

Oppgave 1A:

```
emilbe@itstud: ~/htdocs/Intro20S/oblig/5$ emacs oppgave_1a.c
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ gcc -pthread -o oppgave_1a oppgave_1a.c
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
100000 141840 241840 247312 347312
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
100000 141505 241505 251828 349483
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
100000 140213 241630 254482 349645
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
134011 134490 246580 251106 337335
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
100000 137972 242308 237972 337972
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
103330 139617 239617 247079 347079
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ ./oppgave_1a
100000 147998 253119 337556 350291
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ |
```

Ut ifra utskriften ser vi her at de fem trådene alltid har ulik verdi, med unntak av første tråd som i dette tilfellet har fått like verdier flere ganger. Grunnen til at disse trådene får ulike verdier, fra utskrift til utskrift er fordi at alle trådene har som «mål» å øke value 100 000 ganger. Det som skjer er at alle trådene prøver å oppdatere den globale variabelen samtidig, noe som fører til uforutsigbare resultater.

Selve problemet i denne koden, er at trådene kjører samtidig, og ikke venter på at den ene blir ferdig, før den andre begynner. Eksempelvis rekker ikke alltid tråd 1 å telle seg til 100 000 før tråd 2 starter, noe som fører til at tråd 2 ikke starter på samme «sted» hver gang, som igjen forskyver seg og skjer med resterende tråder. Slik koden er nå, er sjansen for at siste tråd blir 500 000 svært svært liten.

Oppgave 2A:

```
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ time ./oppgave_2a
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m2.581s
user    0m2.575s
sys     0m0.004s
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ |
```

Oppgave 2B:

```
emilbe@itstud:~/htdocs/Intro20S/oblig/5$ time ./oppgave_2b
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m1.093s
user    0m2.178s
sys     0m0.000s
emilbe@itstud:~/htdocs/Intro20S/oblig/5$
```

Grunnen til at det siste programmet går så mye fortere, er fordi at tellingen til 1 milliard blir fordelt på to, istedenfor én. Dette hadde ikke gjort en veldig forskjell på en enkeltkjernet maskin, men ettersom serveren har to kjerner, vil den utnytte fordelene, med å fordele arbeidsoppgavene på to.

Når man bruker flertråding, kan to kjerner arbeide med hver sin tråd samtidig, noe som gir en tilnærmet dobling av ytelsen, i motsetning til enkelttrådet program hvor kun én kjerne brukes.