

Tentamen Operating Systems

Maandag 15 juni 2015, 10:00 – 13:00

- Het tentamen is **gesloten boek**, dus het is niet toegestaan om het tekstboek, slides of eigen gemaakte aantekeningen te gebruiken.
 - Alleen rekenmachines waarin geen teksten kunnen worden opgeslagen zijn toegestaan. Het gebruik van grafische rekenmachines is **niet** toegestaan.
 - De vragen mogen worden beantwoord in het Nederlands en Engels.
 - Beargumenteer al uw antwoorden. Aan antwoorden zonder uitleg of uitwerking worden geen punten toegekend.

 - Het aantal opgaven is 5 met een totaal van 18 onderdelen. Het tentamen bestaat uit 3 pagina's.
 - Achter elk onderdeel staat tussen vierkante haken het te behalen aantal punten.
 - Het totaal aantal te behalen punten is 100.
-

Opgave 1

Beschouw de volgende processen bestaande uit een enkele CPU burst:

Proces	Arrival time	CPU burst
P1	0	4
P2	0	13
P3	3	7
P4	6	10
P5	7	3

1. Illustreer de werking van de volgende vier CPU scheduling algoritmen door de Gantt chart uit te tekenen: FCFS, SJF, SRTF en RR (time slice = 4). Bereken voor elk van deze schedules de gemiddelde wachttijd. [5 punten]
2. Normaal gesproken wordt bij CPU utilisatie ook de context switch tijd meegerekend. Leg uit waarom dit misleidend kan zijn. [5 punten]

Op veel systemen wordt per proces het aantal vrijwillige (“voluntary”) en niet-vrijwillige (“non-voluntary”) context switches per seconde bijgehouden.

3. Omschrijf een multi-level feedback queue scheduler met twee “levels” die gebruik maakt van deze informatie. Geef in de omschrijving ten minste aan: de schedulers die worden gebruikt voor de verschillende levels en de condities voor promotie en degradatie. [10 punten]

Opgave 2

1. Vele gangbare computersystemen ondersteunen “dual-mode operation”. Leg uit hoe dit wordt gebruikt in de implementatie van operating systems. [5 punten]
2. Geef twee consequenties als “dual-mode operation” niet wordt toegepast. [5 punten]

Er bestaan verschillende manieren om operating systems op te bouwen en te structureren. Dit betreft ook de manier waarop de twee “modes” worden toegepast.

3. Beschrijf het verschil tussen het monolithische-kernel model en het microkernel model en bespreek de relatie met “dual-mode operation”. [5 punten]
4. Als u meer dan twee modes ter beschikking zou hebben, hoe zou u deze dan inzetten in de implementatie van een microkernel? Geef aan hoeveel modes u zou willen gebruiken en wat de mogelijkheden en restricties moeten zijn in de verschillende modes. [5 punten]

Opgave 3

Gegeven een disk met cylinders 0 tot en met 399. De leeskop bevindt zich op cylinder 161. De wachtrij voor deze disk bestaat uit de volgende verzoeken:

155, 338, 53, 125, 149, 316, 387, 272, 312, 165, 73, 222

1. Welke cylinders worden bezocht en in welke volgorde, wanneer gebruik wordt gemaakt van de volgende algoritmen: SSTF, C-SCAN en C-LOOK? U hoeft **geen** diagrammen uit te tekenen. [5 punten]
2. Hoe wordt de disk bandbreedte (“disk bandwidth”) berekend? Leg aan de hand van uw antwoord uit waarom disk scheduling algoritmen een belangrijke rol spelen bij het verbeteren van de disk bandbreedte. [5 punten]
3. Bij virtualisatie draait er binnen de gast machine een complete operating system kernel inclusief disk scheduler. Heeft de disk scheduler in de gast naar uw mening een toegevoegde waarde? Beargumenteer uw antwoord. [10 punten]

Opgave 4

1. Leg het mechanisme van memory-mapped files uit aan de hand van een illustratie van de virtuele adresruimte van een proces en het fysieke geheugen. [5 punten]
2. Beargumenteer met een voorbeeld waarom het verwerken van bestanden met het memory-mapped files mechanisme (veel) minder overhead met zich mee brengt dan met het gebruik van standaard system calls als *open()*, *read()* en *write()*. [5 punten]
3. Leg uit welke invloed “TLB reach” heeft op memory-mapped files. Omschrijf de karakteristieken van een programma waarin deze invloed naar voren komt. [5 punten]
4. Aan de hand van welk meetgegeven kunt u “at run-time” vaststellen of een proces te lijden heeft onder de invloed van “TLB reach”? [5 punten]

Opgave 5

1. Leg uit hoe “journaling” (of “log-structured”) file systems het probleem van consistency checking oplossen. [5 punten]
2. Leg uit wat het “trap-and-emulate” mechanisme is en waarom dit van belang is bij de implementatie van virtualisatie. [5 punten]
3. Wat wordt er verstaan onder “copy-on-write”? [5 punten]
4. Omschrijf het probleem van “priority inversion” en een mogelijke manier dit probleem op te lossen. [5 punten]