgm34/edit)

mutexObj.WaitOne(); // is bitene qeder hamini gozledir mutexObj.WaitOne(10) - burada vaxt da verib ne qeder vaxt diger threadin gozlemesini temin ede bilir. Hec ne yazmadiqda is bitene qederdir.

// bu arada her sey izolasiya olunub, basqa bir thread gire bilmez

mutexObj.ReleaseMutex(); - burada stop edir, girise icaze verir