

Noter til "Threading in C#"

2. Concurrency – Hvad?

- Begivenheder, der sker samtidigt eller skifter opgaver hurtigt nok til at give en illusion af samtidighed.

3. Typer af Concurrency

- **Data Parallelism:** Samtidighed opnås ved at udføre den samme beregning på forskellige dataelementer samtidigt.
- **Functional Parallelism:** Samtidighed opnås ved at arbejde på uafhængige funktionelle opgaver samtidigt.

4. Fordele ved Concurrency

- Minimering af latenstid ved reaktion på hændelser
- Forhindring af langsom I/O i at blokere hovedtråden
- Maksimering af gennemløb på multikerne systemer

5. Multithreading Arkitektur i .NET

- Flere applikationsdomæner kan køre i en enkelt proces.
- Tråde kan tilhøre et enkelt applikationsdomæne, men kan ændre domæner over tid.

6. Grundlæggende om Tråde

- Navnerum: `System.Threading`
- Klasser og interfaces: `Mutex`, `Monitor`, `AutoResetEvent`, `Thread`, `ThreadPool`, `Timer`

7. Oprettelse og Start af Tråde

- Oprette og starte tråde ved at instantiere en `Thread`-klasse og kalde `Start` metoden.

8. Passende Parametre til Tråde

- Brug af egenskaber til at overføre parametre til tråde.

9. Styring af Trådens Livscyklus

- **Suspending, resuming and stopping threads:** Undgå brug af `Thread.Suspend` og `Thread.Resume`.

10. Foreground og Background Tråde

- Foreground tråde forhindrer applikationen i at afslutte, mens background tråde tillader applikationen at afslutte, selvom de stadig kører.

11. Tråd Pooling

- Brug af `ThreadPool.QueueUserWorkItem` til at administrere og genbruge tråde.

12. Synkronisering af Tråde

- Brug af låse og monitører for at sikre tråd-sikkerhed.