

ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2015/2016

Εισαγωγή

Η ανακάλυψη πως τα νουκλεϊκά οξέα αποτελούνται από ακολουθίες συγκεκριμένων νουκλεοτιδίων, καθώς επίσης πως οι πρωτεΐνες αποτελούνται από ακολουθίες συγκεκριμένων αμινοξέων, αποτέλεσε τεράστια επιτυχία στον τομέα της βιολογίας. Γρήγορα έγινε σαφές πως συγκρίνοντας τις ακολουθίες αυτές μεταξύ διαφορετικών όντων μπορούσαμε να εξάγουμε πληροφορίες για το κατά πόσο τα όντα αυτά συγγενεύουν μεταξύ τους. Όσο περισσότερο μοιάζουν οι συγκεκριμένες ακολουθίες, τόσο πιο πιθανό είναι τα όντα αυτά να έχουν εξελιχθεί από κάποιον κοινό πρόγονο.

Τα 4 βασικά συστατικά στοιχεία ενός νουκλεϊκού οξέος συμβολίζονται με τα γράμματα A, C, G και T (για Adenine, Cytosine, Guanine και Thymine αντίστοιχα). Ένα νουκλεϊκό οξύ αποτελείται λοιπόν από μία σειρά από αυτές τις βάσεις. Ας υποθέσουμε πως έχουμε τις παρακάτω 2 ακολουθίες:

X: C G G G T A T C C A A
Y: C C C T A G G T C C C A

Το ερώτημα είναι αν θα μπορούσε η μια από αυτές να προκύψει από την άλλη. Για παράδειγμα, αν (για τον οποιοδήποτε λόγο) η βάση G στην ακολουθία X είχε αντικατασταθεί από την βάση C στην ακολουθία Y, τότε οι δύο αυτές ακολουθίες θα έμοιαζαν πολύ περισσότερο. Επιπλέον όμως, η ακολουθία Y είναι πιο μεγάλη. Μήπως λοιπόν προστέθηκε και μια βάση ακόμα κατά την διαδικασία της εξέλιξης;

Σκοπός της ευθυγράμμισης τέτοιων ακολουθιών (sequence alignment) είναι να πετύχουμε το καλύτερο δυνατό ταίριασμα της μιας ακολουθίας με την άλλη. Η ευθυγράμμιση πρέπει να ακολουθεί τους παρακάτω κανόνες:

- Όλα τα σύμβολα των δύο ακολουθιών πρέπει να υπάρχουν στην ευθυγράμμιση και μάλιστα με την ίδια σειρά με την οποία εμφανίζονται στις ακολουθίες.
- Κάθε σύμβολο μιας ακολουθίας πρέπει να ευθυγραμμιστεί με ένα μόνο σύμβολο της άλλης ακολουθίας.
- Ένα σύμβολο μπορεί να ευθυγραμμιστεί με ένα κενό ('-').
- Δύο κενά δεν επιτρέπεται να είναι ευθυγραμμισμένα.

Ένα παράδειγμα ευθυγράμμισης των παραπάνω ακολουθιών θα ήταν το εξής:

X: C G G G T A - - T - C C A A
Y: C C C - T A G G T C C C - A

Παρατηρήστε ότι οι παραπάνω κανόνες δεν απαιτούν ένα σύμβολο να είναι οπωσδήποτε ευθυγραμμισμένο με το κενό ή με το ίδιο ακριβώς σύμβολο. Στο παραπάνω παράδειγμα έχουμε χρωμάτισει με πράσινο τις ευθυγραμμίσεις όπου αυτό ισχύει. Με κόκκινο έχουμε χρωμάτισει τις ευθυγραμμίσεις όπου ένα σύμβολο έχει ευθυγραμμιστεί με άλλο σύμβολο.

Όπως γίνεται κατανοητό, υπάρχουν πολλές ευθυγραμμίσεις που θα μπορούσαμε να έχουμε για τις συγκεκριμένες ακολουθίες. Εκτός από την παραπάνω ευθυγράμμιση, 2 επιπλέον παραδείγματα θα μπορούσαν να είναι τα εξής:

X: C G G G T A - - - T C C A A
Y: C C - - C T A G G T C C C A

X: C - G G G T A - - T C C A A
Y: C C - - C T A G G T C C C A

Εύλογα λοιπόν γεννιέται το ερώτημα ποια είναι η καλύτερη δυνατή ευθυγράμμιση δύο ακολουθιών. Η απάντηση δεν είναι εύκολη, καθώς συνήθως οι ακολουθίες αυτές για πραγματικούς ζωντανούς οργανισμούς αποτελούνται από τουλάχιστον δεκάδες χιλιάδες σύμβολα η καθεμία. Η εύρεση της βέλτιστης ακολουθίας «στο χαρτί» είναι κατά συνέπεια αδύνατη. Όπως σωστά φανταστήκατε, η αξιοποίηση της υπολογιστικής ισχύος των σύγχρονων υπολογιστών δίνει την λύση.

Ο Αλγόριθμος Smith-Waterman

Για να λύσουμε το παραπάνω πρόβλημα χρειαζόμαστε φυσικά τον κατάλληλο αλγόριθμο. Ο πλέον χρησιμοποιούμενος αλγόριθμος προτάθηκε το 1981 από τους Temple F. Smith και Michael S. Waterman και είναι γνωστός με το όνομα Smith-Waterman [1].

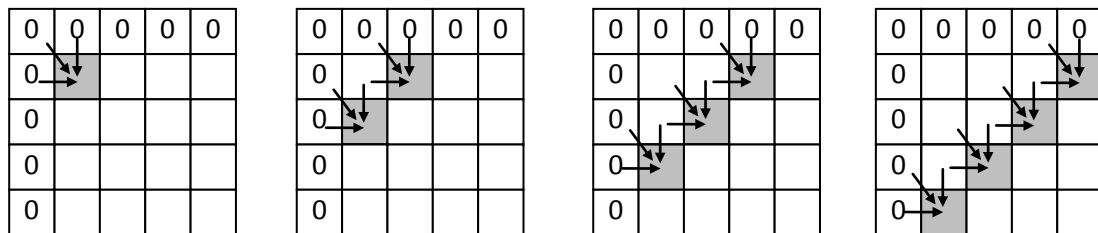
Βασικό στοιχείο στον συγκεκριμένο αλγόριθμο είναι η δημιουργία ενός πίνακα H , γνωστού ως «πίνακα βαθμολόγησης» (scoring matrix). Ουσιαστικά, στον πίνακα αυτό αποθηκεύουμε βαθμολογία για κάθε πιθανή ευθυγράμμιση δύο ακολουθιών. Αν υποθέσουμε πως οι δύο ακολουθίες X , Y έχουν μήκος m και n αντίστοιχα, τότε ο πίνακας H έχει διαστάσεις $(m+1) \times (n+1)$ και κατασκευάζεται βάσει των παρακάτω κανόνων:

- $H(i, 0) = 0, 0 \leq i \leq m$ (Στοιχεία πρώτης στήλης είναι 0)
- $H(0, j) = 0, 0 \leq j \leq n$ (Στοιχεία πρώτης γραμμής είναι 0)
- if $X_i = Y_j$ then $w(X_i, Y_j) = w(Match)$
- if $X_i \neq Y_j$ then $w(X_i, Y_j) = w(Mismatch)$
- $$H(i, j) = \max \begin{cases} 0 & \\ H(i-1, j-1) + w(X_i, Y_j) & Match/Mismatch \\ H(i-1, j) + w(X_i, -) & Deletion \\ H(i, j-1) + w(-, Y_j) & Insertion \end{cases} \begin{cases} 1 \leq i \leq m \\ 1 \leq j \leq n \end{cases}$$

Κατά συνέπεια, **ο πίνακας H κατασκευάζεται αναδρομικά**. Κάθε κελί χρειάζεται τους 3 γείτονες που «προηγούνται» από αυτό για να υπολογιστεί. Η συνάρτηση $w()$ επιστρέφει έναν αριθμό, ο οποίος βαθμολογεί το ταιριασμα δύο συμβόλων στις δύο ακολουθίες. Ας πάρουμε για παράδειγμα τον κλάδο $H(i-1, j-1) + w(X_i, Y_j)$. Ο όρος $H(i-1, j-1)$ αντιστοιχεί στην βαθμολογία της ακολουθίας που έχει ήδη υπολογιστεί. Η συνάρτηση $w()$ θα επιστρέψει έναν αριθμό, ανάλογα με το πως το σύμβολο X_i ταιριάζει με το σύμβολο Y_j . Αν τα σύμβολα είναι ίδια συνήθως επιστρέφεται ένας θετικός αριθμός. Αυτό όμως δεν σημαίνει απαραίτητα ότι ο αριθμός αυτός είναι ίδιος για τα 4 βασικά συστατικά στοιχεία ενός νουκλεϊκού οξέος. Έτσι, αν και οι δύο ακολουθίες έχουν το σύμβολο C στις συγκεκριμένες θέσεις, τότε μπορεί να επιστρέφεται π.χ. ο αριθμός 12, ενώ αν έχουν το G τότε μπορεί να επιστρέφεται ο αριθμός 5. Αυτό συμβαίνει γιατί η ταύτιση για συγκεκριμένα σύμβολα μπορεί να θεωρηθεί πιο σημαντική από την ταύτιση για άλλα σύμβολα. Αν οι δύο ακολουθίες δεν ταιριάζουν στην

συγκεκριμένη θέση, τότε συνήθως επιστρέφεται ένας αρνητικός αριθμός, μειώνοντας έτσι την συνολική βαθμολογία. Αντίστοιχα, για τους δύο τελευταίους κλάδους βαθμολογείται η ταύτιση των συμβολοσειρών όταν προσθέσουμε το κενό σύμβολο σε μία από τις δύο ακολουθίες. Το ποιούς ακριβώς αριθμούς θα επιστρέφει η συνάρτηση $w()$ δεν το καθορίζει ο αλγόριθμος Smith-Waterman. Είναι στην ευχέρεια του καθένα να τους καθορίσει ανάλογα με τις ανάγκες της εφαρμογής.

Ο αλγόριθμος αυτός ακολουθεί το μοντέλο του «κυματομέτωπου» (wave front). Αυτό αποτελεί πολύ σημαντικό στοιχείο για την παραλληλοποίηση του αλγόριθμου. Θα πρέπει να βεβαιώνετε πως κατά τους υπολογισμούς κάθε κελιού έχουν υπολογιστεί ήδη τα κελιά που χρειάζονται. Με λίγα λόγια, στον συγκεκριμένο αλγόριθμο υπάρχουν «εξαρτήσεις δεδομένων». Σχηματικά, ο αλγόριθμος θα ακολουθούσε τα εξής βήματα (για έναν μικρό πίνακα):



Σε κάθε βήμα εκτέλεσης μπορούν να υπολογιστούν τα στοιχεία του πίνακα H που έχουν χρωματιστεί γκρι. Βεβαίως οι υπολογισμοί θα συνεχίζονταν για 3 ακόμα βήματα, μέχρι και το τελευταίο στοιχείο του πίνακα.

Το τελευταίο βήμα του αλγόριθμου είναι να βρεθεί η καλύτερη ακολουθία. Προφανώς, αυτή θα έχει την μεγαλύτερη βαθμολογία. Κατά συνέπεια, θα πρέπει να αναζητηθεί η μέγιστη βαθμολογία στον πίνακα H και στην συνέχεια να ακολουθήσουμε ανάποδα την διαδρομή από την οποία προέκυψε η μέγιστη βαθμολογία (back tracking). Με αυτόν τον τρόπο ολοκληρώνεται η εύρεση της βέλτιστης ευθυγράμμισης.

Αναφορές

- [1] Temple F. Smith and Michael S. Waterman. "Identification of Common Molecular Subsequences". Journal of Molecular Biology 147: 195–197, 1981.

Ζητούμενα της άσκησης

Στα πλαίσια της άσκησης θα σας δοθεί μια απλοποιημένη, σειριακή έκδοση του αλγόριθμου Smith-Waterman γραμμένη σε C++. Για την σειριακή έκδοση της εφαρμογής που θα σας δοθεί ο σκοπός σας θα είναι:

- 1) Να αναλύσετε την συμπεριφορά της εφαρμογής, εντοπίζοντας τις συναρτήσεις του κώδικα που παίρνουν τον περισσότερο χρόνο για να εκτελεστούν.
- 2) Να παραλληλοποιήσετε το πρόγραμμα και να δημιουργήσετε δύο διαφορετικές παράλληλες εκδόσεις:
 - a. Μια με χρήση του προτύπου OpenMP.
 - b. Μια που θα χρησιμοποιεί την έκδοση με OpenMP και επιπλέον θα κάνει διανυσματοποίηση (vectorization).

Θα πρέπει **κατ' αρχάς** να αναλύσετε την απόδοση του σειριακού κώδικα και να δείτε σε ποιες συναρτήσεις καταναλώνεται ο περισσότερος χρόνος εκτέλεσης. Αυτές οι συναρτήσεις είναι προφανώς οι πρώτες υποψήφιες για παραλληλοποίηση. Για τον σκοπό αυτό θα χρησιμοποιήσετε το **εργαλείο Scalasca**, ένα εργαλείο ανάλυσης της απόδοσης εφαρμογών. Στο Παράρτημα Α της εκφώνησης παρατίθενται οδηγίες για την εγκατάσταση των εργαλείων που απαιτούνται στα πλαίσια της εργασίας. Αναλύστε την σειριακή εφαρμογή για τον χρόνο εκτέλεσης και προσπαθείστε να καταλάβετε γιατί οι συναρτήσεις αυτές δεν έχουν καλή απόδοση (π.χ., σύνολο αναφορών στην μνήμη, cache misses ως ποσοστό του συνόλου αναφορών στην μνήμη, TLB misses, αλλά πολλές περισσότερες πράξεις σε σχέση με άλλες συναρτήσεις της εφαρμογής, κλπ) με χρήση των Performance Counters που προσφέρουν οι σύγχρονοι επεξεργαστές (δείτε το Παράρτημα Β για λεπτομέρειες). **Μην μείνετε απαραίτητα στις παραπάνω μετρικές. Μελετήστε τον κώδικα και αποφασίστε ποιες μετρικές είναι οι κατάλληλες που πρέπει να χρησιμοποιήσετε. Είναι μέρος της διαδικασίας αξιολόγησης σας. Συμπεριλάβετε όλα τα συμπεράσματα σας στην αναφορά σας. Συμπεριλάβετε εικόνες (screenshots) από τις αναλύσεις που θα κάνετε με χρήση του εργαλείου Scalasca.**

Στην συνέχεια θα πρέπει να παραλληλοποιήσετε τον κώδικα σας πρώτα με χρήση του προτύπου OpenMP. Ξεκινήστε από τις συναρτήσεις που εντοπίσατε νωρίτερα. Επιπλέον όμως είσατε ελεύθεροι να παραλληλοποιήσετε τον κώδικα με όποιον τρόπο θέλετε. Αν θεωρήσετε πως μια οποιαδήποτε αναδιοργάνωση του κώδικα μπορεί να οδηγήσει σε καλύτερη παραλληλοποίηση και απόδοση είστε ελεύθεροι να το κάνετε. Ωστόσο, θεωρείται αυτονόητο πως η υλοποίηση θα πρέπει να είναι σωστή (να δίνει **πάντα** τα ίδια αποτελέσματα με την ακολουθιακή έκδοση του προγράμματος) και πως **στην αναφορά που θα παραδώσετε θα πρέπει να αιτιολογήσετε γιατί παραλληλοποιήσατε με τον συγκεκριμένο τρόπο την εφαρμογή.**

Τέλος, χρησιμοποιώντας ως βάση την έκδοση του προγράμματος με χρήση του OpenMP, θα δημιουργήσετε μια ακόμα παράλληλη έκδοση του προγράμματος, προσθέτοντας **επιπλέον οδηγίες διανυσματοποίησης (vectorization directives)**. Το Παράρτημα Γ έχει αναφορές που μπορείτε να χρησιμοποιήσετε για να δείτε πως αξιοποιούνται οι εντολές αυτές στα πλαίσια ενός προγράμματος. **Συμπεριλάβετε στην αναφορά τις παρατηρήσεις σας για τις αλλαγές στους χρόνους εκτέλεσης από την εισαγωγή των οδηγιών διανυσματοποίησης.**

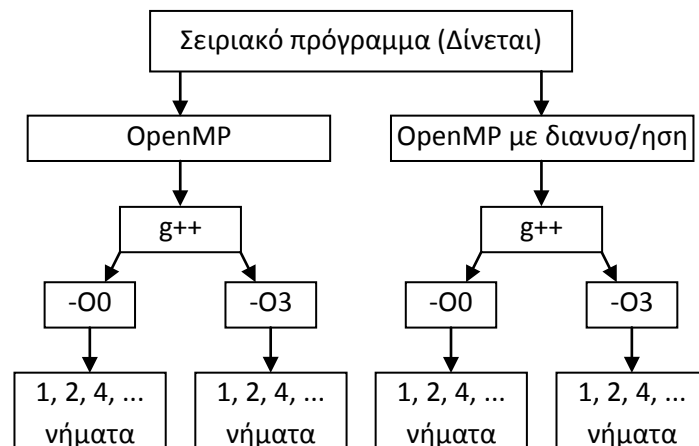
Οδηγίες για την μεταγλώττιση και εκτέλεση του σειριακού κώδικα που σας δίνεται και του παράλληλου κώδικα που θα φτιάξετε δίνονται στο Παράρτημα Β της εκφώνησης. Εκεί επίσης

αναφέρεται για ποιες παραμέτρους θα πρέπει να τρέξετε την κάθε έκδοση του προγράμματος και ποια αποτελέσματα θα συμπεριλάβετε στην αναφορά σας.

Συγκεντρωτικά, θα πρέπει στην αναφορά σας να παραθέσετε μετρήσεις και να σχολιάσετε τις παρακάτω περιπτώσεις:

- 1) Πρόγραμμα με OpenMP
- 2) Πρόγραμμα με OpenMP και οδηγίες διανυσματοποίησης.
- 3) Μεταγλώττιση και εκτέλεση κάθε προγράμματος χωρίς βελτιστοποιήσεις (-O0) και με μέγιστες βελτιστοποιήσεις (-O3).
- 4) 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 του βασικού αλγόριθμου (υπάρχει έτοιμο στον κώδικα που σας δίνεται) και διαγράμματα της χρονοβελτίωσης. Συγκεντρωτικά, όλες οι περιπτώσεις που θα πρέπει να συμπεριλάβετε φαίνονται στο παρακάτω σχήμα:



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 **Τετάρτη, 06/04/2016 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 venetis@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:

[ParPro15-16] 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. Στην συνέχεια θα αναλάβουμε να φτιάξουμε μια ομάδα στο "Open eClass" και θα σας ενημερώσουμε για τον αριθμό της ομάδας σας. Ο φοιτητής που δήλωσε την ομάδα θα πρέπει μετά να εισέλθει στην πλατφόρμα, να μεταβεί στο μάθημα "Παράλληλη Επεξεργασία", στο μενού αριστερά να μεταβεί στο "Ομάδες Χρηστών", να επιλέξει την ομάδα του και τέλος να επιλέξει "Διόρθωση της ομάδας χρηστών". Εκεί θα ορίσει το συνολικό πλήθος των ατόμων που αποτελούν την ομάδα (συμπεριλαμβανομένου του εαυτού του υπενθυμίζεται ότι θα πρέπει να είναι 3 ή 4) και να προσθέσει τα ονόματα των μελών της ομάδας.

Σε περίπτωση που χρειαστεί επιπλέον επικοινωνία μαζί μας μέσω e-mail, αυτή θα πρέπει να γίνει είτε με τον κ. Νικόλαο Νικολουτσάκο (nikoloutsas@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 [ParPro15-16].**

Παραδοτέα

Τα παραδοτέα για την εργασία σας είναι μια γραπτή αναφορά και ο κώδικας της άσκησης που θα αναπτύξετε. **Η προθεσμία παράδοσης της εργασίας ορίζεται η Τρίτη 31/05/2016 και ώρα 23:59:59. Η εργασία θα πρέπει να παραδωθεί 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 δώθηκε. 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).**

Ο βαθμός της εργασίας διατηρείται μέχρι και την άτυπη εξεταστική Φεβρουαρίου 2017. 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. Σε περίπτωση που κάτι δεν πάει καλά (ιδιαίτερα για τα πακέτα που θα πρέπει να εγκαταστήσετε μόνοι σας), διαβάστε καλά τα μηνύματα λάθους. Είναι πολύ πιθανό να πρέπει να εγκαταστήσετε κάποιο επιπλέον πακέτο στην διανομή Linux, για να μπορέσουν να μεταγλωττιστούν και να εγκατασταθούν τα υπόλοιπα πακέτα.

1) Εγκατάσταση Linux

Στα πλαίσια της εργασίας συστήνεται να εγκαταστήσετε την έκδοση 15.10 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 15.10 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 επιπλέον πακέτων της διανομής Linux

Για να μπορέσετε να μεταγλωττίσετε τα προγράμματα και τις βιβλιοθήκες που απαιτούνται για την εργασία θα χρειαστεί να εγκαταστήσετε ορισμένα επιπλέον πακέτα λογισμικού. **Υπενθυμίζουμε πως οι οδηγίες εδώ αφορούν την διανομή Ubuntu 15.10. Αν έχετε εγκαταστήσει άλλη διανομή θα πρέπει να βρείτε τα αντίστοιχα πακέτα και να προσαρμόσετε τις εντολές εγκατάστασης.**

- a. Εγκαταστήστε όλες τις επικαιροποιημένες εκδόσεις των πακέτων του συστήματος εκτελώντας από την γραμμή εντολών (command line) τις εντολές:

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get dist-upgrade
```

- b. Κάντε επανεκκίνηση του υπολογιστή σας. **Αν έχετε εγκαταστήσει το λειτουργικό σύστημα σε VM, τότε όποτε αναφέρουμε πως πρέπει να γίνεται επανεκκίνηση του υπολογιστή θα εννοείται πως η επανεκκίνηση αφορά το VM.**
- c. Εγκαταστήστε τα παρακάτω πακέτα δίνοντας την εντολή:

```
sudo apt-get install bison flex gfortran g++ gcc-5-plugin-dev  
libqt4-dev qt4-qmake zlib1g-dev freeglut3-dev binutils-dev
```

- d. Κάντε επανεκκίνηση του υπολογιστή σας.

3) 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 "Δ".

4) Εγκατάσταση PAPI-5.4.3

Η βιβλιοθήκη PAPI δίνει την δυνατότητα πρόσβασης στους Performance Counters που διαθέτουν οι σύγχρονοι επεξεργαστές. Οι Performance Counters είναι ειδικοί καταχωρητές που μετράνε ένα πλήθος χαρακτηριστικών μιας εφαρμογής που εκτελεί ο χρήστης, όπως για παράδειγμα L1 cache misses ή hits, L2 cache misses ή hits, TLB misses, κλπ. Το τι θα καταμετρηθεί μπορεί να το καθορίσει ο χρήστης. Έτσι δίνεται η δυνατότητα να αναλυθεί με ακρίβεια η απόδοση μιας εφαρμογής και να καθοριστεί σε ποια σημεία της και γιατί δεν έχει την καλύτερη δυνατή απόδοση.

Αν και η βιβλιοθήκη PAPI μπορεί να χρησιμοποιηθεί απευθείας στον πηγαίο κώδικα μιας εφαρμογής με την χρήση του API (Application Programming Interface) που προσφέρει, στα πλαίσια της εργασίας θα χρησιμοποιηθεί μόνο σε συνδυασμό με την αυτοματοποιημένη δυνατότητα αξιοποίησης της από το Scalasca (βλέπε παρακάτω).

Αν και πολλές διανομές Linux προσφέρουν ως πακέτο την βιβλιοθήκη αυτή, συνιστάται να την εγκαταστήσετε μόνοι σας από τον πηγαίο της κώδικα, ακολουθώντας τις οδηγίες που παρατίθενται αμέσως μετά. Ο λόγος είναι πως οι περισσότερες διανομές παρέχουν παλιότερες εκδόσεις της βιβλιοθήκης και είναι πολύ πιθανό νεότεροι επεξεργαστές να μην αναγνωρίζονται από την βιβλιοθήκη και να μην μπορείτε να πάρετε μετρήσεις.

Για να εγκαταστήσετε μόνοι σας την βιβλιοθήκη PAPI:

- a. Μεταβείτε στην ιστοσελίδα <http://icl.cs.utk.edu/papi> και στο μενού αριστερά μεταβείτε στο "Software". Κατεβάστε την τρέχουσα έκδοση της βιβλιοθήκης (5.4.3).
- b. Πριν την εγκατάσταση **αλλά και πριν από κάθε μέτρηση με χρήση της βιβλιοθήκης** θα πρέπει να απενεργοποιείτε το NMI watchdog:

```
sudo su
echo 0 > /proc/sys/kernel/nmi_watchdog
exit
```

Κάθε φορά που κάνετε επανεκκίνηση του Linux που έχετε εγκαταστήσει, θα πρέπει να εκτελείτε την παραπάνω εντολή ως διαχειριστής (root).

- c. Αποσυμπίεστε τον πηγαίο κώδικα της βιβλιοθήκης και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd papi-5.4.3.tar.gz | tar -x
cd papi-5.4.3/src
```

- d. **Υποθέτουμε για την συνέχεια πως θα εγκαταστήσετε όλα τα πακέτα στον κατάλογο "opt" του Home Directory του λογαριασμού σας στο Linux που έχετε εγκαταστήσει. Αν αλλάξετε τους καταλόγους εγκατάστασης, θα πρέπει να προσαρμόσετε αντίστοιχα όλες τις εντολές από αυτό το σημείο και πέρα.**

Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης, μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
./configure --prefix=${HOME}/opt/papi-5.4.3
make
make install
```

Για να δείτε ποιες μετρήσεις με χρήση των Performance Counters είναι διαθέσιμες στο σύστημα σας, μπορείτε να εκτελέσετε την εντολή:

```
${HOME}/opt/papi-5.4.3/bin/papi_avail
```

5) Εγκατάσταση OPARI2-1.1.4

Άλλη μια βιβλιοθήκη που χρειάζεται για την εγκατάσταση του Scalasca. Αυτή δεν δίνεται σαν πακέτο από τις διανομές Linux και θα πρέπει να εγκατασταθεί.

- a. Μεταβείτε στην ιστοσελίδα <http://www.vi-hps.org/projects/score-p> και κοντά στο τέλος της σελίδας ακολουθήστε τον σύνδεσμο για να κατεβάστε την τρέχουσα έκδοση της βιβλιοθήκης (1.1.4). **Στην ίδια ιστοσελίδα μπορείτε να βρείτε και τα εγχειρίδια χρήσης των αντίστοιχων βιβλιοθηκών που εγκαθιστούμε.**
- b. Αποσυμπιέστε τον πηγαίο κώδικα της βιβλιοθήκης και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd opari2-1.1.4.tar.gz | tar -x
cd opari2-1.1.4
```

- c. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης, μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
./configure --prefix=${HOME}/opt/opari2-1.1.4
make
make install
```

6) Εγκατάσταση CUBE-4.3.3

Το πρόγραμμα αυτό χρησιμοποιείται από το Scalasca για να εμφανίσει τα αποτελέσματα των μετρήσεων. Δεν δίνεται σαν πακέτο από τις διανομές Linux και θα πρέπει να εγκατασταθεί.

- a. Μεταβείτε στην ιστοσελίδα <http://www.scalasca.org> και στο μενού που εμφανίζεται στην αρχή της σελίδας επιλέξτε το “Software”. Από το μενού στα αριστερά επιλέξτε “Cube 4.x”. Κατεβάστε την τρέχουσα έκδοση της εφαρμογής (4.3.3).
- b. Βεβαιωθείτε πως έχετε εγκαταστήσει στην διανομή Linux που επιλέξατε το πακέτο που περιέχει το πρόγραμμα “qmake” (ή “qmake-qt4” όπως το ονομάζουν μερικές διανομές). **Αν έχετε εγκαταστήσει το Ubuntu 15.10 τότε αυτό έχει ήδη γίνει στο βήμα 2 νωρίτερα.** Διαφορετικά, προσαρμόστε την αντίστοιχη παράμετρο στην μεθεπόμενη οδηγία ανάλογα με το ακριβές όνομα και το που είναι εγκατεστημένο το συγκεκριμένο πρόγραμμα.
- c. Αποσυμπιέστε τον πηγαίο κώδικα της εφαρμογής και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd cube-4.3.3.tar.gz | tar -x
cd cube-4.3.3
```

- d. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης **(όλα σε μια γραμμή!)**:

```
./configure --prefix=${HOME}/opt/cube-4.3.3
--with-qt=/usr/bin/qmake-qt4
```

- e. Μεταγλωττίστε και εγκαταστήστε το πρόγραμμα:

```
make
make install
```

- f. Προσθέστε στην μεταβλητή περιβάλλοντος PATH τον κατάλογο εγκατάστασης:

```
export PATH=${HOME}/opt/cube-4.3.3/bin:${PATH}
```

7) Εγκατάσταση SCORE-P-1.4.2

Άλλη μια βιβλιοθήκη που χρειάζεται για την εγκατάσταση του Scalasca. Αυτή δεν δίνεται σαν πακέτο από τις διανομές Linux και θα πρέπει να εγκατασταθεί.

- a. Μεταβείτε στην ιστοσελίδα <http://www.vi-hps.org/projects/score-p> και κοντά στο τέλος της σελίδας ακολουθήστε τον σύνδεσμο για να κατεβάσετε την τρέχουσα έκδοση της βιβλιοθήκης (1.4.2).
- b. Αποσυμπιέστε τον πηγαίο κώδικα της βιβλιοθήκης και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd scorep-1.4.2.tar.gz | tar -x
cd scorep-1.4.2
```

- c. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης (όλα σε μια γραμμή!):

```
./configure --prefix=${HOME}/opt/scorep-1.4.2
--with-opari2=${HOME}/opt/opari2-1.1.4
--with-cube=${HOME}/opt/cube-4.3.3
--with-papi-header=${HOME}/opt/papi-5.4.3/include
--with-papi-lib=${HOME}/opt/papi-5.4.3/lib
--without-mpi --disable-gcc-plugin
```

- d. Μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
make
make install
```

- e. Προσθέστε στην μεταβλητή περιβάλλοντος PATH τον κατάλογο εγκατάστασης:

```
export PATH=${HOME}/opt/scorep-1.4.2/bin:${PATH}
```

8) Εγκατάσταση Scalasca-2.2.2

Φτάνουμε έτσι στην εγκατάσταση του κυρίως εργαλείου που θα χρειαστούμε στα πλαίσια της εργασίας, το Scalasca-2.2.2. Εκτός από το ίδιο το πρόγραμμα, μπορείτε να κατεβάσετε από το site που δίνεται παρακάτω και το “User’s Guide”, το οποίο περιέχει πιο αναλυτικά τον τρόπο χρήσης του εργαλείου.

- a. Μεταβείτε στην ιστοσελίδα <http://www.scalasca.org> και στο μενού που εμφανίζεται στην αρχή της σελίδας επιλέξτε το “Software”. Κατεβάστε την τρέχουσα έκδοση της εφαρμογής (2.2.2). Στο μενού που εμφανίζεται οριζόντια μπορείτε να μεταβείτε στο “Documentation” για να κατεβάσετε και το User’s Guide.
- b. Αποσυμπιέστε τον πηγαίο κώδικα της εφαρμογής και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd scalasca-2.2.2.tar.gz | tar -x
cd scalasca-2.2.2
```

- c. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης (όλα σε μια γραμμή!):

```
./configure --prefix=${HOME}/opt/scalasca-2.2.2
--with-opari2=${HOME}/opt/opari2-1.1.4
--with-cube=${HOME}/opt/cube-4.3.3
--with-papi=${HOME}/opt/papi-5.4.3
--with-qmake=/usr/bin/qmake-qt4 --without-mpi
```

d. Μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
make  
make install
```

e. Προσθέστε στην μεταβλητή περιβάλλοντος PATH τον κατάλογο εγκατάστασης:

```
export PATH=${HOME}/opt/scalasca-2.2.2/bin:${PATH}
```

9) Αρχεία/Κατάλογοι που δεν χρειάζονται μετά την εγκατάσταση

Αφού εγκαταστήσετε τα παραπάνω εργαλεία μπορείτε να σβήσετε τους καταλόγους που δημιουργήθηκαν μετά την αποσυμπίεση των αρχείων που κατεβάσατε.

ΠΡΟΣΟΧΗ: Μην σβήσετε τους καταλόγους στους οποίους εγκαταστήσατε τα εργαλεία (\$HOME/opt).

Παράρτημα Β

- 1) Για να μεταγλωττίσετε το πρόγραμμα που σας δίνεται θα πρέπει να εκτελέσετε την εντολή:

```
g++ -O0 -o SmithWaterman SmithWaterman.cpp
```

Η εντολή αυτή θα μεταγλωττίσει το πρόγραμμα χωρίς βελτιστοποιήσεις (-O0) και θα παράξει ένα εκτελέσιμο αρχείο με το όνομα "SmithWaterman", το οποίο μπορείτε να εκτελέσετε με την εντολή:.

```
./SmithWaterman <Παράμετροι γραμμής εντολής>
```

- 2) Για την εκτέλεση του σειριακού προγράμματος που σας δίνεται και των παράλληλων προγραμμάτων που θα φτιάξετε απαιτείται το πέρασμα παραμέτρων από την γραμμή εντολής. Κάποιες από τις παραμέτρους γραμμής εντολών αφορούν χαρακτηριστικά του αλγόριθμου και σε όλα τα πειράματα θα χρησιμοποιήσετε τις ίδιες τιμές που αναφέρονται παρακάτω. Οι δύο βασικές παράμετροι είναι τα αρχεία εισόδου που περιέχουν τις ακολουθίες των συμβόλων που θέλουμε να ευθυγραμμίσουμε. Στον παρακάτω πίνακα αναφέρονται 6 αρχεία με ακολουθίες συμβόλων που σας δίνονται. Όταν θα δοκιμάζετε το πρόγραμμα σας να συγκρίνετε τις ακολουθίες που βρίσκονται στην ίδια σειρά του πίνακα.

Ακολουθία Α	Ακολουθία Β
LITMUS28i.gbk.txt	LITMUS28i-mal.gbk.txt
M13KE.gbk.txt	M13KO7.gbk.txt
Adenovirus-C.txt	Enterobacteria-phage-T7.txt

Κατά την διάρκεια ανάπτυξης και αποσφαλμάτωσης των προγραμμάτων σας μπορείτε να χρησιμοποιείται τις παρακάτω παραμέτρους, οι οποίες οδηγούν σε σχετικά μικρούς χρόνους εκτέλεσης:

```
./SmithWaterman 0.33 1.33 LITMUS28i.gbk.txt LITMUS28i-mal.gbk.txt 1000  
./SmithWaterman 0.33 1.33 M13KE.gbk.txt M13KO7.gbk.txt 1000
```

Με αυτές μπορείτε να ελέγχετε γρήγορα αν οι αλλαγές που κάνετε για την παραλληλοποίηση είναι σωστές. **Για τις μετρήσεις που θα περιλαμβάνονται στην αναφορά σας θα χρησιμοποιήσετε τις παρακάτω παραμέτρους**, που απαιτούν περισσότερο χρόνο:

```
./SmithWaterman 0.33 1.33 Adenovirus-C.txt Enterobacteria-phage-T7.txt 1000
```

- 3) Για την μεταγλώττιση του παράλληλου προγράμματος με OpenMP που θα φτιάξετε θα πρέπει να προσαρμόσετε τις παραμέτρους μεταγλώττισης. Συγκεκριμένα, θα πρέπει να προσθέσετε την παράμετρο "-fopenmp". Στην συνέχεια μπορείτε να ορίσετε το πλήθος των νημάτων που θα δημιουργούνται σε κάθε παράλληλη περιοχή και να εκτελέσετε την εφαρμογή με τις εντολές:

```
export OMP_NUM_THREADS=<Πλήθος νημάτων ανά παράλληλη περιοχή>  
./SmithWaterman <Παράμετροι γραμμής εντολής>
```

4) Για να μεταγλωττίσετε, να τρέξετε και να δείτε μετρήσεις με χρήση του Scalasca:

a. Προσαρμόστε την εντολή μεταγλώττισης ώστε να ξεκινάει με “scorep”:

```
scalasca -instrument g++ -O0 -o SmithWaterman SmithWaterman.cpp
```

b. Θέστε την μεταβλητή περιβάλλοντος “SCOREP_TOTAL_MEMORY”, ανάλογα με την διαθέσιμη μνήμη του υπολογιστή (ή του VM) σας, π.χ.:

```
export SCOREP_TOTAL_MEMORY=1500000000
```

c. Αν θέλετε να χρησιμοποιήσετε τους Performance Counters πρέπει να θέσετε την μεταβλητή περιβάλλοντος “SCOREP_METRIC_PAPI”:

```
export SCOREP_METRIC_PAPI=<Τιμή από πρόγραμμα papi_avail>
```

Για την λειτουργία των παραπάνω (και πολλών άλλων) μεταβλητών περιβάλλοντος που επηρεάζουν την εκτέλεση του Scalasca μπορείτε να δείτε το Appendix B στο User’s Guide του Scalasca.

d. Εκτέλεση:

```
scalasca -analyze ./SmithWaterman <Παράμετροι γραμμής εντολής>
```

Η παραπάνω εντολή εκτελεί το πρόγραμμα και δημιουργεί έναν νέο κατάλογο που περιέχει αρχεία με τα αποτελέσματα των μετρήσεων του Scalasca. Το όνομα του καταλόγου αυτού χρησιμοποιείται στην επόμενη εντολή.

e. Εμφάνιση αποτελεσμάτων:

```
scalasca -examine <Όνομα καταλόγου>
```

Παράρτημα Γ

Μπορείτε να βρείτε το manual του gcc στην ιστοσελίδα <http://gcc.gnu.org/onlinedocs>. Εκτελώντας την εντολή “gcc -v” βλέπετε ποια έκδοση του gcc έχετε εγκατεστημένη. Δείτε το αντίστοιχο manual από την παραπάνω ιστοσελίδα. Οι πληροφορίες για τις οδηγίες διανυσματοποίησης βρίσκονται στην παράγραφο με τίτλο “Loop-Specific Pragmas”.