



Uppgift 1: (10p)

- a) Beskriv, med ord, vad följande shell-script utför (4p):

```
echo    'timestamp' > nyfil  
pr nyfil | lpr
```

- b) (6p)

Skissera huvudstrukturen av UNIX kommandotolkare "sh" i programspråket C. Mindre centrala delar kan anges som kommentarer. Följande funktioner ska hanteras:
utföra kommando sekvensiellt (normal hantering)
utföra kommando asynkront ("i bakgrunden")
omdirigering av standard output (>)
inbyggt kommando *change directory* (cd)
Utförandet av icke inbyggda kommandon ska implementeras som en subprocess.

Ledning: UNIX tillhandahåller bland annat följande systemanrop: fork, exec, wait, waitpid, pipe, close, dup, dup2, creat, open, chdir och exit.

Uppgift 2: (8p)

- a) Redogör kortfattat för skillnaden mellan *systemanrop* och *systemprogram*. (2 p)
b) Redogör för begreppen *process*, *processtillstånd* och *threads*. (4 p)
c) Vad menas med *mekanismer och policies*? (2 p)

Uppgift 3: (6p)

Till en process som kräver 8 sidor allokeras 4 sidramar. Antag följande referenssträng:

1,2,8,3,4,3,8,2,1,4

Hur många sidfel kommer att genereras (demand paging) med en

- a) FIFO-algoritm (2 p)
b) LRU-algoritm (2 p)
c) Optimal algoritm (2 p)

Uppgift 4: (6p)

Ett skivminne har 200 spår numrerade från 1 till 200. Antag att ett krav på spår 143 betjänas och att föregående krav var till spår 125. Antag vidare att följande krav är uppköade av drivrutinen (i FIFO ordning):

63,152,88, 195,175,32,100,164,64

Ange i vilken ordning dessa krav betjänas då diskscheduleringsalgoritmerna SSTF respektive SCAN används.



Uppgift 5: (10p)

I ett filsystem upptar en fil de logiska blocken 234, 875 och 1067. Illustrera (gärna med figurer) denna fil i ett system med allokeringsmetod:

- länkad lista, (exvis 'DOS').
- indexerad allokering (exvis 'UNIX').

Diskutera också eventuella för- /nack- delar med de olika metoderna.

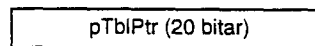
Uppgift 6: (10p)

Ett *demand-paging* system understöds av följande hårdvara:

generellt adressregister:



pekare till sidtabell:



Varje post i sidtabellen (*page-table-entry*) ser ut på följande sätt:



där: *v* = *valid*, 1 anger att sidan finns i minnet

m = *modified*, 1 anger att sidan har modifierats

r = *referenced*, 1 anger att sidan refererats (lästs eller skrivits)

p = *protected*, 0 anger att sidan får skrivas

X = anger sidram

- Vad blir sidstorleken i systemet? (1p)
- Vad är det största logiska adressrum som en process kan tilldelas? (1p)
- Hur många bytes kan processens sidtabell uppta maximalt? (2p)
- Antag att en skrivning görs på adress 0x113284 (given på hexadecimal form), visa hur adressöversättningen går till. Ange också vilka kontroller som utförs. (6p)

Uppgift 7: (5p)

- Antag en uppsättning processer: {P1,P2 och P3} och en uppsättning resurser med vardera en instans: {R1,R2,R3,R4}, antag nu vidare följande scenario (i denna ordning); P1 begär R2; P2 begär R4; P3 begär R3; P1 begär R4; P2 begär R3; P3 begär R2. Illustrera den uppkomna situationen med en resursallokerings-graf, är detta en låsning? (3p)
- Antag nu att, i detta läge, ytterligare en instans av R2 tillförs systemet. Förklara vad som då händer. (2p)

Uppgift 8: (5p)

Förklara kortfattat följande begrepp:

- Overlay (1p)
- Swapping (1p)
- Dynamisk länkning (1p)
- 'thrashing' (2p)