

- i** Denne eksamenen er delt inn i 3 seksjoner, "Kort", "Middels", og "Lang".

**Bruk omtrent halvparten av tiden på de lange oppgavene.**

Maksimalt mulige poeng i hver seksjon:

| Seksjon | Spørsmål | maks poeng |
|---------|----------|------------|
| Kort    | 1-6      | 18         |
| Middels | 7-11     | 16         |
| Lang    | 12,13,14 | 36         |

Tekstsvarene har en anbefalt lengde. Du trenger **ikke** følge den nøyaktig.

Flervalgsoppgaver vil gi **trekk for feil svar** mens **ubesvarte felt gir 0 poeng**.

Hjelpemiddel til eksamen: ingen

Lykke til!

- 1** Hva er forskjellen mellom et program og en prosess?

**Skriv ditt svar her (2-3 setninger)**

Maks poeng: 2

- 2** En enkel OS bruker **base&bounds** til virtuell minne. Bounds inneholder slutt-adressen til minne-regionen. Alle tall nedover har basis 10.

Prosess A har base = 1200. bounds = 1400.

Hva er den fysiske adressen til følgende virtuelle adresser?

Skriv **-1** dersom det er en ugyldig adresse.

| virtuell adresse | fysisk adresse       |
|------------------|----------------------|
| 12               | <input type="text"/> |
| 14               | <input type="text"/> |
| 120              | <input type="text"/> |
| 1400             | <input type="text"/> |

Maks poeng: 2

**3 Finn de begrepene som passer sammen:**

|                | sector                | page                  | page frame            | block                 |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|
| file system    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| virtual memory | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| physical RAM   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| hard disk      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Maks poeng: 2

**4 (a) Hva er systemprogrammet **fsck**, og hva brukes den til?****Skriv ditt svar her (1-2 setninger)**
(b) Det har blitt sjelden nå at man trenger **fsck** som sysadmin. Hvorfor det?**Skriv ditt svar her (1-2 setninger)**

Maks poeng: 3

5 Et **RAID-lagringssystem** slår sammen flere fysiske harddisker til én stor lagringsplass.

(a)

Når vi slår sammen 4 harddisker, med 1TB størrelse hver, i en RAID-4 (parity), hvor mye lagringsplass har vi faktisk tilgjengelig?

TB

(b)

Du skal sammenligne denne RAID-4-løsningen med en løsning som bruker bare én stor disk av tilsvarende størrelse. Velg de setningene som beskriver situasjonen med en **random** I/O workload:

**Velg ett eller flere alternativer**

- ☐ RAID-4 har verre I/O read latency
- ☐ RAID-4 har verre I/O read throughput
- ☐ RAID-4 har verre I/O write latency
- ☐ RAID-4 har verre I/O write throughput

(c)

Etter en diskfeil måtte vi erstatte disk 3 av 4 i et RAID-4-opplegg. Hva er innholdet til den manglende datablokken?

| Disk 1 | Disk 2 | Disk 3               | Disk 4 |
|--------|--------|----------------------|--------|
| 0110   | 1001   | <input type="text"/> | 0010   |

---

Maks poeng: 4

**6 Fem korte svar**

(a)

Når et system har **swap space** tilgjengelig, så kan prosessene bruke

Velg alternativ (mer, like mye, mindre) minne enn uten swap.

(b)

En typisk arbeidsflyt som et **shell**-program bruker til å kjøre andre programmer kalles for

Velg alternativ (exec - wait - fork, exec - fork - wait, wait - fork - exec, wait - exec - fork, fork - exec - wait, fork - wait - exec)

(c)

**context switch** er navnet til prosedyren der man bytter  (fra en prosess til en annen, fra RAID til ZFS, fra minne til swap, fra virtuell til fysisk minne)

(d)

Velg alternativ (C-kode, Assembly-kode, Maskinkode) kan flyttes til en annen maskinarkitektur uten endringer.

(e)

Scheduling-policyen der man går videre til neste steg med jevnt avstand kalles for  (Shortest-Job-First, First-In-First-Out, Round Robin, Least-Used-First)

---

Maks poeng: 5

7 Ada kjører dette C-programmet:

```
#include <stdio.h>
int main() {
    printf("Hello!\n");
    printf("My name is Ada.\n");
    printf("\nGood bye\n");
}
```

Hun kjører programmet under **strace**, og ser at de tre kall til printf() utløste bare **én write-systemkall**:

```
write(1, "Hello!\nMy name is Ada.\n\nGood bye"..., 33) = 33
```

Hva er fordelen med å slå sammen flere write-kall i én? Hva kalles denne proseduren?

**Skriv ditt svar her (3-5 setninger)**

---

Maks poeng: 3

8 Data som er lagret på en disk kan undergå "bit flips" der en tilfeldig bit bytter fra 0 til 1 eller omvendt.

Hva er faren med dette? Hvordan kan en filsystem sikre seg mot dette?

**Skriv ditt svar her (3-5 setninger)**

---

Maks poeng: 3

9 Hva er den store ulempen med **Segmentation** - metoden til adress-translation? Hvordan løses dette i **Paging**?

**Skriv ditt svar her (3-5 setninger)**

---

Maks poeng: 3

- 10** En **atomisk** transaksjon er en transaksjon som enten fullføres i sin helhet, eller har ingen effekt.

Forklar kort hvordan en **log-basert filsystem** kan sikre at **I/O writes** til filsystemet skjer atomisk.

**Skriv ditt svar her (3-5 setninger)**

---

Maks poeng: 4

- 11** I en **inode** finnes det blant annet noen tidsstempler (created/modified/accessed). I **ext2** er tidsstemplene lagret som en 32-bit int, som teller sekunder siden 1970-01-01. Dette valget innebærer et problem som vil ramme flere og flere systemer som fortsatt bruker 32-bit int til tidspunkter. Beskriv kort dette problemet.

**Skriv ditt svar her (3-5 setninger)**

---

Maks poeng: 3

**12** Forklar hvordan en **multi-level feedback queue** (MLFQ) håndterer følgende situasjoner i scheduling av prosesser:

- a) Hvordan bestemmes **rekkefølgen** prosessene blir kjørt i?
- b) Hvordan blir en **ny prosess** håndtert?
- c) Når kan en prosess **flytte opp eller ned** i prioriteten?
- d) Hvordan påvirker MLFQ-scheduling **turnaround** og **responstid**?

Skriv ditt svar her

Format | **B** | *I* | U |  $\times_2$  |  $\times^2$  |  $\mathcal{I}_x$  | | | | | | | | | | | | | | | | | | | |

Words: 0

Maks poeng: 12

**13** Vi har en virtuell adresserom med **7 bits**, tilsvarende 128 virtuelle adresser **fra 0x00 til 0x7f**.

En **multi-level page table** brukes til oversettelsen:

De første 2 bits av en virtuell adresse blir brukt til en page directory,  
de neste 2 til en page table. De siste 3 bits er offset.

De følgende page frames er i bruk, spørsmålene finnes etter denne oversikten.

Page frame 0  
Page directory

| index | page frame |
|-------|------------|
| 00    | 3          |
| 01    | -/-        |
| 10    | -/-        |
| 11    | 7          |

[...]

Page frame 3  
Page table

| index | page frame |
|-------|------------|
| 00    | -/-        |
| 01    | 8          |
| 10    | 12         |
| 11    | -/-        |

[...]

Page frame 7  
Page table

| index | page frame |
|-------|------------|
| 00    | -/-        |
| 01    | 10         |
| 10    | 11         |
| 11    | 13         |

Page frame 8  
Data Page

| address | content |
|---------|---------|
| 000     | 'H'     |
| 001     | 'e'     |
| 010     | 'l'     |
| 011     | 'l'     |
| 100     | 'o'     |
| 101     | ','     |
| 110     | -space- |
| 111     | -space  |

Page frame 9  
Data Page

| address | content |
|---------|---------|
| 000     | 'W'     |
| 001     | 'o'     |
| 010     | 'r'     |
| 011     | 'l'     |
| 100     | 'd'     |
| 101     | 'l'     |
| 110     | -null-  |
| 111     | -null   |

Page frame 10  
Data Page



| address | content |
|---------|---------|
| 000     | -null-  |
| 001     | -null-  |
| 010     | -null-  |
| 011     | -null-  |
| 100     | 0x03    |
| 101     | 0x04    |
| 110     | -null-  |
| 111     | -null   |

Page frame 11  
Data Page

| address | content |
|---------|---------|
| 000     | 0x02    |
| 001     | 0x07    |
| 010     | -null-  |
| 011     | -null-  |
| 100     | 0x05    |
| 101     | 0x0a    |
| 110     | -null-  |
| 111     | -null   |

Page frame 12  
Data Page

| address | content |
|---------|---------|
| 000     | 'A'     |
| 001     | 'I'     |
| 010     | 'i'     |
| 011     | 'c'     |
| 100     | 'e'     |
| 101     | '!''    |
| 110     | -null-  |
| 111     | -null   |

Page frame 13  
Data Page

| address | content |
|---------|---------|
| 000     | 0x03    |
| 001     | 0x04    |
| 010     | 0x05    |
| 011     | 0x0a    |
| 100     | 0x03    |
| 101     | 0x05    |
| 110     | 0x02    |
| 111     | 0x07    |

[...]

### Spørsmål

- (a) Hvor stor er én page i bytes?
- (b) Hvor mange pages trengs til prosessens hele virtuelle adresserom?
- (c) Utifra situasjonen over, kjører vi disse C-programlinjene (en int er 4 byte, lagret little-endian)

```
int * a = 0x74;
printf("%d\n", *a);
```

Hva er det som skrives ut? Inkluder stegene

- (d) Utifra situasjonen over, kjører vi disse C-programlinjene

```
char * greeting = 0x08;
printf("%s\n", greeting);
```

Hva er det som skrives ut? Inkluder stegene

**Skriv ditt svar her**

Maks poeng: 12

- 14 Vi ser på et enkelt filsystem. Root-mappen i filsystemet (dvs. "/") tilsvarer inode 2.

Filsystemet på disken inneholder:

**Bitmap for data**

| block | occupied? |
|-------|-----------|
| 1-5   | 0 0 0 0 1 |
| 6-10  | 0 1 1 0 1 |
| 11-15 | 1 1 0 0 0 |

**Bitmap for inodes**

| inode | occupied? |
|-------|-----------|
| 1-5   | 1 1 1 1 1 |
| 6-10  | 1 0 0 0 0 |

**Inodes**

| Nr. | Type | reference count | data blocks |
|-----|------|-----------------|-------------|
| 2   | dir  | 3               | 5           |
| 3   | file | 1               | 7, 8        |
| 4   | file | 1               | 10          |
| 5   | dir  | 2               | 11          |
| 6   | file | 1               | 12          |
|     |      |                 |             |
|     |      |                 |             |

**Data blocks:**

| Data block 5   |
|--|
| 2 . // 2 .. // 3 worlds.txt // 5 my_data // 4 alice.txt // |
|  |

...

| Data block 7   |
|--|
| No one would have believed in the last years of the nineteenth century that this world was being watched keenly and closely by intelligences greater tha |

| Data block 8  |
|---|
| n man's and yet as mortal as his own; that as men busied themselves about their various concerns they were scrutinised and studied. |

...

| Data block 10   |
|---|
| Alice was beginning to get very tired of sitting by her sister on the bank. |
|   |

| Data block 11                       |
|-------------------------------------|
| 5 . // 2 .. // 6 bergen_rain.csv // |
|                                     |

| Data block 12  |
|--|
| jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec\n255,210,200,140,110,135,155,210,245,270,270,290\n |

## Oppgaver

**OBS!**

Hver deloppgave tar utgangspunktet i situasjonen ovenfor.

Effektene skal ikke kombineres!

**OBS!**

(a)

Ved et uhell inneholder "Bitmap for data" en feil. Istedenfor

|             |           |
|-------------|-----------|
| <b>6-10</b> | 0 1 1 0 1 |
|-------------|-----------|

finnes det

|             |           |
|-------------|-----------|
| <b>6-10</b> | 0 1 0 0 1 |
|-------------|-----------|

Er filsystemet inkonsistent? Hvilken fil er påvirket? Kan dette problemet oppdages automatisk? Hvordan ser dette ut fra brukerens side?

(b)

Når brukeren skrev ut "my\_data/bergen\_rain.csv" krasjet PCen, og blokk 11 ser sånn ut:

| Data block 11 |
|---------------|
| 5 . // 2 ..   |

Er filsystemet inkonsistent? Kan dette problemet oppdages automatisk? Hvordan ser dette ut fra brukerens side?

(c)

Når man kjører en automatisk konsistens-sjekk, hvilke(t) spørsmål trenger svar fra brukeren for å fikse problemet (a)?

(d)

Når man kjører en automatisk konsistens-sjekk, hvilke(t) spørsmål trenger svar fra brukeren for å fikse problemet (b)?

**Skriv ditt svar her**

Format
B
I
U
x<sub>2</sub>
x<sup>2</sup>
I<sub>x</sub>
📄
📋
↶
↷
↺
↻
☰
☷
☰
☷
☰
☷
☰
☷
Ω
🔗
✎
Σ

✖

Words: 0

Maks poeng: 12

**15** Du trenger ikke gjøre noe her.

Vi bruker denne delen for å registrere poengsummen fra delmappene.

Maks poeng: 30