

Oppgave 1

a

```
[halambique@halambique lab_05]$ ./oppgave_1a
72970 93388 111979 151229 165027
[halambique@halambique lab_05]$ ./oppgave_1a
58034 97811 106123 120114 126558
[halambique@halambique lab_05]$ ./oppgave_1a
101515 116710 156685 186031 193180
[halambique@halambique lab_05]$ ./oppgave_1a
42877 98796 135863 185723 188520
[halambique@halambique lab_05]$ ./oppgave_1a
68404 94160 95768 167018 183220
```

Den uforutsigbare outputen fra programmet skjer fordi de ulike trådene vil kunne forsøke å hente og oppdatere variabelen *value* samtidig med **count*-funksjonen.

b

Se vedlagte fil, *oppgave_1_b.c*

Oppgave 2

a

På skolens server:

```
mathibr@itstud:~/htdocs/Intro20S/Lab_05$ time ./oppgave_2_1
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!
real    0m2.719s
user    0m2.657s
sys     0m0.000s
```

På egen maskin (Arch Linux):

```
[halambique@bluefish lab_05]$ time ./oppgave_2_1
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!
real    0m0.837s
user    0m0.833s
sys     0m0.004s
```

b

På skolens server:

```
mathibr@itstud:~/htdocs/Intro20S/lab_05$ time ./oppgave_2_2
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m1.145s
user    0m2.226s
sys     0m0.000s
```

På min egen maskin:

```
[halambique@bluefish lab_05]$ time ./oppgave_2_2
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m0.339s
user    0m0.674s
sys     0m0.000s
```

c

Utskriften fra `lscpu | more` gir også en del informasjon som ikke er direkte relevant i forbindelse med tråder, så jeg har lagt ved en utskrift som bruker `grep` og tar høyde for at det er flere av noe i utskriften til `lscpu` ved `\(s\)`:

```
mathibr@itstud:~/htdocs/Intro20S/lab_05$ lscpu | grep -E 'CPU\(s\)|Thread\(s\) per core|Core\(s\) per socket|Socket\(s\)'
CPU(s):                2
On-line CPU(s) list:    0,1
Thread(s) per core:     1
Core(s) per socket:     2
Socket(s):              1
NUMA node0 CPU(s):      0,1
```

Vi ser her at vi det er to (logiske) prosessorer som arbeider i tandem, og vi kan derfor kjøre to tråder samtidig (2 kjerner per socket). Dette betyr at ved å utnytte hardwaren effektivt i C-programmet, kan hastigheten ~dobles.