CHALMERS LINDHOLMEN

Institutionen för data- och elektroteknik

TENTAMEN

KURS namn	**Parallellprogrammering**
PROGRAM namn, **åk, läsperiod**	**Dataingenjör, 120 p** **3, Lp 2**
KURS beteckning	**LEU 290 0199**
EXAMINATOR	**Alois Goller**
TID FÖR TENTAMEN	**Måndagen den 11 december 2000, kl. 13:30 - 17:30**
HJÄLPMEDEL	**inga**
ANSV LÄRARE: namn **telnr** **besöker tentamen kl**	**Alois Goller** **772 5716** **14:30**
RESULTAT + TID/PLATS FÖR GRANSKNING anslås (datum)	**senast 2001-01-15**
ÖVRIG INFORMATION	**6 uppgifter** **3 sidor**

NAMN (tentand):_____

Uppgift 1: Vad menar man när man säger att en sekvens av satser är en kritisk region? Visa med ett exempel faran i att låta två processer få tillgång till samma kritiska region samtidigt. (4p)

Uppgift 2: Förklara "Manager-Worker" kommunikationsmodellen. Beskriv fördelar och nackdelar samt användningsområden. (4p)

Skriv ett Manager-Worker program is pseudo-code som beräknar

$$\sum_{i=1}^{n} i^2$$

(8p)

Uppgift 3: Betrakta de fyra `select`-satserna:

(a)

```
select
  accept A;
or
  delay 4.0;
end select;
```

(b)

```
select
  accept A;
else
  delay 4.0;
end select;
```

(c)

```
select
  accept A;
or
  delay 2.0;
  delay 2.0;
end select;
```

(d)

```
select
  accept A;
  delay 2.0;
else
  delay 2.0;
end select;
```

Beskriv skillnaderna mellan dessa, i de fall det finns några. (4p)

Uppgift 4: Ge en kort förklaring av var och en av följande termer:

(a) livelock
(b) liveness
(c) svält
(d) deadlock
(e) safety

(10p)

Uppgift 5: Betrakta nedanstående skyddade objekt:

```
protected Barrier is
   entry Wait;
   procedure Open;
private
   entry Entrance;
   Opened : Boolean := False;
end Barrier;

protected body Barrier is
   entry Wait when Entrance'Count<2 and not Opened is
   begin
      requeue Entrance;
   end Wait;
   entry Entrance when Opened is
   begin
      Opened := Entrance'Count /= 0;
   end Entrance;

   procedure Open is
   begin
      Opened := Entrance'Count /= 0;
   end Open;

end Barrier;
```

(a) Beskriv i detalj vad som händer när följande sekvens av händelser inträffar.

```
Process A:  Barrier.Wait;
Process B:  Barrier.Wait;
Process C:  Barrier.Wait;
Process D:  Barrier.Open;
```

De fyra händelserna kommer med så långt mellanrum att allting som skulle kunna inträffa som följd av en av händelserna inträffar innan nästa anrop av det skyddade objektet. Antag att det skyddade objektet inte har använts tidigare i programmet. (6p)

(b) Skriv om det skyddade objektet utan **requeue**. Det nya skyddade objektet skall ha samma funktionalitet som `Barrier`. (8p)

Uppgift 6: En speciell variant av semforer kallas *kvantitativa* semaforer. Skillnaden mellan dessa och vanliga semaforer är att `Wait` och `Signal` tar ett positivt tal som parameter, och semaforen ökas respektive minskas med detta antal enligt följande: Låt semaforen ha värdet S. Om `Wait(N)` anropas:

- Om `N` $\leq S$, minska S med `N`.
- Annars, blockera tasket.

Då `Signal(N)` anropas ökas S med `N`. Om något task är blockerat på semaforen med en parameter mindre än eller lika med semaforens nya värde, väck det första task vars parameter är mindre än eller lika med S, och låt det exekvera `Wait`.

(a) Skriv ett skyddat objekt eller ett task som implementerar kvantitativa semaforer. (10p)

(b) Vilken ködisciplin använder din implementation? (4p)

(c) Kan task utsättas för svält, och i så fall hur? (2p)