Begreber for realtids systemer:

Real-tids systemer er en blanding af Embedded Real-time systemer, Netværk embedded real-tid systemer og real-tid distrubreret systemer.

**Basis elementer af real tids-systemer :**

* Software 🡨🡪
* Hardware interface 🡨🡪
* Sensorer og aktuarer 🡨🡪
* Omgivelser/miljø

Basis enhed af RS er en task (opgave) og der kan være flere tasks i en system.

**En realtids system (RS) har udførsels deadline, når den er igang med at udregne og udfører handlinger.**

* Hard real time systemer
  + En system med meget tæt (kort tid) og svær deadlines
  + En misne deadline er en fejl
  + En system med mindst en hard deadline kaldes for en real tids system eller en sikkerheds-kritisk real tids computer
* Soft real tid systemer
  + Systemer med mere blød (mere tid) deadlines
  + En misne deadline kan være accepteret

Hertil kommer næste begreb, nemlig task prioriteret

* Højere prioriteret task ”stopper” lavere prioriteret task.

**En real tids system kan være:**

* Reactive or event driven (begivenhed drevet)
  + Er en system hvis adfærd drives af reaktioner til ekstern begivenheder
  + Event triggered (ET) tilgang
    - Alle kommunikation og processing aktiviteter er initialiseret hver gang en betydende ændring af state er noteret (eller en event)
    - Signalering af en event kan realiseres ved en interrupt mekanisme
* Tids drevet
  + En system hvis adfærd drives af tidens gang
  + Time triggerede (TT) tilgang
    - Alle kommunikation og processing aktiviteter er initialiseret ved en forudbestemt tidspunkt
    - Initialiseret af en periodisk clock interrupt
* Eller de kan være en kombination af de nævnte

**Real tids systemer skal reagerer på mange uafhængig og mulige sideløbende(samstemmende) begivenheder**

* En løsning til dette kaldes for concurrency (enighed, samvirke, sammentræf sammenfald)

**Definition på slack time:**

* Slack time can be defined as the amount of time an task can be delayed without causing another task to be delayed or impacting the completion date of your project.

**En robust system:**

* En system som bliver ved med at forsætte med gøre det rigtig ting, selvom systemet fejler i momentet
* En robust system skal:
  + Identificerer fejlen
  + Rette fejlen og forsætte med at processing
  + Gentag tidligere udregning/opgave til at genoprette den korrekte system state
  + Tilgå en fejl-sikkert state

**Hvad er en design pattern**

* En design pattern er en generaliseret løsning til almindelige opstående problemer
* Design er som optimering
* Den svær del er hvor der eksisterer så mange ting til at optimerer og hvor der kun er begrænset (kort) tid til rådighed.
* En design pattern har 4 afgørende elementer
* Patterens navn
* Problemet
  + Og dens kontekst
* Løsningen
  + Design patterens elementer (en løsnings skabelon)
* Konsekvenserne
  + Resultatet og trade-offs

**GoF (Gang of four, forfatter til bogen Design pattern book) design pattern kategorisering:**

* Creational patterns
* Structual patterns
  + Er involveret med hvordan klasserne og objekter er fattet til at forme større strukturer
* Behaivoral pattern
  + Er involveret med algoritmer og opgave af forpligtelser mellem objekter

**Andre typer af pattern kategorisering:**

* Arkitekturens patterns
  + Påvirker mest eller hele systemet
  + Er bredt og strategisk anvendt til systemet
  + Er diskturet og gennemgået i bogen ”BDPs real-tids design pattern”
    - De dækkende pattern er relevante to real-tids og indlejret systemer.
* Mekanistisk design pattern
  + GoF pattern med en mere lokal scope
  + De definerer mekanisme for sammenarbejde mellem objekter
  + Er dækket af bogen ”Design Patterns” book
    - All kind of system (OO)
* Idioms
  + Sproget (kode) afhængig design pattern

**Design optimerings kriterier:**

* Performance
* Minimerer krav ressourcer (hukommelse, heat)
* Reusability (Genbruges)
* Portability
* Maintainability
* Safety (sikkerhed)
* Reliability (pålidelighed)
* Security

**Definition på run-time :**

* Beskriver software/instruktioner, der udføres mens programmet kører, især instruktionerne som ikke er skrevet/defineret tydeligt, men er nødvendige for en korrekt udførelse af ens kode.

**Definition på concurrency :**

* Egenskab af systemer, hvor flere beregninger er udførende samtidigt og potentielt ineragerer med hinanden

**Definition på redundancy:**

* En system design hvor en komponent er dubleret, så hvis den fejler, så vil der eksisterer en backup 🡪 Øger reability