

**TENTAMEN**

Operativsystem DVA315, 2022-06-08

Ansvarig lärare: Mats Björkman

Max poäng: 30

Betygsgränser: 3: 15p, 4: 23p, 5: 27p

Hjälpmedel: -

Om du är osäker på vad som efterfrågas, gör rimliga antagen och motivera dessa antaganden innan du löser uppgiften.

**Påbörja varje uppgift på ett nytt papper!**

*Lycka till!*

**Begreppsdel****Uppgift 1 (4p) 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) Race condition (i processammanhang) (0.5p)
- d) Intern fragmentering (i minnessammanhang) (0.5p)
- e) Osäkert tillstånd (i baklåssammanhang) (0.5p)
- f) Master Boot Record (0.5p)
- g) Thrashing (i processammanhang) (0.5p)
- h) DMA (Direct Memory Access) (0.5p)

**Uppgift 2 (4p) Synkronisering**

Ett sätt för processer att kommunicera är via meddelandesystem.

- a) Förklara skillnaderna mellan *indirekt* och *direkt* kommunikation. (2p)
- b) Förklara skillnaderna mellan *asynkron* och *synkron* kommunikation. Ge även exempel på hur dessa fungerar rent implementationstekniskt. (2p)

**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)

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	2	5
E	8	3
F	10	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 att den nyaktiverade processen placeras i kön *innan* processen som råkat ut för preemption placeras i kön.

- a) Schemalägg processerna enligt algoritmen *shortest job first* (SJF). Algoritmen är preemptiv, har ett tidskvantum på 1, och schemaläggs enligt kortast kvarvarande exekveringstid vid varje givet tillfälle. (1p)
- b) Beräkna medelomloppstiden för processerna schemalagda med SJF. (1p)
- c) Schemalägg processerna enligt algoritmen *Round Robin* (RR). Algoritmen är preemptiv och har ett tidskvantum på 1. Vid aktivering ställs den nya processen sist i ready-kön. (1p)
- d) Beräkna medelomloppstiden för processerna schemalagda med RR. (1p)
- e) Schemalägg processerna enligt den preemptiva algoritmen *multipla köer* (MK). Schemaläggaren har tre köer: HÖG med kvantum 1, MELLAN med kvantum 2 samt LÅG med kvantum 4. Vid aktivering ställs den nya processen sist i kön HÖG. Efter att ha exekverat sitt kvantum i HÖG byter processen till kön MELLAN. Efter att ha exekverat sitt kvantum i MELLAN byter processen till LÅG. När en process byter kö ställs den sist i den nya kön. (1p)
- f) Beräkna medelomloppstiden för processerna schemalagda med MK. (1p)

***Eventuella antaganden måste motiveras!***

**Uppgift 5 (4p) Virtuellt minne**

En process har tre tillgängliga ramar i primärminnet och fem sidor i det virtuella minnet. Primärminnet är initialt tomt. Hur många sidfel genererar följande sidaccess-sekvens:

1 2 1 5 1 2 3 2 3 4 2 4 5 1 2 3 4 2 3 5

om man använder följande sidutbytesstrategier (Visa hur du kommer fram till detta):

- a) First in first out (FIFO)? (1p)
- b) Least recently used (LRU)? (1p)
- c) Optimal utkastningsstrategi (OPT)? (1p)
- d) Varför kan OPT inte implementeras? (1p)

*Eventuella antaganden måste motiveras!*

**Uppgift 6 (4p) Bankers Algoritmen**

Vi har ett system med 10 stycken enheter av en viss resurs. Dessa är allokerade till processer enligt följande:

Process A har 4st resurser  
Process B har 2st resurser  
Process C har 1st resurser

Resten är oallokerade. Det maximala antalet resurser som var och en kan allokera är:

Process A max 8st resurser  
Process B max 9st resurser  
Process C max 3st resurser

Visa hur Bankers algoritmen hanterar följande sekvens av allokeringsbegäran:

B begär ytterligare en resurs  
A begär ytterligare en resurs  
C begär ytterligare en resurs  
A begär ytterligare en resurs

Visa också hur du kom fram till detta.

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

- a) Visa hur en teknik med **i-noder** hanterar ovanstående förfrågningar om diskutrymme (och hur den håller reda på de allokerade blocken). Kom ihåg att *visa tillståndet efter varje förfrågan!* (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). Kom ihåg att *visa tillståndet efter varje förfrågan!* (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!***