

TI-2720 Operating System Concepten

21 januari 2013, 14.00 - 17.00 uur.

docent: H.J. Sips

Dit is een tentamen met **9** open vragen

Opgave	Punten
1	9
2	12
3	12
4	12
5	6
6	12
7	12
8	12
9	12
kado	1
Totaal	100

Het gebruik van boeken, dictaten of aantekeningen is **niet** toegestaan

Schrijf **duidelijk** en formuleer **kort en bondig**

Opgave 1 (9 punten)

Veel systemen kennen een user mode en een kernel mode.

- Leg uit wat moet worden verstaan onder user mode en wat onder kernel mode en leg uit waarom het nuttig is deze twee modes te hebben.
- In welke twee situaties gaat een systeem naar kernel mode? Licht uw antwoord kort toe.
- Stel we hebben een systeem dat in principe user mode en kernel mode kent. In een poging de performance te verbeteren heeft een systeemprogrammeur het Operating System zodanig gewijzigd dat er alleen nog maar in kernel mode wordt gedraaid. In hoeverre is dit een bruikbaar systeem?

Opgave 2 (12 punten)

- a. Leg kort uit wat er moet worden verstaan onder een thread.
- b. Threads kunnen binnen een systeem worden ondersteund door de kernel ("kernel threads") of door libraries ("user threads").
Noem het belangrijkste voordeel van user threads. Geef een korte toelichting
- c. Noem het belangrijkste nadeel van user threads. Geef weer een korte toelichting.

Opgave 3 (12 punten)

Er bestaan verschillende policies voor CPU Scheduling, men maakt onder andere onderscheid tussen preemptive en nonpreemptive scheduling policies. Bij sommige policies is starvation mogelijk bij andere niet.

- a. Wat wordt er verstaan onder een preemptive scheduling policy?
- b. Wat wordt er verstaan onder starvation?
- c. Geef voor ieder van de volgende scheduling policies aan of deze preemptive of nonpreemptive is, en of er wel of geen starvation kan optreden. Geef steeds een korte motivering over wel of geen starvation
 - c1. First Come First served
 - c2. Shortest Job First
 - c3. Priority Scheduling
 - c4. Round Robin
- d. Als bij een bepaalde policy in principe starvation mogelijk is, wat kan men dan doen om starvation te verhinderen? Motiveer uw antwoord.

Opgave 4 (12 punten)

Het Readers en Writers probleem betreft het regelen van de toegang van Readers en Writers tot een object, waarbij meerdere Readers tegelijk zijn toegestaan, maar waarbij een Writer exclusief toegang moet hebben. Bij het probleem dat bekend staat als het eerste RW-probleem, krijgt een Reader altijd toegang tenzij een Writer bezig is. Een Writer moet wachten tot er geen Reader meer bezig is.

Een oplossing voor het eerste RW-probleem, maakt gebruik van de volgende shared variabelen.
semafoor mutex initieel (1), semafoor wrt initieel(1), integer readcount initieel(0)

De code voor het Reader proces kan als volgt worden weergegeven:

```
[1]  wait(mutex);
[2]  readcount = readcount + 1;
[3]  if (readcount == 1) wait(wrt); /*eerste lezer wacht tot geen schrijver*/
[4]  signal(mutex);
[5]                                     /* hier wordt gelezen */
[6]  wait(mutex);
[7]  readcount = readcount - 1;
[8]  if (readcount == 0) signal(wrt); /* laatste lezer meldt zijn vertrek */
[9]  signal(mutex);
```

De code voor de Writer kan er dan als volgt uitzien:

```
[10] wait(wrt);
[11]                                     /* hier wordt geschreven */
[12] signal(wrt);
```

beantwoord nu de volgende vragen.

- a. Beschrijf zo precies mogelijk wat in de code van de Reader het doel is van het gebruik van de semafoor mutex?
- b. Gaat er iets fout wanneer in de Reader code regels [4] en [6] worden verwijderd?
Zo ja, wat? Geef een korte toelichting.
zo nee, waarom zijn ze dan aanwezig? geef een korte toelichting.
- c. Mogen in de Reader code [7] en [8] verwisseld worden, als de conditie in de oorspronkelijke regel [8] wordt vervangen door (readcount == 1) ?
Geef een korte toelichting.
- d. Wat gebeurt er als er een Writer bezig is en zowel een Reader (regel [3]) als een Writer (regel [10]) willen beide toegang. Geef een korte toelichting.

Opgave 5 (6 punten)

Gegeven een systeem waarin één instance van het resourcetype X aanwezig is en $2N$ instances van het resourcetype Y. Hierbij is N het maximale aantal processen dat tegelijk in het systeem aanwezig kan zijn. Ieder proces heeft maximaal één resource van het type X nodig en maximaal twee resources van het type Y. Bij de start van een proces is nog niet bekend welke resources het proces nodig zal hebben en ook geen eventuele volgorde. Een proces vraagt een resource pas aan op het moment dat hij deze nodig heeft.

- a. Is in de hier beschreven situatie het gevaar voor deadlock aanwezig? Motiveer uw antwoord.
- b. Is in de hier beschreven situatie het gevaar van starvation aanwezig? Motiveer uw antwoord.

Opgave 6 (12 punten)

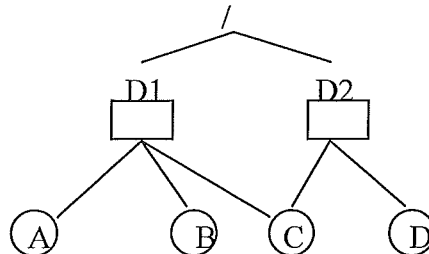
Voor het bestuderen van enkele page-replacement algoritmen gebruiken we in deze opgave de volgende Page Reference String : 0, 1, 0, 2, 0, 3, 0, 1, 2, 0, 3, 2, 0, 3

We nemen aan dat er voor het betreffende proces 3 frames (0, 1, 2) beschikbaar zijn.

- a. Geef voor de volgende page-replacement strategieën algoritmen stap voor stap aan welke pages zich in welke frames bevinden en bepaal voor ieder van deze algoritmen het aantal pagefaults.
 - 1) FIFO
 - 2) Optimal (Min)
 - 3) LRU
- b. Kloppen de resultaten gevonden onder a. met hetgeen op grond van de theorie verwacht mocht worden? Licht uw antwoord kort toe.
- c. Voor het aantal frames dat een proces ter beschikking heet kunnen we uitgaan van een statische verdeling, of we kunnen een dynamische policy toepassen. In geval van een dynamische policy kunnen we bijvoorbeeld de working-set policy gebruiken. Leg uit hoe deze werkt. Geef hierbij ondermeer een precieze beschrijving van het begrip Working Set.

Opgave 7 (12 punten)

Veel filesystemen kennen mogelijkheden om files via verschillende paden door de directory te benaderen. Hieronder is een directory weergegeven, waarin een file C benaderbaar is via D1 en via D2, m.a.w. C kan worden aangeduid zowel door /D1/C als door /D2/C.



- Stel een (bevoegde) gebruiker geeft de opdracht voor het opruimen van C door middel van de opdracht *delete /D1/C*.
Leg uit welk probleem ontstaat als het systeem file C zonder meer opruimt.
- Beschrijf een oplossing voor het hierboven genoemde probleem.
- Een andere manier om files via verschillende paden te bereiken, maakt gebruik van zogenaamde *soft links*. Beschrijf hoe dit werkt.
- Bestaat bij gebruik van soft links het probleem dat onder a) werd gesignaleerd ook? Motiveer uw antwoord.

Opgave 8 (12 punten)

- Beschrijf de werking van respectievelijk een Trojan Horse, een Virus en een Trap Door.
- Beschrijf de werking van Stack of Bufferoverflow mechanisme als beveiligingslek.
- Welke maatregelen (zouden) hiertegen kunnen worden ondernomen?

Opgave 9 (12 punten)

Gegeven een gedistribueerd systeem waar een aantal processen op verschillende systemen draaien. Tussen de events binnen die processen moet een ordeningsrelatie gelden.

- Leg uit wat met de “Happened Before” relatie wordt bedoeld en geef een voorbeeld van waarbij we dit nodig hebben.
- Hoe implementeren we deze relatie op één CPU?
- Hoe implementeren we dit op een gedistribueerd systeem?
- Hoe lossen we tie-breaks op?