ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
1η Εργασία - Τμήμα: Αρτίων Αριθμών Μητρώου
K22: Λειτουργικά Συστήματα – Χειμερινό Εξάμηνο '08
Ημερομηνία Ανακοίνωσης: Τρίτη 7 Οκτωβρίου
Ημερομηνία Υποβολής: Τετάρτη 29 Οκτωβρίου και Ωρα 21:00

Εισαγωγή στην Εργασία:

Ο στόχος αυτής της εργασίας είναι να εξοικειωθείτε με το Solaris/Linux και τα βασικά εργαλεία προγραμματισμού και ανάπτυξης λογισμικού στο εν λόγω περιβάλλον. Στα πλαίσια αυτής της εργασίας θα υλοποιήσετε μια απλή ωστόσο λειτουργική δομή γενεαλογικού δένδρου (rooted tree) με ένα αριθμό από λειτουργίες. Ποιο συγκεκριμένα:

1. Θα κατασκευάσετε ένα μηχανισμό δυναμικής αποθήκευσης γενεαλογικών εγγραφών στην κυρία μνήμη που θα βασίζεται σε rooted-trees. Ενα rooted-tree επιτρέπει την αποθήκευση οικογενειακών δεδομένων και ουσιαστικά είναι ένας ευέλικτος μηχανισμός αποθήκευσης δεδομένων με κλειδιά που παρέχει γρήγορη προσπέλαση σε επερωτήσεις πλαισίου. Η δομή είναι πλήρως δυναμική, παραμένει στην μνήμη για την διάρκεια χρήσης του προγράμματος σας και ο χώρος που καταναλώνει ελευθερώνεται με το τέλος εκτέλεσης του προγράμματος σας.
2. Στο γενεαλογικό δένδρο υπάρχουν εγγραφές που δείχνουν την σχέση μητέρα/πατέρας/παιδιά για πάνω από δυο γενεές Οι εγγραφές κάθε προσώπου που εμφανίζεται στην δομή περιέχουν το κλειδί του (ένα αριθμητικό ID) το όνομα του, την ημερομηνία γέννησης, το ID ενός από τους γονείς του, το ID του/της συζύγου, και τέλος τα IDs όλων των παιδιών της οικογένειας.
3. Τα ονόματα που υπάρχουν στην δομή είναι μοναδικά (για ευκολία) όπως προφανώς και τα αντίστοιχα κλειδιά.
4. Η δεικτοδότηση ενός ατόμου γίνεται στην δομή σε σχέση με τον πατέρα τους ή την μητέρα τους.
5. Η δομή πρέπει να υποστηρίζει βασικές πράξεις εισαγωγής, εύρεσης εγγραφής (επερώτηση μαχαιριού) καθώς επίσης και εξειδικευμένες λειτουργίες (που δίνουμε παρακάτω).

Διαδικαστικά:

Το πρόγραμμά σας θα πρέπει να γραφτεί σε C (ή C++ αν θέλετε αλλά χωρίς την χρήση STL) και να τρέχει στις μηχανές Unix (Workstations Solaris/Linux) του τμήματος. Επίσης το πρόγραμμα σας (source code) πρέπει να αποτελείται από **τουλάχιστον** δυο (και κατά προτίμηση πιο πολλά) διαφορετικά αρχεία (δηλ. πρέπει να κάνετε χρήση separate compilation). Παρακολουθείτε την ιστοσελίδα του μαθήματος για επιπρόσθετες ανακοινώσεις στο URL: *www.di.uoa.gr/~ad* και την λίστα του μαθήματος.

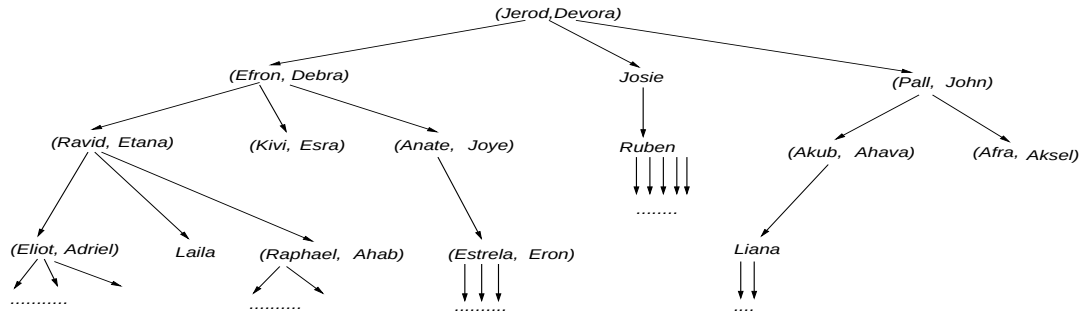
- Υπεύθυνοι για την άσκηση αυτή (ερωτήσεις, αξιολόγηση, βαθμολόγηση κλπ.) είναι ο κ. Βασίλης Μουστάκας (b.moystakas AT di) και η κα. Μαριαλενα Κυριακίδη (std02045 AT di).
- Εγγραφείτε στην ηλεκτρονική λίστα (mailman) και παρακολουθείτε ερωτήσεις/απαντήσεις/διευκρινήσεις που δίνονται σχετικά με την άσκηση.

Αναλυτική Περιγραφή της Εργασίας:

Η Εικόνα 1 δείχνει μια αντιπροσωπευτική δομή. Οι κόμβοι του δένδρου αυτού είναι οικογένειες που έχουν δυο ή ένα γονιό. Κάθε οικογένεια μπορεί να έχει παιδιά που εμφανίζονται στο πιο κάτω επίπεδο από εκείνο των γονιών τους. Στο δένδρο της Εικόνας 1 ο Efron με σύζυγο την Debra έκαναν τρία παιδιά: τον Ravid, τον Kivi, και την Anate που αντίστοιχα με τις Etana, Esra και τον Joye δημιούργησαν οικογένειες κλπ. Ο Ruben υιοθέτησε πέντε παιδιά ενώ το ζεύγος της Estrata και του Eron είχε τρία παιδιά.

Τα βασικά χαρακτηριστικά στοιχεία της δομής είναι τα εξής:

1. Αρχικά το rooted-tree είναι άδειο. Καθώς εγγραφές διαβάζονται από ένα αρχείο εισόδου (text) (η απλά εισέρχονται σε γραμμή εντολής) η δομή μεγαλώνει ανάλογα σύμφωνα με τις σχέσεις γονιός – παιδί – σύζυγος που είναι εμφανείς στις εγγραφές εισόδου. Κάθε άτομο που εμφανίζεται στο rooted-tree έχει ένα



Σχήμα 1: Παράδειγμα ενός Rooted-Tree

μοναδικό ID. Γενικά ένα rooted-tree μπορεί να υλοποιηθεί σαν ένας γραφος που επιτρέπει την μετάβαση από παιδί σε γονιό και αντίστροφα.

2. Κάθε γραμμή εισόδου έχει μια συγκεκριμένη μορφή που περιλαμβάνει τα παρακάτω στοιχεία:
   - ID - Αριθμός που ορίζει μοναδικά ένα πρόσωπο: integer
   - Όνομα: ακολουθία χαρακτήρων 20 characters
   - ID Πατέρα (ή Μητερας) που ορίζει ένα γονιό: integer
   - Ημερομηνία Γέννησης: integer
   - Τόπος Γέννησης: ακολουθία χαρακτήρων 20 characters
   - ID Συζύγου (εάν υπάρχει σύζυγος): integer
   - k : Αριθμός παιδιών στην οικογένεια (μπορεί να είναι και μηδέν): integer
   - ID πρώτου παιδιού αν υπάρχει: integer
   - ID δεύτερου παιδιού αν υπάρχει: integer
   - .....
   - ID κ–ου παιδιού αν υπάρχει: integer
3. Θα πρέπει να υποστηρίζονται λειτουργίες συμπεριλαμβανομένων της εισαγωγής, απλών επερωτήσεων, επερωτήσεων πλαισίου, επερωτήσεων σχέσεων και τέλος (απλές) στατιστικές ερωτήσεις. Οι συγκεκριμένες διεπαφές για τις παραπάνω λειτουργίες δίνονται παρακάτω.

Διεπαφές Εφαρμογής:

Η εφαρμογή σας πρέπει να αλληλεπιδρά με τον χρήστη μέσω ενός κελύφους (δηλ. prompt) δικιάς σας υλοποίησης από το οποίο θα πρέπει να υποστηρίζονται οι παρακάτω ενέργειες:

- initialize
  Αρχικοποίηση της δομής που πρέπει να είναι εντελώς δυναμική και να μην κάνει χρήση καμίας στατικής struct της C συμπεριλαμβανομένων στατικά ορισμένων πινάκων.
- insert <ID><name><ParentID><DOB><BirthPlace><SpouceID> <k> <kidID>... <kidID>
  Εισαγωγή νέας εγγραφής με όνομα name , μοναδικό ID , αριθμό γονιού ParentID, ημερομηνία γέννησης DOB, τόπο γέννησης BirthPlace , αριθμό συζύγου SpouceID , με *κ* παιδιά που έχουν αριθμούς kidID ... kidID.
- load <file>
  Φορτώστε τις εγγραφές που υπάρχουν στο text αρχείο με το όνομα file.
- lookup <ID> || <name>
  Εμφανίζει όλες τις πληροφορίες της εγγραφής που έχει είτε αριθμό ID ή όνομα name.
- print <ID> || <name>
  Τυπώνει όλα τα δεδομένα που συνδέονται με αυτόν τον κόμβο αλλά και όλους που τον ακολουθούν στην

ιεραρχία Για παράδειγμα με την εντολή print <1204> τυπωνουμε όλα τα σχετικά στοιχεία για τον εν λόγω άνθρωπο καθώς επίσης και ΟΛΑ τα στοιχεία για τις έγγραφες που ¨κρέμονται' από τον παραπάνω κόμβο.

- allchildren <ID> || <name>
  Βρειτε όλα τα παιδιά του ατόμου που είτε έχει αριθμό ID ή όνομα name.
- gchildren <ID> || <name>
  Δώστε αριθμητικά πόσα εγγόνια το άτομο με αριθμό ID ή με όνομα name έχει και δώστε τα ονόματά τους.
- gKchildren <ID1> <ID2>
  Είναι το άτομο με ID2 το k–εγγονό του ατόμου με ID1 . Απαντήστε ναι ή όχι και αν ναι, δώστε την αλληλουχία που δείχνει την σχέση.
- anscestry <ID>
  Παρουσιάστε όλους τους προγόνους του ατόμου με ID σε μια σειρά που η μητέρα/πατέρας να είναι πρώτοι, οι παππούς/γιαγιά δεύτεροι κλπ. μέχρι τη ρίζα του γενεαλογικού δένδρου.
- Kcousins <ID1> <ID2>
  Βεβαιώστε οτι το άτομο με <ID1> είναι ξάδελφος με απόσταση K από τον <ID1>.
- avgOffspring <ID1> || <name>
  Για το άτομο που έχει είτε <ID1> D είτε όνομα < name> βρείτε το μέσο όρο όλων των απογόνων του.
- Οποιαδήποτε άλλη εντολή που μπορεί να σας διευκολύνει στην διάρκεια ανάπτυξης της εφαρμογής σας.

Τι πρέπει να Παραδοθεί

1. Μια σύντομη και περιεκτική εξήγηση για τις επιλογές που έχετε κάνει στο σχεδιασμό του προγράμματος σας (1-2 σελίδες σε ASCII κειμένου είναι αρκετές).
2. Οπωσδήποτε ένα Makefile (που να μπορεί να χρησιμοποιηθεί για να γίνει αυτόματα το compile του προγράμματος σας). Πιο πολλές λεπτομέρειες για το (Makefile) και πως αυτό δημιουργείται δίνονται στην ιστοσελίδα του μαθήματος.
3. Ένα tar-file με όλη σας την δουλειά σε έναν κατάλογο που πιθανώς να φέρει το όνομα σας και θα περιέχει όλη σας την δουλειά δηλ. source files, header files, output files (αν υπάρχουν) και οτιδήποτε άλλο χρειάζεται.

Αλλες Σημαντικές Παρατηρήσεις

1. Οι εργασίες είναι **ατομικές.**
2. Αν και αναμένεται να συζητήσετε με φίλους και συνεργάτες το πως θα επιχειρήσετε να δώσετε λύση στο πρόβλημα, αντιγραφή κώδικα (οποιαδήποτε μορφής) είναι κάτι που **δεν επιτρέπεται** και δεν πρέπει να γινει. Οποιοσδήποτε βρεθεί αναμεμειγμένος σε αντιγραφή κώδικά απλά παίρνει μηδέν στο μάθημα. Αυτό ισχύει για **όσους εμπλέκονται** ανεξάρτητα από το ποιος έδωσε/πήρε κλπ.
3. Το πρόγραμμα σας θα πρέπει να τρέχει σε Solaris/Linux αλλιώς **δεν θα βαθμολογηθεί**.
4. Σε καμιά περίπτωση τα MS-Windows **δεν είναι επιλογή** πλατφόρμας για την παρουσίαση αυτής της άσκησης.