

i

Eksamensinformasjon

Bokmål
Eksamensoppgave

Operativsystemer (DATA2500)

Bokmål
31. juli 2018
kl. 9.00-12.00

Hjelpemidler:
Ingen 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 teller 50 poeng, oppgave 5(b) teller 40, oppgave 4(a) teller 8 og oppgave 4(b) teller 12 poeng.

1(a)

Linux kommandolinje

Hvilken kommando flytter filen **/tmp/fil.txt** til katalogen du står i?

Velg ett alternativ

- ☐ mv . /tmp/fil.txt
- ☐ cp -mv /tmp/fil.txt .
- ☐ mv /tmp/fil.txt .
- ☐ mv /tmp/fil.txt
- ☐ rm -r /tmp/fil.txt
- ☐ move /tmp/fil.txt \$HOME
- ☐ cp /tmp/fil.txt ~

Maks poeng: 10

1(b)

Linux kommandolinje

Hvilke brukere kan skrive til filen **/tmp/secret.txt**?

-rw-r----- 1 haugerud drift 9 Sep 5 15:53 /tmp/secret.txt

- ☐ Ingen
- ☐ Brukeren **haugerud** og alle brukere i gruppen **drift**
- ☐ Alle
- ☐ Kun de som er med i gruppen **drift**
- ☐ Kun brukeren **haugerud**
- ☐ Brukeren **drift** og alle brukere i gruppen **haugerud**
- ☐ Alle andre (enn brukeren **haugerud** og alle brukere i gruppen **drift**)

Maks poeng: 10

1(c) **Linux kommandolinje**

Gitt følgende fil med navn **index.html**:

```
$ cat index.html
<!DOCTYPE html>

<html itemscope itemtype="http://schema.org/Article" xmlns="http://www.w3.org/1999/xhtml" xml:lang="no"
lang="no">
<head>
<title>Hjem - HiOA</title>
<meta name="Content-Type" content="text/html; charset=utf-8" />
<meta name="Content-language" content="no-bokmaal" />

<meta name="MSSmartTagsPreventParsing" content="TRUE" />
<meta name="generator" content="eZ Publish" />
<meta name="title" content="Hjem" />
<meta itemprop="name" content="Hjem">
<meta name="author" content="" />
<meta name="p:domain_verify" content="3a11dc5033d5012ab4dce032cb87920c"/>
```

Hvilken **kommando** gir følgende output?

\$ **kommando**

eZ Publish
Hjem
Hjem

3a11dc5033d5012ab4dce032cb87920c

Velg ett alternativ

- ☐ tail -n 5 index.html | grep name | cut -d\" -f 4
- ☐ cat index.html | cut -f 5 | grep name | awk '{print \$2}'
- ☐ grep name index.html | head -n 5 | cut -d\" -f 2
- ☐ cat index.html | grep name | tail -n 5

Maks poeng: 10

1(d) **Linux kommandolinje**

Du har en fil **biler** med følgende innhold:
\$ **cat biler**
student bmw 500000
haugerud berlingo 150000
kyrre elbil 90000

Hvilken Linux-**kommando** fører til følgende resultat?

\$ **kommando**
kyrre elbil 90000
haugerud berlingo 150000
student bmw 500000

Velg ett alternativ

- ☐ sort -n biler
- ☐ sort -k 2 biler
- ☐ sort -r -k 1 biler
- ☐ sort -n -r -k biler
- ☐ sort biler
- ☐ sort -n -k 3 biler

Maks poeng: 10

2 **Bash-scripting**

På en Linux-maskin gir følgende kommando noen linjer med output:

```
haugerud@lap:~$ grep cpu /proc/stat
cpu 342864 2411 54965 9777938 3427 0 1109 0 0 0
cpu0 45600 208 5992 1892824 937 0 566 0 0 0
cpu1 43779 211 6144 1127900 331 0 331 0 0 0
cpu2 43240 404 5631 1127493 353 0 130 0 0 0
cpu3 45704 311 5820 1126367 377 0 51 0 0 0
cpu4 42122 308 6012 1124005 328 0 13 0 0 0
cpu5 41971 467 7858 1127390 309 0 9 0 0 0
cpu6 42817 263 5402 1127460 416 0 4 0 0 0
cpu7 37628 236 12103 1124495 374 0 2 0 0 0
```

(det er to mellomrom mellom **cpu** og **342864** i første linje)
De 10 tallene i første linje er totalt antall jiffies for Linux-maskinen i user mode, nice, system mode, idle og så videre. Verdiene i **/proc/stat** oppdateres kontinuerlig. Skriv et bash-script **proc.sh** som skriver ut hvor mange nye jiffies som har blitt lagt til i løpet av det siste sekundet for en bestemt av kolonnene. Scriptet skal gjenta denne operasjonen hvert sekund og skrive ut verdien 30 ganger og dermed avslutte etter 30 sekunder. Scriptet skal ta ett tall som argument og dette tallet skal tolkes som hvilken av de 10 tall-kolonnene som data skal hentes fra. Et eksempel på starten av en kjøring av scriptet kan se slik ut:

```
haugerud@lap:~$ ./proc.sh 4
798
798
799
800
799
797
799
```

Skriv ditt svar her...

3(a) Prosesser

Skriv ditt svar her...

4/14

3(b) Prosesser

Forklar kort hvordan operativsystemet fordeler CPU-tid mellom fem 100% CPU-intensive prosesser som kjører på en Linux server med 4 kjerner/cores (4 uavhengige ALUer). Forklar spesielt omtrent hvor stor CPU-prosent kommandoen **top** vil vise for hver prosess når alle de fem prosessene kjører samtidig.

Skriv ditt svar her...

Maks poeng: 10

3(c) Prosesser

Et 100% CPU-avhengig sekvensielt regneprogram bruker 4 minutter når det kjøres på en enkelt CPU-core på en Linux server. Hvis fem slike program startes samtidig på samme server og den totalt har 4 CPU-cores (4 uavhengige ALUer), hvor lang tid tar det før den siste regneprosessene er ferdig? (Ingen andre CPU-krevende prosesser enn disse fem kjøres samtidig)

Velg ett alternativ

- ☐ 8 minutter
- ☐ 6 minutter
- ☐ 7 minutter
- ☐ 5 minutter
- ☐ 4 minutter
- ☐ 20 minutter

Maks poeng: 10

3(d) Prosesser

Når man ikke bruker fortegn, hva er det største tallet man kan lagre i et 5 bits register?

Velg ett alternativ

- 127
31
63
5
255
15

Maks poeng: 10

3(e) Prozessor

Når en variabel defineres i et C-program som **int**, settes det av 4 byte til å lagre tallet. Når ett bit brukes til lagre fortegn, hvor stort er det største tallet som kan lagres, uttrykt i potenser av 2? Omtrent hvor stort er dette tallet uttrykt i desimaltall?

Skriv ditt svar her...

Maks poeng: 10

3(f) Prosesser

Anta at du har følgende C-program med navn **sum.c**:

```
#include <stdio.h>
void main()
{
    int i, sum = 0;
    int N = 40000;
    for(i=1;i<=N;i++)
    {
```

DATA2500 konte 2018

```
    sum += i;
}
printf("Sum: %d\n",sum);
}
```

Forklar kort hva programmet regner ut og hvordan du kompilerer og kjører dette programmet på en Linux server.

Skriv ditt svar her...

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

Words: 0

Maks poeng: 10

3(g) **Prosesser**

Du kjører programmet og får resultatet

Sum: 800020000

Så endrer du variabelen **N** til **N = 50000** og får resultatet

Sum: 1250025000

og med **N = 60000** får du

Sum: 1800030000

Til slutt endrer du verdien til **N = 70000** og får resultatet

Sum: -1844932296

Forklar kort hva som gjør at beregningene av summen feiler fullstendig på den siste kjøringen med **N = 70000** .

Skriv ditt svar her...

Maks poeng: 10

3(h) **Prose**

Anta at du kjører det opprinnelige C-programmet med **N = 40000** på en Linux server med 4 CPU'er. Forklar kort hvorfor operativsystemet ikke kan utnytte flere enn en CPU når dette programmet kjøres for å regne ut resultatet raskere.

Skriv ditt svar her...

Maks poeng: 10

3(i) Prosesser

Anta igjen at du har det samme C-program med **N = 40000**. Du kjører programmet på en Linux server med 4

CPU`er. Forklar kort hvordan du som programmerer kan endre programmet slik at operativsystemet kan utnytte alle de fire CPU`ene når dette programmet kjøres for å regne ut resultatet nesten fire ganger raskere (ikke skriv kode, men forklar hvordan du i prinsippet vil endre programmet).

Skriv ditt svar her...

Format

-

B

I

U

x₂

x²

I_x

↶

↷

↺

☰

☷

Ω

Σ

ABC

Words: 0

Maks poeng: 10

3(j) **Prosesser**

Anta at du har et C-program som regner ut Fibonacci-rekken: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, så langt som mulig. I denne rekken er neste tall summen av de to foregående. Forklar kort hvorfor du som programmerer ikke kan endre dette programmet slik at det kan utnytte alle fire CPU`er på en Linux server for å regne ut resultatet raskere.

Skriv ditt svar her...

Format

-

B

I

U

x₂

x²

I_x

↶

↷

↺

☰

☷

Ω

Σ

ABC

Words: 0

Maks poeng: 10

9/14

4(a) Internminne

Velg hvilket tall som tilsvarer hvilke 2er og 10er eksponenter (under er tegnet ^ brukt for å markere eksponent, $10^3 = 10^3$ og $2^{10} = 2^{10}$).

Finn de som passer sammen

	10^6	10^3	2^{20}	2^{10}
1000	☐	☐	☐	☐
1 000 000	☐	☐	☐	☐
1 048 576	☐	☐	☐	☐
1024	☐	☐	☐	☐

Maks poeng: 8

4(b) Internminne

Velg hvilket tall de binære prefiksene tilsvarer (under er tegnet ^ brukt for å markere eksponent, $10^3 = 10^3$ og $2^{10} = 2^{10}$).

Finn de som passer sammen

	10^9	2^{20}	10^3	10^6	2^{10}	2^{30}
K (kilo)	☐	☐	☐	☐	☐	☐
Gi (gibi)	☐	☐	☐	☐	☐	☐
Ki (kibi)	☐	☐	☐	☐	☐	☐
Mi (mebi)	☐	☐	☐	☐	☐	☐
G (giga)	☐	☐	☐	☐	☐	☐
M (mega)	☐	☐	☐	☐	☐	☐

Maks poeng: 12

4(c) Internminne

Gi en omtrentlig oversikt over hvor lang tid det tar å hente data til CPUen fra registre, cache, RAM og disk.

Maks poeng: 10

Forklar kort hva MMU er og hvorfor ikke MMU kan implementeres i software.

Skriv ditt svar her...

Maks poeng: 10

Forklar kort hva som foregår i PDF-figuren og spesielt hvordan Incoming virtual address 8196 blir oversatt til 24580. Hva ville Outgoing physical address blitt om Incoming virtual address var 6666?

Skriv ditt svar her...

Maks poeng: 10

4(f)

Hva ville skjedd i forrige deloppgave om det kom inn en virtuell adresse i virtuell page nummer 7? Forklar kort.

Skriv ditt svar her...

Maks poeng: 10

5(a)

Hva gjør følgende PowerShell kommando?

Get-Process | Where-Object {\$_.StartTime -gt (get-date).AddMinutes(-50)}

Velg ett alternativ

- ☐ Henter alle prosesser som har fått lagt til mer enn 50 minutter av Windows scheduleren.
- ☐ Lister alle prosesser som kjører nå og som har blitt startet de siste 50 minuttene.
- ☐ Henter alle Where-objekter som vil avsluttes innen 50 minutter.
- ☐ Henter alle prosesser som har et PowerShell dateTime objekt og legger til 50 minutter til start-tidspunktet.
- ☐ Lister alle prosesser som har lagret filer på systemet de siste 50 minuttene.

Maks poeng: 10

5(b) PowerShell

Lag et PowerShell script som lager en liste over hvilke ukedager alle filer og mapper under **C:\Users** har blitt laget på(**CreationTime**). Ta også rekursivt med alle mapper og filer under denne mappen. Listen kan se slik ut etter kjøring:

Dag 0: 1353
Dag 1: 4613
Dag 2: 30788
Dag 3: 1747
Dag 4: 825
Dag 5: 5878
Dag 6: 1763

der dag 0 er søndag. PowerShell **DateTime** har en property med navn **DayOfWeek**, men man kan caste dette til en integer ved å sette [int] foran. Da vil søndag bli dag 0 og lørdag bli dag 6. Bruk et array med ukedagene 0-6 som indeks til å telle opp **CreationTime**-dag for alle filer og mapper. Fjern feilmeldinger fra **ls** kommandoen med konstruksjonen **2> \$null**.

NB! Det fungerer ikke å skrive ut verdien av et element i et array med

echo "Dag 0: \$days[0]"

Man må først legge elementet i en variabel og så skrive den ut, slik:

\$verdi = \$days[0]
echo "Dag 0: \$verdi"

NB! Husk at i PowerShell må man først definere et arrayet og deretter legge til elementer med += operatoren

PS C:\> \$tall = @()
PS C:\> \$tall += "null"
PS C:\> \$tall += "en"
PS C:\> \$tall
null
en

Først etter at elementet er definert kan man tilordne verdier uten å bruke += operatoren:

PS C:\> \$tall[0] = "zero"
PS C:\> \$tall
zero
en

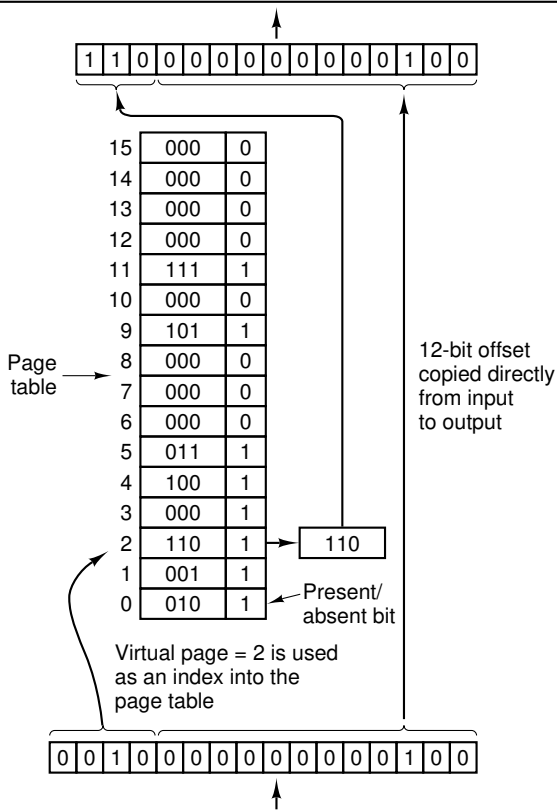
Skriv ditt svar her...

Maks poeng: 40

Question 4.e

Attached





Outgoing
physical
address
(24580)

Incoming
virtual
address
(8196)