ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019/2020

Εισαγωγή

Στα πλαίσια της άσκησης θα ασχοληθούμε με το πρόβλημα της τοπολογικής ταξινόμησης ενός κατευθυνόμενου γραφήματος, το οποίο βρίσκει εφαρμογή (μεταξύ άλλων) στην χρονοδρομολόγηση μιας ακολουθίας εργασιών βάσει των εξαρτήσεων που έχουν μεταξύ τους.

Τοπολογική ταξινόμηση

Η τοπολογική ταξινόμηση (ή αλλιώς τοπολογική διάταξη) ενός κατευθυνόμενου γραφήματος είναι μια γραμμική διάταξη των κόμβων του γραφήματος, τέτοια ώστε για κάθε ακμή e_{uv} από τον κόμβο *u* στον κόμβο *v*, ο *u* να εμφανίζεται πριν από τον *v* στην διάταξη. Για παράδειγμα, οι κόμβοι του γραφήματος μπορεί να αναπαριστούν εργασίες που πρέπει να πραγματοποιηθούν και η κάθε ακμή μπορεί να αναπαριστά τον περιορισμό πως η έναρξη μιας εργασίας πρέπει να ακολουθεί την ολοκλήρωση μιας άλλης. Σε μια τέτοια εφαρμογή, μια τοπολογική ταξινόμηση απλώς υποδεικνύει μια επιτρεπτή σειρά πραγματοποίησης των εργασιών. Μια τοπολογική ταξινόμηση είναι εφικτή μόνο αν το γράφημα δεν έχει κατευθυνόμενους κύκλους, δηλαδή μόνο αν το γράφημα είναι ένα Κατευθυνόμενο Ακυκλικό Γράφημα (Directed Acyclic Graph ή DAG). Κάθε τέτοιο γράφημα έχει τουλάχιστον μια τοπολογική ταξινόμηση.

Μια τυπική εφαρμογή της τοπολογικής ταξινόμησης είναι η χρονοδρομολόγηση μιας ακολουθίας εργασιών βάσει των εξαρτήσεων που έχουν μεταξύ τους. Οι εργασίες αναπαρίστανται μέσω των κόμβων και υπάρχει ακμή από την εργασία x προς την εργασία y αν η εργασία x πρέπει να ολοκληρωθεί πριν ξεκινήσει η εργασία y. Όπως προαναφέρθηκε, μια τοπολογική ταξινόμηση καθορίζει σε αυτήν την περίπτωση την σειρά εκτέλεσης των εργασιών. Συναφείς εφαρμογές της τοπολογικής ταξινόμησης έχουν μελετηθεί ήδη από την δεκαετία του 1960, στα πλαίσια της τεχνικής PERT για την δρομολόγηση εργασιών στην διαχείριση έργων (project management). Στην περίπτωση αυτή οι κόμβοι αναπαριστούν τα ορόσημα (milestones) του έργου και οι ακμές αναπαριστούν εργασίες που πρέπει να πραγματοποιηθούν μεταξύ οροσήμων. Η τοπολογική ταξινόμηση επιτρέπει στην περίπτωση αυτή την εύρεση του κρίσιμου μονοπατιού (critical path) του έργου, δηλαδή μια ακολουθία οροσήμων και εργασιών που καθορίζουν την διάρκεια εκτέλεσης του συνολικού έργου. Στην επιστήμη των υπολογιστών εφαρμογές της τοπολογικής ταξινόμησης εμφανίζονται σε ζητήματα δρομολόγησης εντολών (instruction scheduling), στην σειρά αποτίμησης των εκφράσεων στα κελιά ενός προγράμματος λογιστικών φύλλων (spreadsheet) όταν αλλάξει μια τιμή σε κάποιο κελί, στον καθορισμό της σειράς μετάφρασης αρχείων σε makefiles, στον καθορισμό της σειράς φόρτωσης πινάκων με ξένα κλειδιά (foreign keys) σε βάσεις δεδομένων, κλπ. Αποτελεί επομένως ένα πρόβλημα με ευρύ φάσμα εφαρμογών.

Ο αλγόριθμος του Kahn

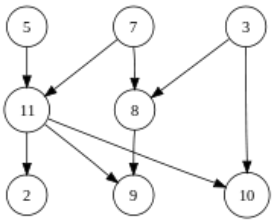
Ο αλγόριθμος που θα υλοποιήσετε στα πλαίσια της εργασίας είναι από τους απλούστερους που επιτυγχάνουν τοπολογική ταξινόμηση. Μια περιγραφή του ακολουθιακού αλγορίθμου μπορεί να βρεθεί στο [1]. Παραθέτουμε ωστόσο τον ψευδοκώδικα προς διευκόλυνση.

```
L ← Empty list that will contain the sorted elements
S ← Set of all nodes with no incoming edge

while S is non-empty do
    remove a node n from S
    add n to tail of L
    for each node m with an edge e from n to m do
        remove edge e from the graph
        if m has no other incoming edges then
            insert m into S

if graph has edges then
    return error   (graph has at least one cycle)
else
    return L       (a topologically sorted order)
```

Θα ξεκινήσουμε με μια γενική περιγραφή του αλγορίθμου και στην συνέχεια θα εστιάσουμε περαιτέρω σε ζητήματα υλοποίησης. Για κάθε κόμβο του γραφήματος διατηρούμε πληροφορία για τον βαθμό εισόδου (πλήθος ακμών που καταλήγουν στον κόμβο) και ξεκινάμε δημιουργώντας ένα σύνολο S που περιέχει τους κόμβους με βαθμό εισόδου 0 (κόμβοι που δεν εξαρτώνται από άλλους και οι εργασίες που αντιστοιχούν σε αυτούς μπορούν αμέσως να πραγματοποιηθούν). Στο σχήμα που ακολουθεί, αυτό αντιστοιχεί στους κόμβους 5, 7 και 3. Επομένως αρχικά S = {5, 7, 3}. Προφανώς η σειρά των κόμβων στο σύνολο δεν έχει σημασία.



Στην συνέχεια, επιλέγεται ένας κόμβος του συνόλου S, μεταφέρεται στο τέλος μιας λίστας L, αφαιρούνται από το γράφημα οι ακμές που ξεκινάνε από τον κόμβο, ενημερώνονται οι βαθμοί εισόδου των κόμβων που επηρεάζονται και προστίθενται στο σύνολο S οι κόμβοι για τους οποίους ο βαθμός εισόδου έγινε 0. Ο αλγόριθμος αυτός επαναλαμβάνεται έως ότου το σύνολο S μείνει κενό. Για παράδειγμα, αν επιλεγεί ο κόμβος 5 από το σύνολο S, τότε αυτός μεταφέρεται στην λίστα L, αφαιρούνται όλες οι ακμές που ξεκινάνε από τον κόμβο (στην περίπτωση μας μόνο προς τον κόμβο 11), ενημερώνονται οι βαθμοί εισόδου των κόμβων που επηρεάζονται (ο βαθμός εισόδου του κόμβου 11 γίνεται 1 από 2 που ήταν αρχικά) και αν κάποιοι από τους κόμβους που επηρεάζονται έχουν πλέον βαθμό εισόδου 0 τότε προστίθενται στο σύνολο S (γιατί δεν έχουν εξαρτήσεις πλέον και μπορούν να εκτελεστούν).

Παρατηρήστε κατ' αρχάς πως σε κάθε επανάληψη του αλγορίθμου δεν έχει σημασία ποιον κόμβο αφαιρούμε (όλοι όσοι είναι στο σύνολο S είναι έτοιμοι προς εκτέλεση), ενώ επιπλέον μπορούμε να επιλέγουμε από το σύνολο S και να επεξεργαζόμαστε παράλληλα περισσότερους κόμβους (ή ακόμα και όλους τους κόμβους, αν υπάρχουν αρκετοί επεξεργαστές διαθέσιμοι). Το αποτέλεσμα (τοπολογική διάταξη των κόμβων) υπάρχει στο τέλος εκτέλεσης του αλγόριθμου στην λίστα L.

Όπως προαναφέρθηκε, η παραπάνω περιγραφή είναι γενική και παρατίθεται ώστε να γίνει σαφής η διαδικασία που ακολουθείται. Υπάρχουν ωστόσο αρκετά ζητήματα που απαιτούν προσοχή σε μια παράλληλη υλοποίηση του αλγορίθμου, τόσο για την διασφάλιση της ορθότητας του παράλληλου αλγορίθμου, όσο και για την επίτευξη ικανοποιητικής επίδοσης. Σε μια παράλληλη υλοποίηση του αλγορίθμου κάθε νήμα του προγράμματος θα πρέπει να εκτελεί τον παραπάνω αλγόριθμο, το οποίο δημιουργεί μια σειρά ζητημάτων. Μια (μη εξαντλητική) λίστα τέτοιων ζητημάτων στα οποία θα πρέπει να δώσετε προσοχή ακολουθεί:

- Για την υλοποίηση του συνόλου S θα μπορούσε να χρησιμοποιηθεί μια κεντρική δομή (π.χ. μια ουρά), προσπελάσιμη από όλα τα νήματα του προγράμματος. Προσέξτε πως η αφαίρεση/προσθήκη κόμβων στο S μπορεί να πραγματοποιείται ταυτόχρονα από περισσότερα νήματα, ενώ το ίδιο ισχύει για την προσθήκη κόμβων στην λίστα L. Απαιτείται επομένως προσοχή ως προς την διατήρηση της συνέπειας των δομών αυτών. Ποια θα ήταν η επίπτωση μιας τέτοιας επιλογής υλοποίησης των δομών αυτών στην επίδοση;
- Εναλλακτικά, κάθε νήμα θα μπορούσε να έχει μια **τοπική** δομή S όπου θα κρατάει τους κόμβους της δικής του ευθύνης των οποίων ο βαθμός εισόδου γίνεται 0. Ποια θα ήταν η επίπτωση μιας τέτοιας επιλογής στον καταμερισμό του φόρτου εργασίας (κατά συνέπεια και της επίδοσης) μεταξύ νημάτων; Θα μπορούσαν να συνδυαστούν οι δύο προσεγγίσεις για την επίλυση των ζητημάτων επίδοσης;
- Στην περιγραφή αναφέρεται πως αφαιρούνται ακμές από το γράφημα. Προγραμματιστικά αυτό δεν είναι απαραίτητο και αρκεί να ενημερώνεται ο βαθμός εισόδου των κόμβων που επηρεάζονται. Προσέξτε όμως την εξής περίπτωση: δύο ή περισσότερα νήματα επεξεργάζονται σε δεδομένη χρονική στιγμή διαφορετικούς κόμβους. Κάποιοι από αυτούς τους κόμβους μπορεί να έχουν ακμή προς έναν (ή περισσότερους) **κοινούς** κόμβους. Σε αυτήν την περίπτωση θα πρέπει να ενημερώσουν ταυτόχρονα τον βαθμό εισόδου του ίδιου, κοινού κόμβου (race condition). Λόγω αυτού του γεγονότος θα πρέπει να διασφαλίσετε επιπλέον πως μόνο ένα νήμα θα μεταφέρει τον κόμβο στο σύνολο S, όταν ο βαθμός εισόδου του γίνει 0.

  Για παράδειγμα, υποθέστε με βάση το παραπάνω σχήμα πως ένα νήμα επεξεργάζεται τον κόμβο 7 και ένα άλλο νήμα τον κόμβο 3. Τότε υπάρχει πιθανότητα τα δύο αυτά νήματα να ενημερώσουν ταυτόχρονα τον βαθμό εισόδου του κόμβου 8. Επιπλέον, μόνο ένα από τα νήματα που επεξεργάζονται τους κόμβους 7 και 3 θα πρέπει να βάλει τον κόμβο 8 στο σύνολο S.
- Όταν ένα νήμα επεξεργάζεται έναν κόμβο από το σύνολο S, με ποια σειρά θα πρέπει να γίνει η αφαίρεση του κόμβου και η προσθήκη των νέων κόμβων για τους οποίους ο βαθμός εισόδου έγινε 0 στο S; Προσέξτε πως την ίδια στιγμή κάποιο άλλο νήμα μπορεί να ελέγχει αν το σύνολο S είναι κενό και αν αυτό (έστω και παροδικά) ισχύει να τερματίσει πρόωρα την εκτέλεση του. Στο παραπάνω σχήμα, το σύνολο S αρχικά θα περιέχει τους κόμβους 5, 7 και 3. Αν έχουμε 4 νήματα στο πρόγραμμα μας, 3 από αυτά θα πάρουν τους κόμβους του S προς επεξεργασία. Το τέταρτο νήμα θα δει πως το σύνολο S είναι κενό και θα τερματίσει, ενώ στην πραγματικότητα δεν έχουμε επεξεργαστεί όλους τους κόμβους του γραφήματος. Πως θα πρέπει να τροποποιηθεί η συνθήκη εξόδου του αλγορίθμου;

Αναφορές

[1] https://en.wikipedia.org/wiki/Topological_sorting

Ζητούμενα της άσκησης

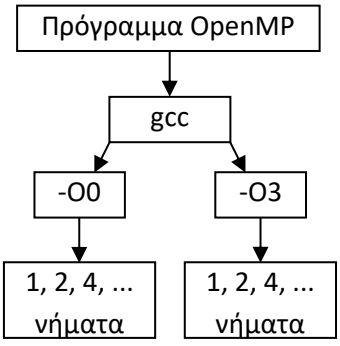
Στα πλαίσια της άσκησης ζητείται να υλοποιήσετε την παράλληλη εκδοχή του αλγορίθμου Kahn όπως αναφέρθηκε παραπάνω σε C **(όχι C++) με χρήση έργων (tasks) του προγραμματιστικού μοντέλου OpenMP**. Η επεξεργασία κάθε κόμβου του γραφήματος θα πρέπει να πραγματοποιείται στα πλαίσια ενός έργου (task).

Θα πρέπει επίσης να ορίσετε κατάλληλες δομές για το σύνολο S, την λίστα L και τους βαθμούς εισόδου κάθε κόμβου. Όπως προαναφέρθηκε, το σύνολο S θα μπορούσε να αναπαρασταθεί ως μια ουρά, προσπελάσιμη από όλα τα νήματα του προγράμματος OpenMP ή ως ένα σύνολο ξεχωριστών ουρών (μια ανά νήμα) ή και συνδυασμό αυτών (με σκοπό την βελτιστοποίηση της επίδοσης). Η λίστα L θα μπορούσε να υλοποιηθεί ως μια διασυνδεδεμένη λίστα. Προφανώς είστε ελεύθεροι να αποφασίσετε να χρησιμοποιήσετε διαφορετικές δομές και γενικότερα είστε ελεύθεροι να βελτιστοποιήσετε τον κώδικα με όποιον άλλο τρόπο θέλετε. Ωστόσο, θεωρείται αυτονόητο πως η υλοποίηση θα πρέπει να είναι σωστή και πως **στην αναφορά που θα παραδώσετε θα πρέπει να αιτιολογήσετε γιατί παραλληλοποιήσατε με τον συγκεκριμένο τρόπο την εφαρμογή.**

Στην αναφορά σας να παραθέσετε μετρήσεις και να σχολιάσετε τις παρακάτω περιπτώσεις:

1) Μεταγλώττιση και εκτέλεση του προγράμματος σας χωρίς βελτιστοποιήσεις (-O0) και με μέγιστες βελτιστοποιήσεις (-O3).
2) Εκτέλεση κάθε περίπτωσης με 1, 2 και 4 νήματα τουλάχιστον. Αν το σύστημα σας διαθέτει περισσότερους πυρήνες, ακόμα καλύτερα!

Για κάθε έναν από τους παραπάνω συνδυασμούς συμπεριλάβετε στην αναφορά σας πίνακες με τους χρόνους εκτέλεσης **και διαγράμματα της χρονοβελτίωσης. Για την καταγραφή του χρόνου εκτέλεσης χρησιμοποιείστε στον κώδικα σας την συνάρτηση gettimeofday() και χρονομετρήστε μόνο την εκτέλεση της παράλληλης περιοχής (μην συμπεριλάβετε διάβασμα αρχείων, δεσμεύσεις μνήμης, αρχικοποιήσεις, κλπ)**. Θα πρέπει δηλαδή να καλέσετε μια φορά την συνάρτηση πριν μπείτε στην παράλληλη περιοχή και μια φορά μετά την παράλληλη περιοχή, όταν θα έχουν τερματίσει όλα τα νήματα. Συγκεντρωτικά, όλες οι περιπτώσεις που θα πρέπει να συμπεριλάβετε φαίνονται στο παρακάτω σχήμα:



Αρχεία εισόδου

Για να ελέγξετε την ορθότητα, αλλά και για να χρονομετρήσετε την υλοποίηση σας, θα χρειαστεί να χρησιμοποιήσετε κατευθυνόμενα γραφήματα. Μπορείτε να βρείτε μια μεγάλη συλλογή τέτοιων γραφημάτων που αναπαρίστανται με μητρώα γειτνίασης στην ιστοσελίδα https://sparse.tamu.edu. Στο πεδίο "Keyword Search" δώστε την λέξη-κλειδί "directed" (χωρίς τα εισαγωγικά). Χρησιμοποιήστε μόνο μητρώα τα οποία στην στήλη "Kind" περιέχουν το "Directed Graph". Προσέξτε επίσης τα μητρώα που θα χρησιμοποιήσετε να είναι τετραγωνικά.

Προτείνεται να χρησιμοποιήσετε την μορφή "Matrix Market". Η μορφή των αρχείων αυτών είναι ως εξής: Στην πρώτη γραμμή του αρχείου περιέχεται το πλήθος των γραμμών του μητρώου, το πλήθος των στηλών του μητρώου και το πλήθος των ακμών. Σε κάθε επόμενη γραμμή αναφέρεται μια σύνδεση μεταξύ δύο κόμβων. Για παράδειγμα, το μητρώο "Tina_AskCal" περιέχει:

```
11 11 29
3 1
10 1
1 2
3 2
4 2
5 2
6 2
8 2
10 2
10 3
11 3
2 4
3 4
7 4
8 4
10 5
1 6
3 6
5 6
11 6
9 7
2 8
3 8
7 8
9 8
11 8
7 9
8 11
9 11
```

Επομένως αναπαριστά ένα γράφημα με 11 κόμβους, υπάρχουν συνολικά 29 ακμές, οι οποίες είναι από τον κόμβο 3 προς τον 1, από τον 10 προς τον 1, από τον 1 προς τον 2, από τον 3 προς τον 2, κλπ. Επομένως το μητρώο γειτνίασης θα είναι:

Κόμβος	1	2	3	4	5	6	7	8	9	10	11
1		1				1					
2				1				1			
3	1	1		1		1		1			
4		1									
5		1				1					
6		1									
7				1				1	1		
8		1		1							1
9							1	1			1
10	1	1	1		1						
11			1			1		1			

Παρατηρήστε πως η αρίθμηση των κόμβων ξεκινάει από το 1 (ενώ οι πίνακες στην C από το 0). Αν υπάρχει σύνδεση μεταξύ δύο κόμβων τότε το αντίστοιχο στοιχείο του μητρώου είναι 1, διαφορετικά είναι 0. Παρατηρήστε πως επειδή το γράφημα είναι κατευθυνόμενο, το μητρώο δεν είναι συμμετρικό (π.χ. υπάρχει η ακμή από τον κόμβο 1 στον 6, αλλά όχι από τον κόμβο 6 στον 1).

Μπορείτε να ξεκινήσετε με μικρά μητρώα για να ελέγξετε την ορθότητα του προγράμματος σας, ωστόσο **για την χρονομέτρηση θα πρέπει να χρησιμοποιήσετε όσο το δυνατόν μεγαλύτερα μητρώα**.

Διαδικαστικά

Η εργασία θα πρέπει να γίνει σε ομάδες των 3 ή 4 ατόμων **ακριβώς**. Δεν θα γίνει δεκτή ομάδα με λιγότερα ή περισσότερα άτομα **για κανέναν λόγο**. Αν δεν συμπληρώσετε ομάδα 3 ή 4 ατόμων, τότε θα προστεθούν άτομα στην ομάδα σας από εμάς.

Η δήλωση των ομάδων θα γίνει στην ηλεκτρονική πλατφόρμα "Open eClass" του Πανεπιστημίου Πατρών (http://eclass.upatras.gr). **Για τον σκοπό αυτό θα πρέπει όλοι οι φοιτητές που επιθυμούν να παραδώσουν εργασία να εγγραφούν πρώτα στην παραπάνω πλατφόρμα.** Στην συνέχεια, ένα άτομο από κάθε ομάδα θα αναλάβει να δηλώσει την ομάδα του μέχρι την **Κυριακή, 05/04/2020 και ώρα 23:59:59**. Το άτομο αυτό θα είναι επίσης υπεύθυνο για όλη την επικοινωνία της ομάδας μαζί μας, καθ' όλη την διάρκεια του εξαμήνου και μέχρι την παράδοση της άσκησης. Η ομάδα θα δηλωθεί μέσω e-mail στην διεύθυνση thanasis@ceid.upatras.gr. Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, το e-mail θα πρέπει να έχει τον εξής τίτλο:

```
[ParPro19-20] Δήλωση ομάδας
```

Το περιεχόμενο του e-mail θα πρέπει να είναι ο Α.Μ. και το ονοματεπώνυμο του φοιτητή που κάνει την δήλωση της ομάδας, καθώς επίσης το πλήθος των ατόμων της ομάδας. Στην συνέχεια θα αναλάβουμε να φτιάξουμε μια ομάδα στο "Open eClass" και θα σας ενημερώσουμε για τον αριθμό της ομάδας σας. Η διαδικασία αυτή μπορεί να χρειαστεί αρκετές ημέρες. Πρώτα θα συγκεντρωθούν όλες οι αιτήσεις για δημιουργία ομάδας και μετά θα δημιουργηθούν οι ομάδες.

Σε περίπτωση που χρειαστεί επιπλέον επικοινωνία μαζί μας μέσω e-mail, αυτή θα πρέπει να γίνει είτε με τον κ. Ιωάννη Θανάση (thanasis@ceid.upatras.gr) είτε με τον κ. Ιωάννη Βενέτη (venetis@ceid.upatras.gr). **Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, ο τίτλος κάθε e-mail θα πρέπει να ξεκινάει με [ParPro19-20].**

Παραδοτέα

Τα παραδοτέα για την εργασία σας είναι μια γραπτή αναφορά (**μην ξεχάσετε να συμπεριλάβετε στο εξώφυλλο της αναφοράς τα ονόματα και τους ΑΜ όλων των μελών της ομάδας**) και ο κώδικας της άσκησης που θα αναπτύξετε. **Η προθεσμία παράδοσης της εργασίας ορίζεται η Κυριακή 31/05/2020 και ώρα 23:59:59. Η εργασία θα πρέπει να παραδοθεί αποκλειστικά μέσω της ηλεκτρονικής πλατφόρμας "Open eClass" (εργασίες που θα αποσταλούν μέσω e-mail δεν θα βαθμολογηθούν) και η πλατφόρμα θα κλειδώσει αυτόματα την δυνατότητα υποβολής εργασιών μετά την λήξη της προθεσμίας. Οργανώστε**

λοιπόν σωστά τον χρόνο σας. Μετά την είσοδο σας στο σύστημα θα πρέπει να μεταβείτε στο μάθημα "Παράλληλη Επεξεργασία" και στο μενού αριστερά να μεταβείτε στο "Εργασίες". **Κάθε ομάδα θα παραδώσει μια φορά μόνο την εργασία (όχι κάθε φοιτητής ξεχωριστά).**

Στην αναφορά **δεν** θα πρέπει να περιλαμβάνεται επεξήγηση του ακολουθιακού αλγόριθμου που σας δόθηκε. Επικεντρωθείτε στην επεξήγηση της παραλληλοποίησης που κάνατε, στις μετρήσεις σας και στα διαγράμματα που θα προσθέσετε. Σχολιάστε τις διαφορές από την χρήση βελτιστοποιήσεων στον χρόνο εκτέλεσης της εφαρμογής και (κυρίως) στην χρονοβελτίωση. Γενικότερα, δώστε ιδιαίτερο βάρος στην αναφορά σε αυτά που κάνατε εσείς.

Ο βαθμός της εργασίας αποτελεί το 30% της τελικής βαθμολογίας. Το υπόλοιπο 70% προκύπτει από την τελική εξέταση. Για να περάσει κάποιος φοιτητής το μάθημα δεν είναι απαραίτητη η παράδοση της εργασίας. **Στην περίπτωση αυτή ωστόσο, θεωρείται πως η εργασία έχει πάρει βαθμό 0 (μηδέν). Ο τελικός βαθμός τότε προκύπτει μόνο από το 70% της τελικής εξέτασης και θα πρέπει να είναι προβιβάσιμος (≥ 5).**

Η εργασία παραδίδεται ΜΟΝΟ κατά την εξεταστική Ιουνίου (υποβολές που θα γίνουν κατά την διάρκεια της εξεταστικής Σεπτεμβρίου ή της άτυπης εξεταστικής Φεβρουαρίου δεν θα γίνουν δεκτές).

Ο βαθμός της εργασίας διατηρείται μέχρι και την άτυπη εξεταστική Φεβρουαρίου 2021. Αν κάποιος φοιτητής δεν περάσει το μάθημα μέχρι τότε θα πρέπει να παρακολουθήσει εξ αρχής το μάθημα και να ανταποκριθεί στις υποχρεώσεις του μαθήματος για το ακαδημαϊκό έτος που θα το παρακολουθήσει ξανά.

Παράρτημα Α

Για την εγκατάσταση των προγραμμάτων που θα χρειαστείτε στα πλαίσια της άσκησης, θα πρέπει να ακολουθήσετε τις παρακάτω οδηγίες.

1) Εγκατάσταση Linux

   Στα πλαίσια της εργασίας συστήνεται να εγκαταστήσετε την έκδοση 18.04.2 LTS της διανομής Ubuntu (http://www.ubuntu.com). Οι οδηγίες που ακολουθούν υποθέτουν την εγκατάσταση της συγκεκριμένης διανομής. Φυσικά μπορείτε να εγκαταστήσετε όποια άλλη διανομή θέλετε, όπως για παράδειγμα Fedora (http://fedoraproject.org) ή openSUSE (http://www.opensuse.org). Σε κάθε περίπτωση επιλέξτε την τελευταία έκδοση που είναι διαθέσιμη για κάθε διανομή. Αν εγκαταστήσετε άλλη διανομή εκτός της Ubuntu 18.04.2 LTS θα πρέπει να προσαρμόσετε κατάλληλα της οδηγίες για την εγκατάσταση των επιπλέον πακέτων που απαιτούνται.

   Οι διανομές συνήθως είναι διαθέσιμες ως αρχεία τύπου ISO. Αν εγκαταστήσετε την διανομή απευθείας στον σκληρό δίσκο του υπολογιστή σας (βλέπε παρακάτω) θα πρέπει να γράψετε το αρχείο ISO σε ένα CD, DVD ή USB stick. Αν εγκαταστήσετε την διανομή σε ένα Virtual Machine (βλέπε παρακάτω) μπορείτε να χρησιμοποιήσετε απευθείας το αρχείο ISO. Η εγκατάσταση της διανομής μπορεί να γίνει με δύο τρόπους:

   a. Απευθείας στον σκληρό δίσκο του μηχανήματος σας

      Αν ακολουθήσετε την μέθοδο αυτή θα πρέπει να φτιάξετε ένα ξεχωριστό partition στον σκληρό σας δίσκο. Όταν ξεκινήσετε την εγκατάσταση της διανομής θα πρέπει να επιλέξετε το συγκεκριμένο partition. Ακολουθήστε αυτή τη μέθοδο αν θέλετε να κρατήσετε την διανομή που θα εγκαταστήσετε και για αργότερα.

   b. Μέσω Virtual Machine (VM)

      "Virtual Machine" είναι στην πραγματικότητα οποιοδήποτε πρόγραμμα που μπορούμε να εγκαταστήσουμε σε ένα λειτουργικό σύστημα που διαθέτουμε και προσομοιώνει έναν υπολογιστή. Αν για παράδειγμα διαθέτουμε Windows, μπορούμε να εγκαταστήσουμε ένα Virtual Machine και στην συνέχεια να εγκαταστήσουμε στον προσομοιούμενο υπολογιστή ένα άλλο λειτουργικό σύστημα, όπως για παράδειγμα Linux.

      Αν ακολουθήσετε αυτή τη μέθοδο θα πρέπει κατ' αρχάς να κατεβάσετε και να εγκαταστήσετε στο λειτουργικό σύστημα που διαθέτετε ένα Virtual Machine. Υπάρχουν αρκετά ελεύθερα προγράμματα διαθέσιμα για αυτό, όπως το Virtual Box (https://www.virtualbox.org), **το οποίο και συστήνουμε για την εργασία**, το VMware Player (http://www.vmware.com/products/player/overview.html) και το QEMU (http://wiki.qemu.org). Επειδή θα χρειαστεί να τρέξουμε παράλληλα προγράμματα, διαβάστε τις οδηγίες εγκατάστασης και δημιουργίας Virtual Machine, ώστε η Virtual Machine στην οποία θα εγκαταστήσετε την διανομή Linux που θα επιλέξετε να υποστηρίζει πολλαπλούς πυρήνες. Επίσης, δηλώστε έναν αρκετά μεγάλο σκληρό δίσκο (τουλάχιστον 10GB). Τέλος, εγκαταστήστε την διανομή Linux που επιλέξατε στην Virtual Machine που φτιάξατε.

2) Εγκατάσταση Guest Additions

Αν έχετε εγκαταστήσει VirtualBox τότε μια σημαντική προσθήκη που μπορείτε να κάνετε είναι τα “Guest Additions”. Τα εργαλεία αυτά προσφέρουν επιπλέον δυνατότητες στον εικονικό υπολογιστή, με κυριότερες την καλύτερη ανάλυση οθόνης και το Clipboard για την αντιγραφή δεδομένων και αρχείων μεταξύ του πραγματικού και του εικονικού υπολογιστή. **Η εγκατάσταση των “Guest Additions” είναι εντελώς προαιρετική. Μπορείτε να ολοκληρώσετε την εργασία και χωρίς αυτά.** Αν θέλετε να τα εγκαταστήσετε ακολουθήστε τα παρακάτω βήματα:

a. Πριν εκκινήσετε την VM εγκαταστήστε στο VirtualBox το Extension Pack: https://www.virtualbox.org/manual/ch01.html#intro-installing
b. Εκκινήστε την VM και εγκαταστήστε τα προαπαιτούμενα πακέτα στο λειτουργικό σύστημα του εικονικού υπολογιστή. Από την γραμμή εντολών εκτελέστε την εντολή:

   ```
   sudo apt-get install dkms
   ```

c. Κάντε επανεκκίνηση του υπολογιστή σας.
d. Στο μενού του Virtual Box επιλέξτε “Devices → Insert Guest Additions CD image…”.
e. Επιτρέψτε την αυτόματη εκτέλεση του προγράμματος εγκατάστασης.
f. Κάντε επανεκκίνηση του υπολογιστή σας.
g. Ανοίξτε το πρόγραμμα Διαχείρισης Αρχείων (File Manager).
h. Κάντε unmount το “Guest Additions CD” κάνοντας κλικ στο σύμβολο “⏏”.