Principper for Samtidighed og Styresystemer

End of Message

René Rydhof Hansen

Maj 2008

Sjov og ballade

- Ekstraforelæsning: tirsdag den 20. maj kl. 10–12 i auditorium 0.1.95 (formentlig).
- Spørgetime: tirsdag den 17. juni kl. 13–15 i auditorium 0.1.95.
- Alternativ spørgetime: den ??. juni

I dag

- Kursusopsummering
- Eksamen
- Mini-Projekter

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!

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
  - Hvad og hvordan er et filsystem en abstraktion
  - Grundlæggende organisationsprincipper for filsystemer
  - Hard links vs. symbolic links
  - Virtuelle filsystemer
  - Grundlæggende egenskaber ved og operationer på filsystemer
  - Allokeringsstrategier for diskblokke
  - API/interface/systemkald for filsystemer
  - Implementation af kataloger
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
  - Definition
  - Grundlæggende træk ved procesoprettelse under UNIX
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
  - Definition og brug
  - Metoder til opnåelse af gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
  - Relativ tid
  - Synkronisering
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
  - Deadlock-begrebet
  - Løsningsstrategier og algoritmer
  - Priority inversion: problem og løsninger
  - Livelock og starvation (husk: Sultne 'soffer)
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
  - Lagerhierarkiet og lokalitet
  - Paging (og segmentering)
  - Organisationsprincipper for sidetabeller
  - Swapping, demand paging
  - Sidefejl og sideerstatningsalgoritmer
    - OPT, LRU, FIFO, Clock
  - Rammeallokering
  - Working set modellen
  - Thrashing
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
  - Instruction-fetch løkken
  - MMU
  - Caching
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
  - User- og kernel-mode
  - Exception (fault, traps, interrupts)
  - I/O (specielt: interruptbaseret I/O)
  - Håndtering af systemkald
  - Proces- og tråd-realisering
  - Schedulering
- Device drivere, Coccinelle

Kursusindhold

- Filer og Filsystemer
- Processer og tråde
- Gensidig udelukkelse
- Samtidighed
- Deadlocks og deadlockhåndtering
- Hukommelsesstyring
- Hardwareunderstøttelse
- Styresystemkernen og Interrupts
- Device drivere, Coccinelle
  - Definition og brug
  - Grænseflader
  - Typer
  - Implementation (Linux)
  - I/O Schedulering (FIFO, CBF, SCAN, CSCAN)

Evaluering

- Masser af begreber og koncepter
- Praktisk erfaring kun i opgaver/mini-projekter
- Overfladisk (primær) litteratur

Eksamen

Datoer

Fra og med mandag 23. juni til og med torsdag 26. juni.

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

Miniprojekt

Formål

At designe og implementere et kørende flertrådet program med ikke-triviel synkronisering.

- Afleveringsdato: formentlig fredag den 30. maj 2008.
- Individuel besvarelse
- Tre emner (detaljer på torsdag)
  - Kalender
  - Hukommelsesallokering
  - Disk-scheduling
- "Blød" kravspecifikation
- Udvikles i C, C# eller Java
- Aflevering af *kort* rapport (to eksemplarer)
- Tre forelæsningsgange reserveres til miniprojektet
- Udgangspunkt for eksamen

Miniprojekt 1: Kalender

- Et antal brugere kan tilgå een fælleskalender
- Kalenderen indeholder slots
  - Tekst
  - BrugerID
  - Status
- Aftaler kan spænde over et eller flere slots.
- Krav
  - Alle brugere kan lave aftaler
  - Den opdaterede kalender skal vises for ejeren kontinuerligt
  - Samtidig læsning skal være muligt
  - Ingen busy waiting tilladt
- Argumenter for at deadlock er umuligt.
- Overvej om udsultning er muligt og argumenter for svaret.

Miniprojekt 2: Disk-schedulering

- Et antal brugere der vil læse/skrive til en disk
- En scheduler der kontrollerer adgangen til disken
  - Skal minimere adgangstid
  - Udsultning skal undgås
  - Brug SCAN algoritmen
- Disken skal modelleres (som en selvstændig process)
- Der skal være mutex på delte variable
- Et request blokerer den kaldende process
- En proces bevarer retten til et spor indtil sporet frigives
- Scheduleren blokerer hvis der ikke er requests
- Shceduleren blokerer imens disken bevæger armen
- Der skal argumenteres for at deadlocks og udsultning ikke kan forekomme

Miniprojekt 3: Hukommelsesallokering

- Et antal processer allokerer, reallokerer og deallokerer hukommelse.
- Allokatoren skal blokere kaldende processer såfremt der ikke er hukommelse nok eller ved fare for deadlock.
- Når hukommelse frigives skal evt. blokerede processer vækkes.
- Der skal indgå deadlock-håndtering, f.eks. Banker's algorithm eller "detection and recovery".
- Overvej om der er mulighed for udsultning og argumenter for svaret.
- Overvej om systemet er fair og argumenter for svaret.

God læselyst!