TENTAMEN
Operativsystem DVA315, 2024-03-18
Ansvarig lärare: Dag Nyström
Max poäng: 30
Betygsgränser: 3: 15p, 4: 23p, 5: 27p
Hjälpmedel: -

Påbörja varje uppgift på ett nytt papper!

Lycka till!

Begreppsdel 12 poäng

Uppgift 1: Processhantering (5p)

a) Rita upp tillståndsmaskinen för processer för den processmodell med 3 tillstånd vi arbetat med i kursen. (1p)
b) Förklara och exemplifiera tydligt de olika tillstånden samt tillståndsövergångarna. (2p)
c) För att byta den exekverande processen används en kontextbytare (context switcher) förklara steg för steg hur denna gör för att byta den exekverande processen. (2p)

Uppgift 2: Minneshantering (3p)

a) För att baklås ska kunna uppstå måste fyra villkor vara uppfyllda. Vilka är dessa. Ange och förklara dessa (2p)
b) Förklara hur en resursgraf fungerar i baklåssammanhang. (1p)

Uppgift 3: I/O (4p)

I/O hantering är egentligen väldigt krånglig, därför erbjuder operativsystem olika abstraktioner såsom, oberoende av enheternas fysiska egenheter, uniform namngivning, felhantering med mer. För att uppnå detta brukar I/O delas upp i **fyra abstraktionsskikt** ovanpå hårdvaruplattformen.

Ange, och förklara, de olika skikten och deras funktion. (4p)

Problemdel 18 poäng

Uppgift 4 (6p) Schemaläggning

Ett system har 5 processer A-E med följande aktiverings- och exekveringstider:

Process	Aktiveringstid	Exekveringstid
A	0	4
B	1	1
C	2	1
D	2	5
E	5	2

När processer har samma aktiveringtid 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 sorteras in i kön innan processen som råkat ut för preemption sorteras in i kön.

a) 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)
b) Beräkna medelomloppstiden för processerna schemalagda med SJF. (1p)
c) 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)
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 kvanta 1, MELLAN med kvanta 2 samt LÅG med kvanta 4. Samtliga processer startar i kön HÖG. (1p)
f) Beräkna medelomloppstiden för processerna schemalagda med MK. (1p)

Eventuella antaganden MÅSTE motiveras!

Uppgift 5 (8p) Virtuellt minne

Anta att man har ett sidindelat virtuellt minne med en sidstorlek på 16 bytes. Vidare har varje process (A och B i vårt exempelsystem nedan) tillgång till 256 bytes virtuellt minne medan det fysiska minnet har en storlek av 128 bytes. Operativsystemet implementerar LRU

a) Hur många bitar i den virtuella adressen krävs för index till sidtabellen? Hur många bitar krävs för offset? (1p)
b) Använd binära minnesadresser och visa hur sidtabellen för process A respektive B ser ut om vi antar att process A har de virtuella sidorna 2, 5, 6 och 14 på ramarna 3, 7, 0 respektive 2 i det fysiska minnet, och process B har de virtuella sidorna 2, 7, 14 och 15 på ramarna 1, 4, 5 respektive 6 i det fysiska minnet. (3p)
c) Till vilken fysisk adress översätts den logiska adressen 11100101 för process A? För process B? Visa hur du kommer fram till detta. (1p)
d) Givet ovanstående allokering av ramar sker sedan följande minnesaccesser:

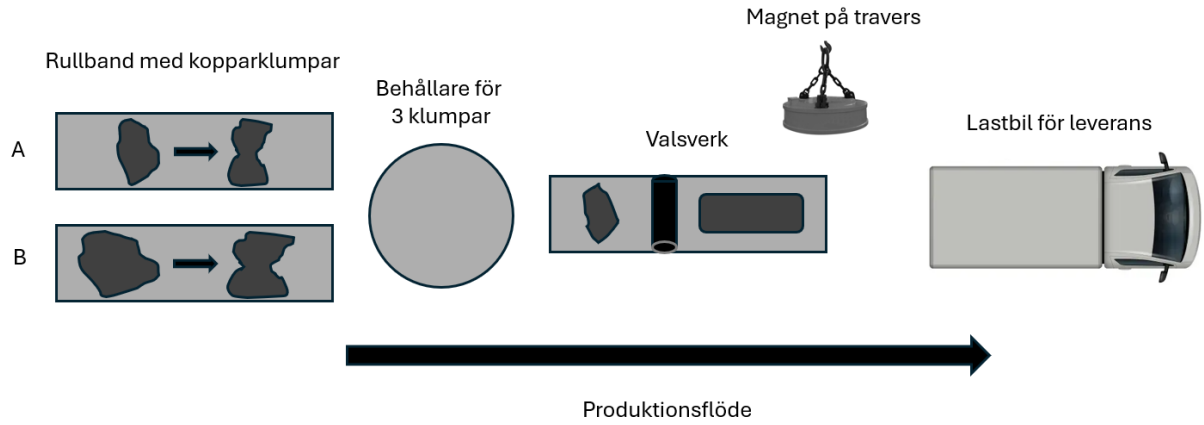
```
A begär: 0010 1001
B begär: 0111 0011
B begär: 0010 0000
A begär: 0010 1101
B begär: 1111 1001
B begär: 1110 0000
A begär: 0110 0011
A begär: 0101 0010
A begär: 1110 0110
B begär: 0101 0101
```

Visa på samma sätt som i b) hur sidtabellerna för A respektive B nu ser ut (3p)

Eventuella antaganden MÅSTE motiveras!

Uppgift 6 (4p) Valsverk

Ni har fått i uppdrag att implementera mjukvaran för ett valsverk som valsar kopparplåtar. Systemet består av två rullband där kopparklumpar kommer in. Dessa ska förflyttas till en behållare för temporär lagring av klumpar. Denna behållare har en max kapacitet på 3 klumpar. Valsverket valsar 1 klump i taget till en plåt som sedan skall förflyttas till en väntande lastbil. All förflyttning sker med en elektromagnet som hänger i en travers i taket.



Följande primitiver finns:

`LastaAvA()`	Magneten lastar en klump från rullband A till behållaren
`LastaAvB()`	Magneten lastar en klump från rullband B till behållaren
`LastaVals()`	Magneten lastar en klump från behållaren till valsverket
`LastaBil()`	Magneten lastar en plåt från valsverket till lastbilen
	Magneten kan bara göra en av ovanstående i taget
`Valsa()`	Valsverket valsar en plåt
`MataFramA()`	Rullband A matar fram nästa plåt
`MataFramB()`	Rullband B matar fram nästa plåt
`s=Sem(n)`	Skapar en semafor `s` med värdet `n`.
`Wait(s), Signal(s)`	Tar respektive släpper semafor s

En tråd skapas med ett funktionsanrop med namn som börjar på `Thread_`, t ex

```
Thread_rullband(){

}
```

Skapar en tråd rullband som automatiskt kör

UPPGIFT: Skapa tre trådar, Thread_rullbandA, Thread_rullbandB och Thread_valsverk som så effektivt som möjligt utför produktionsflöde från rullband till bil utan att arbetuppgifter krockar med varandra (4p)

Eventuella antaganden MÅSTE motiveras!