

EM-091 – Εισαγωγή στους Η/Υ

Εργαστηριακή Άσκηση 3

7 Νοεμβρίου 2007

Σκοπός αυτής τής άσκησης όπως και τής προηγούμενης, είναι η βαθύτερη κατανόηση και εκμετάλλευση τής διαχείρισης διανυσματικών μεταβλητών από την *FORTRAN*. Υπενθυμίζω ότι:

α) Τα στοιχεία μιάς διανυσματικής μεταβλητής τού κυρίως προγράμματος αποθηκεύονται ακολουθιακά σε περιοχή μνήμης που δεσμεύεται αρχικά και έτσι ώστε, μεταξύ οποιονδήποτε δύο διαφορετικών στοιχείων, να προηγείται αυτό με τον αριστερότερα μικρότερο δείκτη, π.χ. τα 105 στοιχεία τής διανυσματικής μεταβλητής $A(3, 5, 7)$ θα αποθηκευτούν ακολουθιακά ώστε το A_{ijk} στοιχείο να βρίσκεται $(i - 1) + 3 * (j - 1) + 3 * 5 * (k - 1)$ θέσεις από την αρχή.

β) Κάθε υποπρόγραμμα αποτελεί αυτόνομη υπολογιστική μονάδα με μοναδικό του σύνδεσμο με κάποιο κυρίως πρόγραμμα, τής μεταβλητέσ-ορίσματά του και τής κοινές μεταβλητές (αυτές που δηλώνονται μέσω κάποιας εντολής *common*). Για κάθε μεταβλητή-όρισμα εκτός από τής διανυσματικού τύπου, δεσμεύεται χώρος μνήμης που ενημερώνεται κατά την κλήση αυτού τού υποπρογράμματος με τής τιμές τών αντίστοιχων ορισμάτων κλήσης, Για τα ορίσματα-διανυσματικού τύπου που δέν δεσμεύεται χώρος, το πρώτο στοιχείο βρίσκεται στην ίδια θέση μνήμης με τού αντίστοιχου ορίσματος, κατά την κλήση τού υποπρογράμματος, ενώ οι θέσεις τών υπολοίπων στοιχείων υπολογίζονται με βάση τον τοπικό ορισμό αυτής τής διανυσματικής μεταβλητής. Τόσο ο αριθμός τών διαστάσεων όσο και το μήκος κάθε διάστασης, μιάς διανυσματικής μεταβλητήσ-όρισμα μπορεί να διαφέρουν από αυτούς του κυρίως προγράμματος χωρίς να επέρχεται πρόβλημα, τουλάχιστον όσο η επεξεργασία περιορίζεται στον δεσμευμένο χώρο από το κυρίως πρόγραμμα, π.χ.

κυρίως πρόγραμμα	υποπρόγραμμα
<i>programm testMatrix</i>	<i>subroutine testrout(B)</i>
<i>real * 8 A(3, 5, 7)</i>	<i>real * 8 B(8, 4)</i>
<i>call testrout(A(2, 4, 6))</i>	:
<i>stop</i>	<i>return</i>
<i>end</i>	<i>end</i>

Σ' αυτό το παράδειγμα, στο κυρίως πρόγραμμα δεσμεύεται χώρος για τα 105 στοιχεία τής τρισδιάστατης διανυσματικής μεταβλητής A ενώ το υποπρόγραμμα *testrout* διαχειρίζεται ένα προσφερόμενο χώρο 32 στοιχείων σαν δισδιάστατη διανυσματική μεταβλητή, την B . Από την συγκεκριμένη κλήση ο χώρος αυτός ξεκινάει $(2-1)+3*(4-1)+3*5*(6-1) = 85$ θέσεις από την αρχή τού δεσμευμένου χώρου στο κυρίως πρόγραμμα, οπότε το $B_{i,j}$ στοιχείο αντιστοιχεί ουσιαστικά στην $85+(i-1)+8*(j-1)$ θέση αυτού του χώρου. Τα τελευταία 12 στοιχεία αντιστοιχούν σε μη δεσμευμένες θέσεις και άρα πιθανή αποθήκευση τιμών στίς μεταβλητές $B(5, 3), \dots, B(8, 3), B(1, 4), \dots, B(8, 4)$ θα δημιουργήσει λειτουργικά προβλήματα.

ΠΑΡΑΤΗΡΗΣΗ: Οποιαδήποτε αναφορά στο περιεχόμενο των θέσεων μνήμης Mem , $Mem+K$ μπορεί να εκτελεστεί τουλάχιστον 5 φορές πιά αργά από αντίστοιχη αναφορά στο περιεχόμενο των θέσεων μνήμης Mem , $Mem+1$ διότι λόγω τής απόστασης τους οι θέσεις μνήμης Mem , $Mem+K$ ίσως βρίσκονται σε άλλη "σελίδα μνήμης". Γενικά πρέπει να υλοποιούμε έναν αλγόριθμο έτσι ώστε η αναφορές σε περιεχόμενα μνήμης να γίνονται με τον συνεκτικότερο δυνατό τρόπο.

Σ' αυτήν την εργασία θα υλοποιήσουμε ένα πρόγραμμα που θα παράγει το γινόμενο $A \times B$ δύο πινάκων $A = (a_{ij}) \in \mathbb{R}^{M,K}$, $B = (b_{ij}) \in \mathbb{R}^{K,N}$, δηλαδή θα παράγει τον πίνακα $C = (c_{ij}) \in \mathbb{R}^{M,N}$ με στοιχεία τους πραγματικούς $c_{ij} = \sum_{m=1}^K a_{im} \cdot b_{mj}$, $i = 1, \dots, M$, $j = 1, \dots, N$.

Παρατηρήστε ότι στους όρους a_{im} του παραπάνω αθροίσματος μεταβάλλεται ο δείκτης στηλών, δηλαδή προκειμένου να παραχθεί το τελικό άθροισμα, σύμφωνα με την υπενθύμιση α) θα χρησιμοποιηθούν οι θέσεις μνήμης Mem , $Mem+M$, $Mem+2M$, ..., $Mem+(K-1)M$ (όπου Mem είναι η θέση τού στοιχείου a_{i1}) κάτι που θα επιφέρει σημαντική καθυστέρηση ειδικά αν λάβουμε υπόψιν ότι θα υπολογιστούν $N \cdot M$ τέτοια αθροίσματα. Για τους όρους b_{mj} δέν υπάρχει πρόβλημα καθυστέρησης αφού στο παραπάνω αθροίσμα μεταβάλλεται ο δείκτης γραμμών και άρα χρησιμοποιούνται διαδοχικές θέσεις μνήμης. Άς παρατηρήσουμε ότι στο ίδιο αποτέλεσμα οδηγεί η πράξη

$$c_{ij} = \sum_{m=1}^K a_{mi}^t \cdot b_{mj}, \quad i = 1, \dots, M, \quad j = 1, \dots, N$$

όπου a_{mi}^t είναι το (m, i) στοιχείο τού ανάστροφου πίνακα $A^t = (a_{ij}^t) \in \mathbb{R}^{K,M}$ που ορίζεται μέσω του αρχικού πίνακα έτσι ώστε $a_{ij}^t = a_{ji}$. Σ' αυτήν την μορφή δέν τίθεται πρόβλημα καθυστέρησης αφού σ' όλους τούς όρους αυτού τού αθροίσματος μεταβάλλονται μόνο οι δείκτες γραμμών.

Στον κατάλογο Fortran τής περιοχής σας, φτιάξτε ένα πρόγραμμα που να αποτελείται από:

- Την συνάρτηση $real*8 \text{ esgi}(A, B, N)$ που δέχεται σάν ορίσματά τον ακέραιο N και τα διανύσματα $A, B \in \mathbb{R}^N$ και επιστρέφει το εσωτερικό τους γινόμενο $\sum_{i=1}^N a_i \cdot b_i$
- Την υπορουτίνα $AepiX(A, x, y, M, K)$ που δέχεται σάν ορίσματά τούς ακεραίους M και K , τον πίνακα $A \in \mathbb{R}^{K,M}$ και τα διανύσματα $x \in \mathbb{R}^K, y \in \mathbb{R}^M$ και υλοποιεί την πράξη $y = A^t \cdot x$, καλώντας κατάλληλα τήν συνάρτηση $esgi$, M φορές.
- Την υπορουτίνα $AepiB(A, B, G, M, K, N)$ που δέχεται σάν ορίσματά τούς ακεραίους M, K και N , και τούς πίνακες $A \in \mathbb{R}^{K,M}$, $B \in \mathbb{R}^{K,N}$ $G \in \mathbb{R}^{M,N}$ και υλοποιεί την πράξη $G = A^t \cdot B$, καλώντας κατάλληλα τήν υπορουτίνα $AepiX$, N φορές.
- Το κυρίως πρόγραμμα στο οποίο ορίζονται τρεις πίνακες $A \in \mathbb{R}^{K,M}$, $B \in \mathbb{R}^{K,N}$ $C \in \mathbb{R}^{M,N}$ με παραμετρικές διαστάσεις K, M, N . Αρχικά γεμίζονται οι πίνακες A, B απο τα αρχεία $pina, pinb$ αντίστοιχα στα οποία έχουμε αποθηκεύσει τα στοιχεία των πινάκων ΚΑΤΑ ΓΡΑΜΜΕΣ (στην i γραμμή των αρχείων αυτών περιέχονται τα στοιχεία τής i γραμμής τού αντίστοιχου πίνακα χωρισμένα με , (κόμμα)). ΠΡΟΣΟΧΗ ο πίνακας A πρέπει να γεμίσει "ανάστροφα". Έπειτα καλείται κατάλληλα η υπορουτίνα $AepiB$ για να γίνει η ζητούμενη πράξη και τέλος εκτυπώνεται στην οθόνη το περιεχόμενο τού πίνακα C ΚΑΤΑ ΓΡΑΜΜΕΣ (στην i γραμμή οθόνης εκτυπώνονται τα στοιχεία τής i γραμμής τού πίνακα C) φορμαρισμένα σε διάστημα 20 χαρακτήρων εκ των οποίων οι 10 είναι για το δεκαδικό μέρος.

ΠΑΡΑΤΗΡΗΣΗ Οι εντολές $format$ δέν ενημερώνονται για τίς τιμές των παραμέτρων οπότε, για την είσοδο δεδομένων διανυσματικών μεταβλητών πρέπει να ΑΠΟΦΕΥΓΟΝΤΑΙ ενώ για την έξοδο διανυσματικών δεδομένων πρέπει να ενημερώνονται σε κάθε αλλαγή παραμέτρου που τίς αφορά (π.χ. διάσταση γραμμής του πίνακα που εκτυπώνεται).

Εξέταση στο εργαστήριο Η/Υ, την Τετάρτη 14 Ιανουαρίου 2004 στις
ώρες 14.00-18.00 και Πέμπτη 15 Ιανουαρίου 2004 στις ώρες 10.00-15.00
Προαπαιτείται η δήλωση του μαθήματος στην Γραμματεία.