Threading is a lightweight process.

Process is use operating system .

Multi Threading maintain a responsive user interface.

A Thread is a small set of executable instructions, which can be used to isolate a task from a process.

Threading helps us to Execute Programme code Paralley.

Type of Threading

1 Foreground threading - foreground threads exists /stayls even if main

2 Badcground threading -Back ground threads die off if main application exits (all thread excuted)

Thread \_ob=new Thread(sp);

\_ob.IsBackground =true;

\_ob.start();

AutoResetEvent -Class achieve synchronous by using signaled mathlogy.

Set every wait one

ex

Static AutoResetEvent \_ob= new AutoResetEvent(False);

s.v.m()

{

New Thread(sp).Start();

// signaled to start again

\_ob.Set();

}

Static void Sp()

{

Console.writeLine(“starting…”);

// wait & hault mode

\_ob.WaitOne();

Console.writeLine(“Finishing…”);

}

---------------------------------------------------------------------------------------------------

ManualResetEvent-All Thread running and not wait.

set one allow all Thead

Static ManualResetEvent \_ob= new ManualResetEvent (False);

========================================================

Thread.Sleep(100);

..........................

Thread thread1 = new Thread(new ThreadStart(DisplayThread1));

thread1.Start();

-----

Killing a Thread:

thread1.Abort();

delegate is type safe function pointer

Syntex

Thread t1 = new Thread(delegate() {sp.s();}):

t1.start();

SYNTEX = Lambda expression

Thread t1 = new Thread(()=> sp.s()):

t1.start();

==================