[04]

1. Use X threads to compute matrix vector product Bnx1 = Amxn x Xnx1

$$\mathbf{B_{nx1}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}$$

2. Use clock_getttime to evaluate total execution time for different X
   #include <time.h>
   int clock_gettime(clockid_t clk_id, struct timespec *tp);
   https://gist.github.com/pfigue/9ce8a2c0b14a2542acd7
3. Use pthread_setaffinity_np to balance threads across all CPUs

Help : Pin a Pthread to CPU

Call pin_cpu(0)from thread function to fix execution to CPU 0

```
void pin_cpu(int cpu) {
cpu_set_t cpuset;
CPU_ZERO(&cpuset);
CPU_SET(cpu, &cpuset);
if (pthread_setaffinity_np(pthread_self(), \
      sizeof(cpu_set_t), &cpuset) < 0)
            err(1, "failed to set affinity");
}
```

ΠΡΟΣΟΧΗ
Ανεβάστε την λύση στο φάκελο Εργασίες/Εργασίες 4 στο ECLASS, π.χ.04_xxxx.c αν ο Αριθμός Μητρώου σας είναι xxxx.

Η προθεσμία **είναι η Δευτέρα 16/11/2020 και ώρα 18:00. Γενικά δεν γίνονται δεκτές εκπρόθεσμες υποβολές.**