

Datamaskinarkitektur

1)

Datamaskinarkitektur

Hva blir det logiske uttrykket $F(A,B)$ for denne kretsen?

Velg ett alternativ

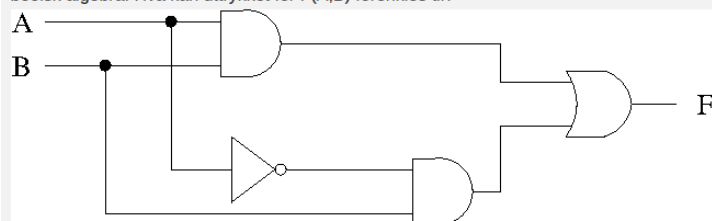
- ☐ $\bar{A} \cdot \bar{B} + A \cdot B$
- ☐ $\bar{A} \cdot B + A \cdot \bar{B}$
- ☐ $(\bar{A} + \bar{B}) \cdot (A + B)$
- ☐ $A + \bar{B}$
- ☒ $A \cdot B + \bar{A} \cdot B$ ✓
- ☐ $(A + B) \cdot (\bar{A} + B)$
- ☐ $\bar{A} \cdot B$

Riktig. 10 av 10 poeng. [Prøv igjen](#)

2)

Datamaskinarkitektur

Uttrykket for den samme kretsen som if forrige oppgave kan forenkles, utifra sannhetstabellen eller ved å bruke boolsk algebra. Hva kan uttrykket for $F(A,B)$ forenkles til?



Velg ett alternativ

- ☐ $A \cdot B$
- ☐ 1
- ☐ 0
- ☐ $\overline{B}$
- ☐ A
- ☒ B
- ☐ $\overline{A}$

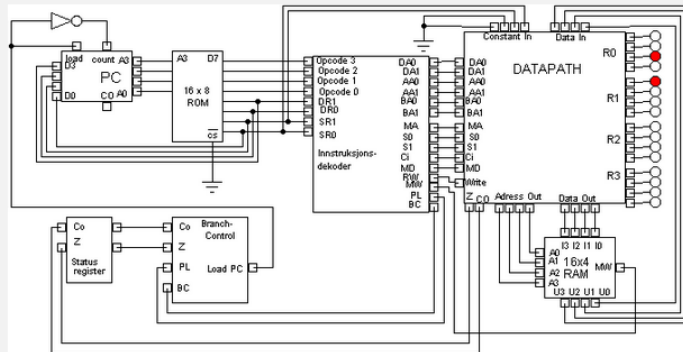


Riktig. 10 av 10 poeng. [Prøv igjen](#)

3)

3 Datamaskinarkitektur

I hvilken del av denne CPU'en ligger all maskinkoden med programmet den kjører?



Velg ett alternativ

- ☐ Ingen av delene
- ☒ ROM
- ☐ Branch-control
- ☐ DATAPATH
- ☐ PC
- ☐ RAM
- ☐ Instruksjonsdekoder



Riktig. 10 av 10 poeng. [Prøv igjen](#)

4)

4 Datamaskinarkitektur

Hvilke konstruksjoner i et høynivåspråk er avhengig av Branch-Control for å kunne utføres?
(det er mulig å velge mer enn ett svar, feil svar gir trekk, men man kan ikke få en negativ score)

Velg ett eller flere alternativer

- ☐ addisjoner
- ☐ multiplikasjoner
- ☐ Initialisering av variabler
- ☒ for-løkker
- ☒ If-tester
- ☒ while-løkker
- ☐ Deklarasjon av variabler



Riktig. 10 av 10 poeng. [Prøv igjen](#)

5) Prosesseringen av en krets av porter går veldig raskt, men det tar litt tid før resultatet er klart og så må mellomlagres før neste operasjon. Resultatet kan være input til neste beregning, men før det lagres må

man være sikker på at beregningen er helt ferdig. Uten en klokke som synkroniserer dette, vil resultatene ukontrollert flyte over i neste beregning. Klokken styrer vipper (som er satt sammen av en master og en slave løs) slik at de venter med å gi resultatet videre til man er helt sikker på at kretsen har regnet seg ferdig. I eksempelet med studenter som var løser (lat ches), så man at informasjonen fløt ukontrollert og usynkront om man ikke organiserte dem to og to med en klokke som sørget for overføring av informasjon med faste intervaller.

Linux i praksis

6)

```
kan@os:~$ cat file1
MpU6t
```

7) Det er riktig at tilgangsrettighetene må gjenopprettes med `chmod`, men det meste annet er galt. Hvis man gjør kommandoen `cd` vil man få en permission denied fordi man ikke har tilgang til hjemmemappen. Det er også galt at dette er roten av filsystemet, det er brukerens hjemmemappe. Og man ville fått 'cannot access' om man gjorde `chmod 755 fengsel` fordi man ikke har tilgang til denne mappen og man kan da heller ikke gå til den med `cd fengsel`. Du kan altså ikke alltid bruke '`chmod 755`' for å gi deg selv tilgang. Og nummereringen burde vært 1, 2, 3; dette var ingen god dag for ChatGPT.

8) Du må bruke `chmod` og full path til hjemmemappen din siden du ikke har aksess fra der du står. Man kan da enten bruke `~` eller `/home/k23`:

```
k23@os:~/fengsel$ chmod 055 ..
k23@os:~/fengsel$ chmod 055 .
k23@os:~/fengsel$ chmod 755 .
chmod: cannot access '.': Permission denied
k23@os:~/fengsel$ chmod 755 ~
k23@os:~/fengsel$ chmod 755 .
chmod: cannot access '.': Permission denied
k23@os:~/fengsel$ chmod 755 ~/fengsel
k23@os:~/fengsel$ cd ..
k23@os:~$
```

9)

```
k23@os:~/run$ for prog in *
> do
> ./$prog | grep HH
> done
HHuva
k23@os:~/run$
```

10)

```
root@os:/home/k23# service docker start
* Starting Docker: docker
root@os:/home/k23#
root@os:/home/k23# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
e3e24c1a03f9   ubuntu:20.04   "tail -f /dev/null"     6 days ago    Exited (137) 6 days ago           ex23k
root@os:/home/k23#
```

ex23k

11)

```
root@os:/home/k23# docker start e3
e3
root@os:/home/k23# docker exec -it e3 bash
root@e3e24c1a03f9:/#
root@e3e24c1a03f9:/# cd
root@e3e24c1a03f9:~# cat xfile2
Ftyrm
root@e3e24c1a03f9:~# tail /etc/shadow
uucp*:19570:0:99999:7:::
proxy*:19570:0:99999:7:::
www-data*:19570:0:99999:7:::
backup*:19570:0:99999:7:::
list*:19570:0:99999:7:::
irc*:19570:0:99999:7:::
gnats*:19570:0:99999:7:::
nobody*:19570:0:99999:7:::
_apt*:19570:0:99999:7:::
os:w7Vk4:19573:0:99999:7:::
root@e3e24c1a03f9:~#
```

Ftyrm

w7Vk4 (W7Vk4 ble også godtatt)

12) De første to kommandoene skriver ut innholdet av filen fil.sh og kjører det som et script. Det sjekker om det ligger en fil med navn file1 i mappen det kjøres fra og øker den udefinerte variabelen \$tall med en. Siden tallet 1 returneres betyr det at det finnes en fil med navn file1 i denne mappen.

Kommandoen `find .` finner alle filer, mapper og linker i mappen `.` og dens undermapper og skriver ut full path til dem.

13)

```
k23@os:~$ cat fil.sh
#!/bin/bash
mappe=$1
for fil in $(find $mappe)
do
    if [ -f "$fil" ]; then
        (( filer += 1 ))
    fi
done
echo $filer
```

Følgende fungerer også:

```
k23@os:~$ cat fil2.sh
find $1 -type f | wc -l
k23@os:~$ ./fil2.sh filer
115
k23@os:~$
```

14)

```
k23@os:~$ ./fil.sh filer
```

```
115
k23@os:~$ ./fil2.sh filer
115
k23@os:~$
```

15) Først listes alle filer og mapper. Deretter er det to tester på om flink er en fil og om flink er en link og begge slår til. De to neste tester om dlink er en mappe og om dlink er en link og begge disse slår også til. Om man bruker dette i et script som teller filer, mapper og linker, må man først sjekke om det er en link og telle det som en link. Hvis det ikke er en link kan man så telle det som fil eller mappe. Hvis ikke kan det føre til en dobbelt-telling av linker.

16)

```
#!/bin/bash
mappe=$1
for fil in $(find $mappe)
do
    if [ -L "$fil" ]; then
        (( linker += 1 ))
    elif [ -f "$fil" ]; then
        (( filer += 1 ))
    elif [ -d "$fil" ]; then
        (( mapper += 1 ))
    fi
done
```

Følgende er også en mulig løsning:

```
#!/bin/bash
find $1 -type L | wc -l
find $1 -type f | wc -l
find $1 -type d | wc -l
```

17)

```
k23@os:~$ ./fil.sh mapper
8 179 223
```

223 mapper

17 Telle mapper

Bruk scriptet du lagde i forrige oppgave til å telle hvor mange mapper som ligger i mappen **mapper** og i alle dens undermapper. Inkluder også mappen **mapper** selv i det totale antallet. Det er også mulig å bruke andre metoder. Mappen **mapper** ligger øverst i Linux-VM. Hvor mange mapper er det?

Velg ett alternativ:

☐ 220

☐ 221

☐ 222

☒ 223



☐ 224

☐ 225

☐ 226

☐ 227

☐ 228

☐ 229

☐ 230

☐ 231

☐ 232

☐ 233

☐ 234

☐ 235

Riktig. 10 av 10 poeng. [Prøv igjen](#)

18)

```
PS /home/k23> (Get-ChildItem -Recurse -File filer).Length
115
PS /home/k23>
```

19)

19 **CPU-avhengige prosesser**

Hvis en 100% CPU-avhengig prosess bruker 60 sekunder på å fullføres på en CPU når den kjører alene, hvor mange sekunder bruker 3 slike prosesser når de kjører på 4 CPUer?

Velg ett alternativ:

☐ 120

☐ 180

☒ 60



☐ 70

☐ 15

☐ 90

☐ 30

☐ 80

☐ 45

Riktig. 10 av 10 poeng. [Prøv igjen](#)

20)

20 **Jiffie**

Omtrent hvor lang er den minste tidsenheten en prosess kan tildels (tidskvantum) i Linux scheduling?

Velg ett alternativ:

☐ 10 nanosekunder (0.00000001 s)

☐ 10 mikrosekunder (0.00001 s)

☒ 10 millisekunder (0.01 s)



☐ 10 sekunder (10 s)

Riktig. 10 av 10 poeng. [Prøv igjen](#)

21)

21

Hardware eller OS

Hva styres av hardware og i liten grad av Operativsystemet?

(det er mulig å velge mer enn ett svar, feil svar gir trekk, men man kan ikke få en negativ score)

Velg ett eller flere alternativer

☒ CPU cache☐ Symmetric Multiprocessing (SMP)☐ Prosess scheduling☒ Branch prediction☐ Context switching☐ Multitasking☒ HyperthreadingRiktig. 10 av 10 poeng. [Prøv igjen](#)

22) Nei, det er feil. Priviligert modus er noe helt annet enn det å være root. Det første gir tilgang til alle maskininstruksjoner og all RAM, mens root er en privilegert bruker som har tilgang til alle filer og devicer, men ikke til OS-kjernen. Derimot finnes det et setuid-bit som gir et program root-rettigheter selvom det kjøres av en vanlig bruker.

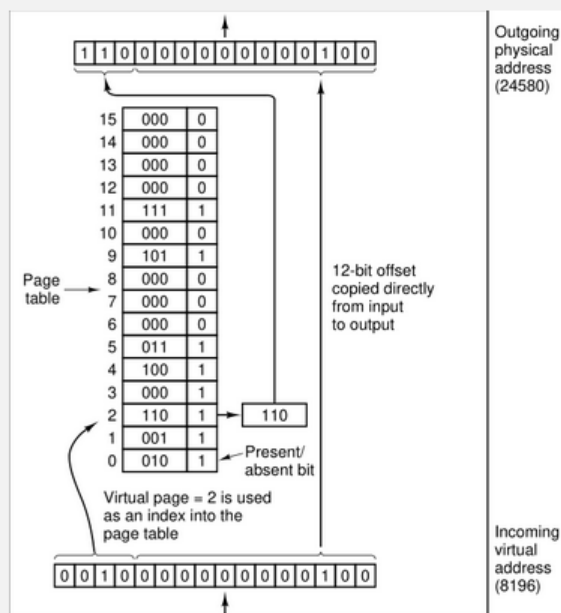
```
$ chmod 4755 program # Setter SETUID-bit
```

23) En virtuell adresse er en logisk/fiktiv adresse som gis til hvert enkelt program som ønsker å bruke RAM og brukes for å abstrahere og administrere det fysiske minnet. En fysisk adresse er den faktiske adressen i RAM hvor data er lagret. MMU brukes for å oversette mellom virtuelle og fysiske adresser.

24)

24 Kopi av MMU

Anta at en innkommende virtuell adresse til MMU er 28678, eller binært med 16 siffer:
0111 0000 0000 0110 (et mellomrom er lagt til mellom hvert 4de siffer for bedre lesbarhet).
Hva blir da den utgående fysiske adressen?



Velg ett alternativ:

- ☐ 24588
- ☐ 12
- ☐ 8204
- ☐ 8200
- ☐ 4096
- ☐ 24580
- ☐ 8196

☒ Gir page fault

- ☐ 24592
- ☐ 4108

Riktig. 10 av 10 poeng. [Prøv igjen](#)

25) VIRT er den totale mengden virtuelt minne som er satt av for en prosess. Alt prosessen kan tenkes å bruke, men som ikke nødvendigvis ligger i RAM. I mem1.c deklarerer det ikke noe ekstra RAM, men i mem2.c deklarerer det et integer array med 1024×1024 integer variabler. Hver av disse heltallsvariablene bruker 4 byte og dermed er den totale mengden ekstra virtuelt minne som deklarerer i mem2.c 4×1024 KByte (siden $1024 \text{ byte} = 1 \text{ KByte}$). Dermed øker VIRT med $4 \times 1024 = 4096$ siden top default rapporterer antall KByte i RAM-kolonnene.

26)

26

Virtualisering

Inntil 2005 var det slik at det fantes noen x86 instruksjoner (som for eksempel POPF) som bare ble oversett om de ble utført i user mode. Det disse instruksjonene utførte skulle bare være lov å gjøre i kernel mode og ingen ting skjedde når de ble utført i user mode. Hvordan ble disse hardware-instruksjonene endret for å støtte x86-virtualisering?

Velg ett alternativ:

- ☐ De ble endret slik at de utføres mye raskere for at virtualiseringen ikke skulle gå for sakte
- ☐ De ble endret slik at det utføres et systemkall når de utføres i user mode
- ☐ De ble endret slik at når de utføres i user mode så hopper CPU'en til neste instruksjon
- ☒ De ble endret slik at de trap'et til kernel mode når de utføres i user mode 
- ☐ De ble endret slik at de trap'et til user mode når de utføres i kernel mode
- ☐ De ble endret slik at disse instruksjonene blir emulert med software

Riktig. 10 av 10 poeng. [Prøv igjen](#)