

ΤΟ ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

α. Ένικά

Όταν λέμε «λογισμικό» εννοούμε το σύνολο των προγραμμάτων που απαιτούνται σ' ένα Η/Υ για να εκτελέσει την επιθυμητή εργασία.

Τα προγράμματα που μέχρηι τώρα έχουμε χρησιμοποιήσει σε γλώσσα μηχανής είναι εξειδικευμένοι λογισμικοί.

Ένικότερα όμως το **ΛΟΓΙΣΜΙΚΟ** μπορεί να διακριθεί σε δύο κατηγορίες. Η πρώτη περιλαμβάνει τα **ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΤΑΧΡΗΣΗΣ** ενώ η δεύτερη τα **ΠΡΟΓΡΑΜΜΑΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ (Sys-tems programs)**.

Τα προγράμματα του Χρήστη στόχο έχουν να επιτευχθούν διάφορες επεξεργασίες που ενδιέφερον τον Χρήστη. Έτσι πολλαές φορές αυτά συνδέονται σε ομάδες προγραμμάτων («suites»). Π.χ. ένα αυ- τοματοποιημένο σύστημα καταγραφής των φοιτητών και των επιδόσεών τους στο Τμήμα Μαθηματικών, μπορεί να κτιστεί με μια ομάδα προ- γραμμάτων που θα δημιουργήσουν τα κατάλληλα αρχεία (files), για την ακαδημαϊκή ενημέρωση των επιδόσεων των φοιτητών στα διάφο- ρα μαθήματα και εργαστήρια, τις κατάλληλες πληροφορίες κατά την υ- λωγή στατιστικών στοιχείων και ωριμότητα των μαθητών και γινώσκοντες και διασκόμμενους, ζήτησε να ελε- γχό να είναι άμεσο και ακριβές, ενώ παράλληλα στη διοίκηση του υπηρέτησαν ωριμότητα των μαθητών με τη μέριμνα των κολλεγιακών υπηρεσιών να είναι πάντα διαθέσιμα και να είναι «δυνατά» με τα διαθέσιμα στοιχεία.

Ο M. Jackson* κατατάσσει τα διάφορα συστήματα προγραμματικών

στις επόμενες τρεις ευρείες κατηγορίες:

Συστήματα μικρά: μέχρι 10000 εγγραφές

Συστήματα μεσαία: μέχρι 100000 εγγραφές

Συστήματα μεγάλα: περισσότερα από 100000 εγγραφές.

Στο σημείο αυτό είναι χρήσιμο ένα ξεκαθάρισμα. Βασικά ο Χρήστης για την οριοθέτηση ανάγκη του μπορεί να συντάξει ένα πρόγραμμα-μάσκα σε γλώσσα μηχανής, να το υποβάλει στον Η/Υ για επεξεργασία και να λάβει τα αποτελέσματά της επεξεργασίας του. Στην πράξη, όμως, έχει αποδειχθεί ότι η προηγούμενη διαδικασία είναι φοβερά απαιτητική και πολύ επιφανής σε σφάλματα, που στη συνέχεια είναι χρονοβόρα στη διόρθωσή τους, καθόσον προκύβουν αλυσόμενες διορθώσεις.

*M. Jackson: Principles of Program Design, Academic Press 1975.

των αρχικών:

Οι εξής εντολές, που θα εκτυπώσαν συγχρόως και τον μέσον όρο
 το εργατήριο 1, που έχετε παραδώσει, στη FORTRAN ακούσαν
 επιφανίζεται στο μέσο έξοδο του συστήματος, που σχετίζεται με
 αριθμούς N αρχικών, οι οποίοι διαβάζονται από τον H/Γ και
 προβάλλονται. Έτσι, για το γινόμενο μας πρόβλημα της εύρεσης του
 τα, που διεκονούει αφάνταστα τον προγραμματισμό επιστημονικών
 FORTRAN (Formula Translator) για τα επιστημονικά πρόβλημα-
 αναφέρουμε τις γλώσσες υψηλού επιπέδου, όπως είναι η γλώσσα
 γραμμάτων των χρηστών. Ως παράδειγμα λογισμικού συστήματος
 για τη συγγραφή, έλεγχο, επεξεργασία και αποθήκευση των προ-
 γραμμάτων που είναι η δημιουργία ενός κατάλληλου περιβάλλοντος
 με «Λογισμικό συστήματος» (systems software), του οποίου ο αποκ-
 υπολογιστικών συστημάτων προκάλεσαν τον εφοδιασμό των H/Γ
 κατασκευαστών να κάνουν όσο το δυνατόν ευρύτερη τη χρήση των
 Έτσι, οι ανάγκες των χρηστών αλλά και η επιθυμία των διαφόρων

(Σημείωση): Το πρόγραμμα υποθέτει την επεξεργασία μέχρι 1000 αριθμών, αλλά και οσονδήποτε άλλων, αρκεί ο αριθμός 1000 να προσαρμοστεί κατάλληλα και τίποτε άλλο! Βάλτε εξάλλου και το ΠΑΡΑΡΤΗΜΑ 1, του βιβλίου σας στη σελίδα 250, 5γ στο FOR-TRAN).

```

PROGRAM ADD
DIMENSION A(1000)
READ *, N, (A(I), I=1,N)
SUM=0
DO I=1,N
SUM=SUM+A(I)
END DO
AVERAGE=SUM/N
PRINT *, SUM, AVERAGE
END PROGRAM ADD

```

σελ. 249).

Φυσικά, τα πρώτα προγράμματα Η/Υ ήταν γραμμένα σε κώδικα μηχανής με χρήση δυαδικού συστήματος και ήταν στη μορφή του πακέτου α, του ΠΑΡΑΡΤΗΜΑΤΟΣ 1, του βιβλίου σας (βλέπε

β. ΣΥΜΒΟΛΟΜΕΤΑΦΡΑΣΤΕΣ (Symbolic Assemblers)

προγράμματος (system program).

Στη συνέχεια θα δούμε μια γενική εικόνα των κύριων τύπων του

α, όπως π.χ. τα WINDOWS της Microsoft Corporation.

Τέλος, το λογισμικό συστήματος, δηλαδή τα προγράμματα που το συνάπτουν συνήθως γράφονται από τους ίδιους κατασκευαστές των Η/Υ και αγοράζονται μαζί με την μηχανή. Τα τελευταία χρόνια όμως βλέπουμε και ανεξάρτητους οίκους, όπως είναι τα SOFTWARE HOUSES κτλ, να συνθέτουν λογισμικά Συστημάτων με μεγάλη επιτυχία.

Αυτό συνεβη στην αρχή και για μικρά μόνο προγράμματα με μικρές δεκάδες εντολές. Πολύ σύντομα όμως εμφανίστηκαν προγράμματα δεκάδες εκατοντάδων εντολών, οπότε η ανάγκη να δοθούν διευθύνσεις μνήμης σε κάθε εντολή εκ μέρους του προγραμματιστή-χρήστη, ιδιαίτερα στις εντολές αλλαγής ποής του προγράμματος (Branch instructions) ήταν φοβερά πολυπλόκο. Ένα παράδειγμα θα δώσει καλύτερη εικόνα των δυσκολιών το παρακάτω πρόγραμμα διαβάσει ένα φυσικό αριθμό N και υπολογίζει το άθροισμα των N πρώτων φυσικών αριθμών.

μών.

0	INA	Αποθήκευσε την τιμή του N στον καταχωρητή A
1	STA A 50	Αποθήκευση του N στη θέση 50
2	ADD B 50	Πρόσθεσε στον Ας B το περιεχόμενο της 50
3	SUB A #1	Αφαίρεσε 1 από τον A
4	BNZ A 1	Πάνε στη θέση 1 εάν $A \neq 0$
5	OUT B	Δώσε στην έξοδο τον B $= \sum_{i=1}^N i$
6	JMP 0	Πάνε στη θέση 0 για επανάληψη του προγράμματος.
7	HLT	Τέλος

(1)

(2):

Το σφάλμα είναι απλό και έχει σχέση με το περιεχόμενο του α-θροιστή B που πρέπει να είναι 0 προτού αρχίσει να σχηματίζεται το περιεχόμενο αθροιστή, πράγμα που ισχύει στην περίπτωση του προ-γράμματος! Στην δεύτερη χρήση όμως είχε περιεχόμενο 6 από την αρχική, που προστέθηκε στο 0 και έτσι έγινε 16. Προφανώς, LDA B#0, πράγμα που τροποποιεί το πρόγραμμα (1) στο ακόλουθο

Τρίτη δοκιμή με N=1 δίνει ΕΞΟΔΟ 17! Αντί του 1.

Στην επόμενη όμως χρήση έχουμε:
• 4 και ΕΞΟΔΟΣ 16, που προφανώς είναι λάθος, αφού $1+2+3+4=10$.

Στην πρώτη χρήση του προγράμματος παίρνουμε:
• 3 και ΕΞΟΔΟΣ 6, που είναι μια ορθή απάντηση αφού $1+2+3=6$.

φορές από αυτά που διόρθωσαν.

θώσες προσπαθώντας να εισαγάγουν περισσότερα σφάλματα ηερικές λειτούργειες και σχεδιαστικές προσπαθώντας, με συνέπεια οι διορθώσεις ο ανθώπινος παράγοντας δεν είναι αποδοτικές σε γραφειοκρατικές, τόσο οι προσπαθώσεις είναι πιο πολύπλοκες, ενώ είναι γρήγορη πείρα είναι προφανές ότι όσο το μέγεθος του προγράμματος αυξάνεται

στη θέση 2 αντί της 1.

αλλά και στην εντολή BNZ που προσάρμοστηκε ο προορισμός της με αλλαγές σε όλες τις εντολές (αναριθμήσεις όλων των εντολών)

```

0 LDA B#0
1 INA A
2 STA A 50
3 ADD B 50
4 SUB A #1
5 BNZ A 2
6 OUT B
7 JMP 0

```

(2)

```

JOHN: INA A
ANN: STA A 50
ADD B 50
SUB A #1
BNZ A ANN
OUT B
JMP JOHN

```

(3)

Είσι τα (1) και (2) γράφονται:

αντί των διευθύνσεων (με κάρδαληγη κατάργηση των διευθύνσεων).

2N, που επιτρέπουν στους προγραμματιστές να κάνουν χρήση ονομάτων

Το πρόβλημα επιλύθηκε με την επινόηση των ΣΤΜΒΟΝΟΜΕΤΑΦΑΣΤ-

Τα ονόματα JOHN και ANN είναι ενδεικτικά, αρκεί να είναι δι-
 αφορετικά, έτσι ώστε ο έλεγχος του προγράμματος να μεταφερθεί
 ορθά. Το σπουδαίο της χρήσης των ονομάτων είναι ότι η πρόσ-
 θεση η αφαίρεση κατοικίας εντολής δεν προκαλεί άλλες αλλαγές.
 Πρωτώς, τα προγράμματα (3) και (4) δεν μπορούν να τρέξουν
 σε Η.Υ. Πρέπει πρώτα να μεταφραστούν σε κώδικα μηχανής από
 ένα κατάλληλο πρόγραμμα (system program) που ονομάζεται **JM-**
BOVOMETΑΦΑΣΤΗΣ (Symbolic Assembler).

```

JOHN: LDA B
      INA A
ANN: STA A 50
      ADD B 50
      SUB A #1
      BNZ A ANN
      OUT B
      JMP JOHN
  
```

(4)

ΕΠΙΣΤΗΜΟΝΙΚΟ ΠΡΟΓΡΑΜΜΑ

Η παρούσα μελέτη πραγματοποιήθηκε στο πλαίσιο του προγράμματος «Επένδυση στην Έρευνα και Καινοτομία» (ΕΠΕΚΕ) του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων. Η μελέτη αφορά στην αξιολόγηση της αποτελεσματικότητας των παρεμβάσεων που υλοποιούνται στο πλαίσιο του προγράμματος.

Η μελέτη πραγματοποιήθηκε με τη βοήθεια της μεθόδου των ερωτηματολογίων και της μεθόδου των συνεντεύξεων. Τα αποτελέσματα της μελέτης παρουσιάζονται στην παρούσα μελέτη.

Η μελέτη πραγματοποιήθηκε στο πλαίσιο του προγράμματος «Επένδυση στην Έρευνα και Καινοτομία» (ΕΠΕΚΕ) του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων. Η μελέτη αφορά στην αξιολόγηση της αποτελεσματικότητας των παρεμβάσεων που υλοποιούνται στο πλαίσιο του προγράμματος.

Η μελέτη πραγματοποιήθηκε με τη βοήθεια της μεθόδου των ερωτηματολογίων και της μεθόδου των συνεντεύξεων. Τα αποτελέσματα της μελέτης παρουσιάζονται στην παρούσα μελέτη.

Η μελέτη πραγματοποιήθηκε στο πλαίσιο του προγράμματος «Επένδυση στην Έρευνα και Καινοτομία» (ΕΠΕΚΕ) του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων. Η μελέτη αφορά στην αξιολόγηση της αποτελεσματικότητας των παρεμβάσεων που υλοποιούνται στο πλαίσιο του προγράμματος.

Η μελέτη πραγματοποιήθηκε με τη βοήθεια της μεθόδου των ερωτηματολογίων και της μεθόδου των συνεντεύξεων. Τα αποτελέσματα της μελέτης παρουσιάζονται στην παρούσα μελέτη.

Η μελέτη πραγματοποιήθηκε στο πλαίσιο του προγράμματος «Επένδυση στην Έρευνα και Καινοτομία» (ΕΠΕΚΕ) του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων. Η μελέτη αφορά στην αξιολόγηση της αποτελεσματικότητας των παρεμβάσεων που υλοποιούνται στο πλαίσιο του προγράμματος.

Η μελέτη πραγματοποιήθηκε με τη βοήθεια της μεθόδου των ερωτηματολογίων και της μεθόδου των συνεντεύξεων. Τα αποτελέσματα της μελέτης παρουσιάζονται στην παρούσα μελέτη.

Οι γλώσσες υψηλού επιπέδου (Γ.Υ.Ε) προσφέρουν ένα άριστο
 εργαλείο προγραμματισμού, που υλοποιούν τους αντίστοιχους αλγόριθμους
 επίσπευσης των διαδικασιών, ενώ η προγραμματιστική
 φιλοσοφία της είναι γενικά απλή και εύκολη στην κατανόηση.

2. ΓΛΩΣΣΕΣ ΨΗΦΙΑΚΩΝ ΕΠΙΠΕΔΩΝ (High-level languages)

στη συνέχεια
 φασικά είναι προγραμματιστικά εργαλεία σε γλώσσα υψηλού επιπέδου πρέπει
 να μεταφραστεί σε γλώσσα μηχανής, από τον ίδιο τον Η.Υ.,
 ή να εκτελεστεί (δηλαδή) ή να μεταφραστεί σε γλώσσα
 (Interpreter), που είναι προγραμματιστικά εργαλεία σε επίπεδο
 μηχανής, τα οποία θα δοθούν.

αποτέλεσμα.

1. Χρήση συμβολικών ονομάτων (symbolic names) αντί διευθετών-
σων ληξών, όπου αποθηκεύονται ενδιάμεσα και τελικά

με τις παρακάτω κατηγορίες:

Τα κοινά χαρακτηριστικά των Γ.Γ.Ε. μπορούν να αποδοθούν στο πρόβλημα που θα αντιμετωπίσει και στον αλγόριθμο επίλυσης ήτοι να χρησιμοποιήσει και στην εισαγωγή της προσοχής του προγράμματος από τις λειτουργίες του υπολογιστικού συστή-
ματος. Πάντως όλες οι Γ.Γ.Ε έχουν κάποια «φιλσοσοφία» και κοινά χαρακτηριστικά, που επικεντρώνονται στην απαλάφη του σκοπού. Για τον παρακείμενο λόγο, που είναι μια γλώσσα γενικού λήματος, ή γλώσσα COBOL στα ευρωπαϊκά πρόβλήματα και η λήματος όπως, π.χ., η γλώσσα FORTRAN στα επιστημονικά προβ-
λήματα και προσφέρουν ειδικές εξυπηρέτησεις σε κατηγορίες προβ-
λήματα υπάρχουν εκάστοτε Γ.Γ.Ε, που όλες έχουν ειδικό προσανα-

$$x1 = (-B + (B^2 + 4 * A * C) * 0.5) / (2 * A)$$

Στις Γ.Υ.Ε. οι μαθηματικές εκφράσεις για τους διαφόρους υπολογισμούς προσαρμοσούν τις γλώσσες μας αλγεβρικές μορφές με τα γνωστά σύμβολα των υπολογισμών για πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση, ενώ αυτές εκχωρούνται με την βοήθεια των συμβόλων = ή του := . Π.Χ. στο FORTRAN εάν θέλουμε να υπολογίσουμε την ρίζα x_1 μιας δευτεροβάθμιας εξίσωσης $Ax^2 + Bx + C = 0$, μπορούμε να γράψουμε:

2. Εκφράσεις και εκχωρήσεις τιμών (Expressions and Assignments)

Τα σύμβολα ονόματα είναι ομάδες αλφαριθμητικών συμβόλων που αρχίζουν από ένα γράμμα και σε ορισμένες περιπτώσεις το μήκος της ομάδας έχει όριο (π.Χ. στο FORTRAN 90, το μήκος είναι 31 χαρακτήρες, ενώ στο BASIC είναι 2, το πρώτο γράμμα και το δεύτερο αριθμός).

Μερικά τυπικά παραδείγματα συμβολικών ονομάτων μεταβλητών για την γλώσσα BASIC είναι:

$B, K, M2, U6,$

ενώ στη γλώσσα FORTRAN είναι:

$ANNA, KOSTAST, TOTAL, SUM2$

Τέλος, στη γλώσσα PASCAL όλα τα συμβολικά ονόματα πρέπει να δηλώνουν στην αρχή του προγράμματος με την ένδειξη Var, όπως στο παραδειγμα που ακολουθεί, και τον προσδιορισμό τους εάν είναι ακέραιες ή πραγματικές μεταβλητές.

Var Algebra, Geometry : integer; x1, y1, z2 : real

Όμοια αν είχαμε να υπολογίσουμε το μέσο όρο, average, τεσσάρων τιμών Q, B, C και D που βρίσκονται στις θέσεις μνήμης 101,102,103 και 104, για μέν το FORTRAN θα μπορούσαμε να γράψουμε απλάως την εντολή:

γραφών των προγραμμάτων.

Στην προηγούμενη παράγραφο περιγράψαμε τον άριστο τρόπο με τον οποίο οι Γλώσσες Υψηλού Επιπέδου (Γ.Υ.Ε.) μας βοηθούν στην

3. Πολλαπλή Επεργείων Ένταξων Γ.Υ.Ε.

για να πάρουμε το ζητούμενο μέσο όρο.

```

LDA A 101 (Τοποθέτηση του  $\Xi$  στο σωσμένο Α)
ADD A 102 (Πρόσθεση του Β στο σωσμένο Α)
ADD A 103 (Πρόσθεση του C στο σωσμένο Α)
ADD A 104 (Πρόσθεση του D στο σωσμένο Α)
DIV A #4 (Διαίρεση δια του 4 του σωσμένου Α)
STA A 105 (Αποθήκευση του μέσου όρου στη θέση 105)

```

ένταξης (τιμή προγράμματος):

Ενώ σε συμβολική γλώσσα θα έπρεπε να γράψουμε τις ακόλουθες

$$Average = (X + B + C + D)/4$$

όνομα x_1 .

και έχει διαβάσει τους 3 συντελεστές της εξίσωσης και τους έχει αποθηκεύσει στις θέσεις μνήμης A, B και C) και το αποτέλεσμα που θα βρει θα το αποθηκεύσει στη θέση μνήμης με το συμβολικό

*READ *, A, B, C*

όπου στην (4), εννοείται ότι η μηχανή θα εκτελέσει τις πράξεις στο βέλος με τις τιμές των συντελεστών, που έχουν αποθηκευθεί στις θέσεις μνήμης με τα συμβολικά ονόματα A, B και C (πράγμα που το γνωρίζει η μηχανή, διότι προτού φθάσει να εκτελέσει την εντολή (4), έχει εκτελέσει μια εντολή της μορφής:

$$(4) \quad x_1 = (-B + (B * B - 4 * A * C) * 0.5) / (2 * A)$$

θμιας εξίσωσης να γράφουμε:

Ιδιαιτέρα στην γράφη των μαθηματικών εκφράσεων και στις εκχώρη-
σεις τιμών, αρκεί λέγαμε στην περίπτωση μιας μιας δευτεροβά-

εντολές της γλώσσας μηχανής είναι βασικό στις Γ.Υ.Ε. τηροτικό της ανάλησης μιας εντολής της Γ.Υ.Ε. σε πολές -**το χάρτα**- ότι το τέλικό άθροισμα το λαμβάνει στη θέση 52. Αυτό **το χάρτα** σε τις τιμές των ψ και Z στις θέσεις 50 και 51, και θα γωρίζε -όπου, φυσικά, έχει φροντίσει ο προγραμματιστής να έχει αποθηκεύ-

LDA A 50 Πρόσθεσε τον ψ στο σσ. A	STA A 52 Αποθήκευσε το άθροισμα στη θέση 52
ADD A 51 Πρόσθεσε τον Z στο σσ.	A ADD A #31 Πρόσθεσε τον 31 στο σσ. A

Είναι ενδιαφέρον, να θυμηθεί κανείς τις εντολές σε συμβολική γλώσσ-
 σα που θα προκαλέσουν την ίδια επεξεργασία όπως η (5):

$$(5) \quad X = Y + Z + 31$$

φής:

Με όμοιο τρόπο, τέλος, θα αντιμετωπίσει και μια εντολή της μορ-

4. Ο χειρισμός των διανυσμάτων (Array handling) και γεικότερα
 α πινάκων. Στην γλώσσα μηχανής για τη χρήση πινάκων και διανυσ-
 μάτων αναγκαστήκαμε να κάνουμε την διεκτιμώμενη αναφορά των
 θέσεων μνήμης, όπως της: **STA A 19B**, στο πρόγραμμα της εύρεση-
 ς του M.K.Δ., που είναι μια αρκετά απαιτητική χρήση. Απεναν-
 τίας σε μια Γ.Γ.Ε. μπορούμε να έχουμε μια άνετη διανυσματική ή
 πινακική (με πολλούς δείκτες - ανάλογα των αναγκών μας) χρήση
 μεταβλητών, όπως ακριβώς στα μηχανικά μπορεί να γωρσούμε
 στον γομπύερ την παροσία αυτής της μεταβλητής με μια εντολή
 τύπου **DIMENSION**, όπως κάναμε στο πρόγραμμα του πρόβληματος
 N αριθμώ, με μια επίστηαση: οι δείκτες της μεταβλητής
 γράφονται στο κάτω μέρος της, αλλά δίπλα της και περιέχον-
 τα εντός παρενθέσων. Σε αυτές τις περιπτώσεις ο δείκτηρα-
 ωτής, φυλάει στη μνήμη του Η/Υ τις θέσεις που καθορίζε η εντολή
 DIMENSION π.χ. στο παράδειγμα που προαναφέραμε 1000 θέσεις
 και τις αναγνωρίζει ως A(1), A(2), ... , A(1000) - στο αντίστοιχο
 όνομα της μεταβλητής με δείκτη.

όπου όπως προηγουμένως είπατε να ελιχάμε φροντίσει να αποθηκεύ-
σουμε τους 1000 αριθμούς A(1000) στις θέσεις 80 έως 1679 το
αριθμια τους στη θέση 50 και τον δείκτη I στη θέση 51.

STA A 50	Τοποθέτησε το SUM στη θέση B
LDA B 51	Αποθήκευσε τον δείκτη I στον B
ADD A 79B	Πρόσθεσε το A(I) στον A
STA A 50	Αποθήκευσε το αριθμια στη θέση 50

αμέσως:

Για να γίνει εντερπότερη η τεραστια η ατλαοποίηση της χρήσης των δι-
ανυσμάτων στην FORTRAN, αρκεί να δει κανείς πως μια εντολή του
παραπάνω προγράμματος που αφορίζει τις τιμές: **SUM=SUM+A(I)**
θα έπρεπε να αποδοθεί στη γλώσσα μηχανής, που δίδεται εντός

μηδέν. Το πρόγραμμα ακολουθεί:

των παραμέτρων, το φασματικό μέτρο της πίεσης θα είναι ίσον με οι 2 πίεες με τη γενική μετρήσιμη τους μορφή που στην περίπτωση της οποίας οι συντελεστές διαβάζονται και στο τέλος εκτυπώνονται

$$Ax^2 + Bx + c = 0,$$

δεντροβάθμιας εξίσωσης:

έκφραση. Στο παρακάτω πρόγραμμα υπολογίζονται οι 2 πίεες μιας η εντολή IF είναι τόσο ευέλικτη που μπορεί με απλή αλγεβρική γλώσσες ψηφίου επιπέδου με βέλτιστο τρόπο - στο FORTRAN, π.χ. «πρωτόγονος» και επιρροής σε σφάλματα. Αυτό εξεπνείται στις εντολών branch (διακλάδωσης): BMI, BPL, BZE που είναι αρκετά χική εκτέλεση των εντολών του προγράμματος είναι δια μέσου των Στη γλώσσα μηχανής ο τρόπος με τον οποίο αλλάζουμε την διαδο-

4. Αλλαγή της ποής του προγράμματος (sequence control)

```

PROGRAM QUADRATIC_EQUATION
  READ *, A, B, C
  IF (A /= 0) THEN
    D = B*B - 4*A*C
    R1 = (-B - D**0.5)/(2*A)
    R2 = (-B + D**0.5)/(2*A)
    PRINT *, "REAL ROOTS", R1, R2
  ELSE IF (B /= 0) THEN
    R = - C/B
    PRINT *, "LINEAR EQUATION WITH THE ROOT R=", R
  ELSE IF (C /= 0) THEN
    PRINT *, "THERE IS NO EQUATION, AND, OF COURSE, THERE IS NO ROOT"
  ELSE
    PRINT *, "THE GIVEN EQUATION IS AN IDENTITY", A, B, C
  ENDIF
ENDDO
END PROGRAM QUADRATIC_EQUATION

```


εξισώσεως που επιλύεται.

θέση και άρα μπορεί να εκτυπωθεί και να δώσει την σειρά της παράλληλα σε κάθε κύκλο του DO η τιμή του δείκτη I είναι δια- και στη θέση της ετολής 100 STOP την ετολή 100 END DO ενώ

DO I=1,1000

θα ακουσε προ της ετολής READ να βάζαμε την ετολή: μένο πρόγραμμα εάν θέλαμε να έχουμε την λυση 1000 εξισώσεων σαφείς, άντεξ και με πληθώρα δυνατοτήτων π.χ. στο προηγούμε- Οι επαναληπτικές διαδικασίες στις γλώσσες υψηλού επιπέδου είναι

5. Επαναληπτικές διαδικασίες (ετολή DO)

Θεωρούμε \mathbb{R}^n ως χώρο διανυσμάτων. Έστω $A \in \mathbb{R}^{n \times n}$ και $\beta \in \mathbb{R}^n$. Η εξίσωση $Ax = \beta$ ονομάζεται γραμμικό πρόβλημα. Η λύση του προβλήματος υπάρχει αν και μόνο αν $\beta \in \text{Im}(A)$.

$$Ax = \beta \quad (\Sigma)$$

Η λύση του προβλήματος (Σ) είναι μοναδική αν και μόνο αν A είναι αντιστρέψιμη.

Η αντιστρεψιμότητα της A εξαρτάται από το αν ο $\det(A) \neq 0$. Αν $\det(A) = 0$, τότε η A δεν είναι αντιστρέψιμη και το πρόβλημα (Σ) μπορεί να μην έχει λύση ή να έχει άπειρες λύσεις.

6. Δομή Προγράμματος (Program Structure)

(I) CALL SIMQ (A, B, N, IND)
 CALL ROOTCP (A, N, E, MAX, X, J, \$L).

επιτολές κλήσεως (Calling sequence):

TA) που έχουν τα ονόματα SIMQ και ROOTCP με τις αντίστοιχες
 σελ. 216 του βιβλίου σας εφαρμόζεις υποπρογράμματα - PAKET-
 γράμματα που αντιμετωπίζουν τα αντίστοιχα προβλήματα (βλέπουμε
 MATHPACK της Univac.) να αναζητήσουμε και να βρούμε τα υποπρο-
 από την βιβλιοθήκη των επιστημονικών υποπρογραμμάτων (π.χ. το
 της (E) στο παράδειγμα επάνω). Στην περίπτωση αυτή, μπορούμε
 της, ενώ με $x_1, x_2, x_3, \dots, x_n$ θα συσχετίσουμε τις N ακριβώς πλές,
 (όπου: οι γινόμενοι συντελεστές της εξίσωσης και ν ο βαθμός

$$(E) \quad C_0 x^n + C_1 x^{n-1} + \dots + C_1 x + C_0$$

Ετσι, εάν στο πρόγραµµά µας στη γλώσσα FORTRAN συµπεριλάβουµε στο αντίστοιχο σηµείο του προγράµµατος µας τις εντοχές (I) και έχουµε φροντίσει προηγουµένως το πρόγραµµά µας να έχει από- θηκεύσει τα απαραίτητα δεδοµένα, και τον Η/Υ να έχει την αντί- τοιχη LIBRARY διαθεσίµη, τότε µε τις εντοχές (I), ο compiler της γλώσσας θα αναζητήσει από την LIBRARY τα αντίστοιχα υποπρο- γράµµατα και θα τα ενσωµατώσει κατάλληλα στο πρόγραµµά µας, µε συνέπεια όταν κληθούν κατά την εκτέλεση του προγράµµατος τότε αυτόµατα ο Η/Υ διαβάζει τα αντίστοιχα δεδοµένα στις υποµονάδες - π.χ. για το σύστηµα την διάσταση του τετραγωνικού πίνακα N, και τα στοιχεία του πίνακα των συντελεστών A και τα στοιχεία του σταθµού διαλύµατος B - οπότε οι υπορουτίνες όταν ολοκληρώσουν την επεξεργασία επιτυχώς επιστρέφουν στο πρό- γράµµα τα αποτελέσµατα που έχουν υπολογίσει - π.χ., για το σύστη- µα τη λύση του ίδιου µέγους µνήµης B που είχε καταρχήν κατα- γραφεί το σταθµό διαλύσιµα, µε τον τρόπο που είναι κατασκευασ-

·531371

Τέλος, για εκτέλεση και ποσό βοηθητική περιγραφή των βασικών στοιχείων του διεκπεραιωτή της FORTRAN μπορείτε να βρείτε στο βιβλίο σας «ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ», στις σελίδες 143-213.

Στις γλώσσες υψηλού επιπέδου η εκσφαλμάτωση των προγραμμάτων γίνεται με απλό τρόπο ενώ παραλλήλα υπάρχουν και εργαλεία που υποβοηθούν, πράγμα που είναι αδύνατο στις γλώσσες μηχανής.

8. Εκσφαλμάτωση (Error detection)

δομημένων (πληκτρολόγιο, δίσκους, διακείμες κλπ).

από τις πιο πρωτόγονες ως τις πιο εξελιγμένες μεθοδούς ελαττώσης. Στις γλώσσες υψηλού επιπέδου υπάρχουν δυνατότητες που ελέγχουν, ελέγχουν και ελέγχουν.

7. Είσοδος δεδομένων και έξοδος αποτελεσμάτων (Input/Output)

Από τα βασικά είναι αυτόνομο ή τερματίζοντας τη λειτουργία του προ-γραμμάτιοι στις γλώσσες υψηλού επιπέδου.

τους αρχικούς - περιφερειακούς.

αλλάξουν προγράμματα, προγράμματα και άλλες
του συστήματος (π.χ. εκτυπώσεις, σκληρούς δίσκους κλπ.) στις
την εκχώρηση και διαθεσιμότητα των περιφερειακών μονάδων
(1). **Έλεγχος των Περιφερειακών - (Peripheral control):** δηλ.

γρήγορα που περιληπτικά φροντίζει για:

διακοπή των χρηστών είναι ένα μέτρο και συνήθως πρό-
« του συστήματος με στόχο **την αποδοτικότητα του και την**
(π.χ., τα windows, το NOVEL, το UNIX) εκτελεί όλη τη γραφειοκρατι-
γρήγορα - χρήση του Η.Υ. από την μηχανή (το H/W). Το Α.Σ.
- (που αποσκοπεί στην απελευθέρωση και απομόνωση του προ-
Γενικά, το Α.Σ. είναι το τμήμα (the «layer») του λογισμικού (συστή-

6. **ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (Α.Σ.) - Operating Systems**

(3). Επίβλεψη Ορθής Εκτέλεσης Διαδοχικών Εργασιών - (Job control): στα μεγάλα συστήματα έτσι οι διαδοχικές εργασίες - φάσεις που απαιτούν τα προγράμματα, να εκτελούνται ακώλυτα. Π.χ. μπορεί κάποιο πρόγραμμα που είναι γραμμένο σε FORTRAN να ζητεί την χρήση υπολογισμάτων, ή ακόμη και Βιβλιοθήκης εισηγμένων υπορουτινών.

(2). Δυνατότητα Πολλαπλών Προσβάσεων - (Provision of multi-access): για τα μεγάλα συστήματα που έχουν την δυνατότητα αυτόχρονης εξυπηρέτησης πολών χρηστών την ίδια χρονική στιγμή, όπου ο καθένας με το προσωπικό του πληκτρολόγιο και οθόνη (εvidence και κτυπιωτή) μπορεί να κάνει χρήση του συστήματος έχοντας την εντύπωση ότι είναι ο μοναδικός χρήστης. Σ' αυτές τις περιπτώσεις, το A.S. φροντίζει να εκχωρεί κατάλληλα τις διάφορες μονάδες του (όπως την CPU, την Memory κλπ.) στους χρήστες, χωρίς κανένα πρόβλημα.

description), που υπέβαλε στο σύστημα.

και το έχει καταργήσει στην υπηρεσία (Job

και ο υπάλληλος που ονομάζεται και υπηρεσία

και τα αποτεύμενα σταδιακά και η υπηρεσία

control section), καθόλη την διάρκεια του

του V.Σ. που είναι αρμόδιο για τον έλεγχο των εργασιών (Job

«φορέσει» στο σύστημα για επεξεργασία. Δηλαδή, το σύστημα

αυτού του συνολικού προγράμματος σε γλώσσα μηχανής, να το

ζητηθέντα υποπρογράμματα και υπορουτίνες και στη συνέχεια,

program) σε γλώσσα μηχανής, την κατάλληλη σύνδεση του με τα

Compiler για την μετάφραση του πηγαίου προγράμματος (Source

Σ αυτή την περίπτωση το V.Σ. φροντίζει την διατήρηση του κατάλληλου

σίων του, λόγω της μεγάλης σπουδαιότητας των εργασιών που εκτελούνται για την αναστήματα και την αποκατάσταση του περιβάλλοντος. Τέλος, εάν υπάρξει διαθεσίμος χρόνος θα επανέλθουμε στα Α.Σ. που έχουν χρησιμοποιήσει τον χώρο που κατέλαβαν, καθώς και τον χώρο που έχουν καταλάβει.

Μεταξύ των διαθεσίμων χώρων που έχουν χρησιμοποιήσει για την αποκατάσταση των διαθεσίμων χώρων, υπάρχουν χώροι που έχουν χρησιμοποιήσει για την αποκατάσταση των διαθεσίμων χώρων. (5). **Εργασίες Νομιστηρίου - (Accounting)**: έτσι ώστε ο κάθε

των του. Α.Σ., μπορεί ο χρήστης να επιλέξει σε οποιοδήποτε χρόνο αρχίσει τον έλεγχο, που πάντοτε διατίθεται από το file control system του προγράμματος, ενώ με την βοήθεια ενός από τους χρησιμοποιούμενους χώρους για αποθήκευση ή επεξεργασία των αρχείων (όπως, π.χ., είναι οι σκληροί δίσκοι) να ελεγχθούν τα αρχεία. (4). **Ελέγχο Αρχείων - (File control)**: έτσι ώστε τα βοηθητικά

της IMSL).

PACK και STAT-PACK της UNIVAC, καθώς και την βιβλιοθήκη εφαρμογών στο βιβλίο σας σελ. 257, βάλτε σχετικά τα MATH-**Μαθηματικά** (σχεδόν για όλα τα προβλήματα των γλωσσών ε-

βιβλιοθήκες αυτές είναι κυρίως στις ακόλουθες περιοχές:

έχει **μία απλοποιημένη χρήση για ένα σύνθετο πρόβλημα**. Οι μετρήσιμα απλή-αλλά κατάλληλη-επικύρωση αυτών μπορεί κανείς να μεταφραστούν που διεκονούν αφάνταστα τους χρήστες, αφού στο διαθεσίμο λογισμικό τους διαφόρες βιβλιοθήκες με ποικίλα-Τα περισσότερα μετρήσιμα υπολογιστικά συστήματα έχουν συνήθως

3. ΒΙΒΛΙΟΘΗΚΕΣ (LIBRARIES)

δηματικών.

Τέλος, στα WINDOWS και στην Compaq Visual Fortran 6, μαζί με τον compiler διατίθεται και η πιοσημονική βιβλιοθήκη της IMSL, που τα εγχειρίδια της υπάρχουν στο εργαστήριο του Τμήματος Μα-

(4) **Γλώσσες** (για έλεγχο γραφής κλπ.).

(3) **Γραφικά** (για την ψηφιακή δημιουργία εικόνων κλπ.).

πολυμεταπισμός κλπ.).

(2) **Επιχειρησιακή Έρευνα** (Ανάυση Κρίσιμου Δρόμου, Αξέπαιος

συγχευσεις κλπ.).

(1) **Επιτόριο** (για κατάλληλη επεξεργασία αρχείων - κατατάξεις /