

i Eksamensinformasjon

Fakultet: Teknologi, kunst og design

Utdanning: Teknologiske Fag

Emnenavn: Operativsystemer

Emnekode: DATA2500

Bokmål

Dato: 29 mai 2024

Tid: kl. 09.00-12.00 (3 timer)

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 to unntak:

Oppgave 5 gir inntil 20 poeng

Oppgave 19 gir inntil 30 poeng

1 Operativsystemet

Hva er IKKE en av operativsystemkjernens viktigste oppgaver?

Velg ett alternativ:

- ☐ Å gjøre det enklere for applikasjonsprogrammer å bruke maskinens ressurser
- ☐ Å administrere ressurser slik at prosesser ikke ødelegger for hverandre når de kjører samtidig på en maskin
- ☐ Å oversette applikasjoners kildekode til maskinkode slik at den kan kjøres av maskinen

Maks poeng: 10

2 Multitasking

Forklar kort prinsippene for hvordan et operativsystem fordeler CPU-tid mellom flere samtidige prosesser på en datamaskin med kun en CPU (med en enkelt ALU/kjerne). Forklar spesielt hvordan OS kan la en prosess kjøre direkte på CPUen uten at prosessen kan ta over kontrollen og hindre andre prosesser å kjøre.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x



















Σ



Words: 0

Maks poeng: 10

3 Trap og interrupt

Hva er den viktigste forskjellen mellom en trap og et interrupt? Forklar kort.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x























Words: 0

Maks poeng: 10

4 Systemkall

Hva er hensikten med et systemkall i et operativsystem? Forklar kort.

Skriv ditt svar her

Format ▾ | **B** *I* U x_2 x^2 | I_x |   |   |   |   |  | Σ |



Words: 0

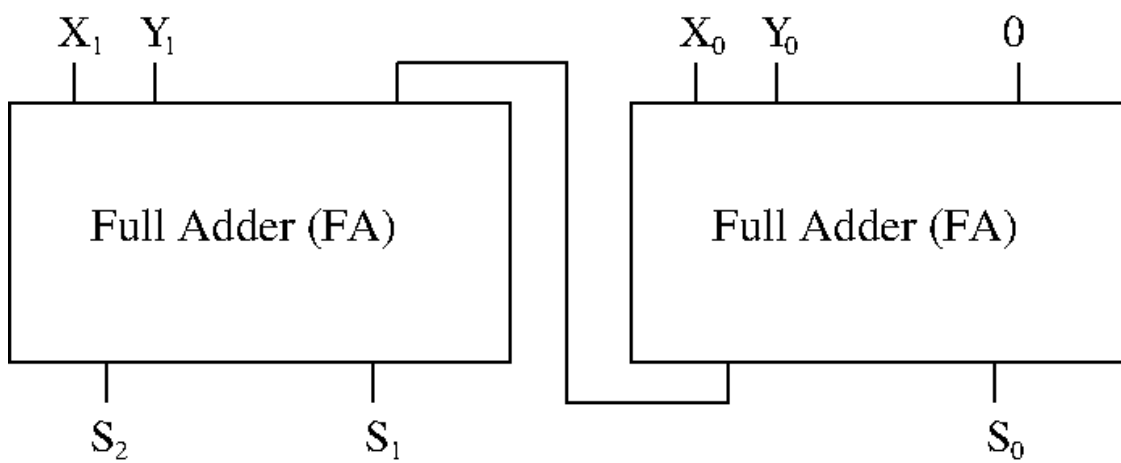
Maks poeng: 10



Words: 0

Maks poeng: 20

6 Full Addder



Anta for denne kretsen at $X_1 = 1, X_0 = 0, Y_1 = 1, Y_0 = 1$

Hva blir da resultatet $S_2S_1S_0$?

Velg ett alternativ

- ☐ 000
- ☐ 101
- ☐ 100
- ☐ 011
- ☐ 010
- ☐ 110
- ☐ 001
- ☐ 111

Maks poeng: 10

7 Datamaskinarkitektur, CPU

Hvilken av følgende enheter er ikke en del av CPUen?
Velg ett alternativ:

- ☐ RAM
- ☐ Datapath
- ☐ Branch-Control
- ☐ ALU
- ☐ Instruksjonsdekoder
- ☐ Program Counter
- ☐ Registere

Maks poeng: 10

8 Simulerings-CPU

For å kunne kjøre koden må CPU'en ha maskinkode. Bruk tabellen nedenfor og oversett linje 5-7 i assemblykoden vist under til maskinkode.

Instruksjonssett				
binært Nr	operand1	operand2	Nr	Navn
0010	DR	tall	2	MOVI
0100	DR	SR	4	ADD
1100	DR	SR	12	CMP
1111	nr	nr	15	JNE

```
5 ADD R0 <- R0 + R1
6 CMP R0 R2
7 JNE 4 (Jump Not Equal 4, hopp til linje 4 hvis R0 != R2)
```

Skriv inn binærkoden for line 5-7 med nuller og enere her:

5

6

7

Maks poeng: 10

9 Linux kommandolinje

Hvilken mappe referer / til i et bash-shell?

Velg ett alternativ

- ☐ Mappen under den du står i
- ☐ Hjemme-mappen din
- ☐ En skjult mappe
- ☐ Mappen over den du står i
- ☐ Hjemme-mappen til root
- ☐ Roten av filsystemet
- ☐ Mappen du står i

Maks poeng: 10

10 Kommandoer find

Hva gjør følgende find-kommando?

```
kan24@os:~$ find mem -executable -type f
mem/mem4
mem/mem1
mem/mem2
mem/mem3
kan24@os:~$
```

Velg ett alternativ:

- ☐ Finner alle filer i mappen mem.
- ☐ Finner alle mapper som kan aksesseres i mappen mem og i alle dens undermapper.
- ☐ Finner alle filer i mappen mem og i alle dens undermapper.
- ☐ Finner alle filer på hele filsystemet.
- ☐ Finner alle kjørbare filer i mappen mem og i alle dens undermapper.
- ☐ Finner alle mapper på hele filsystemet.
- ☐ Finner alle mapper i mappen mem og i alle dens undermapper.
- ☐ Finner alle mapper i mappen mem.

Maks poeng: 10

11 Linux sikkerhet

Anta at du ønsket å komme deg inn på en Linux server hvor du visste at det fantes en bruker med hjemmemappe */home/kan24*. Hvilken fil ville du da helst ha fått tilgang til for å kunne logge deg inn på dette systemet som brukeren *kan24*?

Velg ett alternativ:

- ☐ */home/kan24/.bash_history*
- ☐ */home/kan24/.ssh/id_rsa.pub*
- ☐ */home/kan24/.ssh/id_rsa*
- ☐ */etc/sudoers*
- ☐ */etc/passwd*
- ☐ */home/kan24/.bashrc*
- ☐ */etc/shadow*
- ☐ */etc/network/interfaces*
- ☐ */etc/hosts*

Maks poeng: 10

12 Innlogging på Linux-VM

Øverst i browser-vinduet ved siden av Inspira-fanen er det en fane der det står "Apache Guacamole". Når du trykker på denne får du opp et innloggingsvindu der du skal logge inn med Username: kan24

Password: pwd24shell

og du vil etter 5-10 sekunder få opp et terminal-vindu til en VM som er av samme type som VM-ene som har blitt brukt i kurset, bortsett fra at denne ikke har nett-tilgang. *(hvis du får opp en side hvor teksten "All Connections" står midt på siden, trykk på "osex24" under denne teksten.)*

Du kommer inn i et bash-shell som brukeren "kan24" i dennes hjemmemappe /home/kan24 på din egen Linux-VM som man som en del av eksamen skal bruke for å løse oppgaver. Denne VMen er nesten helt lik Linux-VMene som dere brukte under kurset, med navn som os13, de er også sysbox Docker-containere, men de har ikke nett-tilgang.

Om man ønsker det, kan Linux-VM under eksamen også brukes til å skrive shell-script og teste ut kommandoer. Men hver oppmerksom på at om man lagrer en fil i Linux-VM og så refresher eller kobler seg til Linux-VM på nytt, vil en ny container-VM startes og filen man har lagret vil være borte; filsystemet vil se slik ut som første gang du koblet deg til.

Det er viktig at du bare har en Linux-VM oppe av gangen, ellers blir ressursene på serverene oppbrukt! Ikke start flere samtidige VMer!

Du kan kopiere tekst i Linux-VM ved å markere den med musen og så taste Shift-Ctrl-Alt. Da kommer det opp et clipboard med den avmerkede teksten og så kan du kopiere med Ctrl-C og paste ut her i Inspira-vinduet med Ctrl-V. Internt inne i Linux-VM kan man kopiere tekst ved å markere den med musen og så paste ved å høyreklikke (som i PowerShell).

Filen **file1** ligger øverst i hjemme-mappen til brukeren du er logget inn som og som har brukernavn "kan24". Fyll inn innholdet av filen (5 tegn) i boksen under ved å skrive eller kopiere og paste.

Fyll inn:

Maks poeng: 10

13 Finne fil på Linux-VM

Øverst i hjemme-mappen til brukeren du er logget inn som på Linux-VM og som har brukernavn **kan24** ligger det en mappe som heter **mapper24**. Et sted ned i denne mappen eller nedover i dens undermapper ligger det en fil som heter **fil.txt**. Finn denne filen og skriv eller paste ut tekststrengen den inneholder her (5 tegn):

Maks poeng: 10

14 Filrettigheter

Filen **chmod.txt** ligger øverst i hjemme-mappen til brukeren du er logget inn som og som har brukernavn "kan24". Filrettighetene til denne filen ser slik ut:

```
-rwxr-xr-- 1 kan24 users 0 May 18 19:28 chmod.txt
```

Rettighetene til denne filen har blitt satt med kommandoen:

chmod chmod.txt

Fyll inn hva de tre tall-verdiene i kommandoen må ha vært i boksene over.

Maks poeng: 10

15 Filen -x

Øverst i hjemme-mappen til brukeren du er logget inn som på Linux-VM ressursen og som har brukernavn **kan24** ligger det en fil som heter **-x**. Denne filen inneholder en 5 tegns tekststreng, men det kan av forskjellige årsaker være litt vanskelig å lese denne filens innhold. Finn ut hvordan du kan få tak i innholdet av filen og skriv eller paste innholdet ut her (5 tegn):

Maks poeng: 10

16 Komprimering

Øverst i hjemme-mappen til brukeren du er logget inn som på Linux-VM ressursen og som har brukernavn **kan24** ligger det en fil som heter **koder.tgz**. Denne filen inneholder en mappe som er komprimert med en standard metode ved hjelp av kommandoen **tar**. Pakk ut mappen som denne filen inneholder og gå inn i denne mappen. Der finner du en fil som heter kode1, skriv eller paste innholdet ut her (5 tegn):

Det ligger også en fil inne i mappen som heter kode2.gz som også er komprimert med kommandoen **gzip**. Dekomprimer denne filen og skriv eller paste innholdet ut her (5 tegn):

Maks poeng: 10

17 Kode på systemet

Inne på Linux-VM som du kan logge inn på er en system-kode lagret på en spesiell måte med navnet CODE. Det er mulig å finne koden på tre forskjellige måter. Tenk på hvordan informasjon kan legges inn på en Linux server.

Finn ut hvordan du kan få tak i verdien av CODE og skriv eller paste innholdet ut her (5 tegn):

Maks poeng: 10

18 Kjørbare filer på Linux-VM

Øverst i hjemme-mappen til brukeren du er logget inn som på Linux-VM og som har brukernavn **kan24** ligger det en mappe som heter **mapper24**. I denne mappen og nedover i dens undermapper ligger det blant annet 108 filer som heter **run1**, **run2**, ... , **run108**. Dette er program som kan kjøres og som alle gir en streng på 5 tegn når de kjøres. Ett av programmene gir når det kjøres en streng på 5 tegn som inneholder substrengen "EX" (som for eksempel "ahEXf"). *Legg merke til at denne strengen ikke finnes som en streng inne i programmet men er et resultat av å kjøre programmet.* Finn ut hvilken av de 108 strengene du får når du kjører run-programmene som inneholder "EX" og skriv eller kopier og paste ut denne tekststrengen her (5 tegn):

PS! Du blir i neste oppgave om bash-scripting bedt om å skrive et script som løser denne oppgaven. Dermed vil du for kunne få uttelling for de forsøkene du gjør for å løse denne oppgaven, selvom du ikke finner frem til den riktige strengen.

Maks poeng: 10

19 Script som kjører program

I denne oppgaven skal du laget et bash-script **find.sh** som på en systematisk måte løser problemet som ble gitt i forrige oppgave. Hvis du finner en fungerende løsning, kan den brukes til å løse forrige oppgave, men hvis den ikke virker helt som den skal kan du få noe uttelling for arbeidet du har gjort her.

Som forklart i forrige oppgave, ligger det øverst i hjemme-mappen til brukeren du er logget inn som på Linux-VM og som har brukernavn **kan24**, en mappe som heter **mapper24**. I denne mappen og nedover i dens undermapper ligger det blant annet 108 filer som heter **run1**, **run2**, ... , **run108**. Dette er program som kan kjøres og som alle gir en streng på 5 tegn når de kjøres. Ett av programmene gir når det kjøres en streng på 5 tegn som inneholder substrengen "EX" (som for eksempel "ahEXf"). *Legg merke til at denne strengen ikke finnes som en streng inne i programmet men er et resultat av å kjøre programmet.*

Skriv et bash-script **find.sh** som finner programmet, eventuelt programmene, som ligger i eller under mappen /home/kan24/mapper24, er kjørbare og som gir substrengen "EX" når det (de) kjøres. Scriptet skal gi tilsvarende output som dette når det kjøres:

```
kan24@os:~$ ./find.sh
Filen heter ./mapper24/halt.target.wants/card/run6
Den gir strengen fKzEX
```

Hvis det skulle være to eller flere program som gir output som inneholder "EX", skal to tilsvarende setninger skrives ut for hver av dem. (eksempelet over er fiktivt og gir ikke riktig run-program på Linux-VM). Scriptet skal kun kjøre alle kjørbare filer som ligger under mappen mapper24 og også inkludere andre kjørbare filer enn de 108 som heter **run1**, **run2**, ... , **run108**.

Skriv ditt svar her

Maks poeng: 30

20 MMU

Hva er en MMU og hva er dens funksjon? Forklar kort. (du trenger ikke å vite hva MMU står for, men hva den gjør)

Skriv ditt svar her

Format

B

I

U

x_e

x^2

I_x





















Σ



Words: 0

Maks poeng: 10

21 MMU-beregning

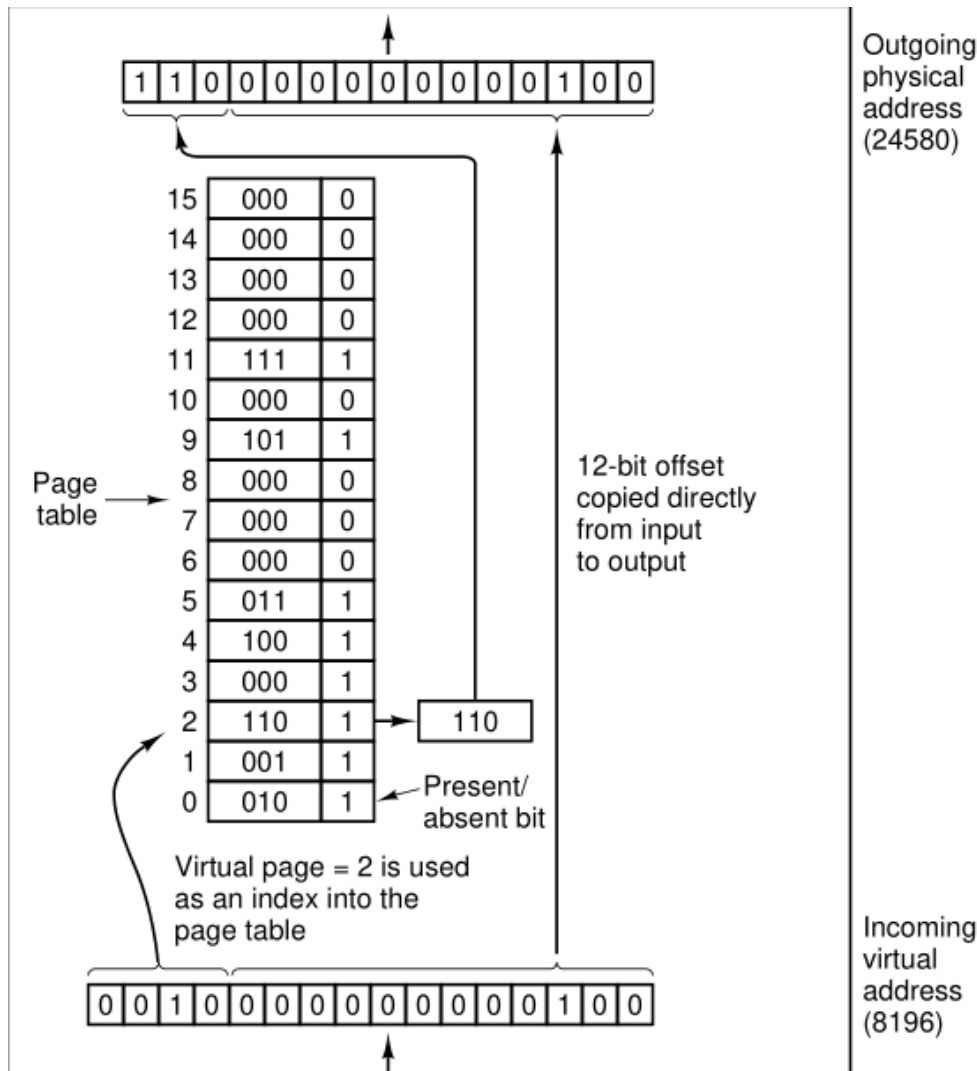
Anta at en innkommende virtuell adresse til MMU er 46512, eller binært med 16 siffer:
 1011 0101 1011 0000 (et mellomrom er lagt til mellom hvert 4de siffer for bedre lesbarhet).
 Hva blir da den utgående fysiske adressen?

Legg merket til at du kan bruke shellen i Linux-VM under ressurser til å for eksempel regne ut den desimale verdien av "Outgoing physical address" i eksempelet under:

```
kan@os:~$ echo 'ibase=2; 110000000000100' | bc
24580
```

Og man kan gå den andre veien for å se hva 28678 er binært:

```
kan@os:~$ echo 'obase=2; 28678' | bc
111000000000110
```



Velg ett alternativ:

- ☐ 30128
- ☐ 34088
- ☐ 11
- ☐ 8196
- ☐ 8200
- ☐ 24592
- ☐ 18
- ☐ 4096
- ☐ 31902
- ☐ 24580
- ☐ Gir page fault
- ☐ 8204

Maks poeng: 10

Maks poeng: 10

24 RAM i praksis

Studer nå kildekoden **mem4.c** i samme mappe **/home/kan24/mem** på Linux-VM og forklar først kort hva som skjer i den delen av koden som er et tillegg til koden i **mem3.c**:

```
for(i=0;i < size;i++)
{
    array[i] = i;
}
```

Forklar deretter kort hva kolonnen RES betyr i output fra **top** og forskjellen på verdien av kolonnen RES for **mem3** og **mem4** som du ser når du starter programmet **mem4** i bakgrunnen og sammenligner output fra kommandoen **top -n 1**. (Men vær oppmerksom på at det tas i bruk betydelig mer RAM enn kun det som er med i arrayet til administrasjon av minnehåndtering, som for eksempel page table entries). Sammenlign også verdiene for **mem3** og **mem4** for kolonnen VIRT og forklar kort.

Skriv ditt svar her

Format

B

I

U

x_e

x^2

I_x

Σ

Words: 0

Maks poeng: 10

25 RAM i praksis

I hvilken minne-region legges data som allokeres av et program, slik som arrayet `array[]` i `mem3.c` i `/home/kan24/mem` på Linux-VM?

Vis hvilken forskjell det gjør på minne-regionene når disse dataene tas i bruk i `mem3` (og eventuelt `mem4`) ved å se på `/proc/pid/maps` som lister en prosess sine minne-regioner. Her må du erstatte 'pid' med pid'en til prosessen `mem3`. Sammenlign med tilsvarende `proc`-fil for `mem2` og finn en minne-region som legges til når arrayet allokeres. (Navnet på denne minne-regionen er identisk med det som det blir spurt etter i første spørsmål i denne oppgaven)

Skriv ditt svar her

Format

B

I

U

x_e

x^2

I_x

Σ

Words: 0

Maks poeng: 10

26 PowerShell-oneliner

Hva gjør følgende PowerShell-kommando?

((Get-date) - (Get-Process dockerd).StartTime).TotalHours

Velg ett alternativ

- ☐ Viser hvilket tidspunkt dockerd-prosessen startet opp
- ☐ Lister alle docker-prosesser som kjører nå og som har blitt startet de siste 60 minuttene.
- ☐ Gir antall timer dockerd-prosessen har kjørt
- ☐ Starter opp dockerd-prosessen
- ☐ Gir antall CPU-timer dockerd-prosessen har brukt
- ☐ Gir antall timer alle docker containere på systemet har kjørt

Maks poeng: 10

27 PowerShell vs Bash

I PowerShell er det laget en rekke alias som gjør det enkelt for en som kjenner til Linux-kommandoer å bruke de samme kommandoene i PowerShell. Merk av hvilke Linux-kommandoer som tilsvarer hvilke PowerShell Cmdlets.

Finn de som passer sammen

	cd	kill	rm	cat	mv	cp	pwd	ls	ps
Get-ChildItem	☐	☐	☐	☐	☐	☐	☐	☐	☐
Stop-Process	☐	☐	☐	☐	☐	☐	☐	☐	☐
Get-Process	☐	☐	☐	☐	☐	☐	☐	☐	☐
Move-Item	☐	☐	☐	☐	☐	☐	☐	☐	☐
Get-Location	☐	☐	☐	☐	☐	☐	☐	☐	☐
Copy-Item	☐	☐	☐	☐	☐	☐	☐	☐	☐
Set-Location	☐	☐	☐	☐	☐	☐	☐	☐	☐
Remove-Item	☐	☐	☐	☐	☐	☐	☐	☐	☐
Get-Content	☐	☐	☐	☐	☐	☐	☐	☐	☐

Maks poeng: 10

