

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών  
1η Εργασία - Τμήμα: Περιττών Αριθμών Μητρώου  
Κ22: Λειτουργικά Συστήματα – Χειμερινό Εξάμηνο '14  
Ημερομηνία Ανακοίνωσης: Πέμπτη 9 Οκτωβρίου 2014  
Ημερομηνία Υποβολής: Πέμπτη 30 Οκτωβρίου 2014 Ώρα 23:59

### Εισαγωγή στην Εργασία:

Ο στόχος αυτής της εργασίας είναι να εξοικειωθείτε με το Linux/Solaris και τα βασικά εργαλεία προγραμματισμού και ανάπτυξης λογισμικού στο εν λόγω περιβάλλον.

Θα πρέπει να υλοποιήσετε μια δομή που να προσπελάσει/εισαγάγει/διαγράφει εγγραφές από ένα σημαντικό όγκο δεδομένων που βρίσκονται αποθηκευμένα στην κυρίως μνήμη με κόστος προσπέλασης  $O(1)$ . Προφανώς κάτι τέτοιο μπορείτε να το επιτύχετε όταν χρησιμοποιείτε δομή κατακερματισμού (hashing). Επίσης θα πρέπει να μπορείτε δυναμικά να απαντήσετε ερωτήσεις που έχουν να κάνουν με συγκεκριμένες ομάδες δεδομένων χρησιμοποιώντας δομές τύπου ανεστραμμένου καταλόγου (inverted index). Οι εγγραφές που θα χρησιμοποιήσετε έχουν να κάνουν με την δαπάνη συνδρομητών σε μια εταιρεία κινητής τηλεφωνίας στο τέλος του μήνα. Η εφαρμογή σας πρέπει να δίνει την δυνατότητα σε υπαλλήλους τις εταιρείας όχι μόνο να βλέπουν τα χρωστούμενα ποσά από συγκεκριμένους πελάτες αλλά να επιτυγχάνουν προσθαφαιρέσεις εγγραφών καθώς επίσης και να κάνουν επερωτήσεις για ομάδες χρηστών με βάση την πόλη διαμονής.

Μερικές βασικές προϋποθέσεις για την παραπάνω εφαρμογή είναι οι εξής:

1. Η βασική δομή των δεδομένων –που είναι το hashing– οργανώνεται με βάση τον αριθμό τηλεφώνου κάθε συνδρομητή. Ο εν λόγω αριθμός είναι το κλειδί για την εγγραφή.
2. Η δομή δέχεται εισαγωγές, διαγραφές και επερωτήσεις.
3. Η εγγραφή κάθε συνδρομητή αποτελείται από το τηλέφωνο, όνομα, επίθετο, πόλη διαμονής και ποσό που ο συνδρομητής οφείλει.
4. Ταυτόχρονα η εφαρμογή χρησιμοποιεί έναν inverted index ώστε να ομαδοποιεί τις εγγραφές που υπάρχουν ανά πόλη και έτσι να μπορούμε να ρωτήσουμε την δομή για χαρακτηριστικά και μεγέθη που αφορούν σε συγκεκριμένη ομάδα/-ες.
5. Η εφαρμογή θα πρέπει να μπορεί να διαβάσει δεδομένα στην εκκίνηση της από ένα αρχείο εισόδου.
6. Το πρόγραμμα σας θα πρέπει –όποτε αυτό απαιτείται– να ελευθερώνει όλη την μνήμη που έχει δεσμεύσει. Το ίδιο ισχύει για τον τερματισμό της εφαρμογής.

Μέθοδοι προσπέλασης σαν και αυτές που αναφέρονται παραπάνω είναι πολύ κοινές σε υλοποιήσεις συστημάτων ανάκτησης πληροφορίας, μηχανών αναζήτησης, και πληροφοριακών συστημάτων

### 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 Linux/Solaris) του τμήματος.

Το πρόγραμμα σας (source code) πρέπει να αποτελείται από **τουλάχιστον** δυο (και κατά προτίμηση πιο πολλά) διαφορετικά αρχεία (δηλ. πρέπει να κάνετε χρήση του separate compilation).

Παρακολουθείτε την ιστοσελίδα του μαθήματος για επιπρόσθετες ανακοινώσεις στο URL <http://www.di.uoa.gr/~ad/OSdir/> και την ηλεκτρονική-λίστα (η-λίστα) του μαθήματος στο URL <https://piazza.com/uoa.gr/fall2014/k22sectiondelis/home>

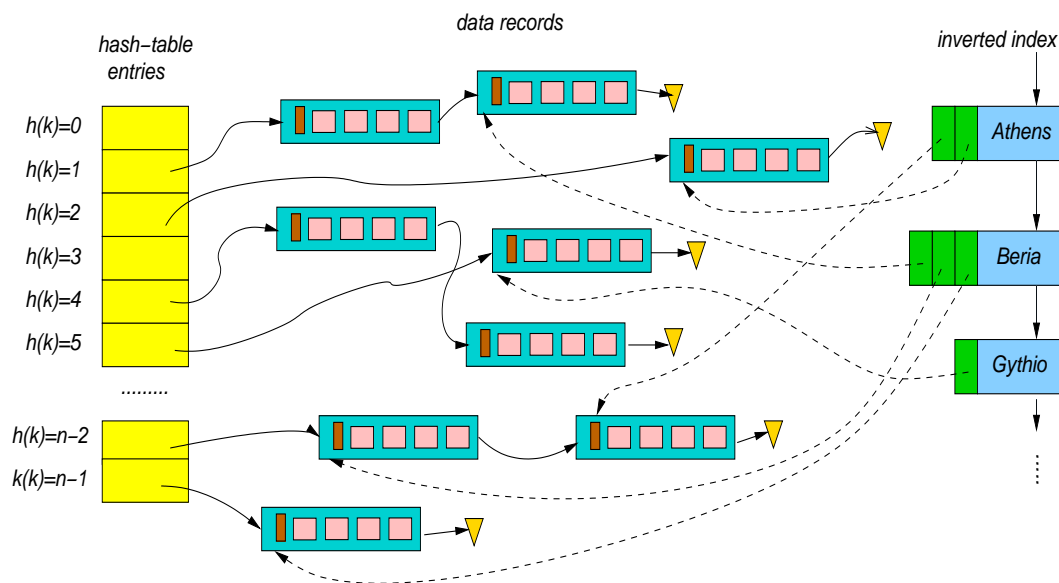
- Υπεύθυνοι για την άσκηση αυτή (ερωτήσεις, αξιολόγηση, βαθμολόγηση κλπ.) ο κ. Γαβριήλ Τζωρτζάκης (g.tzortzakis +AT- di) και ο κ. Αναστάσιος Αντωνιάδης (a.anastasios +AT- di).
- Στην διάρκεια των μαθημάτων των δύο πρώτων εβδομάδων κυκλοφορούμε hard-copy λίστα στην τάξη στην οποία θα πρέπει να δώσετε το όνομά σας και το Unix user-id σας.

Με αυτό το τρόπο μπορούμε να γνωρίζουμε ποιος/-α προτίθεται να υποβάλλει την πρώτη άσκηση και να προβούμε στις κατάλληλες ενέργειες για την τελική υποβολή της άσκησης.

- Με βάση την παραπάνω λίστα, θα σας γράψουμε στο σύστημα piazza.com και πιο συγκεκριμένα στο <https://piazza.com/uoa.gr/fall2014/k22sectiondelis/home>. Μόλις αποδεχτείτε ένα ‘κωδικό’ που θα σας σταλεί, μπορείτε να κάνετε ερωτήσεις και να δείτε απαντήσεις ή/και διευκρινήσεις που δίνονται σχετικά με την άσκηση.
- Η χρήση του piazza.com για το Χειμερινό Εξάμηνο του 2014 παραμένει πιλοτική.

### Η Σύνθεση Δομής που θα Δημιουργήσετε:

Το Σχήμα 1 δείχνει μια μερική αλλά αντιπροσωπευτική κατάσταση της σύνθετης δομής που θα δημιουργήσετε. Οι εγγραφές συνδρομητών είναι οργανωμένες με την βοήθεια ενός πίνακα κατακερματισμού (στο αριστερό



Σχήμα 1: Παράδειγμα της Σύνθετης Δομής για την Εφαρμογή myphones

κομμάτι του σχήματος). Η προσπέλαση στο πίνακα κατακερματισμού γίνεται με την βοήθεια μίας συνάρτησης κατακερματισμού  $h(k)$  όπου  $k$  είναι το κλειδί που είναι ο αριθμός κινητού<sup>1</sup>. Κάθε εγγραφή έχει όλα τα στοιχεία πληροφορίας για τον συνδρομητή που περιγράφει (τηλ., όνομα, πόλη, οφειλή). Η δομή του ανεστραμμένου ευρετηρίου οργανώνει την προσπέλαση στις εγγραφές με βάση το τοπωνύμιο και μπορεί να υλοποιηθεί με χρήση διασυνδεδεμένων λιστών. Έτσι από το στοιχείο της δομής που έχει να κάνει με τους συνδρομητές από την Βέροια έχουμε τρεις δείκτες σε αντίστοιχες εγγραφές (δηλ. των τριών ατόμων που μένουν στην εν λόγω πόλη).

Το μέγεθος του πίνακα κατακερματισμού ορίζεται στην αρχή της εκτέλεσης της εφαρμογής, έχει να κάνει με το πλήθος των εγγραφών που αναμένεται να εισαχθούν στην δομή και παραμένει σταθερό στην διάρκεια εκτέλεσης του προγράμματος. Ο αριθμός των αναμενόμενων εισαγωγών/διαγραφών εγγραφών στην δομή μπορείτε να

<sup>1</sup>Η δομή είναι απλή δομή κατακερματισμού και δεν έχει σχέση με ‘δυναμικό κατακερματισμό’ που πιθανόν έχετε ακούσει σε άλλο μάθημα.

υποθέσετε ότι δεν θα ξεπερνά το 10% των εγγραφών που μαζικά ‘φορτώνονται’ στην δομή όταν ξεκινά η εκτέλεση της εφαρμογής σας.

Η συγκεκριμένη μορφή που παίρνουν ο πίνακας καταμερισμού και ο ανεστραμμένος κατάλογος έχουν να κάνουν με επιλογές σχεδιασμού που θα πρέπει να πάρετε και να περιγράψετε στο σύντομο αρχείο σχεδιασμού που θα υποβάλατε μαζί με τον κώδικα σας. Οι δομές που τελικά θα χρησιμοποιήσετε θα πρέπει να είναι *δυναμικές*.

### Γραμμή Κλήσης της Εφαρμογής:

Η εφαρμογή μπορεί να κληθεί με τον παρακάτω αυστηρό τρόπο:

```
./myphones -l DataFile -o OperationsFile -c config-file
```

όπου

- *myphones* είναι το εκτελέσιμο,
- *DataFile* είναι το (δυαδικό) αρχείο που έχει τις εγγραφές που θα πρέπει αρχικά να εισαχθούν στις δομές της εφαρμογής,
- *OperationsFile* είναι ένα αρχείο σειριακής εισόδου με λειτουργίες που θα πρέπει να εφαρμοστούν στην δομή (δηλ. ερωτήσεις, εισαγωγές, εξαγωγές εγγραφών, κλπ.). Τέτοιες λειτουργίες μπορούν να εισαχθούν και από το prompt της εφαρμογής
- *config-file* είναι ένα προαιρετικό configuration αρχείο που μπορείτε να το χρησιμοποιήσετε ώστε να παραμετροποιήσετε εσείς όπως επιθυμείτε την εφαρμογή σας.

Οι σημαίες *-l/-o/-c* μπορούν να χρησιμοποιηθούν με οποιαδήποτε σειρά στην γραμμή εκτέλεσης του προγράμματος και δεν μπορείτε να κάνετε αλλαγές στη ονομασία τους. Από τις παραπάνω in-line παραμέτρους καμία δεν είναι υποχρεωτική για την επιτυχή εκτέλεση του προγράμματος (αλλά πρέπει να υλοποιηθούν).

### Περιγραφή της Διεπαφής του Προγράμματος:

Η εφαρμογή μέσω ενός prompt επιτρέπει στον χρήστη να αλληλεπιδρά με την δομή και να ανασύρει, αποθηκεύει, ή υπολογίζει διάφορες πληροφορίες με τις παρακάτω εντολές (η μορφή των εντολών είναι αυστηρή και θα πρέπει να χρησιμοποιηθούν όπως ακριβώς ορίζονται παρακάτω):

1. **l(oad) DataFile**: εισήγαγε στην δομή τις εγγραφές που περιγράφονται στο αρχείο *DataFile*. Το πρόγραμμα διαβάζει το παραπάνω (δυαδικό) αρχείο που δεν έχει προκαθορισμένο μέγεθος όσον αφορά στον αριθμό εγγραφών. Οι εγγραφές εισάγονται μία μία και η δομή μεγαλώνει δυναμικά. Ταυτόχρονα με την δημιουργία του πίνακα κατακερματισμού δημιουργείται και ο ανεστραμμένος κατάλογος.
2. **i(nsert) phoneno firstname lastname town amount**: εισήγαγε στην δομή ένα συνδρομητή με τα προφανή παραπάνω στοιχεία που ζει στη πόλη *town* και έχει την οφειλή *amount*. Το *phoneno* λειτουργεί σαν κλειδί για την εισαγωγή σχετικών της εγγραφής στο πίνακα κατακερματισμού.
3. **d(elte) phoneno**: διέγραψε από την δομή την εγγραφή που έχει σαν κλειδί το *phoneno*.
4. **q(uey) phoneno**: ανέσυρε και τύπωσε όλη την εγγραφή του χρήστη με τηλεφωνικό αριθμό *phoneno*.
5. **p(opulation) town**: για την πόλη *town* δώσε το αριθμό των συνδρομητών που μένουν εκεί.
6. **t(op) k**: παρουσίασε τις *k* πόλεις που αριθμητικά διαθέτουν τους πιο πολλούς συνδρομητές.
7. **s(um) town**: για την πόλη *town* παρουσίασε το συνολικό ποσό που οφείλουν οι κάτοικοι της.
8. **f(ind)t(own) town l**: για την πόλη *town* παρουσίασε παρουσίασε τους *l* υψηλότερους οφειλέτες.
9. **f(ind)t(op)s(pender)**: για κάθε πόλη παρουσίασε τον/ην μεγαλύτερο/η οφειλέτη. Το αποτέλεσμα εμφανίζεται σε σε φθίνουσα σειρά ποσού οφειλής.
10. **min / max**: παρουσίασε τον αριθμητικά μικρότερο/μεγαλύτερο τηλεφωνικό αριθμό που βρίσκεται στην δομή.
11. **e(exit)**: το πρόγραμμα απλά τερματίζει αφού ελευθερώσει πρώτα όλο το χώρο που έχει καταλάβει στην

μνήμη.

Κάθε εγγραφή που αποθηκεύεται στην δομή(-ες) αποτελείται από:

- Αριθμός τηλεφώνου (μοναδικός ακεραίος αριθμός ή μοναδική σειρά χαρακτήρων)
- Όνομα συνδρομητή (σειρά χαρακτήρων)
- Επίθετο συνδρομητή (σειρά χαρακτήρων)
- Πόλη Κατοικίας (σειρά χαρακτήρων)
- Ποσό οφειλής (πραγματικός αριθμός)

### Χαρακτηριστικά του Προγράμματος που Πρέπει να Γράψετε:

1. Δεν μπορείτε να κάνετε pre-allocate οποιοσδήποτε χώρο αφού η δομή(-ές) θα πρέπει να μπορεί(-ουν) να μεγαλώσει(-ουν) χωρίς ουσιαστικά κανέναν περιορισμό όσον αφορά στον αριθμό των εγγραφών που μπορούν να αποθηκεύσουν. Η χρήση στατικών πινάκων/δομών που δεσμεύονται στην διάρκεια της συμβολομετάφρασης του προγράμματος σας δεν είναι αποδεκτές επιλογές.
2. Εκτός τις βασικές δομές μπορείτε να χρησιμοποιήσετε οποιαδήποτε άλλη δομή σας δίνει την δυνατότητα να απαντήσετε όλα τα ερωτήματα με πολυπλοκότητα  $O(1)$ . Θα πρέπει να περιγράψετε στην υποβληθείσα γραπτή εξήγηση πώς πετυχαίνετε κάτι τέτοιο.
3. Πριν να τερματίσει η εκτέλεση της εφαρμογής, το περιεχόμενο των δομών απελευθερώνεται συνολικά.
4. Αν πιθανώς κάποια κομμάτια του κωδικά σας προέλθουν από κάποια δημόσια πηγή, θα πρέπει να δώσετε αναφορά στη εν λόγω πηγή είτε αυτή είναι βιβλίο, σημειώσεις, Internet URL κλπ. και να εξηγήσετε πως ακριβώς χρησιμοποιήσατε την εν λόγω αναφορά.
5. Έχετε πλήρη ελευθερία να επιλέξετε το τρόπο με τον οποίο τελικά θα υλοποιήσετε βοηθητικές δομές.

### 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. Οποσδήποτε 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.

### Άλλες Σημαντικές Παρατηρήσεις:

1. Οι εργασίες είναι **ατομικές**.
2. Το πρόγραμμά σας θα πρέπει να τρέχει σε Linux/Solaris αλλιώς **δεν θα βαθμολογηθεί**.
3. 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 γίνει. Οποιοσδήποτε βρεθεί αναμειγμένος σε αντιγραφή κώδικά απλά παίρνει μηδέν στο μάθημα. Αυτό ισχύει για **όσους εμπλέκονται ανεξάρτητα** από το ποιος έδωσε/πήρε κλπ.
4. Προγράμματα που **δεν χρησιμοποιούν** separate compilation χάνουν αυτόματα 5% του βαθμού.
5. Σε καμιά περίπτωση τα MS-Windows **δεν είναι επιλέξιμη** πλατφόρμα για την παρουσίαση αυτής της άσκησης.