Operativsystem för civilingenjörer: Hemtentamen 2020-08-24

Det här är hemtentan som går måndag 24 augusti 2020 i kursen DT513G Operativsystem för civilingenjörer vid Örebro universitet, provkod A001. Ansvarig lärare är Thomas Padron-McCarthy (thomas.padron-mccarthy@oru.se), telefon **070-73 47 013**.

Tid: 08:15 - 13:15

Instruktioner

1. Den här hemtentan ersätter den planerade salstentan, och är endast för de studenter som anmält sig till den tentan.
2. Uppgifterna ska lösas enskilt, dvs inga grupper av två eller flera studenter.
3. **Du får använda dator, böcker och vilka andra hjälpmedel som helst,** men du får inte samarbeta eller fråga någon (utom mig). Exempelvis är det tillåtet att söka och läsa på webbplatser som Stack Overflow, men inte att ställa egna frågor. Du kan alltså provköra kommandon och program, om du vill.
4. **Diskutera inte uppgifterna eller dela med dig av svar förrän tidigast dagen efter tentan.**
5. Lös de angivna uppgifterna och samla svaren på lämpligt sätt, till exempel i en PDF. Skicka sen in lösningarna till mig i Blackboard. Välj **Kursmaterial** i menyn till vänster på kursens Blackboard-sida, gå in i mappen **Hemtentamen**, och välj den rätta tentan. Där kan man sedan välja att skicka in sin lösning. Då blir bedömningen anonym. Glöm inte att klicka på knappen **Skicka** längst ner till höger.
6. Om det skulle bli problem med Blackboard, kan man i nödfall skicka in svaren antingen som ett kursmeddelande i Blackboard eller via vanlig e-post (thomas.padron-mccarthy@oru.se), senast vid tentatidens slut. Då blir bedömningen inte anonym.
7. I Blackboard ser du att du skickat in din lösning. Om du i stället skickade in med e-post, och inte senast en timme efter tentatidens slut fått ett svar från mig med en bekräftelse på att du skickat in svaren, bör du kontakta mig, enklast genom att ringa eller SMS:a mig (ifall det är e-posten som inte fungerar). Tänk på att en del mailtjänster (särskilt Microsoft-tjänster som Hotmail.com, Outlook.com och Live.com) ibland kastar bort brev med bilagor, utan att meddela det.
8. Skriv gärna svaren i ett ordbehandlingsprogram. Rita gärna eventuella diagram i ett ritprogram. Det är inte förbjudet att skriva och rita för hand, men då måste text och bilder scannas in eller fotograferas. Det finnas scanner-appar till Android och iPhone, till exempel Adobe Scan, som ger

bättre resultat än att bara ta vanliga kort med kameran.

9. Tentatiden är utökad med en extra timme för att täcka in problem med e-post, inscanning eller fotografering av diagram, och liknande.
10. Om du behöver fråga något, så kontakta gärna mig. Ring eller skicka SMS, för jag kanske inte kommer att sitta vid datorn hela tiden.
11. Oklara och tvetydiga formuleringar kommer att misstolkas. Lösningar som inte går att läsa eller förstå kan naturligtvis inte ge några poäng.
12. Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
13. 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.
14. Maximal poäng är 26. För godkänt betyg krävs minst 13 poäng.
15. Resultat meddelas senast 15 arbetsdagar efter tentamensdatum. Eftersom svaren skickas in elektroniskt scannas tentorna inte för retur.

Uppgift 1 (5 p)

Behöver alla datorerer ha operativsystem? Diskutera!

Uppgift 2 (5 p)

Operativsystemkärnan kan ha olika arkitekturer. Även om man numera oftast kombinerar dem, är extremerna en mikrokärna ("microkernel" på engelska) och en monolitisk kärna ("monolithic kernel").

a) Vad innebär en monolitisk kärna?

b) Vad innebär en mikrokärna?

c) Vilka fördelar och nackdelar har de två?

d) Jag skrev ovan att man numera oftast kombinerar dem. Hur går det till? Vilka fördelar ger det?

Uppgift 3 (6 p)

Vi konstruerar en processor där fysiska adresser är 20 bitar, medan virtuella adresser är 36 bitar, varav 23 bitar är sidnummer ("page number" på engelska) och 13 bitar är offset.

a) Hur mycket fysiskt minne kan man adressera? Visa också hur du räknat.

b) Hur mycket virtuellt minne kan man adressera? Visa också hur du räknat.

c) Hur stora är minnessidorna? Visa också hur du räknat.

d) Hade det varit rimligt att ha en offsetdel som var 12 bitar i stället för 13? Motivera svaret.

e) Hade det varit rimligt att ha en offset som var 3 bitar? Motivera svaret.

f) Hade det varit rimligt att ha en offset som var 34 bitar? Motivera svaret.

Uppgift 4 (10 p)

Här nedan finns ett program, threads.c, som startar 100 trådar, och var och en av trådarna ökar variabeln **the_big_number** en miljard gånger. Slutresultatet borde förstås bli att variabeln innehåller värdet 100 miljarder.

> Vi antar att alla provkörningar i den här uppgiften görs på en normal, modern dator. Det kan vara en stationär dator eller en bärbar dator. Det är tillåtet att provköra på riktigt när man löser uppgiften, men

det går att besvara alla frågorna även utan att provköra. Man måste i vilket fall som helst förklara sina svar. Det är förklaringarna som är det viktiga.

```
// Many threads, each incrementing the same variable many times
// Note 1: Normal "int" is too small for some of the values,
// so we use "long long int".
// Note 2: An optimizing compiler might remove the actual calculations,
// so don't compile with -O2 or similar.
// Thomas Padron-McCarthy (thomas.padron-mccarthy@oru.se)
// Fri Aug 21 10:22:16 CEST 2020

#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <pthread.h>

#define TOTAL_NR_INCREMENTS (1LL*1000*1000*1000)
#define NR_THREADS 100
#define NR_INCREMENTS_PER_THREAD (TOTAL_NR_INCREMENTS / NR_THREADS)

pthread_t threads[NR_THREADS];

long long int the_big_number = 0;

void *thread_body(void *arg) {
    int thread_number = (int)arg;
    for (long long int i = 0; i < NR_INCREMENTS_PER_THREAD; ++i) {
        the_big_number ++;
    }
    return NULL;
} // thread_body

int main(void) {
    printf("This program increments the same variable %lld times,\n",
           TOTAL_NR_INCREMENTS);
    printf("using %d thread(s), each incrementing the variable %lld times.\n",
           NR_THREADS, NR_INCREMENTS_PER_THREAD);

    printf("Starting the %d threads...\n", NR_THREADS);

    for (int i = 0; i < NR_THREADS; ++i) {
        if (pthread_create(&threads[i], NULL, thread_body, (void*)i) != 0) {
            printf("Couldn't create thread %d.\n", i);
            exit(EXIT_FAILURE);
        }
    }

    printf("%d threads started. Waiting for them to finish...\n", NR_THREADS);

    for (int i = 0; i < NR_THREADS; ++i) {
        if (pthread_join(threads[i], NULL) != 0) {
            printf("Couldn't join thread for thread %d.\n", i);
            exit(EXIT_FAILURE);
        }
    }
```

```
    printf("All threads finished!\n");
    printf("The variable now has the value = %lld.\n", the_big_number);
    printf("It should be %lld.\n", TOTAL_NR_INCREMENTS);
    if (the_big_number == TOTAL_NR_INCREMENTS)
        printf("Ok! As expected!\n");
    else
        printf("Error! Wrong result!\n");

    return EXIT_SUCCESS;
} // main
```

a) Vi kör programmet. Det tar 2,3 sekunder, men variabelns värde blir inte alls 100 miljarder, utan 37226719. Vad beror det på att det blir fel värde?

b) Vi kör programmet en gång till. Det tar 2,3 sekunder den här gången också, men nu blir variabelns värde 39579883, alltså ett annat värde än förra gången. Vad beror det på att det blir ett annat värde?

c) Vi ändrar konstanten **NR_THREADS** till **1**, så det bara körs en enda tråd, och kör programmet ännu en gång. Vad får variabeln för värde nu? Varför?

d) Vi ändrar tillbaka konstanten **NR_THREADS** till **100**. Ändra programmet så det ger rätt resultat, även med 100 parallella trådar! (Det är som sagt tillåtet att provköra, men det räcker med att visa vilka rader som behöver läggas till i programmet.)

e) Vi kör programmet från deluppgift **d** ovan. Om du gjort rätt får variabeln det förväntade värdet. Men ungefär hur lång tid kan man räkna med att det tar att köra programmet? Varför?

f) Jag vill att programmet ska gå fortare att köra, så jag ändrar konstanten **NR_THREADS** till **1000**. Vad händer med körtiden jämfört med när det var 100 trådar? Varför?

Thomas Padron-McCarthy (thomas.padron-mccarthy@oru.se), 21 augusti 2020