Lösningsskisser, tenta OS 2017-06-07

1a) Pseudoparallellism är när OS:et kör flera processer ”samtidigt”, en liten snutt i taget på varje. Det upplevs av användaren som att processerna kör parallellt, men det är en illusion.
b) Relokerbarhet är förmågan att flytta processer i minnet. Relokering kan vara statisk (processer blir kvar där man laddat den) eller dynamisk (processen kan flyttas under exekvering).
c) Race condition är när utfallet fallet av samtidiga (parallella eller pseudoparallella) exekveringar kan bli olika, beroende på exakt i vilken ordning instruktionerna i de samtidiga exekveringarna utförs. Detta är oftast inte bra.
d) Intern fragmentering är när minnesresurser är allokerade till processer men processerna inte använder det allokerade minnet.
e) Osäkert tillstånd är ett tillstånd som riskerar att leda till ett baklås. Osäkra tillstånd bör undvikas.
f) Busy waiting innebär att processen ligger och testar oupphörligt om något visst villkor uppfyllts (exv. att en variabel skall få ett visst värde).
g) Thrashing är när alltför lite nyttigt arbete blir gjort i ett system på grund av att processerna spenderar alltför mycket av sin exekveringstid med att läsa in kod och data som inte finns kvar i minne. Problemet är för lite minne och/eller för många processer och/eller för kort tidskvantum.
h) DMA används för att flytta data utan att processorn behöver vara inblandad. Kräver hårdvarustöd.

2a) Vid indirekt kommunikation namnger sändare/mottagare en brevlåda/mailbox. Vid direkt kommunikation namnger sändare/mottagare den andra processen.
b) Vid asynkron kommunikation lämnar sändaren meddelandet i en buffert och behöver inte vänta på att mottagaren är redo. Denna buffert måste OS:et tillhandahålla. Vid synkron kommunikation (rendezvous) måste sändare invänta att mottagaren är redo, och data kan då kopieras direkt från sändare till mottagare utan mellanliggande buffert.

3a) Trådar saknar egen adressrymd och saknar skydd mellan trådar som exekverar i samma adressrymd.
b) Det är mindre resurskrävande att starta en ny tråd i en existerande process än att starta en ny process. Kommunikationen är enklare mellan trådar i samma process.

4 En bitmapp eller en länkad lista av lediga block. Se bokens avsnitt 4.4.1.

5 lämnas som övning… (Se lösningsförslagen till tentamen 2017-03-22 för principerna.)

6 Rita gärna tabeller av den typ som använts i föreläsningarna.
Primärminnet är initialt tomt, det innebär att även accesserna till de tre första sidorna genererar sidfel. (Dessa är dock ”billigare” att hantera eftersom ingen sida i minnet behöver kastas ut.) Nedan indikerar ett ”O” i raden SF att ett sidfel uppstått.

a) FIFO:

| R\S | 1 | 2 | 1 | 5 | 1 | 2 | 3 | 2 | 3 | 4 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 2 | 3 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 |
| 2 | – | – | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| 3 | – | – | – | – | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 5 |
| SF | | O | O | — | O | – | – | O | – | – | O | O | — | O | O | — | O | O | O | – | O |

FIFO: 12 sidfel

b) LRU:

| R\S | 1 | 2 | 1 | 5 | 1 | 2 | 3 | 2 | 3 | 4 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 2 | 3 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |
| 2 | – | – | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 4 | 4 | 5 |
| 3 | – | – | – | – | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 3 | 3 | 3 | 3 |
| SF | | O | O | — | O | – | – | O | – | – | O | — | – | O | O | O | O | O | – | – | O |

LRU: 11 sidfel

c) OPT:

| R\S | 1 | 2 | 1 | 5 | 1 | 2 | 3 | 2 | 3 | 4 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 2 | 3 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 |
| 2 | – | – | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | – | – | – | – | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 1 | 1 | 3 | 3 | 3 | 3 |
| SF | | O | O | — | O | – | – | O | – | – | O | – | – | – | O | — | O | – | – | – | O |

OPT: 8 sidfel

Kommentar till OPT: Vid sista accessen är det inte givet vilken sida som skall kastas ut när sidan 5 skall läsas in (eftersom framtida accesser inte finns angivna). Välj vilken sida som helst att kasta ut, inget val är fel så länge sidan 5 läses in.

d) Vi kan inte se in i framtiden.

7) Bankers algoritm, 10 resurser

| Process | Har | Max |
|---|---|---|
| A | 4 | 8 |
| B | 2 | 9 |
| C | 1 | 3 |

Lediga: 3

Först: Konstatera om vi är i säkert tillstånd nu. Ja det är vi, C kan köra, sedan A, sedan B.

B begär 1 resurs. Då skulle vi hamna i läget:

| Process | Har | Max |
|---|---|---|
| A | 4 | 8 |
| B | 3 | 9 |
| C | 1 | 3 |

Lediga: 2

Skulle vi då vara i säkert tillstånd? Nej, C kan köra, men sedan varken A eller C.
BEGÄRAN NEKAS, återgår till föregående läge.

A begär 1 resurs. Då skulle vi hamna i läget:

| Process | Har | Max |
|---|---|---|
| A | 5 | 8 |
| B | 2 | 9 |
| C | 1 | 3 |

Lediga: 2

Skulle vi då vara i säkert tillstånd? Ja, C kan köra, sedan A, sedan B.
BEGÄRAN BEVILJAS

C begär 1 resurs. Då skulle vi hamna i läget:

| Process | Har | Max |
|---|---|---|
| A | 5 | 8 |
| B | 2 | 9 |
| C | 2 | 3 |

Lediga: 1

Skulle vi då vara i säkert tillstånd? Ja, C kan köra, sedan A, sedan B.
BEGÄRAN BEVILJAS

A begär 1 resurs. Då skulle vi hamna i läget:

| Process | Har | Max |
|---|---|---|
| A | 6 | 8 |
| B | 2 | 9 |
| C | 2 | 3 |

Lediga: 0

Skulle vi då vara i säkert tillstånd? Nej, ingen kan köra klart.
BEGÄRAN NEKAS, återgår till föregående läge.

Uppgift 8:
a) Bestäm storleken på en inod först. Om den kan hålla minst 4 block blir det enklare att lösa uppgiften. Lämpligt att allokera från block 0 och framåt. Det innebär att (vi skippar delstegen här, se till att era tentasvar redovisar alla delsteg):

inoden för A pekar på block 0, 9, 10
inoden för B pekar på block 1, 2, 3, 5
inoden för C pekar på block 4, 6, 7, 8

b)
(Någonstans finns tre directoryentryn som pekar ut första blocket: A: 0, B: 1, C:4.)
FAT-tabellen ser (efter alla steg, se till att era tentasvar redovisar alla delsteg) ut som (E betyder slut på filen):
0: 9
1: 2
2: 3
3: 5
4: 6
5: E
6: 7
7: 8
8: E
9: 10
10: E