

Tentamen A2 Bedrijfssystemen, 24-8-99

Geef bij al je antwoorden voldoende motivatie: voor alleen 'ja' of 'nee' als antwoord krijg je geen punten.

Vraag 2

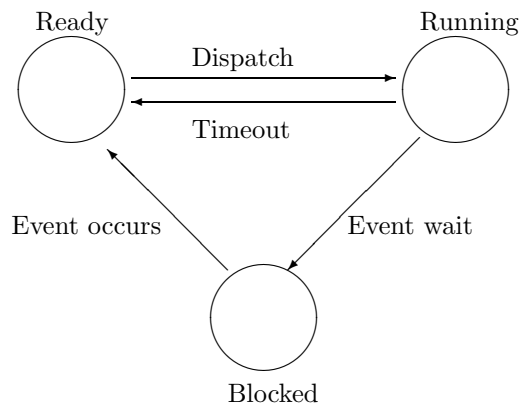
Op een operating systeem met pre-emptive round-robin scheduling draaien 3 processen, die elke uit threads bestaan:

process	threads waaruit dit process bestaat
P_1	T_{11}, T_{12}, T_{13}
P_2	T_{21}, T_{22}
P_3	T_{31}

1. Stel thread T_{11} draait en het quantum loopt af. Het operating system moet dus een nieuwe thread kiezen om te laten draaien. Welke threads zouden er nu direct na T_{11} aan de beurt kunnen komen, als de threads user level threads zijn? En welke als het kernel level threads zijn? Geef een korte motivatie bij je antwoorden.
2. Stel dat T_{11} draait en dat deze thread blokkeert. Welke threads kunnen er nu direct na afloop van T_{11} aan de beurt komen, als de threads user level threads zijn? En welke als het kernel level threads zijn? Geef een korte motivatie bij je antwoorden.

Vraag 3

Het plaatje hieronder toont het toestandsdiagram voor processen van een of ander operating system. Het operating system draait enkel op single processor machines.



We willen dit operating systeem uitbreiden met twee operaties: *laat_slapen* en *maak_wakker*. Met de eerste operatie kan een proces de executie van een *ander*

proces (tijdelijk) onderbreken. Dit andere proces zal dan niet meer draaien – dwz. geen CPU tijd meer krijgen – tot het weer wordt "wakker gemaakt" door een of ander proces.

1. Voeg toestanden en/of toestandsovergangen toe aan het bovenstaande diagram om deze operaties mogelijk te maken. Zeg voor nieuwe toestanden kort waarvoor ze dienen, en voor nieuwe toestandsovergangen waardoor deze veroorzaakt kunnen worden.
2. Als het operating system ook op multi-processor machines moet kunnen draaien, maakt dit wat betreft dit toestandsdiagram enig verschil?

Vraag 5

Verschillende processen, die van type `lezer` of `schrijver` kunnen zijn, delen een globale lijst met data. Processen van type `schrijver` produceren data en zetten dit achteraan in een globale lijst, processen van type `lezer` halen data uit het begin van deze globale lijst en verwerken het.

Voor de globale lijst worden twee pointers bijgehouden, `first` en `last`, die naar het eerste en het laatste element in de rij wijzen. Voor elke pointer `p` in de lijst is `p->inhoud` een of ander data element en `p->next` de pointer naar het volgende element in de lijst. De (pseudo)code voor een `lezer` is

```
while TRUE
{ y = first->inhoud;
  "verwerk data die in de lokale variabele y staat";
  first = first->next;
}
```

en die voor een `schrijver` is

```
while TRUE
{ "produceer data in een lokale variabele x";
  alloc(new_last);
  /* Er is nu geheugen gealloceerd voor een nieuw element in */
  /* de lijst, en new_last wijst naar dit verse stuk geheugen */
  last->next = new_last;
  last = new_last;
  last->inhoud = x;
}
```

We maken ons niet druk over het dealloceren van geheugen.

1. Er kan het een en ander misgaan met deze code. Voeg semafoor-operaties toe om er voor te zorgen dat dit niet meer kan gebeuren.
Zeg duidelijk welke semaforen je gebruikt en wat hun initiële waarden zijn. Je mag semafoor-operaties in de bovenstaande code toevoegen, maar de volgorde van de statements die al in de code staan mag niet veranderen.
2. Zowel schrijvers als lezers moeten in ieder slag van hun repetitie totaal r seconden rekenen. Tijdens het produceren van de data is een schrijver gemiddeld b seconden geblokkeerd (bijv. omdat-ie invoer moet doen),

en tijdens het verwerken van de data is een lezer gemiddeld b seconden geblokkeerd (bijv. omdat-ie uitvoer moet doen). Je mag aannemen dat b groter is dan r .

Geef een zo goed mogelijk ondergrens voor de totale tijd die nodig is voor het produceren en verwerken van 100 data-elementen, als er 1 lezer en 1 schrijver draaien op een single processor machine.

Geef ook een zo goed mogelijk ondergrens voor deze tijd als er 1 lezer en 1 schrijver draaien op een machine met 2 processoren.

Motiveer deze antwoorden!

3. Kun je de code van lezers of schrijvers veranderen om meer concurrency mogelijk te maken? Zoja, leg uit hoe.