Examen besturingssystemen

Zaterdag 12 januari 2019, 8u30

Prof. Koen De Bosschere

Richting: Naam:

Belangrijk

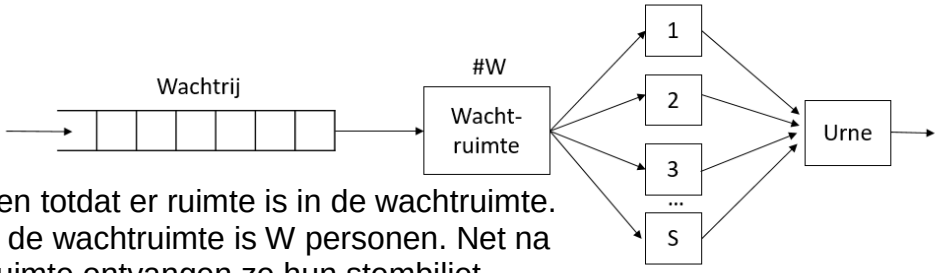
1. Vergeet niet uw naam te vermelden.
2. Schrijf de antwoorden in de daarvoor voorziene ruimte. Schrijf duidelijk en zorg voor voldoende structuur in uw antwoord.
3. Het examen duurt 3 uur.
4. Gelieve geen rode inkt te gebruiken.
5. Het examen is gesloten boek, enkel de leesopdrachten mogen gebruikt worden.
6. U mag geen computer gebruiken bij de oplossing van de vragen, maar wel een eenvoudige rekenmachine.
7. Gelieve uw mobieltje uit te schakelen.
8. Onregelmatigheden worden aan de examencommissie gemeld.

Veel succes!

Ik verklaar op erewoord dat ik noch hulp geboden heb aan, noch hulp ontvangen heb van derden tijdens het oplossen van dit examen. Handtekening:

Schrijf hier eventuele opmerkingen die van belang kunnen zijn bij de quotering (ziekte, topsport, gemaakte afspraken, enz.).

Vraag 1 (4 punten)



Bij de verkiezingen komen de kiezers binnen en wachten totdat er ruimte is in de wachtruimte. De maximale capaciteit van de wachtruimte is W personen. Net na het betreden van de wachtruimte ontvangen ze hun stembiljet (roep dan de functie `ontvangstembiljet()` op). Ze wachten in de wachtruimte totdat er een stemhokje vrijkomt. Er zijn S stemhokjes, en die kunnen slechts door 1 persoon tegelijk gebruikt worden. Na het stemmen (roep daarvoor de functie `stem()` op) deponeren de keizers hun stembiljet één na één in de urne (roep dan de functie `steekinurne()` op), en krijgen ze hun afgestempelde oproepingsbrief terug. Op die oproepingsbrief wordt het nummer van het gebruikte stemhokje genoteerd (roep daarvoor de functie `ontvangoproepingsbrief(hokje)` op). Schrijf een programma dat dit proces implementeert.

```
init() {









}

kiezer () {









}
```

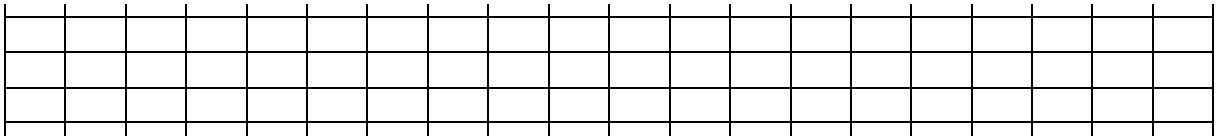
Vraag 2 (4 punten)

Gegeven de volgende proceslijst.

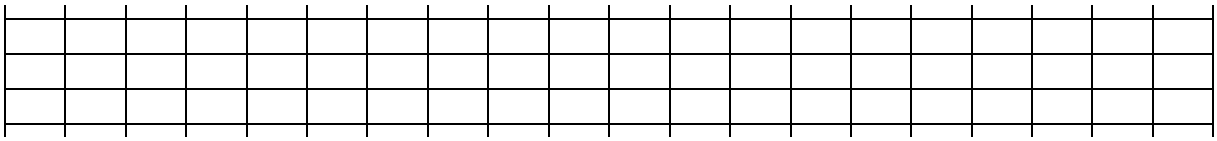
Proces	Aankomst	Burst	IO	Burst
P1	0	3	2-ε	1
P2	1-ε	1	3-2ε	2
P3	2-ε	3	1-3ε	2
P4	3-ε	1	2-4ε	1

Stel een planningsdiagram op voor de volgende planners.

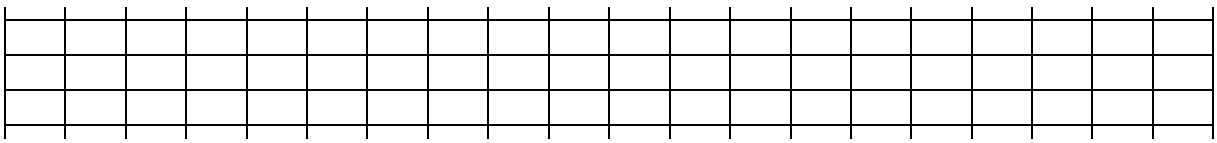
FCFS – op één core



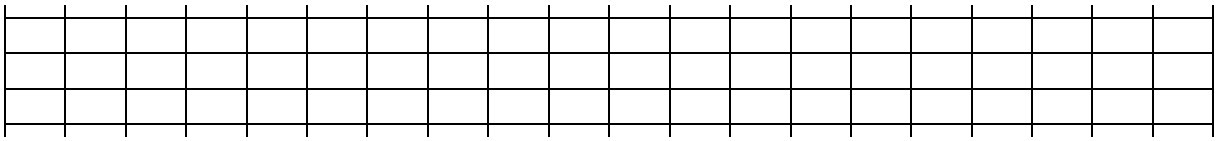
Round Robin (Tijdskwantum = 2) – op één core



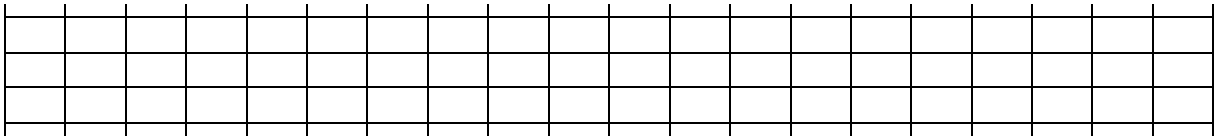
Round Robin (Tijdskwantum = 2) – op twee cores (identificeer duidelijk de cores)



Round Robin (Tijdskwantum = 2) – op vier cores (identificeer duidelijk de cores)



FCFS – op één core, waarbij de tweede burst (persistent) gedeblokkeerd wordt door de eerste burst van een ander proces (het einde van de eerste burst van P1 deblokkeert de tweede burst van P2: P1 → P2; en verder P2 → P4; P3 → P1; P4 → P3) [FCFS dynamisch].



Wat is de gemiddelde belasting van de processor vanaf tijdstip 0 tot het einde van de uitvoering?

FCFS 1 core	RR 1 core	RR 2 cores	RR 4 cores	FCFS dynamisch

Vraag 3 (4 punten)

Om te vermijden dat de informatie van de inodes en de bijhorende datablokken op een schijf te ver van elkaar verwijderd liggen, wordt een partitie opgedeeld in zogenaamde blokgroepen die bestaan uit datablokken van een aantal naast elkaar liggende cilinders. Een dergelijke groep wordt georganiseerd alsof het een kleine partitie was (met een eigen inodetabel, bitmap voor het bijhouden van de vrije ruimte, enz.). Op die manier kan ervoor gezorgd worden dat inodes, directoryinformatie en datablokken in elkaars nabijheid opgeslagen worden en er dus minder verre bewegingen van de schijfarm nodig zijn (in vergelijking met de situatie waarbij alle inodes bv. bij het begin van de partitie opgeslagen zouden liggen).

Je mag uitgaan van een schijf met een opslagcapaciteit van 4TiB (omdat dit gemakkelijker rekent) en met datablokken van 8KiB.

De toestand van de datablokken (vrij/in gebruik) in een blokgroep wordt bijgehouden in een bitmap. Deze bitmap is even groot als één datablok.

Hoeveel datablokken zal een blokgroep bevatten?	
Hoeveel opslagruimte kan een blokgroep maximaal bevatten (in MiB)?	
Hoeveel blokgroepen zullen er op de schijf van 4TiB kunnen?	
Per blokgroep moet er 64 bytes aan metainformatie bijgehouden worden. Hoeveel ruimte zal de beschrijving van alle blokgroepen opnemen op de schijf. Druk de omvang ook uit in MiB.	
De gemiddelde zoektijd op een schijf tussen twee willekeurige plaatsen is evenredig met N/3 met N het totaal aantal cilinders. Als je uitgaat van 50% verplaatsingen tussen blokgroepen, 50% verplaatsingen binnen blokgroepen, hoeveel bedraagt de zoektijdverbetering (bij benadering)?	
Hoeveel ruimte zal de inodetabel per blokgroep innemen (als men ook de minst gunstige situatie waarbij alle bestanden slechts één blok groot zijn wil aankunnen). Ga uit van een inode-grootte van 128 bytes. Druk de totale omvang van de inodetabel ook uit in MiB.	
De gegeven schijf heeft een straal van 1,5 inch, en daarvan wordt enkel de buitenste 1 inch gebruikt om data te slaan. Als de dichtheid van de cilinders 390 000 per inch is, uit hoeveel cilinders zal een blokgroep dan bestaan?	
Uit hoeveel sporen zal een blokgroep bestaan indien de schijf uit drie platen bestaat die aan beide zijden gebruikt worden?	
Wat is dan de gemiddelde datahoeveelheid die op één spoor opgeslagen wordt (in MiB)?	
Wat is de lineaire **bit**dichtheid/inch op een spoor dat 1 inch uit het centrum van de schijf ligt ($O=2\pi R$)	

Vraag 4 (4 punten)

Leg de x86 geneste 32-bit adresvertaling uit (zonder segmenten, dus vertrekkende van de lineaire adressen). Maak duidelijk welk type van adressen er op de verschillende plaatsen gebruikt worden, hoe breed de velden zijn, enz.

Hoeveel extra geheugentoegangen zijn er in het slechtste geval nodig om één adresvertaling te kunnen doen?

Leg vertrekkende van de voorgaande tekening de werking en het nut/noodzaak van schaduwpaginatabellen uit.

Vraag 5 (2 punten)

Wat is een SLA?

Wat zijn de basisprincipes bij het nemen van een reservekopie?

Vraag 6 (2 punten)

Wat is de *Overload-on-Wakeup bug*, en hoe kan hij opgelost worden?

Wat is de wet van Kryder?