

Örebro universitet
Institutionen för naturvetenskap och teknik
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

Tentamen i

Operativsystem för civilingenjörer

måndag 26 augusti 2019

Gäller som tentamen för:
DT513G Operativsystem för civilingenjörer, provkod 0100

Hjälpmedel:	Inga.
Poängkrav:	Maximal poäng är 38. För godkänt betyg krävs 20 poäng.
Resultat:	Meddelas på kursens hemsida eller via e-post senast måndag 16 september 2019.
Återlämning av tentor:	Elektroniskt via Studentforum.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070 - 73 47 013.

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara och tvetydiga formuleringar kommer att misstolkas.
 - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
 - Skriv bara på en sida av papperet. Använd inte röd skrift.
 - Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
 - Skriv gärna förklaringar om hur du tänkt. Även ett svar som är fel kan ge poäng, om det finns med en förklaring som visar att huvudtankarna var rätt.
-

LYCKA TILL!

Uppgift 1 (2 p)

Vad är ett operativsystem?

Uppgift 2 (3 p)

Vilka är operativsystemets viktigaste uppgifter?

Uppgift 3 (5 p)

Här är datorn **basen.oru.se**:



Det är en HP Microserver, med en 2-kärnig Celeron-processor utan hyperthreading. Den har 16 GB primärminne. Sekundärminnet består av en Samsung-SSD med 500 GB.

Att ha den som webb- och databasserver är tråkigt, så jag stänger av de serverprogrammen och använder i stället datorn för att simulera en massa monster som slåss med varandra, ungefär som i labben "monster world" i kursen.

Om jag skriver en flertrådad version av programmet, och gör på rätt sätt, kan jag få bättre prestanda än med en enkeltrådad version.

- Vad är den absolut högsta uppsnabbning jag kan få på den här datorn med en flertrådad version, jämfört med en enkeltrådad version?
- Varför?
- I praktiken får man inte samma uppsnabbning. Vad beror det på?
- Hur bör man skriva det flertrådade programmet för att få så hög uppsnabbning som möjligt?

Uppgift 4 (5 p)

Task-structen, eller PCB som den också kallas, är en av de viktigaste datastrukturerna som finns i operativsystemkärnan.

- a) Vad används den till?
- b) Vad innehåller den?
- c) Task-structen förekommer i flera olika köer i operativsystemkärnan. Vad är det för köer, och vad används de till?

Uppgift 5 (4 p)

- a) Vad är ett systemanrop?
- b) Beskriv hur systemanrop går till.

Uppgift 6 (4 p)

Trådar som tillhör samma process kan dela data, eftersom de har tillgång till samma minnesrymd, men det är svårare för processer att kommunicera, eftersom varje process har sin egen minnesrymd.

Beskriv några olika sätt för processer att kommunicera med varandra!

Uppgift 7 (4 p)

En vanlig sidstorlek ("page size" på engelska) är 4 kilobyte (4096 byte).

- a) Om man ska ange offset inom en sida, hur många bitar behövs?
- b) Om vi antar att en maskin använder virtuella 16-bitarsadresser, med sidor på 4096 byte, hur många bitar i den virtuella adressen blir över till sidnumret?
- c) Vi antar att vi utnyttjar hela den virtuella adressrymden, Vi antar också att vi har 4 gigabyte fysiskt minne, så det behövs 32 bitar för att adressera det. Hur mycket minne går åt till sidtabellen ("page table")?

Uppgift 8 (4 p)

Här är ett C-program för Linux:

```
#include <stdio.h>
#include <unistd.h>

int main(void) {
    printf("Hur många varv? ");
    int antal_varv;
    scanf("%d", &antal_varv);
    for (int i = 0; i < antal_varv; ++i) {
        printf("i = %d\n", i);
        fork();
    }
}
```

- a) Användaren matar in talet **2**. Vad skrivs ut när programmet körs?
- b) Användaren matar in talet **1000**. Du behöver inte ange precis vad som skrivs ut när programmet körs, men vad kommer att hända?
- c) Kan programmet påverka säkerheten i operativsystemet på något sätt? Vad kan operativsystemet göra åt det?

Uppgift 9 (7 p)

Ange för varje påstående om det är sant eller falskt! Lämna in denna sida tillsammans med resten av svaren. Fel svar ger inte minuspoäng.

Påstående	Sant	Falskt
I Linux kan programmet strace användas för att visa vilka systemanrop en process gör.		
Äldre operativsystem var baserade på avbrott (interrupts), så att när det till exempel fanns data att läsa från en disk så skickade disken ett avbrott till operativsystemkärnan. Ett avbrott medför att den körande processen stoppas, och kärnan tar över. I moderna operativsystem använder man i stället pollning, dvs att kärnan hela tiden ligger och kollar om det finns data. Med flerkärniga processorer blir det effektivare, eftersom man kan ha en pollningstråd på varje kärna.		
Systemanropet wait i Linux gör att processen stoppas det antal sekunder som man skickar som argument, och sedan fortsätter köra.		
Virtualisering innebär att trådar kan flyttas mellan olika processer.		
Demand paging innebär att varje process när den startas måste ange hur mycket minne (mätt i antal sidor, eller på engelska "pages") som den maximalt kommer att behöva. Om den skulle begära mer minne, kommer operativsystemet i stället att avsluta processen.		
När primärminnet blir fullt kan operativsystemet behöva kasta ut en del av sidorna ("pages"), och det finns flera olika algoritmer för att välja vilka som ska kastas ut. Det bästa är att kasta ut de sidor som det kommer att dröja längst tid innan de behövs, men man kan normalt inte veta vilka det är. Det näst bästa är att i stället använda en FIFO-kö, och kasta ut den sida som funnits i minnet längst.		
Thrashing är samma sak som garbage collection, eller skräpsamling på svenska, tillämpat på sidorna i primärminnet. Operativsystemet går igenom minnet, och kastar ut de sidor som inte längre går att nå från något av de körande programmen.		