TECHNISCHE UNIVERSITEIT DELFT
Faculteit EWI, Afdeling SCT
Sectie Parallelle en Gedistribueerde Systemen

TI-2720/2725 Operating System Concepten

9 november 2013, 14.00 - 17.00 uur.

docent: H.J. Sips

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

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

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

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

Opgave 1 (12 punten)

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

a. Leg uit wanneer er een interrupt optreedt. Geef een voorbeeld.

b. Leg uit wanneer er een trap optreedt. Geef een voorbeeld

c. Interrupts en traps worden op dezelfde wijze afgehandeld. Geef kort weer wat er gebeurt bij het optreden van een interrupt of een trap

d. Wanneer we de termen synchroon en asynchroon willen gebruiken in verband met interrupts en traps, welke kunnen we dan synchroon noemen en welk asynchroon? Geef een korte toelichting.

Opgave 2 (12 punten)

Bij threading maken we een onderscheid tussen drie categorieën: user threads, kernel threads, en hardware threads.

a. Leg uit wat er onder iedere categorie wordt bedoeld?

b. Wie doet in iedere thread categorie de scheduling?

c. Wat gebeurt er als één van de threads binnen een categorie blijft hangen?

Opgave 3 (12 punten)

Gegeven een time-sharing systeem. Op verschillende tijdstippen komen 3 processen A, B en C binnen. De processen gebruiken alleen CPU-tijd en geven hun uitvoer pas bij afloop. Aangenomen mag worden dat het antwoord onmiddellijk na afloop van het proces op het scherm van de gebruiker verschijnt. Tijdstip van aankomst van het proces en benodigde CPUtijd zijn volgens onderstaande tabel.

	aankomst tijd	CPU tijd
A	0	8
B	2	5
C	4	1

(Alle tijden zijn in ms)

a. Neem aan dat het systeem gebruik maakt van Round-Robin scheduling (RR) met een timeslice q=3 ms. Geef in een tijddiagram zoals hieronder aangegeven, voor elk van de processen A, B en C op de bijbehorende horizontale lijn de intervallen aan waarin zij de CPU hadden (aanduiden met letter c) en waarin zij zich in de Ready-queue bevonden (aanduiden met r).

```
C |-------------------------------------------------------------
  |
B |-------------------------------------------------------------
  |
A |-------------------------------------------------------------
  |
  |----------|----------|----------|----------|----------
  0          5          10         15         20
     --> t (ms)
```

b. Neem nu aan dat het systeem geen gebruik maakt van RR scheduling, maar van scheduling op basis van preemptive Shortest Job First (preemptive SJF). Geef op dezelfde wijze als bij a.) in een tijddiagram aan, wanneer A, B en C bij preemptive SJF de CPU hebben, en wanneer zij zich in de Ready-queue bevinden.

c. Wat is bij a.) de gemiddelde responstijd en wat die bij b.)?
Kloppen de uitkomsten met hetgeen je zonder uitrekenen zou mogen verwachten? Motiveer je antwoord.

d. In hoeverre is preemptive SJF te adviseren als scheduling policy voor een time-sharing systeem? Licht je antwoord kort toe.

Opgave 4 (12 punten)

Mutual exclusion (wederzijdse uitsluiting) kan op een aantal manieren worden gerealiseerd.

a. Leg uit hoe mutual exclusion kan worden bereikt met
- Spinlocks
- Nonpreemptive scheduling
- Interrupt management

b. Wat zijn de voordelen en de nadelen van elk van de bovengenoemde methoden?

c. Werken deze methoden ook bij multi-core processoren (met een shared memory)?

Opgave 5 (12 punten)

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

a. Leg uit wat er verstaan wordt onder een veilige volgorde.

b. Stel we hebben een systeem waarin 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 tot nu gekregen	claim (=maximum)
A	1	4
B	2	3
C	3	7

Laat zien dat dit een veilige toestand is.

c. C vraagt nu zijn vierde buffer. Mag deze worden toegewezen als we deadlocks willen ontwijken? Licht uw antwoord kort toe.

d. 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)

Van een proces hebben we de volgende Page reference String (PRS) gemeten:
1, 2. 3, 4, 1, 3, 5, 1, 2, 6, 1, 2, 4, 2

Voor het proces zijn 4 frames beschikbaar.

a. Laat voor de policies: First-in Fist-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.

b. Kloppen de uitkomsten van a. met hetgeen verwacht mocht worden? Licht uw antwoord kort toe.

c. Het aantal frames dat voor een proces beschikbaar is, hoeft niet constant te zijn. Er bestaan dynamische frame-allocation policies die het aantal frames voor een proces aanpassen afhankelijk van de situatie. Eén van deze policies is de Page-Fault-Frequency policy. Geef een korte uitleg van de werking van deze policy.

Opgave 7 (12 punten)

Het beheer v/d harde schijf heeft een grote impact op de prestaties van het totale computer systeem (denk oa. aan paging).

a. Om de *seek time* te minimaliseren zijn er diverse scheduling algoritmes bedacht. Leg de verschillen uit tussen FCFS (First Come First Served), SSTF (Shortest Seek Time First), SCAN en LOOK.

b. Reken voor de bovenstaande algoritmes de *gemiddelde* response tijd uit voor de volgende situatie: de kop staat op track 53, de wachtrij is "98, 183, 37, 122, 14, 124, 65, 67", en de maximale seek time van track 0 naar 256 bedraagt 8ms. De seek time is proportioneel met het aantal tracks dat wordt gepasseerd.

c. Welk algoritme wordt in de praktijk het meest gebruikt, en waarom?

Opgave 8 (8 punten)

Er bestaan 3 verschillende manieren voor toekennen van schijfruimte aan files. Deze worden aangeduid met de termen: *contiguous*, *linked (chained)* en *indexed* allocation.

a. Beschrijf ieder van de bovenstaande methoden kort.

b. Vergelijk deze methoden op de volgende aspecten, geef steeds een korte toelichting.
1) toegangsmogelijkheden (sequentieel en direct access)
2) Interne en externe fragmentatie op de disk
3) uitbreidingsmogelijkheden van de files

Opgave 9 (8 punten)

a. Wat houdt het CAP theorema in voor een gedistribueerd systeem?

b. Gegeven een aantal Servers met ieder één Client. Een van deze Servers beheerd data records die voor alle Clients nodig zijn en die door alle Clients gelezen en/of beschreven kunnen worden. Geef aan wat dit voor de drie CAP scenario's betekent. Welke additionele mechanismen zijn dan noodzakelijk?

Einde tentamenopgave