

Game of Life

Σε αυτήν την άσκηση θα εξοικειωθείτε με τη δημιουργία και το χειρισμό threads, shared memory, καθώς επίσης και με semaphores/mutexes για τον συγχρονισμό διαφορετικών threads και τον αμοιβαίο αποκλεισμό.

Συγκεκριμένα, θα υλοποιήσετε το Παιχνίδι της Ζωής. Ο κόσμος του παιχνιδιού είναι ένας πίνακας δύο διαστάσεων 100x100. Κάθε κελί μπορεί να βρίσκεται σε μία κατάσταση, «νεκρό» ή «ζωντανό», ενώ έχει ακριβώς 8 γείτονες που βρίσκονται δίπλα του οριζόντια, κάθετα και διαγώνια. Σε κάθε χρονικό βήμα παρατηρούνται οι ακόλουθες αλλαγές:

- Κάθε ζωντανό κελί με λιγότερους από δύο ζωντανούς γείτονες πεθαίνει.
- Κάθε ζωντανό κελί με δύο ή τρεις ζωντανούς γείτονες παραμένει ζωντανό.
- Κάθε ζωντανό κελί με περισσότερους από τρεις ζωντανούς γείτονες πεθαίνει.
- Κάθε νεκρό κελί με ακριβώς τρεις ζωντανούς γείτονες ζωντανεύει.

Η εξέλιξη του παιχνιδιού καθορίζεται μόνο από τις αρχικές συνθήκες. Ο παίκτης εισάγει την αρχική διάταξη και παρατηρεί πώς αυτή εξελίσσεται.

Καλείστε να υλοποιήσετε το παραπάνω παιχνίδι χρησιμοποιώντας 100 threads. Κάθε thread αναλαμβάνει την εξέλιξη των κελιών ενός υποπίνακα διαστάσεων 10x10.

Το πρόγραμμά σας θα πρέπει να δέχεται ως είσοδο τον αρχικό πίνακα του παιχνιδιού σε μορφή ακεραίων, με το 0 να συμβολίζει ένα νεκρό κελί, ενώ το 1 ένα ζωντανό κελί. Θα σας δωθούν input files για testing.

Η εξέλιξη όλων των κελιών γίνεται ταυτόχρονα, δηλαδή η κατάσταση ενός κελιού στην επόμενη γενεά εξαρτάται αποκλειστικά από την κατάσταση αυτού και των γειτόνων του στην τρέχουσα γενεά.

Επίσης, θα πρέπει να τυπώνετε τον πίνακα του παιχνιδιού όταν ολοκληρώνεται η κάθε γενεά, ώστε να φαίνεται η εξέλιξη των κελιών. Το παιχνίδι δεν σταματά ποτέ, παρά μόνο αν το υποχρεώσει ο χρήστης.

Reference - man pages

Ένα man page περιγράφει τον τρόπο λειτουργίας ενός προγράμματος, ενός system call ή μιας library function. Η εμφάνιση ενός man page γίνεται με τη χρήση της εντολής:

```
man(1)
```

Για να δείτε το man page (σε Linux) που αναφέρεται στη foo(N), κάνετε:

```
% man -S N foo
```

Ο συμβολισμός foo(N) αναφέρεται στο man page που περιγράφει τη foo στη κατηγορία (section) 'N'.

Λίστα με χρήσιμα man pages

Σας παραθέτουμε man pages με system calls που μπορεί να χρειαστείτε για την υλοποίηση της άσκησης. Η παρακάτω λίστα δεν είναι δεσμευτική. Μπορείτε να χρησιμοποιήσετε και εναλλακτικούς τρόπους.

pthread_create(3)
pthread_join(3)
pthread_exit(3)
sem_init(3)
sem_destroy(3)
sem_post(3)
sem_wait(3)
sem_getvalue(3)
pthread_mutex_lock(3)
pthread_mutex_unlock(3)
pthread_mutex_init(3)
pthread_mutex_destroy(3)

Παρατηρήσεις

1. Η άσκηση είναι ατομική. Τυχόν αντιγραφές μπορούν να ανιχνευθούν εύκολα από κατάλληλο πρόγραμμα και θα μηδενιστούν. Συμπεριλάβετε το όνομα σας και το λογαριασμό σας (account) σε όλα τα αρχεία.
2. Κατασκευάστε ένα αρχείο Makefile, έτσι ώστε πληκτρολογώντας make all να γίνεται η μεταγλώττιση (compilation) του προγράμματος και να παράγεται το εκτελέσιμο αρχείο. Επίσης πληκτρολογώντας make clean να καθαρίζονται όλα τα περιττά αρχεία, και να μένουν μόνο τα αρχεία που χρειάζονται για τη μεταγλώττιση.
3. Επιπλέον, γράψτε και ένα αρχείο readme.txt το πολύ 30 γραμμών που να περιέχει επεξηγήσεις για τον τρόπο υλοποίησης.
4. Τοποθετήστε σε ένα κατάλογο όλα τα αρχεία που χρειάζονται για την άσκηση 2. Παραδώστε τα παραπάνω αρχεία χρησιμοποιώντας το πρόγραμμα turnin (πληκτρολογήστε turnin assignment_2@hy345 directory_name από τον κατάλογο που περιέχει τον κατάλογο directory_name με τα αρχεία της άσκησης).
5. Σε πολλές περιπτώσεις τα ονόματα των συναρτήσεων βιβλιοθήκης είναι ενδεικτικά. Μπορείτε να χρησιμοποιήσετε όποια σας βολεύουν.