Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος: Υπολογιστής αριθμητικών παραστάσεων μεταβλητής ακρίβειας

(near infinite multi-threaded calculator-inator)

Δημήτρης Κολιός (ΑΜ: 2338)

Επιβλέπων καθηγητής: Ιωάννης Ξεζωνάκης

ΗΡΑΚΛΕΙΟ 2012

Περιεχόμενα

Περίληψη

Έγραψα κώδικα για ένα πρόγραμμα το οποίο θα διαβάζει αριθμητικές παραστάσεις και θα βγάζει το αποτέλεσμα.

Summary

I coded a program that reads arithmetic representations and outputs the results.

Λέξεις-κλειδιά

Αριθμομηχανή, κομπιουτεράκι, υπολογιστής, μεγάλη ακρίβεια, μεγάλοι αριθμοί

Keywords

Calculator, bignum, big arithmetic, infinite precision

Εισαγωγή

Σκοπός αυτής της πτυχιακής εργασίας είναι η δημιουργία ενός προγράμματος, το οποίο θα δέχεται σαν είσοδο μία αριθμητική παράσταση σε μορφή όσο το δυνατόν κοντινότερη σε αυτή που θα έγραφε ένας άνθρωπος στο χαρτί και θα υπολογίζει το αποτέλεσμα για πολύ μεγάλους αριθμούς και κάνοντας χρήση όλων των πυρήνων του επεξεργαστή.

Αριθμοί του Προγράμματος

Το πρόγραμμα φτιάχτηκε με σκοπό να μπορεί να κάνει πράξεις μεταξύ πολύ μεγάλων αριθμών, αριθμών πολύ μεγαλύτερων απ' όσο μπορούν να κρατήσουν οι απλές μεταβλητές. Για το λόγο αυτό δημιουργήθηκε μία δομή, με ένα δείκτη σε ακέραιο και 3 ακεραίους.

Ο δείκτης λειτουργεί ως πίνακας ακεραίων μεταβλητού μεγέθους. Ένας αριθμός μπορεί να αποθηκευτεί σειριακά στον πίνακα, με κάθε στοιχείο του πίνακα να μπορεί να κρατήσει μέχρι 8 ψηφία του αριθμού. Ένας ακέραιος δείχνει το πλήθος των στοιχείων του πίνακα που κρατάνε το ακέραιο μέρος του αριθμού. Ένας άλλος δείχνει το πλήθος των στοιχείων που κρατάνε το δεκαδικό μέρος. Ο τελευταίος δείχνει μόνο το πρόσημο του αριθμού.

Για να αποθηκευτεί σωστά ένας αριθμός σε αυτή τη δομή, πρέπει να φροντίσουμε ο αριθμός να αποθηκευτεί έτσι ώστε κάθε ένα στοιχείο του πίνακα να περιέχει μόνο ψηφία του ακέραιου ή μόνο ψηφία του δεκαδικού μέρους του αριθμού.

Με αυτά ως δεδομένα, το πρόγραμμα μπορεί να αποθηκεύσει αριθμούς μέχρι $2^{32} * 2 * 8 =$ 68.719.476.736 ψηφία σε κάθε αριθμό (για την 32 bit έκδοση του προγράμματος). Με πολύ λίγες αλλαγές στον κώδικα και με διαφορετικό compiler μπορεί να δημιουργηθεί η 64 bit έκδοση, που μπορεί να αποθηκεύσει μέχρι $2^{64} * 2 * 18 =$ 664.082.786.653.543.858.176 ψηφία.

Υπάρχουν όμως αλγοριθμικοί περιορισμοί σε διάφορες συναρτήσεις του προγράμματος, που στις περισσότερες περιπτώσεις αποτρέπουν τη δημιουργία τόσο μεγάλων αριθμών, αλλά επειδή ο περιορισμός είναι διαφορετικός ανά περίπτωση (καθώς και γιατί πρακτικά με τους υπολογιστές που υπάρχουν σήμερα απέχουμε πολύ από τον υπολογισμό τόσο μεγάλων αριθμών με αυτό το πρόγραμμα), το πρόγραμμα δεν ειδοποιεί το χρήστη και θα κρασάρει (αντί να προστεθεί περιττός και πολύπλοκος κώδικας για να ελέγχει αυτές τις περιπτώσεις που δε θα εμφανιστούν ποτέ στην πραγματικότητα). Σε κάθε περίπτωση, ο μεγαλύτερος περιορισμός είναι στο διάβασμα της αριθμητικής παράστασης, όπου μπορεί να διαβάσει μέχρι 2^{32} (ή 2^{64}) χαρακτήρες.

Είσοδος Δεδομένων

Το πρόγραμμα φροντίζει να μην αφήνει το χρήστη να γράψει κάτι που δε θα είναι συντακτικά σωστό. Αυτό επιτυγχάνεται εφαρμόζοντας κάποιους κανόνες, οι οποίοι ελέγχονται αν ισχύουν κάθε φορά που διαβάζεται ένας καινούριος χαρακτήρας από το πληκτρολόγιο. Αν ισχύουν, τότε αυτός ο χαρακτήρας μπαίνει στη συμβολοσειρά και τυπώνεται στην οθόνη. Κάθε φορά που ένας χαρακτήρας σβήνεται, τότε ελέγχουμε τι χαρακτήρας ήταν αυτός και τι υπήρχε πριν απ' αυτόν, ώστε να αλλάξουν τα διάφορα flags που χρησιμοποιούνται για την εφαρμογή των παραπάνω κανόνων, σα να μην είχε γραφτεί ποτέ αυτός ο χαρακτήρας.

Όταν διαβαστούν τα δεδομένα, τότε τα έχουμε σε μία μορφή που είναι ευανάγνωστη από τον άνθρωπο, αλλά την οποία το πρόγραμμα δεν καταλαβαίνει. Τα δεδομένα είναι ακόμα σε μορφή συμβολοσειράς. Για το λόγο αυτό, πρέπει να τα μετατρέψουμε σε αριθμούς που καταλαβαίνει το πρόγραμμα και να το κάνουμε να εκτελεί τις πράξεις με τη σωστή σειρά.

Γι' αυτό υπάρχει μία συνάρτηση που μετατρέπει την παράσταση από ένθετη (η μορφή στην οποία γράφει ο χρήστης) σε μεταθεματική μορφή. Στη μεταθεματική μορφή, μία παράσταση περιλαμβάνει μόνο αριθμούς και τελεστές. Διαβάζεται από τα αριστερά προς τα δεξιά και στον πρώτο τελεστή που συναντάται, εκτελείται η αντίστοιχη πράξη με τους πλησιέστερους, αριστερά από αυτό, αριθμούς (στην περίπτωση του παραγοντικού, χρησιμοποιείται μόνο ένας αριθμός. Στις υπόλοιπες πράξεις, χρησιμοποιούνται οι 2 προηγούμενοι).

Κατά τη μετατροπή, δημιουργούνται 2 πίνακες. Ο ένας περιέχει τους αριθμούς με τη σειρά που θα τους συναντούσαμε στη μεταθεματική μορφή. Ο άλλος είναι μία συμβολοσειρά, που αναπαριστά τη μεταθεματική μορφή, αλλά επειδή δεν μπορεί να αποθηκεύει τους αριθμούς, χρησιμοποιεί το χαρακτήρα '0' για να δείξει ότι εκεί υπάρχει ένας αριθμός. Άρα τα μηδενικά στη συμβολοσειρά είναι όσα και οι αριθμοί στον άλλον πίνακα και αντιστοιχίζονται ένα προς ένα.

Εκτέλεση των Πράξεων

Τα στοιχεία της συμβολοσειράς με τη μεταθεματική παράσταση ελέγχονται από την αρχή της συμβολοσειράς μέχρι να εντοπιστεί κάποιος τελεστής. Τότε γίνεται η πράξη που υποδεικνύει ο τελεστής, η οποία επιστρέφει κάθε φορά έναν καινούριο αριθμό (το αποτέλεσμα της πράξης). Η πράξη θα γίνει με τους αριθμούς που αντιστοιχούν στα 2 προηγούμενα μηδενικά (στην περίπτωση της πράξης του παραγοντικού, χρησιμοποιείται ο αριθμός που αντιστοιχεί στο τελευταίο μηδενικό), οι οποίοι θα φύγουν από τον πίνακα και τη θέση τους θα πάρει το αποτέλεσμα της πράξης. Στη συμβολοσειρά θα αντικατασταθούν τα μηδενικά κι ο τελεστής που χρησιμοποιήθηκαν από ένα μηδενικό, που θα αντιστοιχεί στο αποτέλεσμα της πράξης που πραγματοποιήθηκε.

Οι πράξεις που υποστηρίζονται είναι οι εξής:

Πρόσθεση (+)

Αφαίρεση (-)

Πολλαπλασιασμός (*)

Παραγοντικό (!)

Δύναμη (^)

Διαίρεση (/)

Ρίζα (r)

Binomial coefficient $\binom{n}{k}$ (:) όπου n>k

Στην πράξη του παραγοντικού, της δύναμης και της ρίζας, η τάξη της δύναμης και της ρίζας, όπως και η βάση του παραγοντικού, δεν μπορούν να είναι δεκαδικοί αριθμοί (αν δοθεί δεκαδικός, τότε το πρόγραμμα βλέπει μόνο το ακέραιο μέρος του) και δεν μπορούν να υπερβαίνουν το ένα ψηφίο (8 πραγματικά ψηφία του δεκαδικού).

Πρόσθεση – Αφαίρεση

Το πρόγραμμα αντιλαμβάνεται τους αρνητικούς αριθμούς, οπότε η πράξη της πρόσθεσης μπορεί να οδηγήσει σε αφαίρεση, όπως και σε πρόσθεση αρνητικών αριθμών. Γι' αυτό πριν κληθεί η συνάρτηση που θα εκτελέσει την πράξη, γίνονται έλεγχοι, ώστε να δοθούν οι αριθμοί με τη σωστή σειρά και να ξέρει η συνάρτηση αν θα γίνει πρόσθεση ή αφαίρεση και αν το αποτέλεσμα θα είναι θετικό ή αρνητικό.

Στην αρχή υπολογίζει ποιος από τους 2 αριθμούς έχει μεγαλύτερο ακέραιο και ποιος έχει μεγαλύτερο δεκαδικό μέρος (σε πλήθος ψηφίων, όχι σε τιμή). Αυτές οι δύο τιμές αθροίζονται, για να ξέρει τι μέγεθος να περιμένει από το αποτέλεσμα. Αυτό το μέγεθος το διαιρεί με το πλήθος των επεξεργαστών στον υπολογιστή. Έτσι χωρίζει το αποτέλεσμα (το οποίο ακόμα δεν υπάρχει) σε όσο πιο ίσα κομμάτια μπορεί (για να μοιραστεί η δουλειά στους επεξεργαστές όσο καλύτερα γίνεται). Για κάθε κομμάτι απ' αυτά, βρίσκει σε ποιο κομμάτι των 2 αριθμών που αθροίζονται (ή αφαιρούνται) αντιστοιχίζεται και εκτελεί την πράξη με αυτά τα κομμάτια. Στο τέλος ενώνει αυτά τα κομμάτια προσθέτοντας κρατούμενα όπου υπάρχουν και σχηματίζει το αποτέλεσμα.

Πχ

-5.114+63.8=

Πρόσθεση με έναν αρνητικό και ένα θετικό, οπότε η πράξη που θα γίνει είναι 63.8-5.114.

Το μεγαλύτερο ακέραιο μέρος των 2 αριθμών είναι 2 ψηφία και το μεγαλύτερο δεκαδικό μέρος 3 ψηφία. Οπότε αναμένονται 3+2=5 ψηφία στο αποτέλεσμα.

Έστω ότι το πρόγραμμα εκτελείται σε υπολογιστή με 2 επεξεργαστές. Το αποτέλεσμα θα χωριστεί σε όσο πιο ίσα κομμάτια γίνεται, δηλαδή 3 και 2 ψηφία. Το αποτέλεσμα που αναμένεται θα είναι στη μορφή 00.000.

Η πρόσθεση θα γίνει ταυτόχρονα στα 2 κομμάτια.

1) 6 3 8 – 0 5 1 = 5 8 7
2) 0 0 – 1 4 = -1 (το κρατούμενο) 8 6

Στη διαδικασία της ένωσης παίρνει το πρώτο κομμάτι και του προσθέτει το κρατούμενο του επόμενου κομματιού (587-1=586). Μετά του κολλάει το δεύτερο κομμάτι και γίνεται 58686. Τέλος, πρέπει να το φέρει στη μορφή που έχει ήδη υπολογίσει (00.000), οπότε γίνεται 58.686 το οποίο είναι και το σωστό αποτέλεσμα.

Πολλαπλασιασμός

Όταν πολλαπλασιάζονται 2 αριθμοί, τότε επιλέγεται ένας από τους 2 για να χωριστεί σε όσο γίνεται ίσου μήκους κομμάτια. Για να έχουμε όσο γίνεται περισσότερα κομμάτια (σε περίπτωση που ο ένας από τους δύο αριθμούς είναι πολύ μικρός για να σπάσει), καθώς και για να γίνει όσο καλύτερα γίνεται η μοιρασιά, σπάμε το μεγαλύτερο από τους 2 αριθμούς.

Κάθε του κομμάτι πολλαπλασιάζεται με ολόκληρο τον άλλο αριθμό και τα αποτελέσματά τους ενώνονται όπως και στην πράξη της πρόσθεσης – αφαίρεσης. Δηλαδή είναι γνωστή η θέση που καταλαμβάνει το κάθε μέρος του αποτελέσματος και αφού κεντραριστούν σωστά, προσθέτονται τα ψηφία μεταξύ των αποτελεσμάτων που βρίσκονται στην ίδια θέση, γίνεται έλεγχος για κρατούμενα και μπαίνει η υποδιαστολή στο σωστό μέρος (στην ουσία, το μήκος του ακεραίου μέρους των 2 αριθμών προστίθεται και έτσι υπολογίζεται το μήκος του ακεραίου μέρους που αναμένεται από το γινόμενό τους).

Πχ
2.63123*30=

Μεγαλύτερος από τους 2 αριθμούς είναι ο 2.63123. Για υπολογιστή με 2 επεξεργαστές, χωρίζεται σε 263 και 123. Γίνονται οι πολλαπλασιασμοί:

1) 263*30=7890
2) 123*30=3690

Έπειτα τα αποτελέσματα μπαίνουν στη σωστή θέση κατά μήκος του τελικού αποτελέσματος και αθροίζονται όσα ψηφία είναι σε κοινή θέση:

7890

\+ 3690

7893690

Οι αριθμοί έχουν ακέραιο μέρος μήκους 1 και 2. Το αποτέλεσμα αναμένεται να έχει ακέραιο μέρος μήκους 1+2 (ή 1+2-1 αν η πράξη με το πρώτο κομμάτι του σπασμένου αριθμού δε βγάζει κρατούμενο). Οπότε ο αριθμός παίρνει την τελική μορφή 78.9369.

Δύναμη

Η ύψωση σε δύναμη συμβολίζει μία σειρά από πολλαπλασιασμούς, οπότε για κάθε έναν από αυτούς τους πολλαπλασιασμούς καλείται η αντίστοιχη συνάρτηση για να τον εκτελέσει. Για να μειωθεί ο χρόνος υπολογισμού, η δύναμη σπάει στους πρώτους παράγοντές της. Αυτό σημαίνει ότι θα γίνουν λιγότερες πράξεις, αλλά με μεγαλύτερους αριθμούς (εκτός αν η δύναμη είναι πρώτος αριθμός).

Πχ

2^12=4096

Η παραπάνω πράξη μπορεί να γραφτεί ως 2*2*2*2*2*2*2*2*2*2*2*2=4096

Αν παραγοντοποιήσουμε το 12, βγάζουμε 2*2*3

Οπότε θα κάνουμε ((2^2)^2)^3=

2*2=4 4*4=16 16*16*16=4096

Έτσι το αποτέλεσμα υπολογίζεται σε 4 πράξεις αντί για 11.

Παραγοντικό

Το πλήθος των νημάτων περνιέται σαν όρισμα, στο κάθε νήμα που δημιουργείται, μαζί με τον αρχικό αριθμό. Το δεύτερο νήμα θα πάρει τον αρχικό αριθμό μειωμένο κατά ένα, το τρίτο κατά δύο κοκ. Το κάθε νήμα μειώνει τον αριθμό που έχει πάρει κατά το πλήθος των νημάτων. Μετά πολλαπλασιάζει τον αριθμό που έχει πάρει με το αποτέλεσμα αυτό. Ξαναμειώνει τον αριθμό και τον πολλαπλασιάζει πάλι με το αποτέλεσμα. Αυτό επαναλαμβάνεται μέχρι να δει ότι ο αριθμός έχει πέσει κάτω από 2.

Στο τέλος, το αποτέλεσμα του κάθε νήματος επιστρέφεται και πολλαπλασιάζονται μέσω της συνάρτησης πολλαπλασιασμού για να βγάλουν το αποτέλεσμα.

Πχ

9!=

$1^{ο}$ νήμα) αριθμός: 9 νήματα: 2 9*7*5*3=945

$2^{ο}$ νήμα) αριθμός: 8 νήματα: 2 8*6*4*2=384

945*384=362880

Διαίρεση

Για την πράξη της διαίρεσης το πρόγραμμα χρησιμοποιεί μία υλοποίηση του αλγορίθμου Fourier division. Ο αλγόριθμος αυτός είναι πολύ δυνατός, αλλά δυστυχώς δεν επιτρέπει πολυπύρηνη υλοποίηση.

Ρίζα

Ο αλγόριθμος που χρησιμοποιείται για τη ρίζα, είναι προσεγγιστικός αλγόριθμος. Δηλαδή ξεκινάει από μία τιμή και κάθε φορά που εκτελείται, η τιμή αυτή πλησιάζει όλο και περισσότερο στο αποτέλεσμα. Όταν το πρόγραμμα δει ότι το τελευταίο αποτέλεσμα δεν έχει διαφορά από το προηγούμενο, τότε το κρατάει ως τελικό.

Ο αλγόριθμος έχει ως εξής:

$X_{i+1}=((n-1)*x_i+A/x_i^{n-1})/n$

n: η δύναμη της ρίζας

A: το υπόριζο

x_i: το προηγούμενο αποτέλεσμα (παίρνουμε ως $x_0=1$)

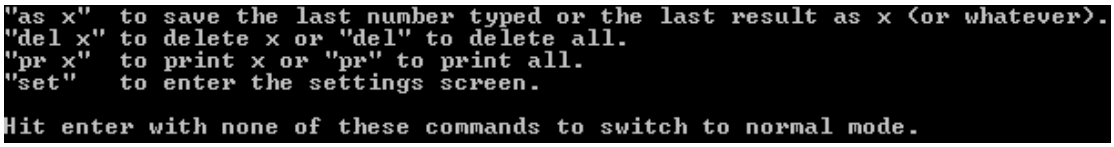
Το κακό με αυτή τη συνάρτηση είναι πως δημιουργήθηκε μόνο για λόγους πληρότητας του προγράμματος και δεν ήταν στις αρχικές προβλέψεις, οπότε υλοποιήθηκε βασισμένη στις υπόλοιπες, έτοιμες συναρτήσεις. Αυτό έχει ως αποτέλεσμα να είναι αρκετά αργή.

Binomial coefficient $\binom{n}{k}$

http://en.wikipedia.org/wiki/Binomial_coefficient

Μενού Εντολών

Με τη χρήση του χαρακτήρα backslash (\ ) οποιαδήποτε στιγμή, εμφανίζεται το μενού εντολών:



Οι 3 πρώτες εντολές έχουν να κάνουν με τη διαχείριση μεταβλητών που προσφέρει το πρόγραμμα στο χρήστη. Η τελευταία οδηγεί στο μενού ρυθμίσεων.

*Η πρώτη εντολή δεν εμφανίζεται αν το μενού ανοίξει την ώρα που υπολογίζεται μία αριθμητική παράσταση.

Μεταβλητές

Το πρόγραμμα δίνει τη δυνατότητα στο χρήστη να αποθηκεύσει έναν οποιοδήποτε αριθμό σε μεταβλητή με ένα γράμμα (case sensitive) του λατινικού αλφαβήτου για όνομα. Το γράμμα αυτό δεν μπορεί να είναι το r, επειδή χρησιμοποιείται για την πράξη της ρίζας.

Η εντολή "as x" δημιουργεί μεταβλητές και τους δίνει τιμή. Το x είναι το όνομα της μεταβλητής (θα μπορούσε να είναι οποιοδήποτε γράμμα εκτός από το r). Η τιμή της μεταβλητής εξαρτάται από το ποιος ήταν ο τελευταίος χαρακτήρας στην παράσταση, πριν πατηθεί το \. Αν ο χαρακτήρας ήταν αριθμητικός, τότε ο τελευταίος αριθμός που είχε γραφεί, θα είναι η τιμή της μεταβλητής. Σε οποιαδήποτε άλλη περίπτωση, ο αριθμός που θα αποθηκευτεί, είναι το τελευταίο αποτέλεσμα που υπολογίστηκε.

Η εντολή "del x" διαγράφει τη μεταβλητή με το όνομα x.

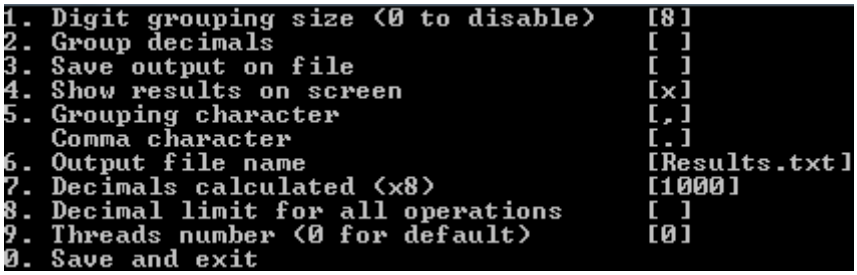
Η εντολή "del" διαγράφει όλες τις μεταβλητές.

Η εντολή"pr x" τυπώνει την τιμή της μεταβλητής x.

Η εντολή "pr" τυπώνει όλες τις μεταβλητές.

Μενού Ρυθμίσεων

Με τη χρήση της εντολής "set" εμφανίζεται το μενού ρυθμίσεων:



Αν πιέσετε το:

1) Το πρόγραμμα περιμένει έναν αριθμό, ο οποίος δείχνει το πλήθος των ψηφίων που θα γράφονται κολλητά, πριν χωριστούν από το grouping character (για να είναι πιο ευανάγνωστοι οι αριθμοί). Με τις τιμές 0 ή 8 επιτυγχάνεται η πιο γρήγορη εκτύπωση.
2) Καθορίζει αν θα χωρίζονται τα δεκαδικά ψηφία σε ομάδες ή όχι.
3) Καθορίζει αν τα αποτελέσματα θα αποθηκεύονται σε αρχείο ή όχι.
4) Καθορίζει αν τα αποτελέσματα θα εμφανίζονται στην οθόνη ή όχι.
5) Αλλάζει τους χαρακτήρες που χρησιμοποιούνται για την ομαδοποίηση ψηφίων και για την υποδιαστολή.
6) Το πρόγραμμα περιμένει μία συμβολοσειρά, η οποία θα χρησιμοποιηθεί ως όνομα για το αρχείο αποτελεσμάτων (δε θα αλλάξει το όνομα του ήδη υπάρχοντος αρχείου).
7) Το πρόγραμμα περιμένει έναν αριθμό, ο οποίος δείχνει το πόσες οκτάδες δεκαδικών θα υπολογίζονται σε πράξεις όπως η διαίρεση και η ρίζα, που μπορούν να βγάλουν άπειρα δεκαδικά.
8) Καθορίζει αν το παραπάνω όριο θα ισχύει και για τις υπόλοιπες πράξεις.
9) Καθορίζει το πλήθος των νημάτων που θα προσπαθήσει το πρόγραμμα να ανοίξει σε κάθε πράξη. Στην τιμή 0 το πρόγραμμα προσπαθεί να ανοίξει ένα νήμα για κάθε επεξεργαστή του υπολογιστή.
0) Αποθηκεύονται οι ρυθμίσεις στον αρχείο ρυθμίσεων και ξεκινάνε να ισχύουν οι αλλαγές.

Το πρόγραμμα κάθε φορά που ξεκινάει ψάχνει για το αρχείο ρυθμίσεων. Αν δεν το βρει, τότε βάζει τις default ρυθμίσεις (που είναι αυτές που φαίνονται στην εικόνα πιο πάνω).

Συγκρίσεις

Ο παρακάτω πίνακας περιέχει τιμές με το χρόνο που χρειάστηκε να υπολογιστούν κάποιες πράξεις στο δικό μου πρόγραμμα (με χρήση δύο νημάτων και ενός νήματος) και στο bigal. Το bigal τρέχει πάντα σε ένα νήμα, αλλά μετά από πολύ ψάξιμο στο Google ήταν το μόνο που βρήκα να μπορεί να εκτελεί τόσο μεγάλες πράξεις. Η πλειοψηφία των αποτελεσμάτων ήταν επιστημονικά κομπιουτεράκια και λίγα ακόμα, που δεν έβγαζαν ούτε ένα εκατομμύριο ψηφία.

Οι τιμές αυτές παρατηρήθηκαν στο laptop μου, με επεξεργαστή Intel i3 M380 @2.53 (2 πυρήνες) με το hyperthreading απενεργοποιημένο και 2 κανάλια μνήμης στα 1066MHz.

Πράξεις	2 threads	1 thread	bigal
50000!	5.5	13.5	11.5
80000!	14.5	36.5	30
100000!	23.5	58	50.5
1234567890^100000	56	112.5	130
1234567890^65536	19	37	54.5
1234567890^7919	29	99	1
1/3 (800000 decimal digits)	-	0.053	109
2345026345023560687516027563206320576123087561203712035612035716023571 / 1328561032857613271562936510365015615610556018568912597326501273650175601875017508 7 (240000 decimal digits)	-	0.076	10.5
25r1234567890 (160 decimal digits)	1.1	1.1	0-0.5
50r1234567890 (160 decimal digits)	6.5	9	0-0.5
100r1234567890 (160 decimal digits)	40	74	0-0.5

Για να χρησιμοποιήσετε το bigal, τρέχετε cmd, πηγαίνετε στη διαδρομή του bigal.jar και γράφετε java –jar bigal.jar. Καλύτερα την πρώτη φορά να το τρέξετε συμπληρώνοντας help. Έτσι θα σας εμφανίσει όλες τις δυνατότητές του και πώς να τις χρησιμοποιήσετε.

Παράδειγμα πολλαπλασιασμού με το bigal:

java –jar bigal.jar 5 * 5

(Μην ξεχνάτε τα κενά ανάμεσα στα ορίσματα)

Σύνδεσμοι

μέγεθος παραγοντικού

http://inder-gnu.blogspot.com/2009/08/find-number-of-digits-in-factorial-of.html

αλγόριθμος διαίρεσης

http://en.wikipedia.org/wiki/Fourier_division

αλγόριθμος ρίζας

http://en.wikipedia.org/wiki/Nth_root_algorithm

παραγοντοποίηση

http://en.wikipedia.org/wiki/Prime_factor

ένθετη – μεταθεματική μορφή

http://en.wikipedia.org/wiki/Reverse_Polish_notation

bigal

http://bigal.sourceforge.net/

νήματα

Όταν δούλευα σαν αμειβόμενος φοιτητής υπό τον κύριο Μανιφάβα, είχα φτιάξει σημειώσεις για εκμάθηση και κατανόηση των νημάτων για το μάθημά του «Λειτουργικά Συστήματα». Σας τις παραθέτω στο αρχείο threads.pdf.

http://en.wikipedia.org/wiki/Thread_(computing)