

TENTAMEN

Operativsystem DVA315, 2019-03-27 kl. 08:10-12:30

Ansvarig lärare: Mats Björkman

Max poäng: 30

Betygsgränser: 3: 18p, 4: 22p, 5: 26p

Hjälpmedel: -

Påbörja varje uppgift på ett nytt papper!

Lycka till!

Begreppsdel**Uppgift 1 (3p) Allmänt**

- a) Ett operativsystem kan sägas vara en *virtuell maskin*. Förklara och exemplifiera *virtuell maskin* i detta sammanhang, alltså för generella OS och inte bara hypervisors. (1p)
- b) Ett operativsystem kan även sägas vara en *resursadministratör*, vad innebär detta begrepp i detta sammanhang? Ge exempel på resurser som administreras. (1p)
- c) En processor har oftast två exekveringslägen, *supervisor mode* och *user mode*. Förklara hur ett systemanrop byter mellan dessa lägen, samt förklara syftet med detta. (1p)

Uppgift 2 (5p) Allmänt

Förklara kortfattat följande operativsystemsrelaterade begrepp:

- a) Pseudoparallelism (till skillnad från sann parallelism) (0.5p)
- b) Relokerbarhet (Ability of relocation) för processer (0.5p)
- c) Räknande semafor (0.5p)
- d) Extern fragmentering (i minnessammanhang) (0.5p)
- e) Intern fragmentering (i minnessammanhang) (0.5p)
- f) Osäkert tillstånd (i låsningssammanhang) (0.5p)
- g) Journalförande filsystem (0.5p)
- h) Master boot record (0.5p)
- i) Asynkront meddelandesystem (0.5p)
- j) Thrashing (i OS-sammanhang) (0.5p)

Uppgift 3 (4p) Baklås

Man brukar säga att det krävs *fyra villkor* för att ett system skall kunna vara i baklås.

- a) Ange, samt förklara kort innebörden av, dessa 4 villkor. (2p)
- b) Ge två konkreta exempel på hur man kan omöjliggöra baklås i ett system genom att eliminera något av de fyra villkoren. (2p)

Problemdel

Uppgift 4 (6p) Schemaläggning

Ett system har 6 processer A-F som med följande aktiverings- och exekveringstider:

Process	Aktiveringstid	Exekveringstid
A	0	3
B	1	1
C	2	1
D	4	6
E	8	2
F	11	1

När processer har samma aktiveringstid antas de komma till skeduleraren i bokstavsordning. Om aktivering av en ny process sker vid samma tid som en omskedulering p.g.a. preemption, antas den nyaktiverade processen ställas i kö först, innan processen som råkat ut för preemption ställs i kö.

- Schemalägg processerna enligt algoritmen *shortest job first* (SJF). Algoritmen är preemptiv, har ett tidskvanta på 1, och schemaläggs enligt kvarvarande exekveringstid vid varje givet tillfälle. (1p)
- Beräkna medelomloppstiden för processerna schemalagda med SJF. (1p)
- Schemalägg processerna enligt algoritmen *Round Robin* (RR). Algoritmen är preemptiv och har ett tidskvanta på 1. Vid aktivering ställs en ny process sist i ready-kön. (1p)
- Beräkna medelomloppstiden för processerna schemalagda med RR. (1p)
- Schemalägg processerna enligt den preemptiva algoritmen *multipla köer* (MK). Schemaläggaren har tre köer: HÖG med kvanta 1, MELLAN med kvanta 2 samt LÅG med kvanta 4. Samtliga processer startar i kön HÖG. (1p)
- Beräkna medelomloppstiden för processerna schemalagda med MK. (1p)

Eventuella antaganden MÅSTE motiveras!

Uppgift 5 (4p) Baklås (Deadlock)

I ett operativsystem har man implementerat baklåsdetektering med hjälp av en algoritm som använder E, A, C och R matriser för att periodiskt kontrollera om några processer är i baklås eftersom systemet stödjer multipla resurser av samma typ.

Vid ett givet tillfälle befinner sig systemet i följande tillstånd:

Existerande resurser: w: 4st
 x: 6st
 y: 4st
 z: 3st

Aktiva Processer: p1, p2, p3 och p4

Nuvarande ägandeskap: w: p1 äger 1st, p2 äger 2st, p3 äger 1st
(Claimed Resources) x: p1 äger 1st, p2 äger 1st, p3 äger 1st, p4 äger 1st
 y: p2 äger 1st, p4 äger 2st
 z: p1 äger 1st, p3 äger 1st, p4 äger 1st

Begärda resurser: w: p1 begär 2st, p3 begär 2st, p4 begär 3st
(Requested Resources) x: p1 begär 3st, p2 begär 1st, p3 begär 1st, p4 begär 2st
 y: p1 begär 2st, p2 begär 1st, p3 begär 3st
 z: p4 begär 2st

- a) Konstruera matriserna E, A, C och R för ovanstående tillstånd. (2p)
- b) Är systemet i baklås? Visa hur du kom fram till detta m.h.a matriserna. (2p)

Eventuella antaganden MÅSTE motiveras!

Uppgift 6 (4p) Virtuellt minne

Antag att man har ett sidindelat virtuellt minne med en sidstorlek på 8 bytes. Vidare har varje process (A och B i vårt exempelsystem nedan) tillgång till 128 bytes virtuellt minne medan det fysiska minnet har en storlek av 64 bytes.

- a) Visa hur sidtabellen för process A respektive B ser ut samt hur det fysiska minnet ser ut (med binära minnesadresser) om vi antar att process A har sidorna 1, 3, 6, 7 och 11 på ramarna 2, 4, 3, 6 respektive 0 i det fysiska minnet. Process B har sidorna 3, 6 och 13 på ramarna 1, 7 respektive 5 i det fysiska minnet. (2p)
- b) Till vilken fysisk adress översätts den logiska adressen 0011110 för process A? Förklara hur du kommer fram till detta. (1p)
- c) Till vilken fysisk adress översätts den logiska adressen 0011110 för process B? Förklara hur du kommer fram till detta. (1p)

Eventuella antaganden MÅSTE motiveras!

Uppgift 7 (4p) Filhantering

I filhantering finns många olika tekniker för att hålla reda på vilka block på disken som tillhör vilken fil. Två av dessa tekniker är **i-noder** och **länkade listor med index (t.ex. FAT-tabeller)**.

I ett system med en tom disk med 16 lediga block och tre olika filer (A, B och C), sker följande förfrågningar efter diskutrymme:

- 1. Fil A begär 2 block
- 2. Fil B begär 1 block
- 3. Fil C begär 4 block
- 4. Fil B begär 2 block
- 5. Fil C begär 3 block
- 6. Fil A begär 1 block

- a) Visa hur en teknik med **i-noder** hanterar ovanstående förfrågningar om diskutrymme (och hur den håller reda på de allokerade blocken) (2p)
- b) Visa hur en teknik med **länkade listor med index (t.ex. FAT-tabeller)** hanterar ovanstående förfrågningar om diskutrymme (och hur den håller reda på de allokerade blocken) (2p)

Kom ihåg att förklara, för var och en av teknikerna, vilken fil de allokerade blocken tillhör efter varje förfrågan. Om teknikerna använder några speciella datastrukturer för att hålla reda på blocken, beskriv även dessa strukturers tillstånd *efter varje förfrågan*.

Eventuella antaganden MÅSTE motiveras!