

Institutt for datateknikk og informasjonsvitenskap

Løsningsforslag til

## **Kontinuasjoneksamensoppgave i TDT4186 Operativsystemer**

**Faglig kontakt under eksamen:**

Svein Erik Bratsberg: 9953 9963

**Eksamensdato: 17. august 2013**

**Eksamenstid (fra-til): 09:00 - 13:00**

**Hjelpemiddelkode/Tillatte hjelpemidler:**

D – Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

**Annen informasjon:**

**Målform/språk: Norsk bokmål**

**Antall sider: 3**

**Antall sider vedlegg: 0**

**Kontrollert av:**

---

Dato

Sign.

## Oppgave 1 – Systemkall – 5 %

For selve programutførelseslogikken betyr det ingenting om bibliotekskallet resulterer i et systemkall. Men for ytelsen er det en fordel om det ikke resulterer i et systemkall. Systemkall gjør at maskinen må skifte til kjernekontekst. I tillegg kan det hende at vi får en reschedulering (av en annen prosess) når det skjer et systemkall.

## Oppgave 2 – Tråder, synkronisering og tidsdeling – 20 %

a) I den venstre figuren viser brukernivåtråder. Til høyre vises kjerne-tråder. Forskjellen er at kjernen må vite om trådene i det siste tilfelle.

b) **Fordeler:** Kan håndtere blokkerende systemkall (kan da schedulere en annen tråd i samme prosess). Den kan også håndtere sidefeil i en tråd. Tillater parallellitet i samme prosess.

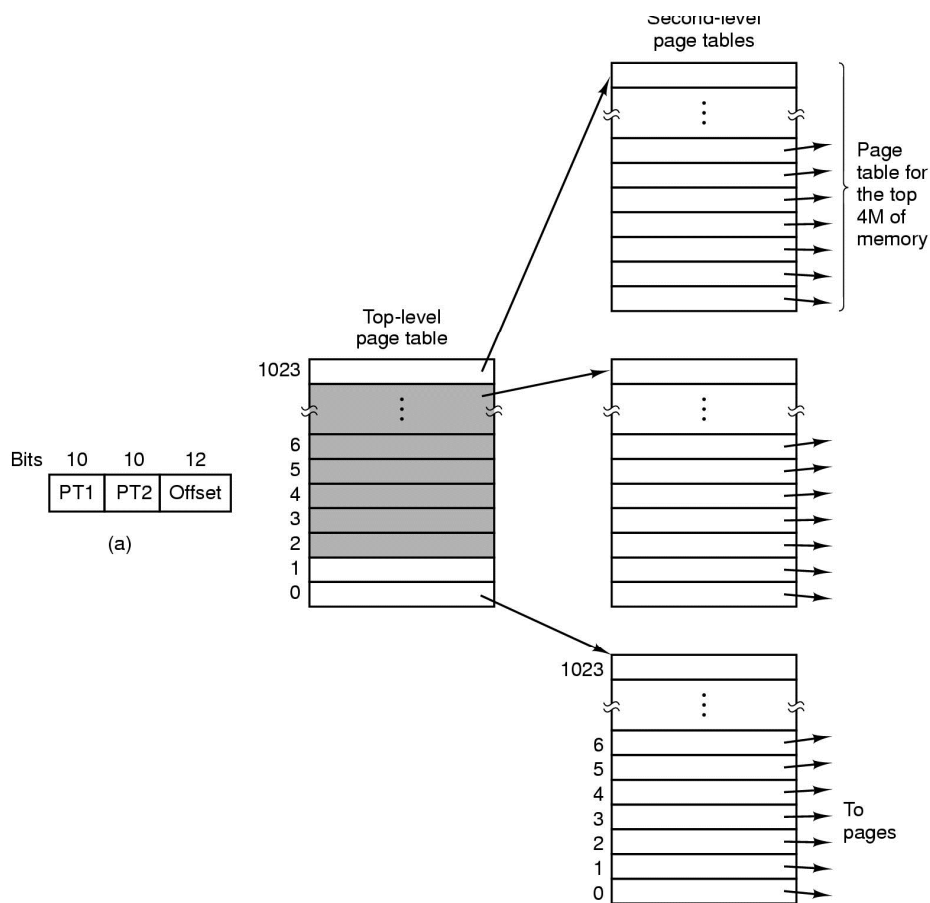
**Ulemper:** Man må inn i kjernemodus for trådskifte (dyrt). Tidsdeling (scheduling) er bestemt av OSet. Ikke noe applikasjonsbestemt tidsdeling.

c) Gjennomstrømmingen ved round-robin-tidsdelingen styres av overheadet ved prosesskifte. Et stort tidskvantum gir få prosesskifter og da lite overhead. Hvis du ikke har overhead, så har tidskvantumet ingenting å si.

## Oppgave 3 – Synkronisering – 10 %

Petersons løsning er en programvarebasert løsning for å beskytte en kritisk region. Det er to prosedyrer `enter_region` og `leave_region` som brukes for å gå inn i og ut av kritisk region. Det brukes to variabler og en aktiv venteløkke for å test på om man komme inn. Se side 121 i læreboka.

## Oppgave 4 – Minnehåndtering– 15 %



a)

- i) NRU: 2
- ii) FIFO: 2
- iii) LRU: 1
- iv) Second chance: 2

b) Hver sidereferanse vil i utgangspunktet endre LRUen. Det å endre LRUen vil forårsake mange sidereferanser ingen. Rekursivt problem og dyrt.

## Oppgave 5 – Filsystemer – 15 %

- a) Fordelen med liten blokkstørrelse er at man kaster bort lite plass når filene er korte. Ulempen er at throughput'en mot filsystemet blir liten.
- b) Man må to forskjellige strukturer. En hashinngang på blokkid og en liste for LRU..

### **Oppgave 6 – Vranglåser – 15 %**

- a) Logiske objekter som ikke kan brukes av mange prosesser/tråder samtidig, f.eks. låser, kan forårsake vranglåser.
- b) Man kan bruke algoritmen beskrevet på side 441 i læreboka.

### **Oppgave 7 – Sikkerhet – 10 %**

Companion virus, executable program virus, memory resident virus, boot sector virus, device driver virus, macro virus, source code virus er omtalt på side 670-680 i læreboka.

### **Oppgave 8 – Linux – 10 %**

Se side 754-767 i læreboka. Litt om organiseringen av adresserommet i hver prosess og paging bør være med.