

SIF8041 OPERATIVSYSTEMER OG DATABASES

våren 2001

ØVING 1 : Prosesser og CPU-tildeling (OS)

INNLEVERINGSFRIST : Onsdag 07.02.00

GRUPPESTØRRELSE : 2 -3. Lever svarene på papir (ikke håndskrevet) i boks 2 eta. E-blokka.

1. Diverse

- (a) Forklar kort (2-3 setninger) med egen ord hva "polling" og "avbrudd" ("interrupts") med hensyn på I/O i et datasystem er? Sammenlign disse to.
- (b) Hva er forskjellen mellom et "exception" og et "avbrudd" ("interrupt")?
Avbruddsprosessering krever en spesiell stakk ("kernel stack") hvorpå den aktuelle (current) PC'en vil bli lagret. Bruker også "exception handling" denne stakken? Forklar svaret.
- (c) x86 har en instruksjon kalt "halt" som hovedsakelig gjør at CPU'en ikke bruker tid på noen andre aktiviteter mens "halt"-instruksjonen utføres. Kan det tenkes at en "stygg" prosess kan ta over kontrollen (d.v.s. at OS mister kontrollen) av systemet ved å utføre en slik "halt" instruksjon? Forklar.

2. Tråd (thread)

Oppgave 4.5 i læreboka.

Hvilke ressurser brukes når en tråd (thread) opprettes? Hva er forskjellen i forhold til de ressursene som brukes når en prosess opprettes?

3. Kontekst-svitsj

Oppgave 4.6 i læreboka.

Beskriv hva kjernen (kernel) gjør ved kontekst-svitsjing:

- (a) mellom tråder.
- (b) mellom prosesser.

4. Tidsperspektiv

Vi kan ha CPU-tildeling med ulike tidsperspektiv: kortsiktig og langsiktig.

(a) Hva er forskjellen på disse ?

(b) Noen mulige algoritmer for å dele prosessortid mellom flere prosesser:

- First-Come-First-Served (FCFS).
- Round-robin (RR).
- Shortest-Job-First (SJF).
- Highest-Priority-First (HPF).
- Longest-Job-First (LJF).
- Last-Come-First-Served (LCFS).

Hvilken av disse algoritmene er det hensiktsmessig å benytte til henholdsvis kortsiktig og langsiktig CPU-tildeling? *Er det noen av algoritmene som man neppe vil benytte?*

5. Sammenlikning av ulike algoritmer

Anta en en-prosessor maskin som skal kjøre følgende jobber:

Jobb	Totaltid	Prioritet
1	10	3
2	1	1
3	2	3
4	1	4
5	5	2

Jobbene er ankommet i rekkefølge 1, 2, 3, 4 og 5. Totaltid angir prosessortid for hver jobb hvis uavbrutt kjøring.

- (a) Tegn et diagram som viser utførelsen av disse jobbene med henholdsvis FCFS, RR (med tidskvant lik 1), SJF og en ikke-avbrytbar (non-preemptive) prioritets-tildeling. Merk at i det siste tilfellet starter utførelsen av første jobb etter at jobb 5 er ankommet slik at *alle* tas i betraktning.
- (b) Hva blir omløpstiden for hver jobb med de ulike algoritmene ?
- (c) Hva blir ventetiden for hver jobb med de ulike algoritmene ?
- (d) Hvilken algoritme gir minimal midlere ventetid ?

- (e) I disse beregningene er "overhead" som skyldes prosess-skifte ignorert. Hvordan vil slik kontekst-svitsjing påvirke resultatet ? Er det likegyldig hvilket tidskvant vi velger for round-robin fordeling?

6. Oppgave 5.4 i læreboka

Løs oppg. 5.4 a), b) og c) i læreboka. (s.151)

7. Flernivåkøer

- (a) Hvorfor kan det være nyttig å benytte flere fordelingsalgoritmer i samme system ?
- (b) Vis hvordan ulike jobbtyper kan grupperes med tanke på CPU tildeling.
- (c) Vis hvordan en statisk gruppering som i (b) kan gjøres mer effektiv.