

Principper for Samtidigighed og Styresystemer

kursusintroduktion og Introduktion til Styresystemer

René Rydhof Hansen

Februar 2008

Kursusmål (studieordning)

Den studerende skal ved den afsluttende prøve kunne:

- *dokumentere* kendskab til og overblik over de berørte temaer og begreber inden for samtidighed og operativsystemer
- *benytte* korrekt fagterminologi og notation i såvel skrift som tale
- *dokumentere* forståelse for opbygning, strukturering, funktionalitet og virkemåde af operativsystemer
- *anvende* berørte emner til udvikling af systemnære simple programmer, der benytter sig af samtidighed og synkronisering

Hvorfor?

- Iflg. studieordningen:

“Alle interessante systemer anvender i en eller anden udstrækning parallelitet og samtidige processer.”

- Systemnær programmering ofte en forudsætning for indlejrede systemer
- Fundamentale begreber
- ... sjovt!

- Forelæsningsformat: 3x30min. med 10min. pauser
- Pensum [English], forelæsninger, slides, opgaver samt mini-projekt.
- Opgaveregning
 - Mestendels grupperum; af og til samlet
 - Minimér spildtid: brug hjælpelærer!
- Undervisere
 - René Rydhof Hansen.
 - Email: rrh@cs.aau.dk
 - Kontor: 1.2.36
 - Morten Kühnrich (hjælpelærer).
 - Email: mokyhn@cs.aau.dk
 - Kontor: 1.2.59

[English] John English “Introduction to Operating Systems — Behind the Desktop”. **Primær** litteratur (obligatorisk).

[Love] Robert Love “Linux Kernel Development”. Sekundær litteratur (valgfri).

Samt eventuelle relevante artikler og lign.

- Individuel besvarelse
- To emner (foreløbige)
 - Hukommelsesallokering
 - Disk-schedulering
- Udvikles i C eller Java
- Aflevering af *kort* rapport
- Tre forelæsningsgange reserveres til miniprojektet
- Udgangspunkt for eksamen

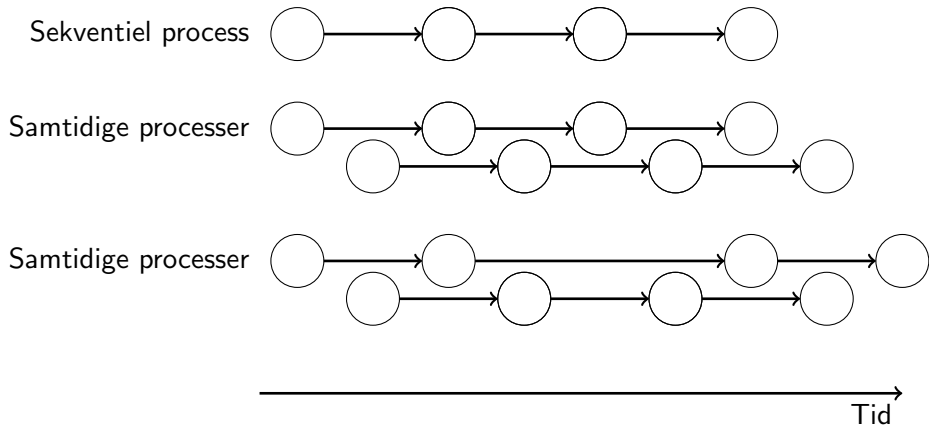
- Mundtlig eksamen med forberedelse
- Ekstern censur
- 7-trinsskalaen
- Hovedspørgsmål (kendt) og sidespørgsmål (ukendt(e))
- Udgangspunkt i miniprojekt

Resten af kurset (foreløbig plan)

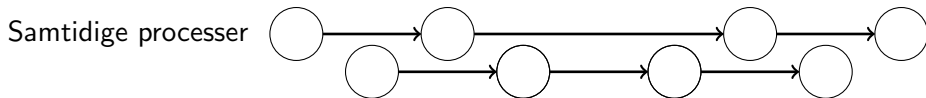
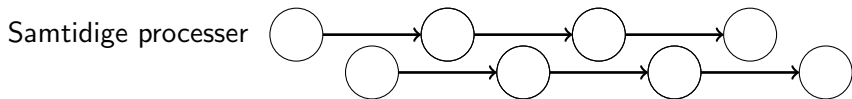
- Filsystemer
- Processer og tråde
- Samtidighed, deadlocks og synkronisering
- Virtuelt og udvidet lager
- Kernen
- Device drivere og I/O
- Virtualisering
- Scheduling i Linux kernen (gæsteforelæsning)

Hvad er samtidighed?

"... systemer karakteriseret ved flere samarbejdende processer"



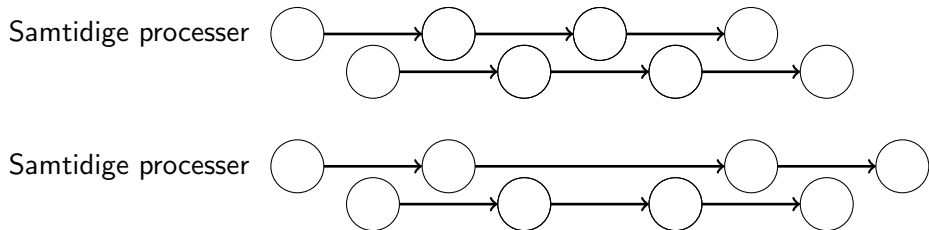
Hvad er samtidighed? ... problematisk



- Den præcise rækkefølge og timing for processernes samlede udførelse er **uforudsigelig**
- Potentielt problem: hvis de enkelte processers resultat (utilsigtet) er afhængigt af andre processers udførelse: **race condition**
- Eksempel: filsystem
- Eksempel: TOCTTOU

Race conditions og synkronisering

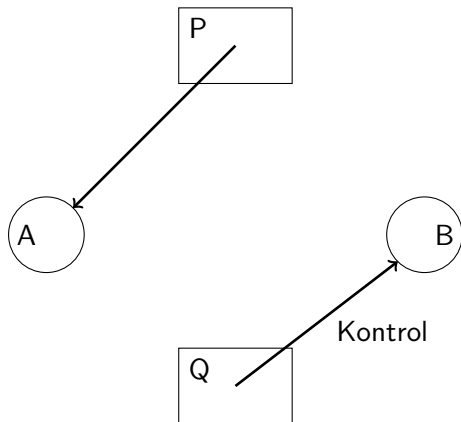
Race conditions opstår som følge af **ukontrolleret** samtidig tilgang til delte ressourcer.



Løsning: **synkronisering**

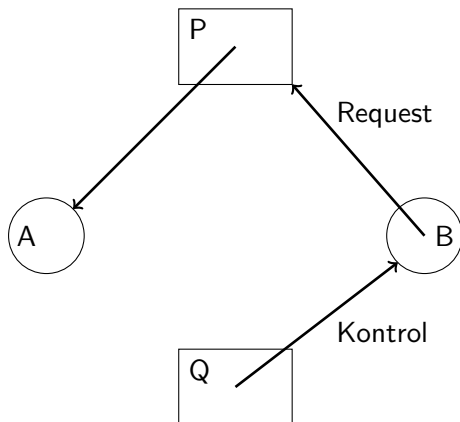
Synkronisering

Synkronisering kræver **kontrol** over ressourcer:



Synkronisering

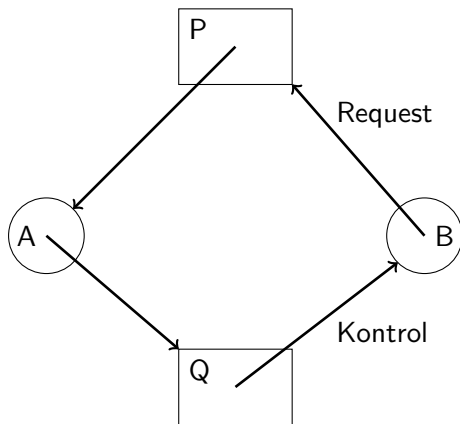
Synkronisering kræver **kontrol** over ressourcer:



Kontrol opnås
gennem **requests**

Synkronisering

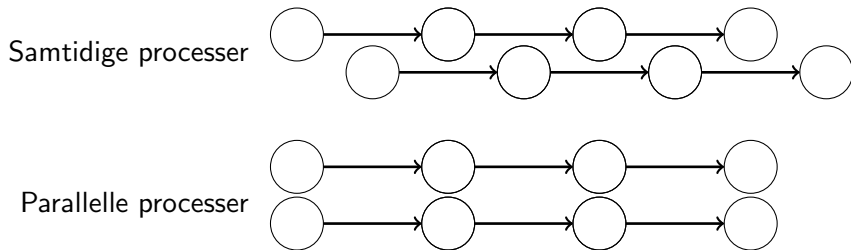
Synkronisering kræver **kontrol** over ressourcer:



Kontrol opnås
gennem **requests**

Cykliske requests fører til **deadlock**

Samtidig vs. parallelitet



Samtidig \subset Parallel

- MVP **Parallel** udførelse kræver hardware-understøttelse, e.g., multi-core processorer
- PSS **Samtidig** udførelse kan implementeres i software, e.g., virtualiseringssoftware, styresystem

Eksempler på samtidighed

- Parallel beregning
- Parallel I/O
- Kontrolsystemer i f.eks. biler og fly
- En CPU (multi-core)
- En computer (CPU vs. GPU)
- Computernetværk
- Kommunikationsnetværk
- **Multitasking**
- Virkeligheden

Multitasking

- Et styresystemdesign der tillader flere **processer** at blive udført på en (eller flere) delte processorer
- En process kan opfattes som en “virtuel maskine” der udfører et program
- Styresystemet skifter mellem aktive processer
 - (Multiprogramming; vent på I/O)
 - Cooperative
 - Preemptive

Hvorfor multitasking?

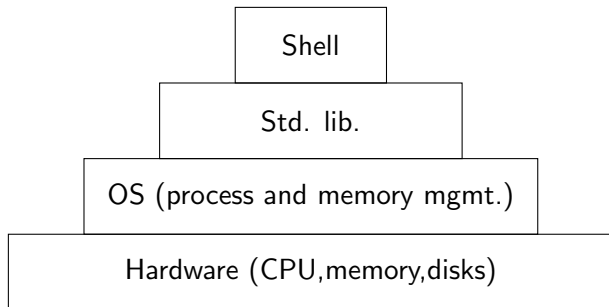
- Batch-systemer
 - Ressource-maksimering
 - Throughput-maksimering
 - Skift udløses af I/O
- Time-sharing
 - Interaktiv brug
 - Illusion af parallelle processer
 - Skift udløses af timer
- Real-time
 - Reagere på eksterne hændelser
 - Real-time krav (maksimal absolut ventetid)

Hvad er et styresystem?

- Hvad styres?
- Hvad er en computer? Hvad er en virtuel computer?
- Er styresystemet hardware eller software?
- Er det et program?
- Er en “shell” en del af styresystemet?
- Er en GUI en del af styresystemet?
- Design
 - Monolitisk vs. mini-kerne
 - Torvalds vs. Tanenbaum

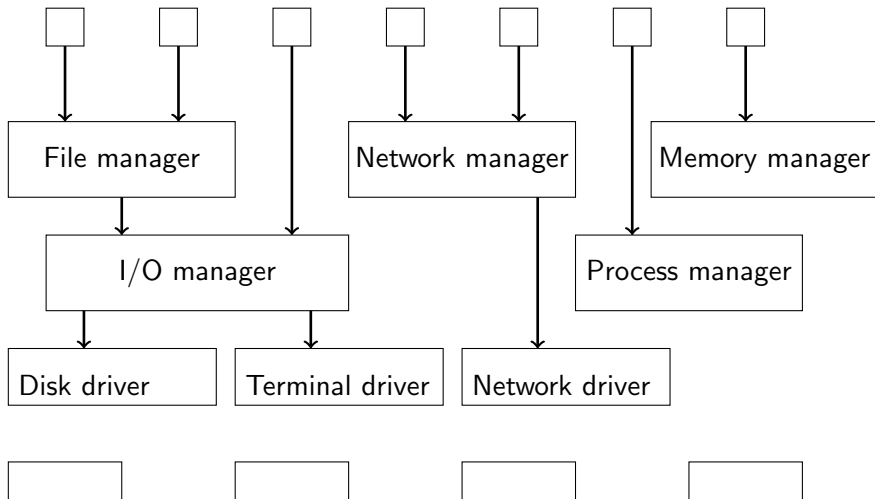
Hvad er et styresystem? En abstraktion

- Administration af ressourcer
 - Processor
 - Hukommelse
 - Navne
 - ...
- Koordinering
 - Hændelser
 - Aktiviteter



Ressourcemanager

System service interface



Hardware interface

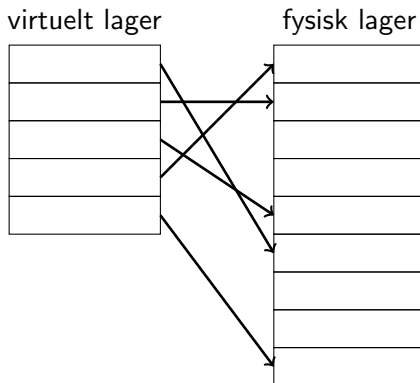
Ressource: Filsystemer

- Allokering
- Navnerum
- Pålidelighed
- Sikkerhed
- Hastighed

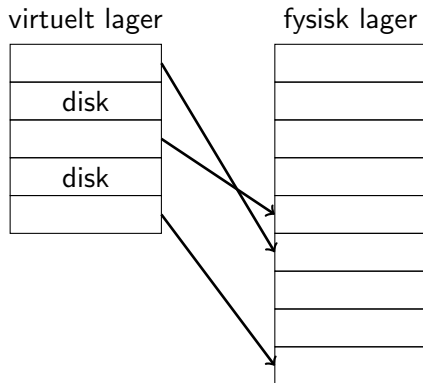
- Programmer
 - Instruktionssekvens
 - Oftest en fil
 - Statisk
- Process
 - Et kørende program
 - En sekvens af tilstande
 - Dynamisk
 - "Virtuel maskine"
 - Isoleret
- Processer består af
 - Programteksten: maskinkode
 - Data: stak og heap
 - Kontekst: cpu-registre etc.

Virtuelt Lager (paging)

- Styresystemet virtualiserer det fysiske lager



Virtuelt Lager (demand paging)



Opsummering og næste gang

- Introduktion til samtidighed
- Introduktion til styresystemer
- Næste gang: Filsystemer