

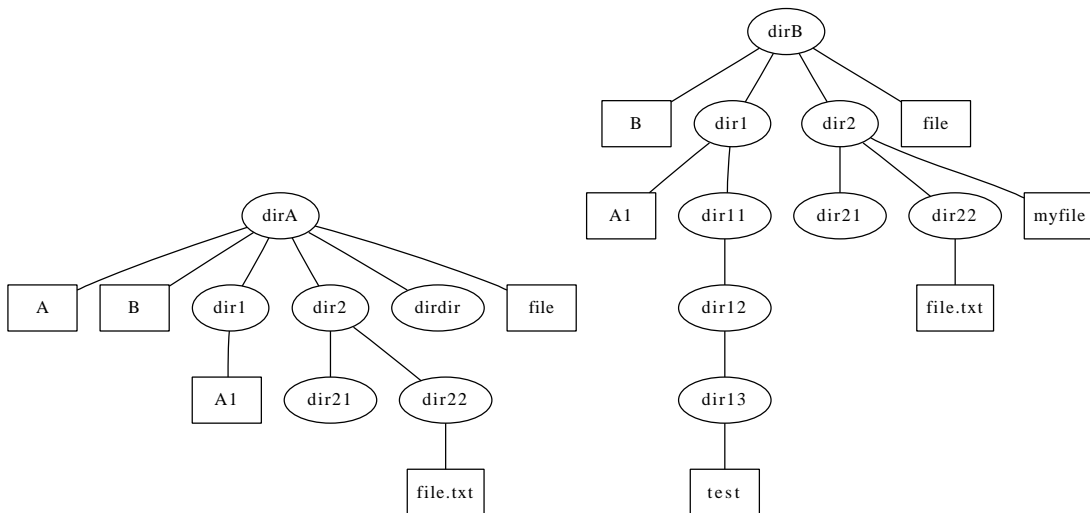
**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**  
**Τμήμα Πληροφορικής και Τηλεπικοινωνιών**  
**Κ22: Λειτουργικά Συστήματα - Περίττοι Αριθμοί**  
**4η Προγραμματιστική Εργασία**  
**Ημερομηνία Ανακοίνωσης: 30 Ιανουαρίου 2012**  
**Ημερομηνία Υποβολής: 5 Μαρτίου 2012**

### Εισαγωγή στην Εργασία:

Στα πλαίσια αυτής της άσκησης πρέπει να γράψετε ένα πρόγραμμα γραμμής εντολών που να συγκρίνει ιεραρχίες καταλόγων μεταξύ τους. Το πρόγραμμα που ονομάζεται `catalogs` θα μπορεί επίσης να δημιουργεί μια νέα ιεραρχία καταλόγων συγχωνεύοντας τους δύο υπό σύγκριση καταλόγους σε ένα νέο.

Αν το `catalogs` συγκρίνει δύο ίδιες ιεραρχίες καταλόγων δεν εκτυπώνεται τίποτα. Διαφορετικά, για κάθε διαφορά που παρατηρείται, θα πρέπει να εκτυπώνεται το αντίστοιχο μήνυμα, είτε πρόκειται για διαφορετικούς φακέλους, είτε για διαφορετικά αρχεία και συνδέσμους.

Το πρόγραμμα πρέπει να μπορεί (1) να συγκρίνει δύο καταλόγους και να εμφανίζει τις διαφορές τους καθώς και (2) να δημιουργεί ένα νέο κατάλογο ο οποίος προκύπτει από τη συγχώνευση των δύο καταλόγων. Στη πρώτη περίπτωση η κλήση του προγράμματος γίνεται με ορίσματα τα ονόματα των δύο φακέλων, ενώ στη δεύτερη περίπτωση καλείται με ένα επιπλέον όρισμα, το όνομα του φακέλου που θέλουμε να δημιουργηθεί μετά τη συγχώνευση. Για παράδειγμα: για τις ιεραρχίες καταλόγων του Σχήματος 1, μία ενδεικτική εκτέλεση του προγράμματος για σύγκριση των ιεραρχιών



Σχήμα 1: Δύο καταλόγοι υπό σύγκριση

είναι η εξής:

```
$ catalogs dirA dirB
In dirA :
  A
  dirdir
In dirB:
  dir1/dir11
  dir1/dir11/dir12
  dir1/dir11/dir12/dir13
  dir1/dir11/dir12/dir13/test
  dir2/myfile
```

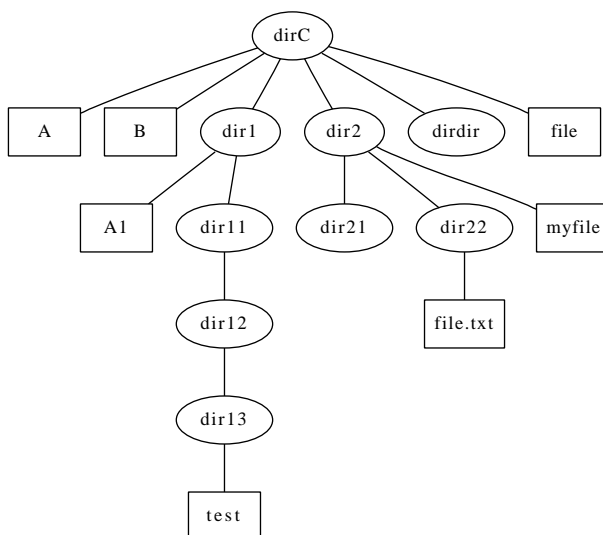
Η σειρά εμφάνισης των διαφορών δεν παίζει κάποιο ουσιαστικό ρόλο. Είστε ελεύθεροι να επιλέξετε ότι διαγνωστικά μηνύματα κρίνετε απαραίτητο ότι θα πρέπει να παρουσιαστούν καθώς επίσης και την σειρά τους (ωστόσο όλα να

εμφανίζονται με κάποιο ευανάγνωστο και κατανοητό τρόπο).

Το αποτέλεσμα εκτέλεσης:

```
$ catalogs dirA dirB dirC
```

εμφανίζεται στο Σχήμα 2.



Σχήμα 2: Αποτέλεσμα συγχώνευσης των δύο καταλόγων του Σχήματος 1

### Διαδικαστικά:

- ◊ Το πρόγραμμά σας θα πρέπει να γραφτεί σε C (ή C++ χωρίς όμως STL extensions) και να τρέχει στις μηχανές Unix (Linux/Solaris) του τμήματος
- ◊ Παρακολουθήστε την ιστοσελίδα του μαθήματος για επιπρόσθετες ανακοινώσεις στο <http://www.di.uoa.gr/~ad/OSdir/index.html> όπως και την λίστα mailman του μαθήματος
- ◊ Υπεύθυνοι για την άσκηση (δηλ. ερωτήσεις, αξιολόγηση, καθώς επίσης και βαθμολόγησή) είναι οι κ. Αλέξανδρος Αντωνιάδης (a.antoniadis-AT+di) και Γιάννης Γκερμπεσιώτης (j.gerbesiotis+AT-di).

### Διατύπωση του Προβλήματος:

Το πρόγραμμα σας θα πρέπει να δέχεται στη γραμμή εντολών τις αντίστοιχες παραμέτρους, ανάλογα αν η χρήση είναι για τον έλεγχο των διαφορών, ή και για την συγχώνευση αρχείων. Αν δε δοθούν σωστά τα ορίσματα αυτά θα πρέπει να τυπώνεται κάποιο μήνυμα λάθους και πληροφορίες για τη σωστή χρήση του προγράμματος. Στη περίπτωση της συγχώνευσης, αν ο κατάλογος προορισμού δεν υπάρχει, τότε το πρόγραμμα σας θα πρέπει να τον δημιουργεί.

Στην συνέχεια της περιγραφής θεωρείστε ότι οι δύο ιεραρχίες καταλόγων είναι οι A και B. Το πρόγραμμά σας θα πρέπει να διατρέχει τις δύο ιεραρχίες και να συγκρίνει καταλόγους και αρχεία, με το τρόπο που περιγράφεται παρακάτω, ώστε να βρίσκει διαφορές. Στην περίπτωση της σύγκρισης το πρόγραμμά σας απλά θα πρέπει να εκτυπώνει τις διαφορές αυτές, ενώ στην περίπτωση της συγχώνευσης θα πρέπει να χειρίζεται κατάλληλα την πληροφορία αυτή ώστε να δημιουργεί τον νέο κατάλογο χωρίς πλεονασμούς.

Οι πληροφορίες που θα χρειαστούν για την σύγκριση των αρχείων και των καταλόγων θα αντλούνται από τα i-nodes με την κλήση συστήματος stat. Κάθε φορά πρέπει να εξετάζονται i-nodes που βρίσκονται στο ίδιο σχετικό μονοπάτι.

Παρακάτω θεωρούμε  $\alpha$  ένα  $i$ -node κάτω από την ιεραρχία  $A$  και  $\beta$  ένα  $i$ -node κάτω από την ιεραρχία  $B$  στο αντίστοιχο σχετικό μονοπάτι. Για το επίπεδο αυτό, θα θεωρούμε ότι τα  $\alpha$  και  $\beta$  είναι διαφορετικά, αν ισχύει έστω και ένα από τα εξής:

1. Αν δεν υπάρχει το  $\beta$
2. Αν δεν υπάρχει το  $\alpha$
3. Αν  $\alpha$ ,  $\beta$  ανήκουν σε διαφορετικές κατηγορίες (κανονικά αρχεία, συμβολικοί σύνδεσμοι, κατάλογοι)
4. Αν υπάρχουν  $\alpha$ ,  $\beta$  και δεν είναι ίδια σύμφωνα με τις συνθήκες που περιγράφονται αναλυτικά παρακάτω για αρχεία, συνδέσμους και καταλόγους.

Δύο αρχεία είναι ίδια αν και μόνο αν:

1. Έχουν το ίδιο σχετικό μονοπάτι από την αρχή της ιεραρχίας.
2. Έχουν το ίδιο όνομα.
3. Έχουν το ίδιο μέγεθος.

Παρότι μπορεί να μην ισχύει πάντα, αγνοήστε άλλες περιπτώσεις και οι πληροφορίες που χρειάζονται για τις συγκρίσεις υπάρχουν στα  $i$ -nodes και δε πρέπει να διαβάσετε τα ίδια τα αρχεία (δηλ. περιεχόμενο).

Δυο σύνδεσμοι (links) θεωρούνται ίδια αν και μόνο αν:

1. Έχουν το ίδιο σχετικό μονοπάτι από την αρχή της ιεραρχίας.
2. Έχουν το ίδιο όνομα.
3. Δείχνουν σε ίδια αρχεία

Δεν χρειάζεται να λάβετε υπόψιν links που δείχνουν εκτός των ιεραρχιών για λόγους απλότητας.

Δύο κατάλογοι θεωρούνται ίδιο αν και μόνο αν:

1. Έχουν το ίδιο σχετικό μονοπάτι από την αρχή της ιεραρχίας.
2. Έχουν το ίδιο όνομα.

Σε κάθε άλλη περίπτωση όλα τα παραπάνω (κατάλογοι, αρχεία, σύνδεσμοι) θεωρούνται διαφορετικά.

Στην περίπτωση που τα  $\alpha$  και  $\beta$  αντιστοιχούν σε καταλόγους, πρέπει τα παραπάνω βήματα να εφαρμοστούν αναδρομικά και στα περιεχόμενα των καταλόγων.

Κατά τη διαδικασία της συγχώνευσης πρέπει να δημιουργήσετε μια νέα ιεραρχία αντιγράφοντας τα αρχεία που είναι κοινά και στις δύο ιεραρχίες και στην συνέχεια αντιγράφοντας τις αλλαγές που βρέθηκαν με τη προηγούμενη διαδικασία. Επίσης, θα πρέπει να χειριστείτε τις περιπτώσεις που δύο αρχεία ή σύνδεσμοι με το ίδιο όνομα δεν είναι ίδια. Στην περίπτωση αυτή θα πρέπει να κρατάτε μόνο αυτό που τροποποιήθηκε τελευταίο, δηλαδή έχει μεταγενέστερη ημερομηνία τροποποίησης.

Επιπλέον, θα πρέπει να χειριστείτε σωστά την περίπτωση των συμβολικών και ισχυρών συνδέσμων (symbolic/hard links). Οι σύνδεσμοι θα πρέπει να διατηρούνται και στη νέα ιεραρχία. Πιο συγκεκριμένα, οι συμβολικοί σύνδεσμοι πρέπει να δείχνουν στο αντίστοιχο μονοπάτι στη νέα ιεραρχία. Στη περίπτωση των ισχυρών συνδέσμων, πρέπει τα δεδομένα των αρχείων να μην αντιγράφονται στην τελική ιεραρχία πάνω από μία φορά.

## Υπόβαθρο:

Τα αρχεία στο λειτουργικό σύστημα Linux (Unix) καταγράφονται σε δύο διαφορετικά επίπεδα, των ονομάτων και των κόμβων-δεικτών ( $i$ -nodes). Κάθε  $i$ -node συνδέεται με τα δεδομένα ενός και μόνο αρχείου και τηρεί όλες τις πληροφορίες εκτός από το όνομα του αρχείου (π.-χ. αριθμός  $i$ -node, ημερομηνία τελευταίας αλλαγής, μέγεθος). Η σχέση είναι αμφιμονοσήμαντη, καθώς κάθε αρχείο δεικτοδοτείται από ένα και μόνο  $i$ -node. Όμως ένα  $i$ -node μπορεί να είναι συνδεδεμένο με πολλαπλά ονόματα αρχείων (αυτό επιτυγχάνουν οι σύνδεσμοι, τα λεγόμενα hard links). Τα ονόματα δημιουργούν ένα κατευθυνόμενο άκυκλο γράφο (δέντρο) καθώς:

- Οι κατάλογοι, που και αυτοί είναι ονόματα, περιλαμβάνουν με τη σειρά τους άλλα ονόματα.
- Δεν επιτρέπονται οι σύνδεσμοι σε καταλόγους, οπότε αποκλείονται οι κύκλοι.

Ένας τρόπος να δούμε το i-node ενός αρχείου είναι η εντολή `ls -i` (σε επίπεδο προγράμματος συστήματος) όπως και η `stat` (σε επίπεδο κλήσης συστήματος). Ειδικά στο Linux υπάρχει και το πρόγραμμα συστήματος με το ίδιο όνομα `stat` που κάνει την ίδια δουλειά σε επίπεδο κελύφους.

Η δημιουργία ισχυρών συνδέσμων σε υπάρχον αρχείο γίνεται με την εντολή `ln <existing_name> <new_name>` και με παράμετρο `-s` για συμβολικούς συνδέσμούς. Για τον έλεγχο των συμβολικών συνδέσμων θα πρέπει να χρησιμοποιηθεί η εντολή `readlink` τόσο σε επίπεδο προγράμματος συστήματος αλλά και κλήσης συστήματος.

Προφανώς, η χρήση οποιουδήποτε προγράμματος συστήματος (π.χ. `cp -R`) για την παροχή της απαιτούμενης λειτουργικότητας δεν επιτρέπεται.

### Τι πρέπει να παραδοθεί:

1. Όλη η δουλειά σας σε ένα tar-file που να περιέχει όλα τα source files, header files, output files, makefile.
2. Μια σύντομη περιγραφή (1-2 σελίδες) για τις επιλογές που κάνατε στο σχεδιασμό της άσκησης.
3. Ένα Makefile που χρησιμοποιήσατε για να γίνει αυτόματα compile το πρόγραμμα σας.

### Άλλες 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 γίνει. Οποιοσδήποτε βρεθεί αναμειγμένος σε αντιγραφή κώδικα απλά παίρνει μηδέν στο μάθημα. Αυτό ισχύει για **όλους όσους εμπλέκονται** ανεξάρτητα από το ποιος έδωσε/πήρε κλπ.
3. Το πρόγραμμα σας θα πρέπει να τρέχει σε Solaris ή Linux αλλιώς **δεν θα βαθμολογηθεί**.
4. Σε καμιά περίπτωση τα MS-Windows **δεν είναι επιλογή** πλατφόρμας για την παρουσίαση αυτής της άσκησης.