

A.M 1115200900115

Έχω υλοποιήσει την άσκηση σε c++11 λόγο χρησιμοποίησης της `high_resolution_clock` για να κρατάμε τον χρόνο. Όλη η εργασία βρίσκεται στο `main.cpp`. Στην `int main()` ελέγχω ότι ο χρήστης έχει βάλει 3 ορίσματα, δηλαδή έχει αρχικοποιήσει τα `M N` όπου στη συνέχεια μετατρέπονται σε `int` με την χρήση της `atoi`. Σε περίπτωση λάθους εκτυπώνετε το αντίστοιχο μήνυμα. Έχω υλοποίηση την `run(N,M)`. Στην αρχή αρχικοποιούμε τους σημαφόρους `semempty & semfull` όπου και ελέγχουμε αν έγινε πήγε κάτι στραβά. Και τους αρχικοποιούμε με τιμές 0 και N αντίστοιχα. Στη συνέχεια κάνουμε το `setup` για το `up & down` των σημαφόρων. Χρησιμοποιούμε την `k= fork()` όπου αν `k=0` δημιουργούμε τα παιδιά. Κάθε παιδί κάνει `down` τον σημαφόρο ότι πήρε το νούμερο γράφει το νούμερο και κρατάει τον συνολικό χρόνο που έκανε για όλα τα νούμερα καθώς και στο τέλος κάνει `up` τον σημαφόρο ώστε να χρησιμοποιηθεί η μνήμη από άλλο παιδί η από τον feeder. Υπολογίζω το μέσο ορό του χρόνου και εκτυπώνω τα κατάλληλα μηνύματα καθώς χρησιμοποιώ την `ofstream` για την εγγραφή των δεδομένων. Τέλος παρουσιάζετε ο feeder όπου το `number` κάθε φορά δείχνει στον επόμενο τυχαίο αριθμό αλλά και το `chrono` μέσο της `high_resolution_clock::now()` στον χρόνο. Τέλος το πρόγραμμα περιμένει να τερματίσουν τα παιδιά αλλά και διαγράφουμε τους σημαφόρους αλλά και την μνήμη.

ΕΝΔΕΙΚΤΙΚΗ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ:

Το πρόγραμμα κάνει compile με την εντολή: `g++ -std=c++11 main.cpp -o myprogram` λόγο c++11.

```
Admins-iMac:Ask1_0900115 admin$ make
g++ -std=c++11 -c main.cpp
g++ -std=c++11 main.o -o myprogram
Admins-iMac:Ask1_0900115 admin$ ./myprogram 4000 3
M   = 4000
N    = 3
NB   = 12
Child PID: 757 : 6039.49 nanoseconds .
Child PID: 756 : 6104.76 nanoseconds .
Child PID: 758 : 6600.57 nanoseconds .
name = stats0.log
name = stats1.log
name = stats2.log
Admins-iMac:Ask1_0900115 admin$
```

Links:

`high_resolution_clock`: <https://stackoverflow.com/questions/2808398/easily-measure-elapsed-time>