Προγραμματιστική εργασία εαρινού εξαμήνου 2014
ΗΥ-240 Δομές Δεδομένων

Παράδοση 28/3/2014 23:59

Γενική περιγραφή

Το IMDb (Internet Movie Database) είναι η πιο διαδεδομένη πλατφόρμα και βάση δεδομένων σχετικά με ταινίες και τηλεοπτικά προγράμματα. Οι πληροφορίες που παρέχονται βοηθούν τους χρήστες στην επιλογή ταινιών που θα δουν, με βάση διάφορα κριτήρια αναζήτησης ή συσχέτισης. Στα πλαίσια της προγραμματιστικής εργασίας σας θα κληθείτε να υλοποιήσετε ένα (μικρό) υποσύνολο της λειτουργικότητας του IMDb βασιζόμενοι σε πραγματικά δεδομένα από ταινίες.

Το σύστημα θα περιλαμβάνει μια **διπλά συνδεδεμένη λίστα** από ταινίες για τις οποίες είναι γνωστά τα εξής χαρακτηριστικά: **αναγνωριστικό**, **τίτλος, έτος προβολής, βαθμολογία, πλήθος βαθμολογήσεων, χρονική διάρκεια** (σε λεπτά) και **είδος** (ένα η περισσότερα). Παράλληλα θα υπάρχει μια **απλά συνδεδεμένη λίστα** από χρήστες του συστήματος, για τον καθένα από τους οποίους θα υπάρχει συσχέτιση με τις ταινίες που επιθυμεί να παρακολουθήσει.

Αρχεία που θα σας δοθούν

- **Αρχείο ταινιών**
  Το συγκεκριμένο αρχείο θα περιέχει όλα τα δεδομένα για να δημιουργήσετε τη λίστα ταινιών. Θα έχει το εξής format:
  [movie_id] [title] [year] [rating] [votes] [duration]
  genre_name1 genre_name2 genre_name3 …
  [movie_id] [title] [year] [rating] [votes] [duration]
  genre_name1 genre_name2 genre_name3 …
  ……

- **Αρχείο events**
  Το συγκεκριμένο αρχείο θα περιέχει όλα τα δεδομένα για να δημιουργήσετε τη λίστα χρηστών. Θα περιλαμβάνει 2 τύπους γεγονότων:
  - Γεγονός εισαγωγής στη λίστα με τις ταινίες που επιθυμεί να παρακολουθήσει ο συγκεκριμένος user.
  - Γεγονός διαγραφής από τη λίστα με τις ταινίες του user – θεωρούμε ότι την παρακολούθησε και δεν την επιθυμεί πλέον στη λίστα του - .

  Το αρχείο θα έχει το εξής format:
  [I] [username] [movie_id] -> Γεγονός που σηματοδοτεί την εισαγωγή μίας ταινίας με movie_id στην watch_list του χρηστή με username .
  [D] [username] [movie_id] -> Γεγονός που σηματοδοτεί τη διαγραφή της ταινίας με movie_id από την watch_list του χρήστη με username .

Δομές που θα υλοποιήσετε

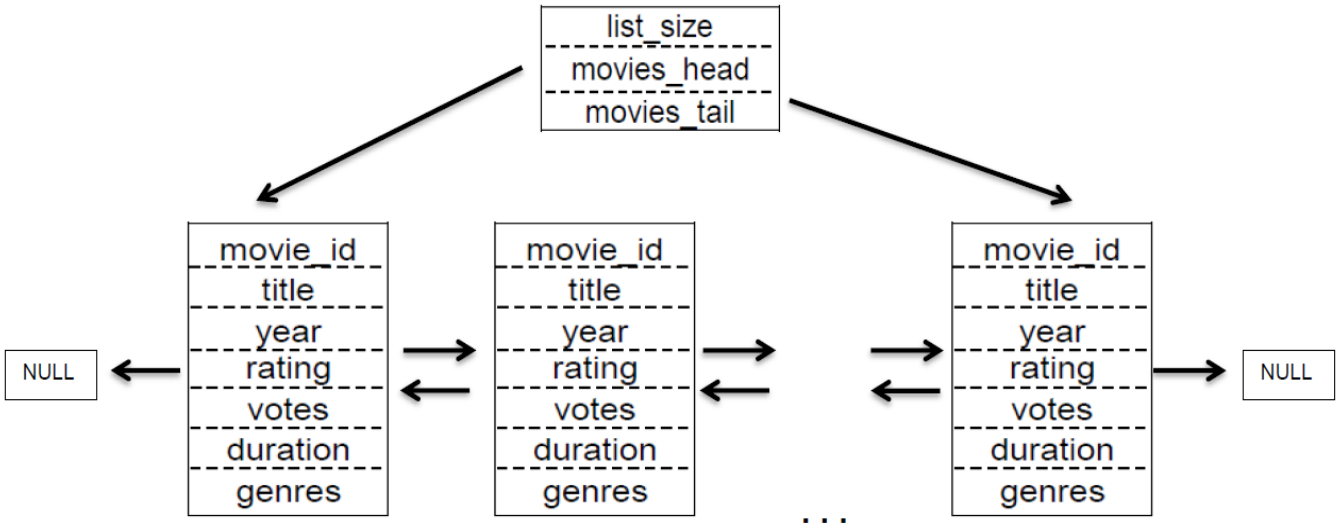
- Μια **διπλά συνδεδεμένη** λίστα ταινιών **movies_list**, η οποία θα περιέχει όλες τις ταινίες που υπάρχουν στη βάση δεδομένων και θα πρέπει να υπάρχει πρόσβαση στα στοιχεία της τόσο από την αρχή της (head) όσο και από το τέλος της (tail). Θα πρέπει να κάνετε 3 διαφορετικές υλοποιήσεις της λίστας αυτής:

  - Μία θα είναι **unsorted,** όπου τα στοιχεία θα εισάγονται με οποιαδήποτε σειρά.
  - Μία θα είναι **sorted,** και κάθε insert θα πρέπει να γίνεται με τέτοιο τρόπο ώστε η λίστα να είναι ταξινομημένη ως προς το έτος προβολής (year) - αύξουσα ταξινόμηση.
  - Μία θα είναι **self-adjusting,** και κάθε φορά που αναζητάμε ένα κόμβο αυτός θα πρέπει να έρχεται στην αρχή της λίστας.

Κάθε κόμβος της λίστας ταινιών θα έχει τα εξής πεδία:

1. Αναγνωριστικό ταινίας **movie_id (int)**
2. Τίτλος ταινίας **title (string)**
3. Έτος προβολής ταινίας **year (int)**
4. Βαθμολογία ταινίας **rating (double)**
5. Πλήθος βαθμολογήσεων **votes (int)**
6. Διάρκεια σε λεπτά **duration (int)**
7. Είδη στα οποία ανήκει η κάθε ταινία **genres (array)**
8. Δείκτη στο επόμενο στοιχείο της λίστας ταινιών **next**
9. Δείκτη στο προηγούμενο στοιχείο της λίστας ταινιών **prev**

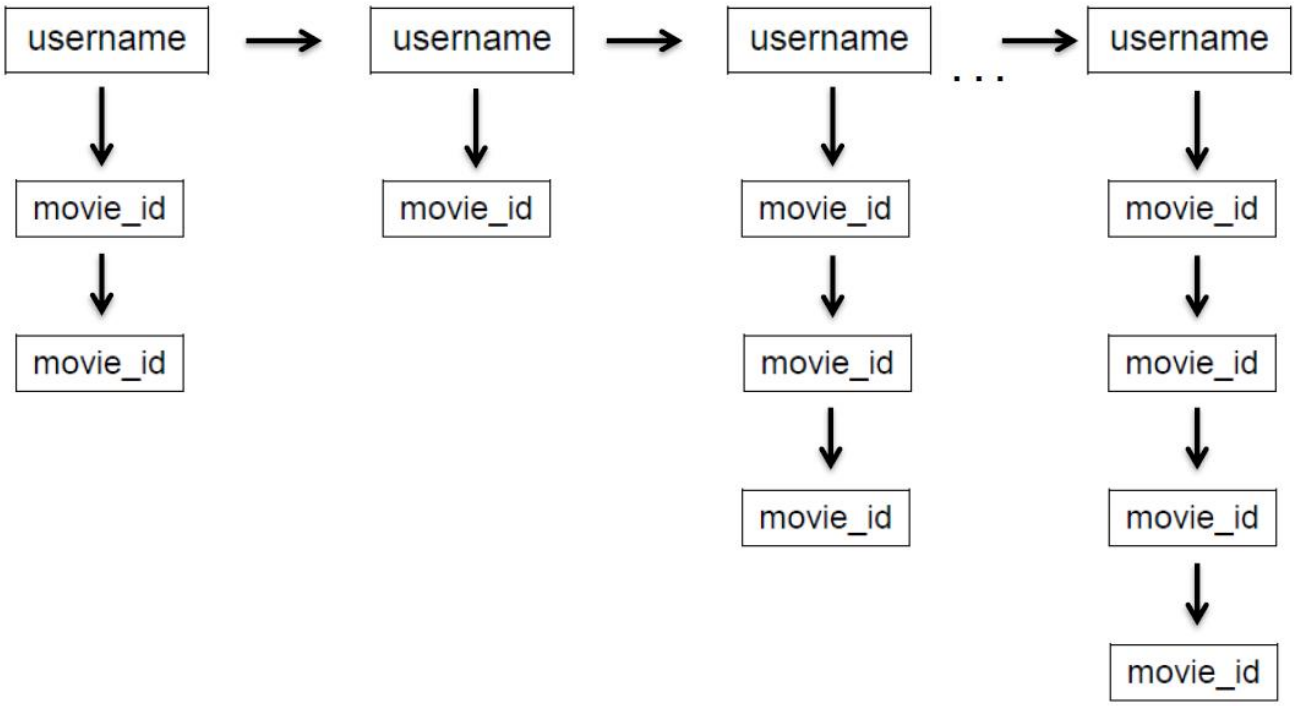
Σχηματικά η δομή αυτή θα έχει ως εξής:



- Μία **απλά συνδεδεμένη** λίστα χρηστών **users_list**, **αταξινόμητη**, που θα περιέχει όλους τους users που βρίσκονται στη βάση δεδομένων και τις ταινίες που επιθυμεί να παρακολουθήσει ο καθένας. Κάθε κόμβος της θα περιέχει τα εξής πεδία:

  1. Το όνομα χρήστη του user **username (string)**
  2. Δείκτη στο επόμενο στοιχείο της λίστας χρηστών **next**
  3. Τον πρώτο κόμβο της απλά συνδεδεμένης λίστας με τις ταινίες του κάθε χρήστη **watchMovie_list**. Κάθε κόμβος αυτής της λίστας θα περιέχει τα πεδία :
     3.1 Αναγνωριστικό ταινίας **movie_id (int)**
     3.2 Δείκτη στον επόμενο κόμβο αυτής της λίστας **next**

Συνολικά λοιπόν η δομή αυτή θα έχει ως εξής:



Περιγραφή υλοποίησης

- Αρχικά θα διαβάσετε το αρχείο των ταινιών και θα δημιουργήσετε τις αντίστοιχες λίστες: unsorted, sorted, self-adjusting εκτυπώνοντας τους χρόνους δημιουργίας κάθε μίας.
- Στη συνέχεια, θα διαβάσετε το αρχείο χρηστών και θα δημιουργήσετε την απλά συνδεδεμένη λίστα users_list. Μετά, θα επεκτείνετε αυτή τη λίστα δημιουργώντας τη λίστα watchMovie_list σε κάθε κόμβο (η παραπάνω δομή)
- Αφού υλοποιήσετε τις δομές με λειτουργίες εισαγωγής, αναζήτησης και διαγραφής, θα πρέπει για κάθε event που διαβάζετε από το αρχείο χρηστών, να ελέγχετε αν το event είναι έγκυρο ή όχι. Πιο συγκεκριμένα, ένα event:
  - Εισαγωγής μιας ταινίας στο watchMovie_list θα πρέπει να τυπώνει:
    - "Ο χρήστης [username] πρόσθεσε την ταινία [title] στο watch list του" και να βάζει την ταινία στο watch list του αν η ταινία υπάρχει στη movies_list.
    - "Η ταινία με id [movie_id] δεν υπάρχει στη βάση" και να μην βάζει την ταινία στο watch list του αν η ταινία δεν υπάρχει στη movies_list.
  - Διαγραφής μιας ταινίας στο watchMovie_list θα πρέπει να τυπώνει:
    - "Ο χρήστης [username] αφαίρεσε την ταινία [title] από το watch list του" και να βγάζει την ταινία από το watch list του αν η ταινία υπάρχει στη movies_list.
    - "Η ταινία με id [movie_id] δεν υπάρχει στη watch list του χρήστη [username]" και να μην αλλάζει τίποτα στη δομή.

  Στη λίστα των users, όταν δεν έχει απομείνει καμία ταινία προς παρακολούθηση στην watchMovie_list, **θα διαγράφεται ο αντίστοιχος κόμβος** της users_list .

- Μετρήστε και παρατηρήστε τι χρόνος απαιτείται για τις λειτουργίες της εισαγωγής, αναζήτησης, διαγραφής στις τρεις διαφορετικές υλοποιήσεις (sorted, unsorted, self-adjusting) για τη λίστα ταινιών.
- Θα πρέπει να υποστηρίζονται κάποια ερωτήματα, που θα βασιστούν στη λειτουργία της αναζήτησης με συγκεκριμένα κριτήρια. Πιο αναλυτικά:
  1. Ποιος είναι ο τίτλος της ταινίας με το μεγαλύτερο rank; ***
  2. Πόσες ταινίες γυρίστηκαν μεταξύ των ετών XXXX και YYYY; (XXXX <= YYYY)
  3. Ποιες ταινίες ανήκουν στα είδη X και Y ταυτόχρονα;
  4. Πόση χρονική διάρκεια έχουν συνολικά οι ταινίες που θέλει να παρακολουθήσει ο χρήστης με username X;
  5. Ποιος είναι ο user που επιθυμεί να δει τις περισσότερες ταινίες του είδους X;

Bonus

6. Ποιες είναι οι 10 ταινίες με το μεγαλύτερο rank *** στο είδος X;
   Υπόδειξη: Θα υλοποιήσετε μία ταξινομημένη ως προς το rank λίστα, μεγέθους 10 κόμβων. Θα εισάγετε τις 10 πρώτες ταινίες του είδους X που θα συναντήσετε στη βάση, στη λίστα αυτή. Στη συνέχεια, για κάθε επόμενη ταινία του είδους θα συγκρίνετε το rank της, με το ελάχιστο rank της λίστας αυτής. Αν αυτό είναι μεγαλύτερο, θα αφαιρείτε την ταινία με το minimum rank και θα εισάγετε τη νέα ταινία στην κατάλληλη θέση, ώστε να παραμείνει λίστα ταξινομημένη.

*** Το rank υπολογίζεται ως εξής (τύπος από IMDb):

$$rank = \left(\frac{votes}{(votes + 25000)}\right) * rating + \left(\frac{25000}{(votes + 25000)}\right) * 6.9$$

Αφού, θα έχετε υλοποιήσει όλα τα ερωτήματα και τις ζητούμενες δομές το πρόγραμμά σας θα πρέπει να μπαίνει σε ένα ατέρμονα βρόχο – while(true) – και να τυπώνει ένα menu, το οποίο θα δίνει τη δυνατότητα στο χρήστη να:

- Τυπώνει ολόκληρη τη movies_list με όλα τα πεδία της κάθε ταινίας .
- Τυπώνει ολόκληρή τη users_list με όλους τους χρήστες και τις ταινίες που επιθυμεί να παρακολουθήσει ο καθένας (watchMovie_lists) .
- Δίνει το username κάποιου χρήστη και να τυπώνει τη watchMovie_list του .
- Δίνει τις παραμέτρους αναζήτησης για να εκτελεί οποιοδήποτε από τα παραπάνω ερωτήματα και να τυπωθούν τα αντίστοιχα αποτελέσματα εκτυπώνοντας τους χρόνους εκτέλεσης των ερωτημάτων για κάθε μία από τις 3 υλοποιήσεις της λίστας ταινιών.

Οι υπογραφές των συναρτήσεων που πρέπει να υλοποιήσετε θα σας δοθούν και πρέπει να τηρηθούν αυστηρά για κάθε λειτουργία. Είστε ελεύθεροι να προσθέσετε και δικές σας συναρτήσεις, αλλά σε καμία περίπτωση δεν πρέπει να αλλάξετε τις υπογραφές αυτών που σας δίνονται. Επίσης η συνάρτηση main() θα πρέπει να βρίσκεται σε ξεχωριστό αρχείο μόνη της και θα σας δίνεται ο parser που θα επιστρέφει από τα αρχεία όλα τα δεδομένα τα οποία χρειάζεστε για να δημιουργήσετε τις ζητούμενες δομές .