

TI-2720 Operating System Concepten

8 november 2011, 14.00 - 17.00 uur.

docent: H.J. Sips

Dit is een tentamen met **8** open vragen

Opgave	Punten
1	9
2	12
3	12
4	12
5	12
6	12
7	10
8	9
9	12
Totaal	100

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

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

Opgave 1 (9 punten)

De kernel van een Operating System kan worden aangeroepen door middel van een interrupt of trap.

- Leg uit wanneer er een interrupt optreedt. Geef een voorbeeld.
- Leg uit wanneer er een trap optreedt. Geef een voorbeeld.
- Interrupts en traps worden op dezelfde wijze afgehandeld. Licht kort toe wat er gebeurt bij het optreden van een interrupt of trap.

Opgave 2 (12 punten)

- Wat wordt er in een Operating Systeem verstaan onder een proces?
- Processen kunnen child processen creëren. Noem twee redenen waarom het nuttig kan zijn dat een proces niet zelf alle taken uitvoert, maar dit doet door middel van child proces. Motiveer uw antwoord kort.
- Soms wordt er geen gebruik gemaakt van child proces, maar van threads. Beschrijf kort wat er moet worden verstaan onder een thread.
- Noem twee belangrijke voordelen van het gebruik van threads in plaats van child processen.

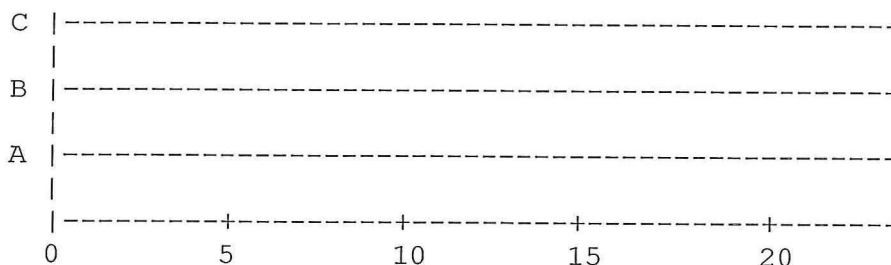
Opgave 3 (12 punten)

- Wat wordt er verstaan onder een preemptive scheduling policy?
- Onderstaande tabel geeft voor de processen A, B en C aan op welk tijdstip deze binnenkomen in de Ready queue en hoeveel CPU-tijd ze nodig hebben. Deze processen doen geen I/O of andere activiteiten die aanleiding zouden zijn tot een blocked toestand.

proces	aankomst (sec)	CPU-tijd (sec)
A	0	6
B	1	3
C	2	4

De short term scheduler in het systeem maakt gebruik van de scheduling policy die bekend staat als preemptive Shortest Job First (preemptive SJF).

Geef in een schema zoals hieronder aangegeven (teken het schema over op uw papier) aan, wanneer ieder proces de CPU had (aangeven met C) en wanneer deze Ready (aangeven met R) was.



- Wat is de gemiddelde responstijd voor de hierboven beschreven processen? Laat de berekening zien.
- Kunt u preemptive SJF aanbevelen als een goede scheduling policy voor timesharing systemen? Licht uw antwoord kort toe.

Opgave 4 (12 punten)

Gegeven een systeem met één processor. In dit systeem lopen N processen P_1 t/m P_N . Deze processen maken gebruik van de volgende shared variabelen.

```
var    a : array[1..N] of integer;
       max : integer;
```

Ieder proces P_i bevat onder meer de volgende code (de getallen vooraan de regel dienen om in uw antwoord eenvoudig naar het regelnummer te kunnen verwijzen)

```

      .
      .
(1)   j := a[i];
(2)   a[i] := func1(j);
(3)   if a[i] > max then max := a[i]; else no-op;
(4)   print(max);
      .
      .

```

Hierbij is `func1` een functie die geen gebruik maakt van shared variabelen.

- Welke regels in de bovenstaande code vormen een kritieke sectie waarvoor wederzijdse uitsluiting moet gelden? Motiveer uw antwoord.
- Beschrijf hoe de kritieke sectie door middel van één of meer semaforen beschermd kan worden. Geef hierbij de initiële waarde(n) van de semafoor (of semaforen) en geef aan op welke plaats in de code welke operaties op de semafoor (semaforen) moeten worden uitgevoerd. Geef een korte toelichting.
- Werkt de oplossing met semaforen ook in een systeem met meerdere processoren? Licht uw antwoord kort toe.

Opgave 5 (12 punten)

Een van de begrippen die een rol spelen bij het ontwijken (avoidance) van deadlocks is het begrip veilige volgorde.

- Leg uit wat er verstaan wordt onder een veilige volgorde.
- Stel we hebben een systeem met 8 buffers. Onderstaande tabel geeft aan hoe de buffers op tijdstip $t=0$ zijn verdeeld over de processen A, B en C, en geeft de maximale behoefte van de processen (hun *claim*)

proces	heeft	claim
A	3	7
B	2	5
C	1	3

- Laat zien dat dit een veilige toestand is.
- B vraagt nu zijn derde buffer. Mag deze worden toegewezen als we deadlocks willen ontwijken? Licht uw antwoord kort toe.
- Als een toewijzing niet zou mogen plaatsvinden en het zou toch gebeuren, gaat er dan gegarandeerd een deadlock optreden? Motiveer uw antwoord.

Opgave 6 (12 punten)

Een computer systeem maakt gebruik van paging. De pagesize is 4096 bytes en een logisch adres (=virtueel adres) is 32 bits lang.

- Hoeveel entries zijn er in de pagetable als de volledige logische adresruimte wordt gebruikt? Laat de berekening zien.
- Leg kort maar precies uit hoe de pagetable wordt gebruikt om een logisch adres te vertalen in een fysiek adres.
- Wat is de functie van de Translation Look Aside Buffer (TLB) ?
- Wat gebeurt er bij een contextswitch met de pagetable en wat met de TLB?

Opgave 7 (10 punten)

Van een proces hebben we de volgende Page Reference String (PRS) gemeten:

<0, 2, 1, 3, 2, 0, 2, 3, 2, 0, 1, 3>

Voor het proces zijn 3 frames beschikbaar, genummerd 0, 1 en 2.

- Laat voor de policies: First-in First-out (FIFO), Least Recently Used (LRU), en de Optimale Policy (MIN) stap voor stap zien, welke pages in welke frames aanwezig zijn, en bepaal het aantal pagefaults.
- Kloppen de uitkomsten van a. met hetgeen je mocht verwachten? Licht je antwoord kort toe.

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) tegen het in b. genoemde mechanisme 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?

Einde tentamenopgaven