

i **Kopi av Eksamensinformasjon**

Fakultet: Teknologi, kunst og design
Utdanning: Teknologiske Fag
Emnenavn: Operativsystemer
Emnekode: DATA2500

Bokmål

Dato: 30 juli 2020

Tid: kl. 09.00-12.00 (3 timer)

Hjelpemidler:

Alle hjelpemidler er tillatt.

Andre opplysninger:

Les nøye gjennom oppgavene før du begynner og pass på å besvare alle spørsmålene.

Sett gjerne egne forutsetninger dersom du synes oppgaveteksten er uklar. Beskriv forutsetningene og løs oppgaven utifra dem.

Maks poeng på de fleste av deloppgavene er 10 og de bidrar 3.33 % hver til sluttpoengsummen som maksimalt er 300. Men legg merke til at det er noen unntak:

Oppgave 2 om datamaskinarkitektur gir inntil 20 poeng.

Følgende oppgaver gir inntil 30 poeng:

Oppgave 23 om bash-scripting

Oppgave 25 om PowerShell-scripting

1 Kopi av Datamaskinarkitektur

Hva må input til en OR-port være om output skal bli 0?

Velg ett alternativ:

- ☐ false
- ☐ 1
- ☐ 0
- ☐ 01
- ☐ true
- ☐ 10
- ☐ 11
- ☐ 00

Maks poeng: 10

2 Kopi av Datamaskinarkitektur

På en Linux-server har du følgende C++ program som du kaller **sum.cpp**. Det leser heltall fra en fil med navn **tall.txt** som inneholder 20000 tilfeldige og usorterte heltall mellom 0 og 20000 og summerer ved hjelp av en if-test opp alle tallene som er større enn 10000:

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int i,c;
    int arraySize = 20000;
    int data[arraySize]; // deklarerer et data-array

    // Leser tall fra fil inn i data-array
    ifstream in("tall.txt");
    for (c = 0; c < arraySize; ++c)
    {
        in >> data[c];
    }
    in.close();
    // Legger sammen alle tall større enn 10000
    long sum = 0;

    // Ytre løkke for at kjøringen skal ta litt tid
    for (i = 0; i < 10000; ++i)
    {
        // Indre løkke
        for (c = 0; c < arraySize; c++)
        {
            if (data[c] > 10000)
                sum += data[c];
        }
    }
}
```

```
    cout << "sum = " << sum << "\n";  
}
```

Du kompilerer programmet og tar tiden på hvor lang tid det bruker på å kjøre med

```
Linux$ g++ sum.cpp  
Linux$ time ./a.out  
sum = 1504654300000  
Real:1,071 User:1,071 System:0,000 100,01%
```

Deretter sorterer du numerisk tallene som programmet leser inn og tar på nytt tiden på programmet:

```
Linux$ sort -n tall.txt > sort.txt  
Linux$ cp sort.txt tall.txt  
Linux$ time ./a.out  
sum = 1504654300000  
Real:0,399 User:0,393 System:0,004 99,54%
```

Ta utgangspunkt i det du har lært om datamaskinarkitektur og forklar kort hvorfor det går mer enn dobbelt så raskt å legge sammen alle tall som er større enn 10000 i det andre tilfellet til tross for at programmet i begge tilfeller utfører nøyaktig like mange instruksjoner.

Skriv ditt svar her

Format

B

I

U

$\times_e$

$\times^2$

$\int_x$

























Words: 0

Maks poeng: 20

3 Kopi av Threads

Hvilket utsagn om fordelene ved threads (tråder) er **IKKE** riktig?

Velg ett alternativ:

- ☐ Ressursdeling: flere prosesser deler på kode, data og delvis PCB innenfor samme tråd.
- ☐ Respons: interaktive applikasjoner kan ha en tråd med høy prioritet som kommuniserer med brukere og lavprioritetstråder som gjør grovarbeid.
- ☐ Multiprosessering: hver tråd kan tildeles en egen CPU
- ☐ Felles variabler: enkelt å sette opp for threads
- ☐ Effektivitet: tar mindre tid å lage nye threads og mindre tid å context-switche mellom threads

Maks poeng: 10

4 Kopi av Prosesser

Hvordan gjør et moderne OS det mulig å kjøre flere prosesser samtidig på én CPU?

Velg ett alternativ:

- ☐ Bruker en hardware timer til å gi begrensede tidsintervall til brukerprosessene
- ☐ Hver prosess gir frivillig fra seg tilgangen til CPU til en annen prosess etter en timeslice
- ☐ Venter til prosessen som kjører gjør I/O og gir da CPU'en til neste prosess i ready-list
- ☐ Overlater timesharingen til hardware-algoritmer
- ☐ Kjørør en while-løkke som plukker prosesser fra ready-list og sender med en tidsparameter til hver prosess som startes

Maks poeng: 10

5 Kopi av Prosesser

Når du kjører en CPU-intensiv regnejobb, hva slags modus er for det meste CPU'en i?

Velg ett alternativ:

- ☐ running mode
- ☐ process mode
- ☐ memory mode
- ☐ normal mode
- ☐ user mode
- ☐ kernel mode
- ☐ brute force mode

Maks poeng: 10

6 Kopi av Serialisering

Hvilken av de følgende begrepene er **IKKE** en metode eller løsning for å serialisere tråder eller prosesser og unngå at en race condition oppstår?

Velg ett alternativ:

- ☐ Semaforer
- ☐ Spinlocks
- ☐ lock
- ☐ synchronized()
- ☐ EnterCriticalSection()
- ☐ Atomiske operasjoner
- ☐ blockRaceCondition()
- ☐ testAndSet()

Maks poeng: 10

7 Kopi av Serialisering

Hva er busy-waiting?

Velg ett alternativ:

- ☐ At en prosess utfører CPU-intensive kode mens den ligger i en I/O-kø
- ☐ At en prosess blir lagt i en wait-kø
- ☐ At en prosess venter i en while-løkke til en hendelse inntreffer
- ☐ At en prosess venter på et interrupt som er generert av data-bussen
- ☐ Metode for å unngå at prosesser som er veldig viktige ikke ligger i samme kø som andre prosesser
- ☐ Metode for å unngå at det oppstår deadlock

Maks poeng: 10

8 Kopi av Internminne

Du gjør følgende kommando på en Linux-server:

```
linux$ getconf PAGE_SIZE  
4096
```

Hvis en prosess på denne serveren bruker en mebibyte (1 MiB) interminne, hvor mange pages (sider) utgjør dette?

Velg ett alternativ:

- ☐ 512
- ☐ 32
- ☐ 2048
- ☐ 16
- ☐ 128
- ☐ 256
- ☐ 4096
- ☐ 1024
- ☐ 64

Maks poeng: 10

9 Kopi av Internminne

Hva er en dirty page?

Velg ett alternativ:

- ☐ En side som har blitt endret slik at den må skrives til disk om den må ut av minnet
- ☐ En side som har en annen TLB referanse enn den som ligger i page-tabellen i RAM
- ☐ En side som på grunn av en RAM-feil har fått endret innhold
- ☐ En side som inneholder informasjon av støtende natur og som derfor fjernes av OS
- ☐ En side som har blitt delvis overskrevet av en annen side når den lå i cache på grunn av en race condition

Maks poeng: 10

10 Kopi av Internminne

Hvorfor kan ikke MMU implementeres i software og styres av OS?

Velg ett alternativ:

- ☐ Fordi RAM må kontrolleres av hardware, OS kan kun styre CPU
- ☐ Fordi det ikke ville være plass i RAM til hele page-tabellen
- ☐ Fordi det da ikke ville være mulig å bruke cache
- ☐ Fordi minne-aksess da ville ha tatt alt for lang tid
- ☐ Fordi også vanlige brukerprosesser må kunne aksessere RAM
- ☐ Fordi man da ikke kunne finne mappingen for en virtuell adresse siden den ligger i fysisk RAM

Maks poeng: 10

11 Kopi av Kompilere C-program

De neste 14 oppgavene er relatert til samme problemstilling, men selvom man ikke får til eller forstår noen av de første oppgavene kan man likevel klare å løse oppgavene som kommer etter disse.

Du har en Linux-server med Docker installert og ønsker å kjøre en applikasjon som starter opp flere uavhengige samtidige prosesser og ønsker å teste ut om Docker-containere fullt ut kan utnytte alle de 4 CPU'ene som serveren din har. Siden du har lest mange steder at man bare skal ha en enkelt tjeneste eller prosess per container, ønsker du å finne ut om en Docker-container kun bruker en enkelt CPU når den kjører eller om den kan utnytte alle serverens CPUer. I denne og de følgende oppgavene prøver du å finne et svar på dette.

For å teste ut dette starter du med å finne et C-program på nettet som heter **prime.c**, som regner ut antall primtall under en viss størrelse og skriver ut resultatet og som er 100% CPU-avhengig. Skriv ned Linux-kommandoer som kompilerer C-programmet optimalisert for at det skal kjøres raskest mulig og som så tar tiden på hvor lang tid programmet bruker på å kjøre på Linux-serveren din.

Skriv ditt svar her

Format | B | I | U | x₂ | x² | I_x | | | | | | | | | | |

Words: 0

Maks poeng: 10

12 Kopi av Docker

Du skriver så følgende Dockerfile for å ta tiden på programmet i en container:

```
root@osG70:~/prime# cat Dockerfile
FROM ubuntu
COPY prime.c prime.c
RUN gcc -O prime.c
RUN time ./a.out > res.txt
```

Forklar kort hva hver linje betyr og angi en docker-kommando som bygger dette imaget og gir det navnet **prime**.

Skriv ditt svar her

Format | **B** | *I* | U | x₂ | x² | I_x | | | | | | | | | | | | |

Words: 0

Maks poeng: 10

13 **Kopi av Docker**

Du kjører nå docker-kommandoen for å bygge imaget og du får følgende output:

```
Sending build context to Docker daemon 32.26kB
Step 1/4 : FROM ubuntu
latest: Pulling from library/ubuntu
a4a2a29f9ba4: Pull complete
127c9761dcba: Pull complete
d13bf203e905: Pull complete
4039240d2e0b: Pull complete
Digest: sha256:35c4a2c15539c6c1e4e5fa4e554dac323ad0107d8eb5c582d6ff386b383b7dce
Status: Downloaded newer image for ubuntu:latest
--> 74435f89ab78
Step 2/4 : COPY prime.c prime.c
--> 78f39f910b62
Step 3/4 : RUN gcc -O prime.c
--> Running in cd8b1323c69b
/bin/sh: 1: gcc: not found
The command '/bin/sh -c gcc -O prime.c' returned a non-zero code: 127
```

Forklar kort hva denne feilen skyldes og hvordan du kan endre Dockerfile slik at byggingen vil fungere.

Skriv ditt svar her

Format | - | B | I | U | x_o | x² | I_x | [] | ↶ | ↷ | ⌵ | ⌶ | Ω | [] | ✎ | Σ | [] | ✖

Words: 0

Maks poeng: 10

16 Kopi av Bash-script

Skriv et lite bash-script **run.sh** som ved hjelp av eksperimenteringen fra forrige oppgave tar tiden på programmet **a.out** og skriver resultatet av kjøringen til filen **res.txt** og tiden det tar til filen **tid.txt**. Først i scriptet skal environment-variabelen **\$TIMEFORMAT** gis verdien "Real:%R User:%U System:%S %P%%".

Skriv ditt svar her

1	
---	--

Maks poeng: 10


```
root@osG70:~/prime# docker container ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
31c5c5fbde8b	prime	"/run.sh"	46 seconds ago	Up 44 seconds	goofy_stallman

Forklar kort hva disse fire kommandoene gjør og hva de viser. Kan du ut ifra disse resultatene si noe om hvor effektivt det er å kjøre en CPU-avhengig prosess i en container sammenlignet med å kjøre den direkte på en server?

Maks poeng: 10

20 Kopi av Bash-script

Nå har du endelig kommet så langt at du kan prøve å kjøre fire prosesser samtidig inne i en container og se om de utnytter alle de fire CPUene som serveren har. Skriv en ny versjon av scriptet **run.sh** som er slik at det starter fire uavhengige og samtidige prosesser som kjører **a.out** og som skriver resultatene til hver sin fil. Etter at scriptet har kjørt skal det derfor lages fire filer **res1.txt**, **res2.txt**, **res3.txt** og **res4.txt** og tilsvarende fire filer **tid1.txt**, **tid2.txt**, **tid3.txt** og **tid4.txt**.

Skriv ditt svar her

1	
---	--

Maks poeng: 10

21 **Kopi av Linux kommandolinje**

Du bygger en ny container med den nye parallelle versjonen av **run.sh**, kjører den og får følgende output når du lister den:

```
root@osG70:~/prime# docker container ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    NAMES
4d89632bba3a   prime    "/run.sh"  37 seconds ago  Up 35 seconds  xenodochial_agnesi
```

Skriv en Linux-kommando som bruker en for-løkke til å skrive ut innholdet av de fire filene tid1.txt, tid2.txt, tid3.txt og tid4.txt i denne containeren (du kan om du ønsker det skrive kommandoen på tre eller fire linjer slik du ville gjort det i et script og slik du kunne ha gjort på kommandolinjen). Bruk alternativt en annen metode og skriv en alternativ kommando for å oppnå det samme.

Skriv ditt svar her

Format

B

I

U

x_o

x^o

I_x

Words: 0

Maks poeng: 10

```
Real:14.324 User:14.192 System:0.049 99.42%
Real:14.394 User:14.211 System:0.045 99.04%
Real:14.579 User:14.474 System:0.021 99.42%
Real:14.238 User:14.127 System:0.058 99.62%
```

```
root@osG70:~/prime#
Real:14,323 User:14,227 System:0,067 99,80%
Real:14,280 User:14,241 System:0,031 99,94%
Real:14,421 User:14,331 System:0,083 99,95%
Real:14,436 User:14,349 System:0,083 99,98%
```

Skriv ditt svar her

23 Kopi av Bash-script

```
$ vid=29583758204
$ token=30ab56c57d37f244db35a23c4b
$ url=https://owner-api.teslamotors.com/api/1/vehicles/$vid/data_request/drive_state
$ curl -s -X GET $url -H "Authorization: Bearer $token" | jq '.response | .speed'
```

19/26

```
$ time curl -s -X GET $url -H "Authorization: Bearer $token" | jq '.response | .speed'  
23  
Real:1.322 User:0.048 System:0.012 4.53%
```

Du ønsker å finne ut hvor fort din kone og andre som bruker bilen kjører (se i denne oppgaven bort fra moralske skrupler). Skriv et bash-script som ved hjelp av kommandoene over kjører i en evig løkke og skriver ut så ofte som mulig hastigheten til bilen din med et tidsstempel for hver måling til loggfilen **tesla.log**. Resultatet kan se ut tilsvarende som dette etter at programmet har kjørt noen sekunder:

```
$ cat tesla.log  
5 ma. 27. juli 11:06:04 +0200 2020  
5 ma. 27. juli 11:06:05 +0200 2020  
5 ma. 27. juli 11:06:06 +0200 2020  
9 ma. 27. juli 11:06:08 +0200 2020  
15 ma. 27. juli 11:06:09 +0200 2020  
20 ma. 27. juli 11:06:11 +0200 2020  
21 ma. 27. juli 11:06:12 +0200 2020
```

Hvis det finnes noe gammelt innhold i **tesla.log** før scriptet startes, skal det først fjernes.

Skriv ditt svar her

1	
---	--

Maks poeng: 30

25 Kopi av PowerShell

Du har et PowerShell-script **run1.ps1** som du tar tiden på med følgende kommando:

```
PS C:\Users\los> Measure-Command{.run1.ps1}
```

```
Days       : 0
Hours      : 0
Minutes    : 0
Seconds    : 13
Milliseconds : 129
Ticks      : 131299664
TotalDays  : 0,000151967203703704
TotalHours : 0,003647212888888889
TotalMinutes : 0,218832773333333
TotalSeconds : 13,1299664
TotalMilliseconds : 13129,9664
```

Du har også et annet PowerShell-script som heter **run2.ps1** og som bruker litt mindre tid på å kjøre. Skriv et PowerShell-script **diff.ps1** som tar tiden på begge run-scriptene og skriver ut hvor mye mer tid **run1.ps1** bruker sammenlignet med **run2.ps1**. Det kan se slik ut når du kjører scriptet:

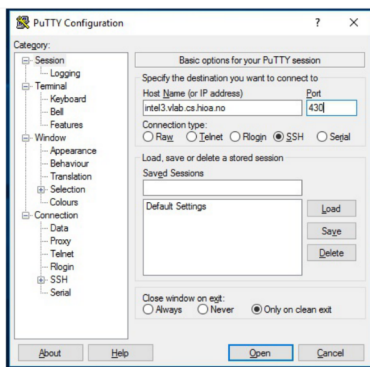
```
PS C:\Users\los> .diff.ps1
run1.ps1 bruker 0.2018713 sekunder mer enn run2.ps1
PS C:\Users\los>
```

Skriv ditt svar her

1	
---	--

Maks poeng: 30

26 Innlogging på Linux-VM



Disse oppgavene er laget spesielt for prøve-eksamen 2021 og kommer i tillegg til de oppgavene som var en del av korte-eksamen 2020. På ordinær eksamen vil det derfor være noe reduksjon av den 'vanlige' typen av oppgaver.

På StudentWeb har du fått oppgitt et kandidatnummer, det kan for eksempel være et tall som 430 og vi vil bruke dette tallet som eksempel, men bytt ut 430 med ditt eget kandidatnummer når du gjør disse oppgavene.

På serveren intel3.vlab.cs.hioa.no har hver kandidat sin egen Linux-VM som man som en del av eksamen skal logge inn på for å løse oppgaver (disse er helt like Linux-VMene som dere brukte under kurset, med navn som os13, og de er også egentlig Docker-containere). Alle disse Linux-VMene har forskjellig innhold, slik at svaret på noen av oppgavene vil være forskjellig for hver kandidat. For eksempel vil svaret på spørsmål som "Totalt hvor mange filer finnes det på din Linux-VM" variere for hver kandidat, fordi det ikke ligger samme antall filer på hver VM.

Anta at du er kandidat nummer 430. Da må du først laste ned riktig passord fra denne web-siden:

<https://www.cs.hioa.no/~haugerud/os.php>

Skriv inn kandidatnummeret (ditt eget og ikke 430) og trykk på 'Send inn nummeret'-knappen. Hvis du skriver inn et gyldig kandidatnummer får du da et passord som er av samme type som for os-VMene og kan være noe lignende som **bien6tuft**. Det er viktig at du bare henter passord for din egen VM, hvilken IP som henter hvilket passord blir loggført. Det er viktig at du ikke går inn på andre kandidaters servere.

Logg så inn på din VM med kommandoen

```
ssh -p 430 kan430@intel3.vlab.cs.hioa.no
```

hvis du logger inn fra et bash-shell (Mac eller Linux) på din hjemme-PC og bruk så passordet du fikk fra websiden os.php på den samme hjemme-PCen (se instruksjoner for putty fra Windows lenger ned). Opsjonen -p 430 betyr at du bruker port-nummer 430 og ikke standard ssh-portnummer som er 22. Dette er laget slik for at hver kandidat skal nå sin egen VM. kan430 er brukernavnet du logger inn med. For både port-nummeret og brukernavnet må du bytte ut 430 med ditt eget kandidatnummer.

Hvis du logger deg på direkte fra putty (se figur) må du bruke Host Name intel3.vlab.cs.hioa.no, skriv inn ditt kandidatnummer som port number (430 i eksempelet), bruk brukernavn kan430 og passordet du lastet ned (igjen, bytt ut 430 med ditt kandidatnummer).

For å svare på denne oppgaven, skriv inn følgende:

Kandidatnummer

Passordet du hentet fra websiden os.php

"Din IP-adresse" som du fikk fra websiden os.php

Innholdet av filen file1 som ligger øverst i hjemme-mappen til din bruker (kan430)

27 Fil på Linux-VM

I filsystemet til brukeren du logget på med på Linux-VM ligger det en tekst-fil med navn **file2**. Finn denne filen, kopier og paste ut innholdet av filen her:

24/26

28 **Filer på Linux-VM**

I filesystemet til brukeren du logget på med på Linux-VM er det en mappe som heter **files**. Hvor mange filer inneholder denne mappen?

Skriv ditt svar her

Format

-

B

I

U

x₂

x²

I_x

Words: 0

Maks poeng: 5

29 **xfile på Linux-VM**

I hjemme-mappen til brukeren root ligger det en fil med navn **xfile**. Kopier og paste ut innholdet av denne tekstfilen her:

Skriv ditt svar her

Format

-

B

I

U

x₂

x²

I_x

Words: 0

Maks poeng: 5