

## HY342 - Παράλληλος Προγραμματισμός

### Άσκηση 2

Σε αυτή την άσκηση θα αναπτύξετε ένα παράλληλο πρόγραμμα για το “Game of Life”. Το παιχνίδι “Game of Life” εφευρέθηκε το 1970 από τον John Conway και είναι παιχνίδι με 0 παίκτες (world simulation), γνωστό και ως *κυτταρικό αυτόματο* (cellular automaton). Το παιχνίδι αποτελείται από ένα δισδιάστατο κόσμο που χωρίζεται σε κελιά. Κάθε τέτοιο κελί είναι είτε “ζωντανό” είτε “νεκρό” σε μια δεδομένη γενιά. Το παιχνίδι αποτελείται από ένα σύνολο κανόνων που περιγράφουν πώς εξελίσσονται τα κελιά από μια γενιά στην επόμενη. Οι κανόνες αυτοί υπολογίζουν την κατάσταση ενός κελιού στην επόμενη γενιά, ως συνάρτηση των καταστάσεων των γειτονικών κελιών στην τρέχουσα γενιά. Στο δισδιάστατο κόσμο του παιχνιδιού, γειτονικά κελιά είναι τα 8 κελιά που βρίσκονται οριζόντια, κατακόρυφα, ή διαγώνια δίπλα στο δεδομένο κελί.

Οι κανόνες του Conway για το παιχνίδι συνοψίζονται ως εξής:

1. Ένα ζωντανό κελί με λιγότερους από δύο ζωντανούς γείτονες πεθαίνει από μοναξιά: στην επόμενη γενιά θα είναι νεκρό.
2. Ένα ζωντανό κελί με περισσότερους από τρεις ζωντανούς γείτονες πεθαίνει από υπερπληθισμό: στην επόμενη γενιά θα είναι νεκρό.
3. Ένα ζωντανό κελί με ακριβώς δύο ή τρεις ζωντανούς γείτονες επιζεί: στην επόμενη γενιά παραμένει ζωντανό.
4. Ένα νεκρό κελί με ακριβώς τρεις ζωντανούς γείτονες αποκτά ζωή: στην επόμενη γενιά θα είναι ζωντανό.

Μπορείτε να διαβάσετε περισσότερα για το Game of Life στη Wikipedia: [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life).

Σε αυτή την άσκηση θα υλοποιήσετε το Game of Life του Conway σε C με παραλληλισμό OpenMP, με τον περιορισμό ότι ο δισδιάστατος κόσμος είναι πεπερασμένων διαστάσεων. Για κάθε κελί στην άκρη του κόσμου, υποθέστε ότι οι γείτονες που θα βρίσκονταν πέρα από το όριο του κόσμου είναι νεκροί. Το πρόγραμμά σας θα πρέπει να παίρνει την αρχική κατάσταση του κόσμου από ένα αρχείο κειμένου (`argv[1]`), να υπολογίζει την κατάσταση του κόσμου μετά από ορισμένες γενιές (`argv[2]`) και να αποθηκεύει το αποτέλεσμα σε ένα αρχείο κειμένου (`argv[3]`).

Η μορφή του αρχείου που περιγράφει τον δισδιάστατο κόσμο είναι όπως στο παράδειγμα:

[illegible]

Με το πρόγραμμά σας παραδώστε και μια αναφορά με μετρήσεις για την απόδοση του προγράμματός σας. Για τις μετρήσεις χρησιμοποιήστε το αρχείο εισόδου `glid1.txt` και `glid2.txt`, και μετρήστε το χρόνο εκτέλεσης με την εντολή `time` (του shell) για 100 και για 1000 γενιές. Επαναλάβετε κάθε μέτρηση 5 φορές.

1. Ποιό είναι το average speedup για παραλληλισμό 1, 2 και 4;
2. Υπολογίστε τη διακύμανση του χρόνου εκτέλεσης για κάθε πειραματικό configuration (παραλληλισμός  $\times$  input-size), και τη διακύμανση των χρόνων εκτέλεσης για το σειριακό πρόγραμμα.
3. Πού οφείλεται το overhead του OpenMP προγράμματος με παραλληλισμό 1, σε σχέση με το σειριακό πρόγραμμα (δηλαδή μεταγλωττισμένο χωρίς -fopenmp);
4. Περιγράψτε περιληπτικά πώς αλλάζει το speedup που παρατηρείτε ανάλογα με τις επαναλήψεις (100-1000), και πώς ανάλογα με το μέγεθος της εισόδου (glid1.txt, glid2.txt), και γιατί;