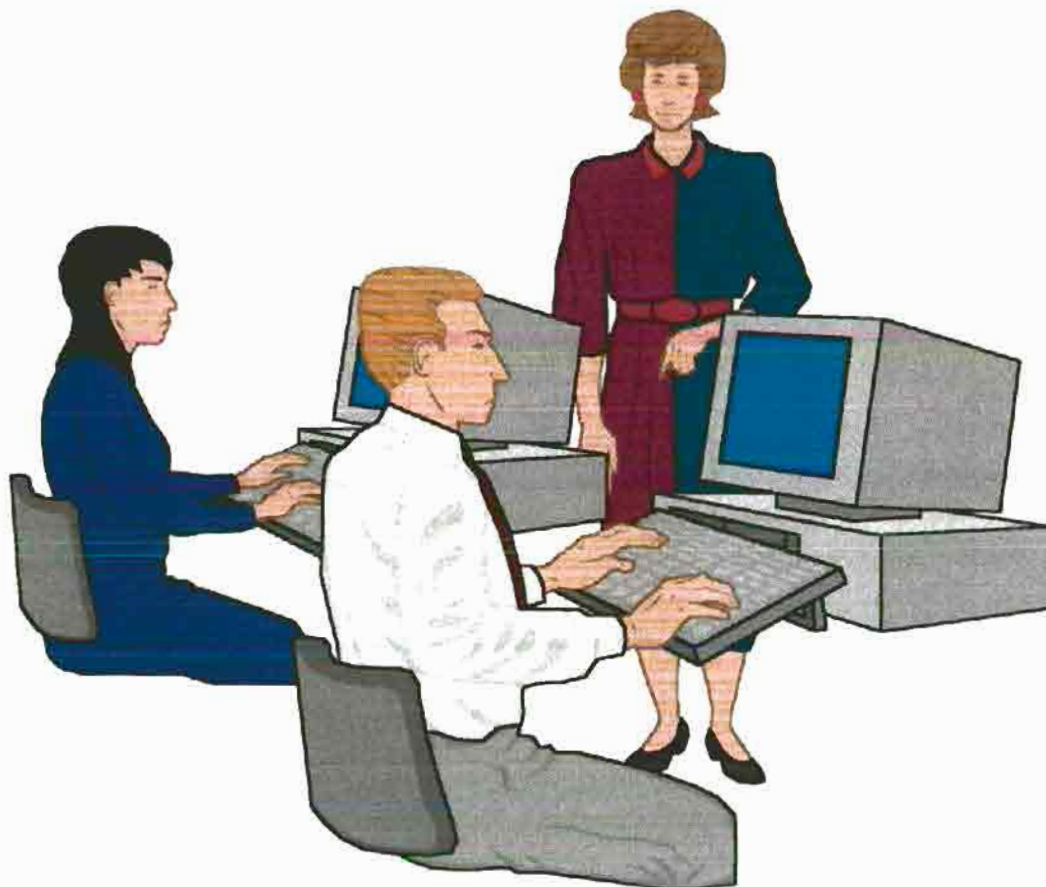


ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Πρόγραμμα τιμολόγησης προϊόντων και
σύνδεση του με το πρόγραμμα παραγωγής
(Γλώσσα Clipper)



Εισηγητής:

Γεώργιος Ραβασόπουλος

Σπουδάστριες:

Κυριακοπούλου Χριστίνα

Μπαλτούμα Μαρία

Πατσιά Δήμητρα

Πάτρα 1997



ΤΟΜΟΣ
ΕΙΣΑΓΩΓΗΣ

2741

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος

4

Εισαγωγή

Βασικές Έννοιες για τον Ηλεκτρονικό Υπολογιστή

5

Μέρος Α

Κεφάλαιο 1ο

Εξοπλισμός --Hardware

11

1.1	Κεντρική Μονάδα	11
1.1.1	Η μνήμη του Ηλεκτρονικού Υπολογιστή	11
	Πως επικοινωνεί ο άνθρωπος με τον υπολογιστή.....	
	Τι μπορούμε να βάλουμε στη μνήμη και πώς	
	Χαρακτηριστικά στοιχεία της μνήμης	
	Είδη μνήμης Chip Ram -Rom	
1.1.2	Ο Επεξεργαστής και πως λειτουργεί	17
1.1.3	Επικοινωνία επεξεργαστή και μνήμης	19
	Τύποι επικοινωνίας	
1.2	INPUT/OUTPUT-ΜΟΝΑΔΕΣ I/O	22
1.2.1	Μονάδες εισόδου στοιχείων -INPUT.....	22
	Πληκτρολόγιο	
	Ποντίκι	
	Scanner	
1.2.2	Μονάδες εξόδου στοιχείων-OUTPUT	23
	Εκτυπωτές-Έξοδος σε άμεσα αναγνώσιμη μορφή	
	Εξοδος σε μη αναγνώσιμη μορφή	
1.2.3	Μονάδες εισόδου-εξόδου Στοιχείων-I/O	27
	Σκληροί δίσκοι	
	Δισκέτες	

Κεφάλαιο 2ο

Προγράμματα - Software

32

- 2.1 Προγράμματα Υπολογιστών 33
- 2.2 Πρόγραμμα και Προγραμματισμός 33
 - 2.2.1 Ειδικά Προγράμματα ανάπτυξης εφαρμογών 36

Κεφάλαιο 3ο

Συστήματα πολλών υπολογιστών

38

- Γενικά για τα Δίκτυα υπολογιστών 38
- 3.1 Ανάλυση χαρακτηριστικών 41
 - 3.1.1 Τοπολογίες Δικτύων 41
 - 3.1.2 Ιεραρχία Ελέγχου και ροής Δεδομένων 42
 - 3.1.3 Περιοχή κάλυψης του Δικτύου 43
 - 3.1.4 Ταξινόμηση ως προς την κυριότητα 44
- 3.2 Αναφορά στα μέρη του δικτύου 45
 - 3.2.1 Μονάδες εξυπηρέτησης του δικτύου 45
 - 3.2.2 Δίαυλοι Επικοινωνίας 45
 - 3.2.3 Ενδιάμεσες μονάδες Επικοινωνίας 45
 - 3.2.4 Λογισμικό Επικοινωνιών 46
 - 3.2.5 Τερματικά 46
- 3.3 Περιήγηση Novell Netware 47
 - 3.3.1 Δομικά Στοιχεία 47

ΜΕΡΟΣ Β

Κεφάλαιο 4

Ανάλυση - Σχεδιασμός του Συστήματος Τιμολόγησης

53

- 4.1 Ανάλυση προδιαγραφών Συστήματος 59
- 4.2 Σχεδιασμός Συστήματος Τιμολόγησης 60
- 4.3 Υλοποίηση Συστήματος Τιμολόγησης 71
- 4.4 Ανάλυση εντολών προγραμματισμού 86

Παράρτημα

Βάσεις δεδομένων

Κώδικας PRG

Εκτυπώσεις

Βιβλιογραφία

Πρόλογος

Το βιβλίο που κρατάτε στα χέρια σας αναλύει με απλά λόγια την δημιουργία ενός Πληροφοριακού Συστήματος Τιμολόγησης για την μηχανογραφική έκδοση παραστατικών . Επίσης περιλαμβάνει λίγα λόγια για το Hardware και Software του υπολογιστή . Η όλη εργασία μας χωρίζεται σε δύο τμήματα . Στο πρώτο τμήμα ασχοληθήκαμε με την καθεαυτού δημιουργία του προγράμματος στον υπολογιστή (σχεδιασμός και ανάλυση των επιθυμητών λειτουργιών του προγράμματος και καταγραφή του κώδικα) . Στο δεύτερο τμήμα ασχοληθήκαμε με τη δημιουργία των σημειώσεων και του εγχειριδίου χρήσης του προγράμματος με σκοπό να δώσουμε στους αναγνώστες του να κατανοήσουν τον τρόπο σκέψης μας και τα βήματα που ακολουθήσαμε ώστε να ολοκληρωθεί αυτή η εργασία επιτυχώς .

Ελπίζουμε ότι μέσω των σημειώσεων αυτών σας δίνουμε την δυνατότητα να καταλάβετε τη διαδικασία δημιουργίας ενός πληροφοριακού συστήματος . Μπορούμε να πούμε ότι οι δυσκολίες που αντιμετωπίσαμε ήταν πολλές , αλλά τα οφέλη που αποκομίσαμε ήταν πολύ περισσότερα .

Στο σημείο αυτό αισθανόμαστε την ανάγκη να ευχαριστήσουμε όλους όσους μας βοήθησαν στην επίτευξη του στόχου μας και κυρίως τον εισηγητή μας κ. Γεώργιο Ραβασόπουλο για την βοήθεια και τις γνώσεις που μας πρόσφερε . Η συμβολή του για την ολοκλήρωση αυτής της εργασίας ήταν σημαντική .

Εισαγωγή

Σκοπός του βιβλίου αυτού είναι να ερμηνεύσει με απλά λόγια και παραδείγματα την διαδικασία δημιουργίας ενός πληροφοριακού συστήματος .

Στην αρχή θεωρήσαμε σκόπιμο να αναλύσουμε τα βασικά μέρη του ηλεκτρονικού Υπολογιστή (Hardware) και να αναφερθούμε στα δίκτυα , έτσι ώστε οι αναγνώστες του βιβλίου αυτού να αποκτήσουν μια ολοκληρωμένη εικόνα για τον Η/Υ και στην συνέχεια να περάσουμε στον προγραμματισμό . Με άλλα λόγια το βιβλίο αυτό χωρίζεται σε δύο βασικά μέρη : Το πρώτο που ασχολείται με το σύστημα του Η/Υ και το δεύτερο με τον προγραμματισμό .Το πρώτο μέρος ίσως θεωρηθεί κουραστικό για κάποιους οι οποίοι γνωρίζουν τον Η/Υ . Για εκείνους όμως που έρχονται για πρώτη φορά σε επαφή , αυτό είναι απαραίτητο . Όπως όμως και να έχει το πράγμα , σας προτείνουμε να μην μπείτε κατευθείαν "στο ψητό" αλλά να διαβάσετε πρὶν τις πληροφορίες που αναφέρονται στο σύστημα του Η/Υ γιατί είναι χρήσιμες και σύντομες .

Ο Άνθρωπος και ο υπολογιστής

Από πολύ παλιά ο άνθρωπος άρχισε να ασχολείται με μεγάλες και πολύπλοκες εργασίες οι οποίες απαιτούσαν πολύ σκέψη , ταχύτητα αλλά και πολύ χρόνο για να γίνουν . Για τη διεκπαιρέωση αυτών των εργασιών χρησιμοποιούσε τις πνευματικές του ικανότητες καθώς και διάφορα βοηθητικά μηχανήματα . Όσο περνούσε ο καιρός και τα πράγματα εξελισσόταν όλο και περισσότερο , ανακάλυψε ένα ακόμα μηχανήμα τον **Ηλεκτρονικό Υπολογιστή (Η/Υ)**. Στις ημέρες μας ο Η/Υ έχει μπει για τα καλά στην ζωή μας και σχεδόν τα πάντα γίνονται με αυτόν. Οπότε και εμείς πρέπει να προσαρμοστούμε σε αυτόν να τον γνωρίσουμε από κοντά και να τον αγαπήσουμε.

Ο Ηλεκτρονικός Υπολογιστής και η αρχιτεκτονική του

Ο υπολογιστής προορίζεται να βοηθήσει τον άνθρωπο στις διανοητικές του λειτουργίες , είναι δηλαδή ένα μηχανικό βοήθημα διανοητικής λειτουργίας ,όπως υπάρχουν μηχανικά βοηθήματα όρασης , ακοής κλπ. Τον βοηθάει στο να φέρει σε πέρας μία δουλειά διανοητική : διεκπεραίωση μιας διαδικασίας , λήψη μιας απόφασης κλπ. Για να κάνει ο Η/Υ αυτή την δουλειά θα χρειαστεί μηχανήματα εισαγωγής στοιχείων , επεξεργασίας , εξόδου αποτελεσμάτων ή με άλλα λόγια μηχανήματα επικοινωνίας με τον έξω κόσμο (είσοδος-έξοδος) , εσωτερικής επεξεργασίας καθώς και μηχανήματα αρχείων .

Τα μηχανήματα **εισόδου** έχουν περάσει πολλές αλλαγές από τότε που πρωτοπαρουσιάστηκε ο Η/Υ και ασφαλώς θα περάσουν και άλλες . Την στιγμή αυτή υπάρχουν αρκετά μηχανήματα εισόδου αλλά το πιο διαδεδομένο είναι το λεγόμενο **τερματικό** εφοδιασμένο με **οθόνη** και **πληκτρολόγιο**. Το καθαρά μηχανήματα εισόδου αν θέλουμε να κυριολεκτήσουμε είναι το πληκτρολόγιο με το οποίο ο άνθρωπος πληκτρολογεί ό,τι θέλει να βάλει μέσα στον Η/Υ (προγράμματα , στοιχεία). Το πληκτρολόγιο είναι συνδεδεμένο με τον κυρίως Η/Υ και έτσι τα στοιχεία περνάνε κατευθείαν στο εσωτερικό του Η/Υ για να τα επεξεργαστεί .

Το εσωτερικό αυτό του Η/Υ , τα μηχανήματα δηλαδή **επεξεργασίας** (το μυαλό) του Η/Υ ονομάζονται **κεντρική μονάδα(K/M)** . Η K/M αποτελείται από τη λεγόμενη **Μνήμη** (memory) όπου αποθηκεύονται τα προγράμματα και τα στοιχεία που πρόκειται να επεξεργαστούν και τον **επεξεργαστή** (Processor) ή κεντρική μονάδα επεξεργασίας -**KME**(Central Processing Unit -**CPU**) . Ο επεξεργαστής αποτελείται από την λεγόμενη μονάδα ελέγχου-**ME**(Control Unit) που έχει την αποστολή να παίρνει τις εντολές του προγράμματος από τη μνήμη μία-μία και να τις εκτελεί . Τέλος έχουμε την αριθμητική μονάδα -**AM** (Arithmetic Unit) που αναλαμβάνει να κάνει τις αριθμητικές πράξεις που ζητάνε οι εντολές .

Σαν μηχανήματα **εξόδου** θεωρούμε εκείνα τα μηχανήματα που επιτρέπουν στον Η/Υ να μας παρουσιάζει τα αποτελέσματα της επεξεργασίας των στοιχείων που έκανε στην Κεντρική Μονάδα . Τέτοια μηχανήματα είναι ο γνωστός **εκτυπωτής** όπως επίσης και η **οθόνη** του **τερματικού** .

Σαν μηχανήματα **αρχείων** για την αποθήκευση μεγάλων όγκων δεδομένων χρησιμοποιούνται οι μαγνητικοί **δίσκοι** και οι μαγνητικές **ταινίες** . Κάτι ανάλογο των δίσκων του πικ-απ και των ταινιών του μαγνητοφώνου ή του κασετοφώνου αντίστοιχα .

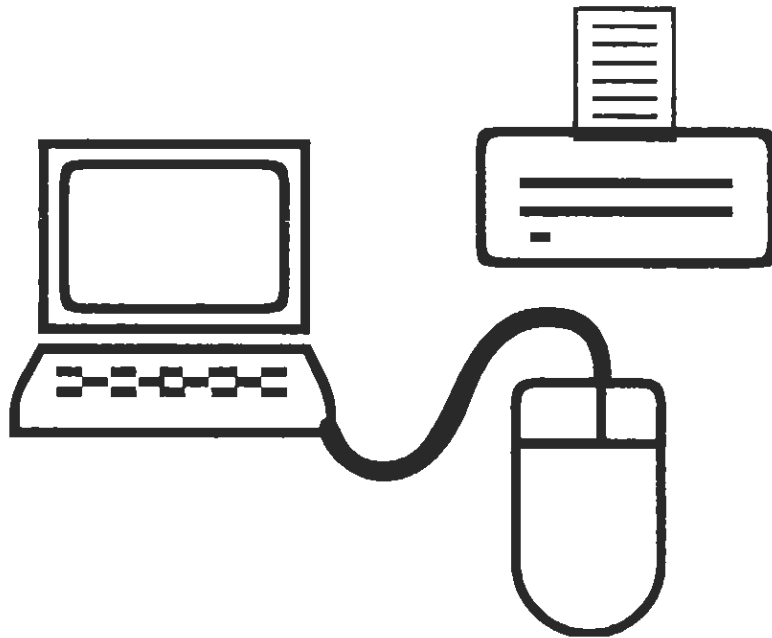
Καταλήγοντας μπορούμε να πούμε ότι κάθε σύστημα υπολογιστή από το πιο μεγάλο μέχρι το πιο μικρό αποτελείται αναγκαστικά από μηχανήματα εισόδου , επεξεργασίας , εξόδου και (αν χρειάζεται) αρχείων . Εκείνο που διαφέρει και χαρακτηρίζει τα μεγάλα και τα μικρά συστήματα υπολογιστή είναι η πολυπλοκότητα , ο όγκος και ο αριθμός μηχανημάτων από την κάθε κατηγορία , που αποτελούν το σύστημα του υπολογιστή . Το σύνολο των μηχανημάτων αυτών λέγεται με μια λέξη εξοπλισμός ή υλικό του συστήματος του Η/Υ . Ακόμα συναντάται συχνά με τον αγγλικό - διεθνή όρο **hardware** (χάρντγουερ) που υποδηλώνει στα Αγγλικά το σκληρό (hard) , το σιδερικό .

Σε αντιδιαστολή τα προγράμματα ονομάστηκαν **software** (σόφτγουερ) λόγω της "μαλακής" -soft φύσης τους (μπορούμε εύκολα να τα αλλάξουμε όταν αλλάξουμε κάποιες εντολές τους) . Ελληνικά αποδόθηκαν σαν λογικό ή λογισμικό .

Στην αρχή οι ξεχωριστές αυτές μονάδες τοποθετούνταν σε ξεχωριστά κουτιά που συνδέονταν μεταξύ τους με ειδικά μηχανήματα διασύνδεσης και επικοινωνίας . Η νέα τεχνολογία όμως κατασκευής Η/Υ με τα μικροκυκλώματα υψηλής πυκνότητας που οδήγησαν στην σμίκρυνση των διαστάσεων των διαφόρων μερών του Η/Υ έφερε και σημαντικές απλοποιήσεις . Η αρχή έγινε με τους μικρουπολογιστές . Εδώ οι διαστάσεις των διαφόρων μονάδων είναι πραγματικά μικρές και τα διάφορα μηχανήματα διασύνδεσης που προαναφέραμε αντικαταστάθηκαν σε ένα μεγάλο ποσοστό από ένα αγωγό επικοινωνίας ή διάδρομο ή **bus** . Πάνω στο bus "κρεμιούνται" οι διάφορες μονάδες του Η/Υ . Όταν κάποια από τις μονάδες αυτές θέλει να στείλει στοιχεία σε άλλη τα τοποθετεί σε bus μαζί με την διεύθυνση προορισμού τους . Η χρησιμοποίηση αυτού του αγωγού επικοινωνίας έφερε μια σημαντική απλοποίηση και τυποποίηση στην διασύνδεση των διαφόρων και λειτουργικά ανόμοιων μονάδων ενός Η/Υ . Η ιδέα έπιασε και άρχισε να εφαρμόζεται βελτιωμένη και επταυξημένη και στους μεγαλύτερους Η/Υ .

"Μέρος Α"

Κεφάλαιο 1ο
Εξοπλισμός-Hardware



1.1 Κεντρική Μονάδα

1.1.1 Η Μνήμη του Ηλεκτρονικού Υπολογιστή

Ένα βασικό κομμάτι του Η/Υ είναι η **μνήμη** . Αυτή κρατάει μέσα της όλες τις απαραίτητες πληροφορίες για το τι θέλουμε να επεξεργαστούμε (δεδομένα) και πως (πρόγραμμα) . Στην αρχή μπορούμε να θεωρήσουμε την μνήμη σαν μια μεγάλη ντουλάπα με ράφια . Κάθε ράφι το λέμε θέση **μνήμης** και μπορούμε να θεωρήσουμε πως σε κάθε τέτοια θέση μπαίνει και ένα στοιχείο που βάζουμε στην μνήμη , είτε δεδομένο είναι αυτό(αριθμός , λέξη) είτε εντολή προγράμματος . Η θέση μνήμης λέγεται στην μηχανογραφική -πληροφορική ορολογία και **λέξη** (αγγλικά word) . Οι θέσεις μνήμης αριθμούνται με αύξουσα αρίθμηση από την πρώτη μέχρι την τελευταία . Ο αύξων αριθμός κάθε θέσης μνήμης την προσδιορίζει μονοσήμαντα μέσα στην μνήμη και λέγεται **διεύθυνση** (address) της υπόψη θέσης ή λέξης της μνήμης . Οι διευθύνσεις βοηθάνε τον Η/Υ να κρατάει λογαριασμό για το που τοποθετείται το κάθε τι που εισέρχεται στην μνήμη . Οι θέσεις μνήμης είναι χωρισμένες σε **κυψελίδες** σε κάθε μια από τις οποίες τοποθετείται ένα ψηφίο ή ένας χαρακτήρας από ότι πληκτρολογεί ο άνθρωπος . Οπότε έπρεπε να βρεθεί το κατάλληλο υλικό που θα ήτανε φτιαγμένες αυτές οι κυψελίδες να μπορούν να κρατάνε ότι τοποθετείτε σε αυτές .

Για να βρούμε ένα υλικό για κάθε κατάσταση ήτανε πολύ δύσκολο . Έτσι ήτανε φανερό πως θα έπρεπε να αξιοποιηθεί η αρχή του **διακόπτη** που αναγνωρίζει δύο καταστάσεις (ανοικτός κλειστός) . Η κατασκευή λοιπόν κυψελίδων δύο καταστάσεων ή δυαδικών κυψελίδων (μέσα στις θέσεις μνήμης) θα απλοποιούσε τα πράγματα γιατί η πρώτη ύλη υπάρχει άφθονη στην φύση και οι δύο καταστάσεις που θα παράγουν οι κυψελίδες αυτές είναι ξεκάθαρες και εύκολο να ανιχνευτούν .

Ένα τέτοιο στοιχείο πολύ δημοφιλές τα τελευταία χρόνια στην κατασκευή Η/Υ είναι το **τρανζίστορ** . Το τρανζίστορ είναι ένας κρύσταλλος (πυρίτιο) που αν διοχετεύσεις ρεύμα κατά μία φορά το αφήνει να περάσει ενώ αν του το δώσεις από την άλλη φορά το διακόπτει . Βλέπουμε δηλαδή ότι ανάλογα με την φορά του διοχετευόμενου ρεύματος το τρανζίστορ είτε αφήνει το ρεύμα να περάσει (είναι αναμμένο) είτε όχι (είναι σβηστό) . Πράγματι αν "δούμε μια λέξη της μνήμης από πιο κοντά θα διαπιστώσουμε πως δεν είναι μονοκόμματη αλλά αποτελείται από ένα αριθμό τέτοιων τρανζίστορς από τα οποία άλλα είναι "αναμμένα" και άλλα "σβηστά" .

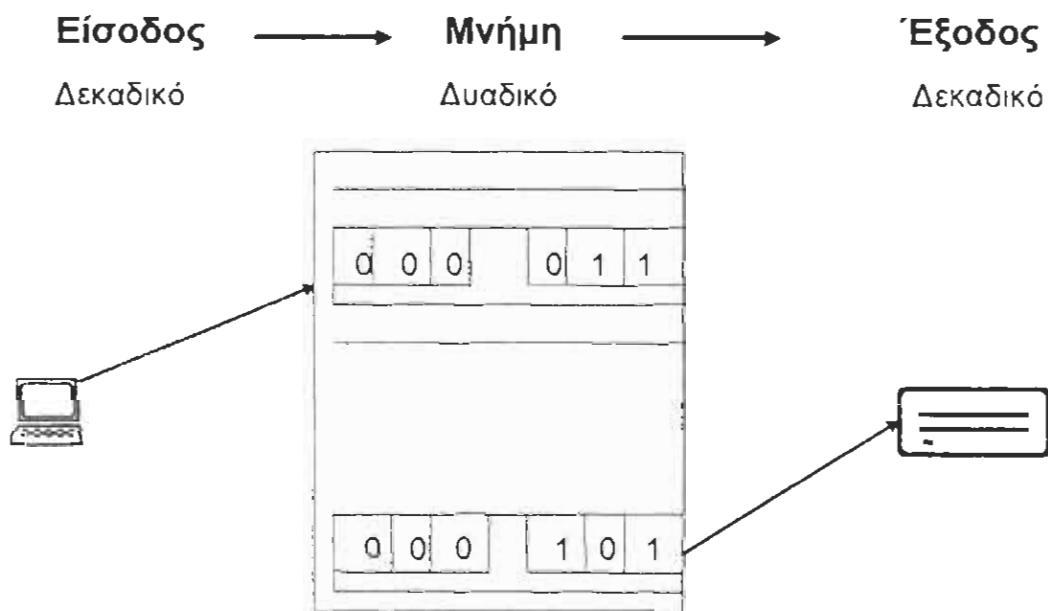
Πως επικοινωνεί ο άνθρωπος με τον υπολογιστή

Για να αντιστοιχίσουμε την φυσική αυτή κατάσταση των διακοπών - τρανζίστορς με τα στοιχεία του έξω κόσμου που θέλουμε να καταχωρήσουμε αντιστοιχούμε την "αναμμένη" κατάσταση του τρανζίστορ με τον αριθμό "1" και την σβηστή με το "0" . Δηλαδή από την έννοια του δυαδικού φυσικού στοιχείου οδηγούμαστε στην έννοια του αριθμητικού δυαδικού ψηφίου (0 ή 1) . Το δυαδικό αυτό ψηφίο και κατ'επέκταση και το αντίστοιχό του δυαδικό στοιχείο λέγεται διεθνώς **bit** .

Έτσι η αλληλουχία των "αναμμένων"και "σβηστών" τρανζίστορ μιας λέξης , αντιστοιχείται σε μια αλληλουχία από "1" και "0" bits ., πχ 1001110 . Αυτή η αλληλουχία ψηφίων είναι ένας αριθμός . Ο αριθμός αυτός λέγεται δυαδικός γιατί είναι εκφρασμένος σε μια αριθμητική γλώσσα που έχει δύο μόνο ψηφία το "0" και το "1" και λέγεται **δυαδικό σύστημα** . Το δυαδικό σύστημα είναι σύστημα που **"καταλαβαίνει" ο Η/Υ** , ενώ ο άνθρωπος καταλαβαίνει το δεκαδικό σύστημα . Οπότε η επικοινωνία του ανθρώπου και της μηχανής γίνεται μεταξύ του δυαδικού και δεκαδικού συστήματος . Ο άνθρωπος πληκτρολογεί είτε αριθμούς που ανήκουν στο δεκαδικό σύστημα είτε χαρακτήρες όπου κάθε ένας από αυτούς αντιστοιχεί σε έναν αριθμό του δεκαδικού συστήματος . Το

μηχάνημα παίρνει αυτά τα στοιχεία τα μεταφέρει στην μνήμη όπου τα μετατρέπει στο δυαδικό σύστημα μέσω κάποιου προγράμματος . Εκεί γίνεται η επεξεργασία τους και μετά η μετατροπή τους πάλι στο δεκαδικό σύστημα για να εξαχθούν τα αποτελέσματα στην οθόνη ή στον εκτυπωτή και να γίνουν κατανοητά από τον άνθρωπο .

Επικοινωνία Ανθρώπου - Μηχανής



Τι μπορούμε να βάλουμε στην Μνήμη και πώς

Στην μνήμη μπορούμε να καταχωρήσουμε αριθμητικά στοιχεία, χαρακτήρες-αλφαβητικά στοιχεία , γραφικές παραστάσεις , εικόνες , φωτογραφίες , άλλα δεδομένα όπως είναι οι μουσικές νότες , η κωδικοποίηση των οποίων επέτρεψε στον υπολογιστή να παίζει ολόκληρες συμφωνίες , προγράμματα . Όλα αυτά για να μπούν στην μνήμη πρέπει να μετατραπούν από την δική μας ανθρώπινη "μορφή" σε δυαδικό αριθμό .

Χαρακτηριστικά στοιχεία μνήμης

Μήκος λέξης . Είδαμε παραπάνω πως μια θέση μνήμης λέγεται λέξη της μνήμης και αποτελείται από τα bits . Ο αριθμός των bits που αποτελούν μια λέξη ποικίλει από Η/Υ σε Η/Υ και ονομάζεται **μήκος** της λέξης . Όσο πιο μεγάλο είναι το μήκος της λέξης τόσο πιο "δυνατός" θεωρείται ο Η/Υ . Ιδιαίτερη θέση στην ορολογία της μνήμης έχει το λεγόμενο **byte** . Ένας χαρακτήρας μπαίνει σε 8 bits για τον λόγο αυτό τα bytes λέγονται συχνά και χαρακτήρες .

Χωρητικότητα . Το πλήθος των λέξεων της μνήμης ονομάζεται **χωρητικότητα** της μνήμης . Η χωρητικότητα της μνήμης εκφράζεται σε χιλιάδες ή εκατομύρια λέξεις ή bytes .

Είδη της μνήμης - Chips RAM ΚΑΙ ROM

Σε ένα σύστημα Η/Υ συναντάμε διάφορα είδη μνήμης ανάλογα με τον σκοπό που εξυπηρετούν . Τα πιο βασικά είδη μνήμης είναι εκείνα που επιτρέπουν διάβασμα και γράψιμο και εκείνα που επιτρέπουν μόνο διάβασμα των περιεχομένων τους. Οι πιο γνωστές είναι οι λεγόμενες μνήμες RAM, ROM .

Από τις μνήμες αυτές εκείνη που ενδιαφέρει άμεσα τον χρήστη είναι η RAM , οι άλλες αναφέρονται περισσότερο στην λειτουργία του συστήματος .

Κύρια μνήμη ή RAM - Διάβασμα / Γράψιμο

Το είδος της μνήμης με το οποίο ασχοληθήκαμε είναι η μνήμη όπου ο χρήστης μπορεί να τοποθετήσει τα δεδομένα του ,τα προγράμματα που θα τα επεξεργασθούν και τα αποτελέσματα που θα προκύψουν . Κατά την διάρκεια εκτέλεσης των προγραμμάτων του ο χρήστης μπορεί με τις κατάλληλες εντολές να επιλέξει οποιαδήποτε θέση μνήμης και να διαβάσει το περιεχόμενό της ή να γράψει ένα αποτέλεσμα σε αυτήν .

Κάτι που πρέπει να έχει υπόψην του ο χρήστης είναι πως τα περιεχόμενα της μνήμης αυτής διατηρούνται όσο είναι ανοιχτός ο Η/Υ, ενώ μόλις κλείσει χάνονται . Οι μνήμες με την ιδιότητα αυτή χαρακτηρίζονται σαν μνήμες **volatile** . Αυτό σημαίνει πως όταν τελειώσει μια επεξεργασία και κλείσει ο Η/Υ . θα πρέπει ο χρήστης να μεταφέρει τα αποτελέσματα και ό,τι άλλο χρειάζεται από την δουλειά που έκανε στη μνήμη , σε ένα πιο μόνιμο μέσον (δίσκο, δισκέτα , κασέτα) που διατηρεί τα περιεχόμενά του και όταν ακόμα δεν περνάει ρεύμα . Τα μέσα αυτά χαρακτηρίζονται σε αντίθεση σαν **non-volatile** . Η μνήμη αυτή με άλλα λόγια χρησιμεύει σαν ένα είδος πρόχειρου σημειωματάριου , όπου ο χρήστης γράφει και σβήνει μέχρις ότου λύσει το πρόβλημά του , οπότε περνάει τα αποτελέσματα στο καθαρό για να τα χρησιμοποιήσει αργότερα . Το είδος αυτό της μνήμης , που είναι και το πιο βασικό σε ένα σύστημα Η/Υ είναι η λεγόμενη **Κύρια Μνήμη** του Η/Υ .

Η μνήμη αυτή υλοποιείται με την βοήθεια τσίπς μνήμης που λέγονται **RAM chips**-Random Access Memory και επιτρέπουν στον χρήστη να διαβάσει τα περιεχόμενά τους και να τα αλλάξει με την βοήθεια του προγράμματός του .

Μνήμη ROM – Μόνο για Διάβασμα

Η μνήμη ROM ή Read Only Memory χρησιμοποιείται για μόνιμη καταχώριση στοιχείων.

Επιτρέπει το διάβασμα των περιεχομένων τους αλλά αντίθετα με την RAM δεν επιτρέπει στον χρήστη να γράψει σε αυτήν ή να αλλάξει τα περιεχόμενά της με τα προγράμματά του, επιβάλλοντας έτσι ένα είδος προστασίας των περιεχομένων της από θελημένη ή αθέλητη επέμβαση του χρήστη. Ακόμα δεν χάνει το περιεχόμενό της με το κλείσιμο του Η/Υ. Η μνήμη με τις ειδικές αυτές ιδιότητες είναι πιο ακριβή από την RAM και διατίθεται σε μικρότερες χωρητικότητες από την RAM.

Η ROM έρχεται "γραμμένη" από τον κατασκευαστή του Η/Υ και χρησιμοποιείται για την τοποθέτηση των πιο βασικών για την λειτουργία του Η/Υ προγραμμάτων και στοιχείων. Η μνήμη αυτή υλοποιείται με την βοήθεια ειδικών τσίπς μνήμης που λέγονται rom chips.

1.1.2 Ο Επεξεργαστής και πώς λειτουργεί

Πώς εκτελούνται τα προγράμματα από την Μονάδα Ελέγχου και την Κεντρική Μονάδα.

Η **Μονάδα Ελέγχου** και η αριθμητική μονάδα από κοινού αποτελούν τον επεξεργαστή του Η/Υ . Είναι επιφορτισμένες με την επεξεργασία των δεδομένων του χρήστη . Ο επεξεργαστής με άλλα λόγια είναι η μονάδα που "εκτελεί" τα προγράμματα του Η/Υ .

Η ΜΕ ουσιαστικά αποτελεί το νευρικό κέντρο του υπολογιστή . Παρακολουθεί και ελέγχει την καλή λειτουργία όλων των μονάδων του Η/Υ και κάτω από τον έλεγχό της γίνεται η ροή εντολών και δεδομένων από και προς την μνήμη . Για να πετύχει αυτό παίρνει από την μνήμη τις εντολές του προς εκτέλεση προγράμματος μία μία, τις αποκωδικοποιεί και τις εκτελεί . Εάν η εντολή αφορά στην εκτέλεση μιας αριθμητικής ή λογικής πράξης ενεργοποιεί την ΑΜ και πετυχαίνει την εκτέλεσή της . Εδώ υπάρχει ένας καταχωρητής για την καταχώριση των δεδομένων και ένας για την καταχώριση των εντολών .

Ακόμα υπάρχει ο καταχωρητής για την πρόσβαση στην επιθυμητή θέση μνήμης . Είναι ο καταχωρητής διευθύνσεων όπου τοποθετείται η διεύθυνση της ζητούμενης λέξης της μνήμης .

Τέλος υπάρχει ο καταχωρητής που εξασφαλίζει την σειριακή εκτέλεση των εντολών του προγράμματος . Αυτός δείχνει ανά πάσα στιγμή την διεύθυνση της εντολής που πρόκειται να εκτελεσθεί .

Η **Αριθμητική Μονάδα** λειτουργεί κάτω από τον έλεγχο της ΜΕ και έχει σαν αποστολή την εκτέλεση των υπολογισμών που ζητάει ο χρήστης μέσα από

το πρόγραμμά του . Οι υπολογισμοί αυτοί μπορεί να είναι είτε αριθμητικοί είτε λογικοί .

Με τις απλές αυτές πράξεις που εκτελεί η **AM** δίνει την δυνατότητα της λεγόμενης **λήψης αποφάσεων** στον Η/Υ. Η λήψη αποφάσεων στηρίζεται στην σύγκριση δύο ποσοτήτων που γίνετε στην **AM** . Ανάλογα με το αποτέλεσμα της σύγκρισης γίνεται η εκτέλεση μιας συγκεκριμένης σειράς εντολών που έχει προκαθορίσει ο προγραμματιστής στο πρόγραμμά του .

Η **Μονάδα** αυτή χαρακτηρίζεται σαν **Μονάδα εκτέλεσης** . Η **AM** χρησιμοποιεί τρεις βασικούς καταχωρητές . Στους πρώτους δύο τοποθετούνται τα δεδομένα στα οποία θα γίνει η ζητούμενη πράξη και ονομάζονται **Accumulator** ή **ACC** και **Operand** ή **OPR** . Ο τρίτος καταχωρητής λέγεται **MQR** είναι βοηθητικός και χρησιμοποιείται στους πολλαπλασιασμούς (αν το γινόμενο είναι πολύ μεγάλο) και στις διαιρέσεις (γαι να κρατάει το υπόλοιπο) . Ο πιο σημαντικός και γνωστός καταχωρητής είναι ο **ACC** που κρατάει ανά πάσα στιγμή το αποτέλεσμα της αριθμητικής πράξης . Μεταξύ των καταχωρητών αυτών γίνεται διακίνηση πληροφοριών επομένως απαιτείται ένας αγωγός που ονομάζεται **εσωτερικό bus** δεδομένων του επεξεργαστή και αποτελείται από γραμμές - καλώδια τόσα όσα και τα bits του **MDR** .

1.1.3 Επικοινωνία Επεξεργαστή και Μνήμης

Ο επεξεργαστής προκειμένου να πάρει τις εντολές του προγράμματος και τα δεδομένα εισόδου και να στείλει πίσω στη μνήμη τα αποτελέσματα επικοινωνεί με τη μνήμη με έναν αγωγό που λέγεται **bus μνήμης** . Το bus μνήμης αποτελείται από γραμμές και καλώδια μέσα στα οποία κυκλοφορούν τα σήματα δεδομένων , τα σήματα διευθύνσεων των επιθυμητών μονάδων και τα σήματα ελέγχου . Τα σήματα αυτά έχουν δυαδική μορφή , δηλαδή αποτελούνται από bits . Όλα αυτά τα σήματα δεν περνάνε από τις ίδιες γραμμές του bus , αλλά το κάθε ένα έχει τις δικές του για την αποκλειστική του χρήση . Οι γραμμές του bus λοιπόν διακρίνονται σε τρεις ομάδες γραμμών : τις γραμμές για τη μεταφορά δεδομένων (bus δεδομένων - data bus) , τις γραμμές για τη μεταφορά διευθύνσεων (bus διευθύνσεων - address bus) και τις γραμμές για τη μεταφορά σημάτων ελέγχου (bus ελέγχου - control bus).

Μέσα στο bus αυτό κυκλοφορούν τα δεδομένα και οι εντολές από :

- ⇒ τη μνήμη στον επεξεργαστή για επεξεργασία . Αυτό λέγεται και **Διάβασμα - READ** μιας συγκεκριμένης θέσης μνήμης από τον επεξεργαστή .
- ⇒ από τον επεξεργαστή στη μνήμη γίνεται το **Γράψιμο - WRITE** του αποτελέσματος επεξεργασίας σε μια συγκεκριμένη θέση στη μνήμη .

Ο επεξεργαστής καθορίζει από ή προς ποιά θέση μνήμης θα γίνει η διακίνηση των δεδομένων δίνοντας την διεύθυνση της επιθυμητής θέσης μνήμης με τη βοήθεια του address bus .

Αφού τώρα ο επεξεργαστής προσδιορίσει επακριβώς τη λέξη της μνήμης που τον ενδιαφέρει θα πρέπει να ειδοποιήσει τη μνήμη τί θέλει να κάνει με αυτή τη λέξη και συγκεκριμένα αν θέλει να διαβάσει το περιεχόμενό της ή να γράψει σε αυτήν κάποιο αποτέλεσμα που έχει επεξεργαστεί .

Για να πάρει η μονάδα ελέγχου , δηλαδή "διαβάσει" κάποιο στοιχείο (δεδομένο ή εντολή) από τη μνήμη :

⇒ "φωνάζει" το όνομα (διεύθυνση) της λέξης της μνήμης που την ενδιαφέρει .
Για να το κάνει αυτό τοποθετεί στον MAR την διεύθυνση και φορτώνει το περιεχόμενο στο address bus για μεταφορά στη μνήμη . Με τον τρόπο αυτό επιλέγεται η λέξη .

⇒ ειδοποιεί την μνήμη ότι επιθυμεί να πάρει (διαβάσει) το περιεχόμενο της λέξης . Για να το κάνει αυτό ενεργοποιεί τη γραμμή READ του control bus .
Αφού φτάσουν και τα δύο παραπάνω σήματα στη μνήμη αυτή ενεργοποιείται και μεταφέρει το ζητούμενο περιεχόμενο (αφού το φορτώσει στο data bus) στον MDR από όπου το παραλαμβάνει η ME .

Για να πάρει η μνήμη στοιχεία από τον επεξεργαστή ακολουθείται η αντίστροφη πορεία . Τώρα το προς μεταφορά αποτέλεσμα είναι τοποθετημένο στον MDR . Η ME λοιπόν :

- ⇒ "φωνάζει" το όνομα (διεύθυνση) της λέξης της μνήμης όπου πρέπει να τοποθετηθεί το αποτέλεσμα
- ⇒ ειδοποιεί τη μνήμη να παραλάβει το στοιχείο που της έρχεται με το data bus και να το τοποθετήσει στη διεύθυνση που έχει επιλεγεί
- ⇒ φορτώνει το περιεχόμενο του MDR στο data bus .

Τύποι Επικοινωνίας

σειριακή / παράλληλη . Στη σειριακή επικοινωνία υπάρχει ένας μόνο αγωγός μεταξύ πομπού και δέκτη μέσα από τον οποίο περνάνε όλα τα bits το ένα μετά το άλλο . Στην παράλληλη επικοινωνία αντίθετα υπάρχουν πολλοί αγωγοί ένας για κάθε bit .

Είναι προφανές πως η παράλληλη επικοινωνία είναι πιο γρήγορη από τη σειριακή . Είναι όμως και πιο ακριβή γιατί απαιτεί πολλούς αγωγούς . Σε μικρές αποστάσεις αυτό δεν είναι πρόβλημα για το λόγο αυτό στο εσωτερικό του Η/Υ έχουμε παράλληλη επικοινωνία δηλ. το bus είναι ένας παράλληλος αγωγός . Σε μεγαλύτερες όμως αποστάσεις προτιμάται η σειριακή επικοινωνία και για μεγάλες είναι ο κανόνας . Εκτός όμως από τους καθαρά πρακτικούς λόγους για τους οποίους προτιμάται η σειριακή μεταφορά θα πρέπει να σημειώσουμε και κάποια προβλήματα που παρουσιάζει η παράλληλη μεταφορά . Συγκεκριμένα τα bits στην παράλληλη επικοινωνία δεν ταξιδεύουν με την ίδια ταχύτητα στις αντίστοιχες γραμμές μεταφοράς οπότε δεν θα φτάνουν συγχρόνως στον προορισμό τους και χρειάζεται ειδικές διατάξεις για να διασφαλίσουμε τον συγχρονισμό και την αξιοπιστία των μεταφερομένων πληροφοριών . Βλέπουμε λοιπόν πως η σειριακή επικοινωνία είναι η πιο συνηθισμένη μορφή επικοινωνίας του Η/Υ με τον έξω κόσμο .

1.2 Μονάδες INPUT / OUTPUT

1.2.1 Μονάδες εισόδου στοιχείων - INPUT

Πληκτρολόγιο : Είναι μια συσκευή εισόδου και μέσω αυτής επικοινωνούμε με τον Ηλεκτρονικό Υπολογιστή πληκτρολογώντας τις διάφορες εντολές ή τα διάφορα στοιχεία .

Mouse :Είναι συσκευή εισόδου . Η αρχή λειτουργίας του είναι η εξής .Στην οθόνη εμφανίζονται διάφορα γραφήματα , που το καθένα παριστάνει κάποια συγκεκριμένη εργασία που μπορεί να εκτελέσει ο Η/Υ . Κουνώντας το mouse αυτόματα κουνιέται και κάποιος δείκτης (συνήθως ο δείκτης είναι ένα βέλος). Όταν θέλουμε να εκτελεσθεί κάποια διαδικασία τότε πηγαίνουμε τον δείκτη (κουνώντας κατάλληλα το mouse) πάνω στο ανάλογο γράφημα και στην συνέχεια για να εκτελέσουμε την διαδικασία πατάμε το αριστερό κουμπί που βρίσκεται πάνω στο mouse .

Scanner: Είναι συσκευή εισόδου .Είναι μια μηχανή η οποία επιτρέπει την απευθείας εισαγωγή στον Η/Υ προτυπωμένων εντύπων χωρίς να απαιτείται η πληκτρολογήσή τους . Ο scanner συνδέεται με τον Υπολογιστή και αφού ψηφιοποιήσει το κείμενο το στέλνει στον Η/Υ . Στον Η/Υ υπάρχει ένα πρόγραμμα (το οποίο συνεργάζεται με τον scanner) για την ανάγνωση , καταχώρηση των χαρακτήρων στην μνήμη , καθώς και για την μετέπειτα επεξεργασία τους .

1.2.2 Μονάδες εξόδου στοιχείων - OUTPUT

Έχουμε λοιπόν τα στοιχεία μας : δεδομένα , προγράμματα ή αποτελέσματα στη κύρια ή βοηθητική μνήμη που είδαμε στην προηγούμενη ενότητα .

Τώρα χρειαζόμαστε ένα μηχάνημα που θα μας βοηθάει να τα παίρνουμε από τα bits και τα bytes και τα τρακς όποτε τα χρειαζόμαστε και μάλιστα σε μια μορφή που θα μπορούμε να τα διαβάζουμε και να τα κρατάμε . Ένα απαραίτητο μηχάνημα εξόδου σε κάθε σύστημα Η/Υ είναι ο **Εκτυπωτής** .

Έξοδος στοιχείων: Διακρίνουμε δύο κύριες μορφές εξόδου στοιχείων , την έξοδο σε άμεσα αναγνώσιμη μορφή , δηλαδή σε μορφή που μπορούμε να διαβάσουμε απευθείας και την έξοδο σε μη αναγνώσιμη μορφή δηλαδή σε μορφή που για να διαβάσουμε πρέπει να υποστεί κάποια επεξεργασία .

Εκτυπωτές έξοδος σε άμεσα αναγνώσιμη μορφή

Η πιο διαδεδομένη μορφή είναι εκείνη που μας επιτρέπει την άμεση ανάγνωση των εξερχομένων στοιχείων είτε στην **οθόνη** του τερματικού είτε στον **εκτυπωτή** .

Οι εκτυπωτές είναι το πιο διαδεδομένο μηχάνημα εξόδου . Η ανάπτυξη των Η/Υ γενικότερα και κυρίως των προσωπικών υπολογιστών έφερε μια κάθετη αύξηση στην ζήτηση εκτυπωτών στην οποία η αγορά ανταποκρίθηκε με πλήθος μοντέλων από τα πιο απλά και φθηνά μέχρι τα πιο ακριβά και εξεζητημένα .

Οι εκτυπωτές χωρίζονται σε τρία κύρια είδη : τους εκτυπωτές χαρακτήρων ή σειριακούς εκτυπωτές , τους εκτυπωτές γραμμών ή line printers και τους εκτυπωτές σελίδων με κυριότερους εκπροσώπους τους γνωστούς laser printers . Μια άλλη διαφοροποίηση εκτυπωτών τους διακρίνει σε εκτυπωτές που χρησιμοποιούν μελανοταινία και τυπώνουν με κτύπημα του χαρακτήρα πάνω στο χαρτί και σε αυτούς που χρησιμοποιούν μελάνι και τυπώνουν φτύνοντας το μελάνι πάνω στο χαρτί .

Εκτυπωτές χαρακτήρων: Είναι οι πιο κατεξοχήν εκτυπωτές των προσωπικών υπολογιστών . Είναι φτηνοί , αξιόπιστοι εκτυπωτές αλλά δεν είναι κατάλληλοι για εκτυπώσεις μεγάλων όγκων. Είναι σειριακοί εκτυπωτές που τυπώνουν χαρακτήρα - χαρακτήρα . Υπάρχουν δύο κύρια είδη σειριακών εκτυπωτών : οι εκτυπωτές ολόκληρων χαρακτήρων ή μαργαρίτας και οι εκτυπωτές σημείων .

Οι **εκτυπωτές μαργαρίτας** λειτουργούν με τρόπο ανάλογο της γραφομηχανής . Διαθέτουν ένα σετ από στάνταρτ χαρακτήρες κατάλληλο αποτυπωμένο σε μεταλλικό ή πλαστικό μέσο και με τον κατάλληλο μηχανισμό χτυπούν το χαρτί και αποτυπώνουν τον αντίστοιχο χαρακτήρα .

Οι **εκτυπωτές σημείων** τυπώνουν με την βοήθεια μιας κεφαλής που αποτελείται από ένα πλέγμα ακίδων. Εκτύπωση ενός γράμματος γίνεται με την κατάλληλη ενεργοποίηση εκείνων των ακίδων που αντιστοιχούν στο σχήμα του υπόψη γράμματος .

Εκτυπωτές γραμμών: Είναι οι επαγγελματικοί εκτυπωτές,- δηλαδή οι εκτυπωτές που συναντώνται κυρίως στα μηχανογραφικά κέντρα . Είναι μεγάλης αξιοπιστίας και μπορούν να φέρουν σε πέρας μεγάλους όγκους εκτυπώσεων αλλά είναι ακριβοί βασικό τους χαρακτηριστικό είναι ότι μαζεύουν

Έξοδος σε μη άμεσα αναγνώσιμη μορφή

Όταν τα προς έξοδο στοιχεία είναι μικρού σχετικά όγκου τότε έχει νόημα να τυπωθούν με τους εκτυπωτές ,όταν όμως οι όγκοι είναι μεγάλοι δεν έχει νόημα να τυπώνουμε χιλιάδες σελίδες . Εκτός αυτού δεν συμφέρει ιδίως τους μεγάλους Η/Υ να απασχολούνται επί τόσο μεγάλο χρονικό διάστημα με το να τυπώνουν στοιχεία σε ένα αργό περιφερειακό όπως είναι οι εκτυπωτές .

Έξοδος σε δίσκο - SPOOL: Στην περίπτωση αυτή μπορούμε να τοποθετήσουμε τα στοιχεία που προορίζονται για έξοδο , σε ένα μαγνητικό μέσο (δίσκο ή ταινία) και να τυπώσουμε από εκεί σε εκτυπωτή επιλεκτικά όποια τμήματα μας ενδιαφέρουν σε κάθε δεδομένη στιγμή . Η διαδικασία αυτή εκτός του ότι μας εξοικονομεί χώρο παίρνει και λιγότερο χρόνο μιας και τα μαγνητικά μέσα είναι πολύ πιο γρήγορα μηχανήματα από τους εκτυπωτές .

Λόγω των παραπάνω πλεονεκτημάτων οι μονάδες μαγνητικών μέσων σε συνδυασμό με τους εκτυπωτές αποτελούν ένα βασικό μηχάνημα εξόδου για τους μεγάλους Η/Υ ακόμα και στην περίπτωση που απαιτείται άμεση εκτύπωση των αποτελεσμάτων . Η διαδικασία εξόδου είναι απλή : ο Η/Υ γράφει τα αποτελέσματα στο μαγνητικό μέσο και από εκεί τυπώνονται στον εκτυπωτή λίγα λίγα χωρίς να απασχολείται άμεσα ο Η/Υ, ο οποίος κατά την διάρκεια της εκτύπωσης απασχολείται με άλλες εργασίες . Η μέθοδος αυτή λέγεται **SPOOL**.

1.2.3 Μονάδες εισόδου - εξόδου στοιχείων -- I/O

Μηχανογραφικά μέσα αρχειοθέτησης

Τα κυριότερα μηχανογραφικά μέσα αρχειοθέτησης είναι οι μαγνητικές ταινίες και οι μαγνητικοί και οπτικοί δίσκοι. Για μικρούς υπολογιστές έχουμε τις κασσέτες , τα cartridges και τις εύκαμπτες δισκέτες (floppy discs) .

Για την αρχειοθέτηση σε καθένα από τα παραπάνω μέσα χρησιμοποιείται η αντίστοιχη μονάδα - μηχανήμα (μονάδα ταινιών , δίσκων κτλ.) . Λέγοντας αρχειοθέτηση σε ένα κατάλληλο μέσο εννοούμε το να τοποθετήσουμε όλες τις εγγραφές ενός αρχείου στο μέσο αυτό την μία μετά την άλλη .

Μαγνητικές Ταινίες : Μια μηχανογραφική μαγνητική ταινία μοιάζει εξωτερικά με μια ταινία μαγνητοφώνου , αποτελείται από πλαστική ταινία πλάτους μισής ίντσας (1/2 in ή 1.27 cm) καλυμμένη στην μια πλευρά της από μαγνητιζόμενο υλικό . Σε κάθε κάθετη διατομή της ταινίας υπάρχουν συνήθως 9 σημεία μαγνήτισης καθένα από τα οποία μπορεί να είναι ή να μην είναι μαγνητισμένο . Αυτές τις καταστάσεις μαγνήτισης μπορούμε εύκολα να αντιστοιχήσουμε με το δυαδικό bit 1 και 0 αντίστοιχα με την ίδια συλλογιστική που χρησιμοποιήσαμε στην περίπτωση της μνήμης . Βλέπουμε λοιπόν ότι κάθε κάθετη τομή της ταινίας αποτελείται από 9 bits που είναι αρκετά για να παραστήσουμε ένα χαρακτήρα . Συγκεκριμένα από τα 9 bits τα 8 δηλαδή 1 byte χρησιμοποιούνται για την απεικόνιση του χαρακτήρα και το ένατο είναι βοηθητικό (bit ελέγχου). Οι νοητές οριζόντιες γραμμές μαγνήτισης που διατρέχουν όλο το μήκος της ταινίας λέγονται **tracks** της ταινίας .

Ο αριθμός των χαρακτήρων που μπορούν να γραφτούν σε μια ταινία μας δίνει τη **χωρητικότητα** της ταινίας . Το **μήκος** της ταινίας εκφράζεται σε πόδια . Η **πυκνότητα** εγγραφής εκφράζεται σε bytes per inch - **BPI** .

Όπως είναι φυσικό σε μια ταινία μπορούν να γραφτούν περισσότερα από ένα αρχεία . Το κάθε αρχείο αναγνωρίζεται μέσα στην ταινία από το όνομα που του έχει δώσει ο χρήστης που το δημιούργησε . Στην αρχή της ταινίας υπάρχει συνήθως μια περιοχή στην οποία είναι γραμμένες γενικές πληροφορίες για τα αρχεία που περιέχει και ονομάζεται **Label** (ταμπέλα ,τίτλος) της ταινίας.

Ταινίες Streamer : Μια ειδική περίπτωση ταινίας είναι η λεγόμενη ταινία streamer η οποία είναι απλουστευμένη έκδοση ταινίας , μοιάζει με κασέτα , δεν ξεχωρίζει blocks και χρησιμοποιείται για αποτύπωση - διάσωση (dump)των περιεχομένων της μνήμης .

Μηχανογραφικοί Δίσκοι : Οι μηχανογραφικοί δίσκοι μοιάζουν σε γενικές γραμμές με τους κοινούς δίσκους του πικ-απ και ανάλογα με τα κατασκευαστικά χαρακτηριστικά τους διακρίνονται στους παραδοσιακούς **μαγνητικούς δίσκους** και στους πιο μοντέρνους **οπτικούς δίσκους** .

Οι μαγνητικοί δίσκοι διακρίνονται στους μεταλλικούς τους λεγόμενους σκληρούς δίσκους και στους πλαστικούς ή εύκαμπτους δίσκους τους γνωστούς floppy που χρησιμοποιούνται συνήθως στους προσωπικούς micro υπολογιστές .

Μαγνητικοί δίσκοι (σκληροί) : Η τοποθέτηση των στοιχείων στους δίσκους γίνεται σε ομόκεντρες περιφέρειες , σε αντίθεση με την ταινία όπου γίνεται κατά μήκος της ταινίας . Η επιφάνεια ενός δίσκου αποτελείται από ομόκεντρες περιφέρειες μαγνητικού υλικού τα **tracks** στα οποία

τοποθετούνται τα προς αρχειοθέτηση στοιχεία σε διαδοχικά bytes - χαρακτήρες . Όταν λέμε λοιπόν πως ένα αρχείο είναι στον δίσκο εννοούμε πως όλες οι εγγραφές του η μια μετά την άλλη χαρακτήρα - χαρακτήρα καταχωρούνται σε διαδοχικά τρακς του δίσκου .

Ένα τρακ χωρίζεται σε ισομεγέθη τόξα που ονομάζονται **sectors** κάθε ένα από τα οποία αποτελείται συνήθως από 512 bytes . Το sector αποτελεί την μονάδα μεταφοράς μεταξύ δίσκου και μνήμης .

Λέγοντας δίσκο στην μηχανογραφική διάλεκτο εννοούμε συνήθως ένα σύνολο ομοαξονικών δίσκων που έρχονται σε κατάλληλη συσκευασία και θεωρούνται σαν ένας δίσκος που λέγεται disk - rack . Ένα χαρακτηριστικό στοιχείο του disk - rack είναι ο κύλινδρος που ορίζεται σαν το σύνολο των τρακς που έχουν την ίδια ακτίνα σε όλους τους δίσκους του disk - rack . Υπάρχουν τόσοι κύλινδροι σε ένα disk - rack όσα και τα τρακς κάθε δίσκου . Οι κύλινδροι αριθμούνται από την εξωτερική περιφέρεια του δίσκου προς την εσωτερική . Η τοποθέτηση των χαρακτήρων ενός αρχείου στον δίσκο γίνεται διαδοχικά στα τρακς του ίδιου κυλίνδρου και όταν γεμίσει ο κύλινδρος συνεχίζεται στον αμέσως επόμενο .

Η ανάγνωση / εγγραφή πληροφοριών του δίσκου από και προς την μνήμη γίνεται όταν τον τοποθετήσουμε στο κατάλληλο μηχανήμα - **μονάδα δίσκων** . Το μηχανήμα αυτό περιστρέφει τον δίσκο με σταθερή ταχύτητα και οι πληροφορίες διαβάζονται ή γράφονται με την βοήθεια κατάλληλα κινούμενων **κεφαλών** . Οι κεφαλές αυτές ανάγνωσης / εγγραφής βρίσκονται στο άκρο βραχιόνων για να διευκολύνεται η πρόσβαση σε οποιαδήποτε περιοχή του δίσκου . Συνήθως υπάρχει μία κεφαλή για κάθε επιφάνεια του δίσκου αλλά σε μερικές περιπτώσεις μπορεί να υπάρχουν πολλές κεφαλές σε κάθε βραχίονα .

Οπτικοί δίσκοι :Τα τελευταία χρόνια όλο και περισσότερο συζητούνται οι οπτικοί δίσκοι κυρίως λόγω της τεράστιας ικανότητας καταχώρησης στοιχείων που διαθέτουν . Η πρώτη γενιά των οπτικών δίσκων ανακαλύφθηκε από την εταιρεία Φίλιπς βασισμένη στην ίδια τεχνολογία με τους γνωστούς κόμπακτ ντίσκς και ονομάστηκε **CD-ROM** .

Ο τρόπος κατασκευής των δίσκων αυτών είναι ανάλογος με εκείνο των κόμπακτ ντίσκς :με την χρησιμοποίηση ισχυρής δέσμης ακτίνων laser δημιουργούμε μικροσκοπικές οπές στην επιφάνεια ενός δίσκου - προτύπου .

Οι οπές αυτές ονομάζονται pits ενώ τα ανέπαφα από ακτίνες σημεία ανάμεσά τους ονομάζονται lands . Στην ανάγνωση τα pits από τα lands διαφοροποιούνται γιατί έχουν διαφορετική ανακλαστικότητα οπότε στα μέν πρώτα αντιστοιχούμε το 1 και στο άλλο το 0 . Έτσι καταχωρούμε πληροφορίες υπό δυαδική οπτική μορφή .

Δισκέτες : Οι δισκέτες τα γνωστά φλόπις ή μαλακοί δίσκοι είναι τα πιο διαδεδομένα μέσα αρχειοθέτησης των PC . Είναι πλαστικοί δίσκοι καλυμμένοι με μαγνητικό υλικό . Οι μεγαλύτεροι δίσκοι σήμερα είναι διαμέτρου 8 ιντσών (8") . Οι πιο συνηθισμένες μορφές όμως είναι διαμέτρου 5.25" και 3.5" .

Μια τυπική δισκέτα MS -DOS είναι η λεγόμενη διπλής όψης και διπλής πυκνότητας . Κάθε όψη έχει 40 τρακς , αριθμημένα από το 0 μέχρι το 39 .Η αρίθμηση γίνεται από έξω προς το κέντρο του δίσκου . Κάθε τρακ έχει 9 sectors και κάθε sector 512 bytes.

Επικοινωνία με την μνήμη

Στοιχεία που είναι τοποθετημένα στην βοηθητική μνήμη δεν μπορούν να επεξεργασθούν εκεί αλλά πρέπει να μεταφερθούν προηγουμένως στην κύρια μνήμη . Για να μεταφερθούν τα στοιχεία από την ταινία στην κύρια μνήμη θα πρέπει να τοποθετήσουμε την ταινία στην ειδική μονάδα - μηχανήμα που είναι συνδεδεμένη με την κεντρική μονάδα ελέγχου .

Η μεταφορά από και προς την μνήμη δεν γίνεται συνήθως με μία εγγραφή αλλά καθ'ομάδας εγγραφών τα blocks .

Όσον αφορά τα στοιχεία που φυλάγονται στους δίσκους πρέπει και αυτά να μεταφερθούν στην μνήμη για να γίνει η επεξεργασία τους .Η μεταφορά αυτή γίνεται όταν ο δίσκος τοποθετηθεί στο κατάλληλο μηχανήμα δίσκων(disk drive). Η μικρότερη ποσότητα που μπορεί να μεταφερθεί είναι ένα sector. Πρωτού όμως γίνει η μεταφορά θα πρέπει να εντοπισθεί η θέση της πληροφορίας μέσα στον δίσκο και να δοθεί εντολή από το σύστημα στην κεφαλή να τοποθετηθεί στο αντίστοιχο σημείο .

Κεφάλαιο 2ο

Προγράμματα-Software



2.1 Προγράμματα Υπολογιστών

Το λογικό ή λογισμικό (software) του Η/Υ. Όσα αναφέραμε στα προηγούμενα κεφάλαια αφορούσαν το μηχανισμό του Η/Υ (Hardware) . Το Hardware όμως θα μπορούσαμε να πούμε με απλά λόγια ότι είναι ένα "κουτί" γεμάτο μηχανήματα , εξαρτήματα και καλώδια .

Ο Η/Υ όμως σαν "κουτί" απλά , δεν λειτουργεί . Για να λειτουργήσει χρειάζεται κάποιο υλικό , το λεγόμενο λογικό ή λογισμικό (**software**) . Το υλικό αυτό είναι τα διάφορα **προγράμματα** που ουσιαστικά αντιπροσωπεύουν όλη τη γνώση του υπολογιστή για τον έξω κόσμο, που προσδιορίζουν την νοημοσύνη του και του επιτρέπουν να "επικοινωνεί" με το περιβάλλον του. Η νοημοσύνη δίνεται στον Η/Υ με τη μορφή οδηγιών - προγραμμάτων που τον "διδάσκουν" πώς θα αντιδρά σε κάθε συγκεκριμένη περίπτωση .

2.2 Πρόγραμμα και προγραμματισμός

Το πρόγραμμα και ο προγραμματισμός εκφράζει και κατά κάποιο τρόπο ενσαρκώνει την κυριαρχία του ανθρώπου πάνω στην μηχανή .

Ένα πρόγραμμα δεν είναι τίποτα άλλο από ένα κείμενο γραμμένο σε μια γλώσσα προγραμματισμού . Όπως ένα κείμενο σε οποιαδήποτε γλώσσα έτσι και το πρόγραμμα αποτελείται από προτάσεις , λέξεις , σημεία στίξεως και άλλα σύμβολα αποδεκτά από την υπόψη γλώσσα και κατάλληλα διατυπωμένα ώστε να γίνονται κατανοητά από εκείνους που πρόκειται να τα διαβάσουν δηλ. τους Η/Υ .

Πώς γράφουμε ένα πρόγραμμα σε μια γλώσσα προγραμματισμού ; Μα όπως γράφουμε ένα κείμενο σε οποιαδήποτε γλώσσα ! Στην αρχή

προετοιμαζόμαστε για το τί θέλουμε να γράψουμε δηλ. οργανώνουμε τις ιδέες που θέλουμε να αναπτύξουμε . Στη συνέχεια τις διατυπώνουμε όσο μπορούμε καλύτερα χρησιμοποιώντας τις κατάλληλες λέξεις οργανωμένες σε προτάσεις που να έχουν νόημα και να είναι συντακτικά σωστές . Ιδιαίτερη σημασία δίνεται στην εμφάνιση του κειμένου , τα διαστήματα , τα κενά , τον χωρισμό των παραγράφων , κλπ. Αν δεν ξέρουμε άριστα την γλώσσα που χρησιμοποιούμε για να γράψουμε το κείμενο συνήθως δίνουμε το κείμενο να το διαβάσει κάποιος που ξέρει τη γλώσσα καλά , κάνουμε τις διορθώσεις που μας υποδεικνύει και τότε μόνο λέμε πώς τελειώσαμε και το κείμενο μπορεί να σταλεί στους αποδέκτες του .

Ορισμένα από αυτά τα προγράμματα πρέπει να εγκατασταθούν στον Η/Υ από τον κατασκευαστή του .Είναι τα προγράμματα συστήματος . Είναι άκρως απαραίτητα για να τεθεί ο Η/Υ σε λειτουργία και αποδίδονται με τον όρο **Λειτουργικά Συστήματα** . Αυτά εγκαθίστανται στον σκληρό δίσκο (c:) του Η/Υ κατά την αγορά του , τον συνοδεύουν πάντα και όταν ανοίγει ο Η/Υ είναι έτοιμα να του δώσουν τις απαραίτητες οδηγίες ανάλογα με τη δουλειά που προορίζεται να κάνει . Το πιο βασικό από αυτά τα προγράμματα είναι το **MS-DOS**.

Μια ματιά στο MS-DOS : Το **MS-DOS** είναι ένα πρόγραμμα άκρως απαραίτητο για την εκκίνηση του Η/Υ . Για να τεθεί σε λειτουργία πρέπει να περάσει από το αποθηκευτικό μέσο στο οποίο βρίσκεται στην μνήμη του Η/Υ . Όταν βρίσκεται φορτωμένο στον Σκληρό Δίσκο , φορτώνει μόνο του τον εαυτό του από το δίσκο στη μνήμη , όπου και παραμένει στη συνέχεια , αναμένοντας τις διαταγές σας . Αν δεν χρησιμοποιείται σκληρό δίσκο , το **MS-DOS** πρέπει να αντιγραφεί στη μνήμη του Η/Υ από τη δισκέτα του οδηγού A . Σ' αυτό μπορείς να δημιουργήσεις διάφορους καταλόγους και αρχεία . Μπορείς να δεις ανά πάσα στιγμή , δίνοντας την κατάλληλη εντολή, την ημερομηνία και την ώρα του συστήματος , καθώς και να αλλάξεις αυτά . Το **MS-DOS** είναι σε επίπεδο

διαταγών , έτοιμο να δεχτεί μια διαταγή όταν στην οθόνη βλέπουμε το σήμα του συστήματος **C:** .

Εκτός από τα προγράμματα συστήματος υπάρχουν και προγράμματα που βοηθούν τον χρήστη να γράψει ή να διορθώσει τα προγράμματά του , να τα μεταφράσει σε γλώσσα μηχανής και να τα εκτελέσει . Επίσης να δημιουργήσει αρχεία δεδομένων , να τα διορθώσει , να τα ταξινομήσει κλπ. Τα κυριότερα από τα προγράμματα αυτά είναι τα εξής :

⇒ **Μεταφραστές** (compilers). Τα προγράμματα αυτά δίνουν στον Η/Υ τη γνώση των γλωσσών (basic , cobol , fortran , κλπ.)και τον βοηθούν να επικοινωνήσει με τον άνθρωπο .

⇒ **Utilities** . Τα προγράμματα αυτά βοηθούν τον χρήστη να κάνει δουλειές που χρειάζεται να γίνονται συχνά κυρίως με τα προγράμματα του και τα αρχεία του.

Για τα **προγράμματα** του χρήστη υπάρχουν οι εξής Utilities :

⇒ editors για γράψιμο και διόρθωση πηγαίων προγραμμάτων

⇒ Tracing, debbygging για λογικό έλεγχο και διόρθωση των προγραμμάτων κατά την ροή τους

⇒ Για τα αρχεία του χρήστη υπάρχουν τα :

⇒ sort για την ταξινόμηση αρχείων ως προς το κλειδί της επιλογής του χρήστη

⇒ file copying για αντιγραφή αρχείων

⇒ file maintenance για δημιουργία αρχείων , εισαγωγή , διαγραφή και τροποποίηση εγγραφών

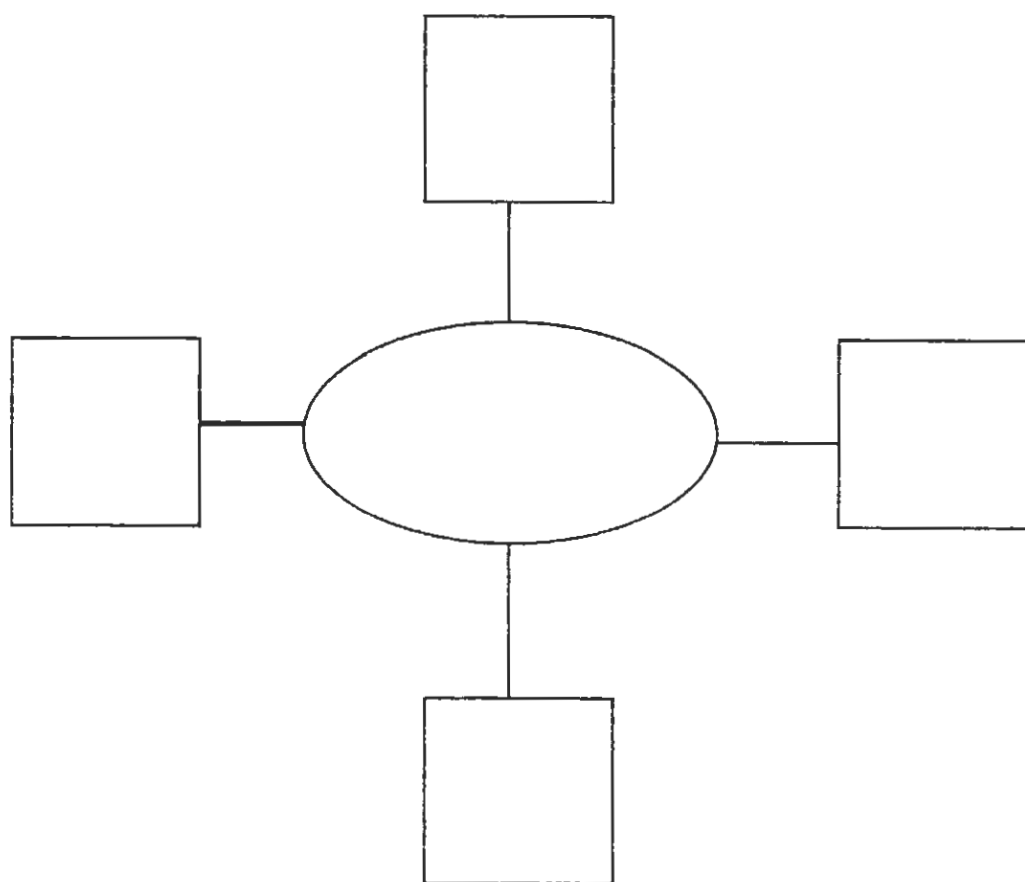
⇒ reportgenerators για την δημιουργία εκτυπωτικών καταστάσεων της αρεσκείας μας από στοιχεία που βρίσκονται σε διάφορα αρχεία .

2.2.1 Ειδικά προγράμματα ανάπτυξης εφαρμογών

Όλοι σχεδόν οι Η/Υ ακόμη και οι σημερινοί μικρουπολογιστές εκτός από τις δυνατότητες που δίνουν στους χρήστες να αναπτύσσουν οι ίδιοι τις εφαρμογές τους γράφοντας τα αντίστοιχα προγράμματα σε γλώσσες υψηλού επιπέδου τους παρέχουν και κάποιους εναλλακτικούς πιο ξεκούραστους τρόπους ανάπτυξης εφαρμογών . Πραγματικά διατίθενται στην αγορά έτοιμα προγράμματα , πακέτα που επιτρέπουν στους χρήστες να αναπτύσσουν τις εφαρμογές τους χωρίς να χρειάζεται να γράφουν προγράμματα . Οι κυριότερες περιοχές εφαρμογών που καλύπτονται από τέτοια πακέτα είναι :

- ⇒ Επεξεργασίας δεδομένων .Εδώ έχουμε τα λεγόμενα πακέτα
- ⇒ Διαχείρισης Βάσεων Δεδομένων
- ⇒ Επεξεργασίας κειμένων .Εδώ έχουμε τα πακέτα word processing με τα οποία μπορούμε να φτιάξουμε πάσης φύσεως κείμενα (γράμματα , άρθρα , βιβλία κλπ).
- ⇒ Επεξεργασίας γραφικών .Ιδιαίτερη σημασία έχουν οι εφαρμογές γραφικών που δίνουν την δυνατότητα στους χρήστες να σχεδιάζουν σε ελεύθερο και γραμμικό σχέδιο
- ⇒ Επεξεργασίας και διαμόρφωσης σελίδων . Τα πακέτα που διατίθενται στην αγορά δίνουν την δυνατότητα στους χρήστες τους να διαμορφώνουν σελίδες σε μορφή βιβλίου , περιοδικού , ανάμειξη κειμένων και σχημάτων κλπ .
- ⇒ Ακόμα έχουμε πακέτα για στατιστικές επεξεργασίες .
- ⇒ Ιδιαίτερη σημασία έχουν αποκτήσει τα τελευταία χρόνια οι εφαρμογές τεχνητής νοημοσύνης και έμπειρων συστημάτων .

Κεφάλαιο 3ο
Συστήματα πολλών
Υπολογιστών



Γενικά για τα Δίκτυα Υπολογιστών

Στην ενότητα αυτή θα ασχοληθούμε με την ανάλυση ορισμένων βασικών θεμάτων για τα δίκτυα επικοινωνίας μέσω Ηλεκτρονικών Υπολογιστών . Πριν αναφερθούμε σε κάποια θέματα γενικά για τα δίκτυα πρέπει να αναφέρουμε ότι το σύστημα που δημιουργήσαμε , έχει την δυνατότητα να λειτουργήσει σε περιβάλλον δικτύου και συγκεκριμένα στο Novell Netware .

Δίκτυο επικοινωνίας γενικά μπορούμε να ορίσουμε ένα οργανωμένο σύνολο υπολογιστών και τερματικών που επικοινωνούν μεταξύ τους .Ο όρος αυτός χρησιμοποιείται συνήθως για να υποδηλώνει ότι η επικοινωνία γίνεται κυρίως μεταξύ τερματικών και ενός κεντρικού υπολογιστή ενώ όταν αναφερόμαστε σε ένα σύνολο διασυνδεμένων , ανεξάρτητων και αυτόνομων υπολογιστών που επικοινωνούν κατά τρόπο αυτόματο μεταξύ τους τότε ορίζουμε ένα δίκτυο υπολογιστών (computer network) .Η επικοινωνία αυτή επιτυγχάνεται με την ανταλλαγή δεδομένων μεταξύ των υπολογιστών και τη κατανομή του υπολογιστικού έργου .Πιο συγκεκριμένα κάθε χρήστης έχει την δυνατότητα να αντλεί πληροφορίες από ένα δίκτυο , να στέλνει πληροφορίες και δεδομένα για επεξεργασία και να λαμβάνει σε οποιοδήποτε χρονικό σημείο τις απαραίτητες απαντήσεις . Για παράδειγμα μερικοί χρήστες ασχολούνται με το πρόγραμμα αποθήκης, πωλήσεων και αποστολής αγαθών, παράλληλα με την έκδοση των αντίστοιχων εγγράφων πρέπει να ενημερώνεται κάθε κίνηση εισόδου ή εξόδου τις αποθήκης και όλες οι πληροφορίες να είναι διαθέσιμες σε όλους τους χρήστες των αντίστοιχων τμημάτων .

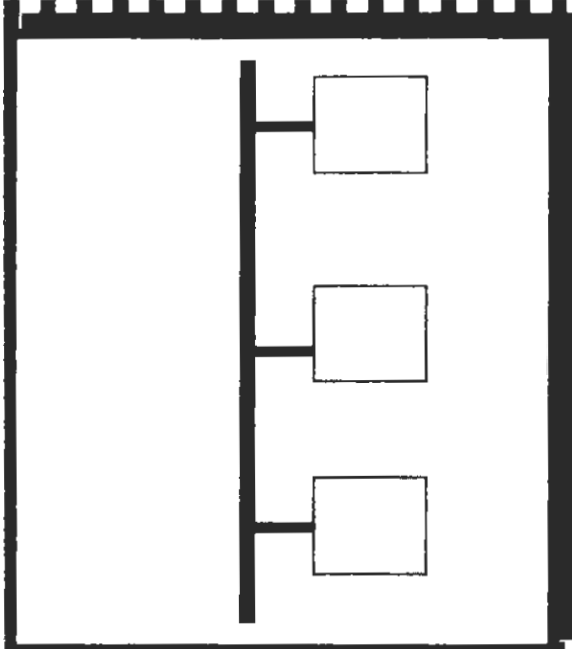
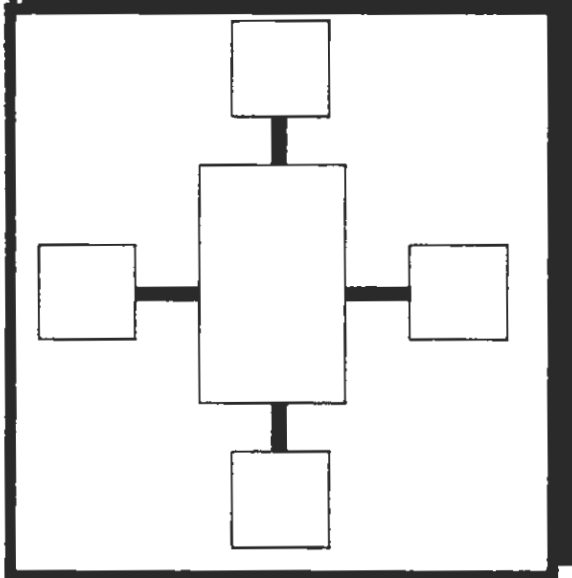
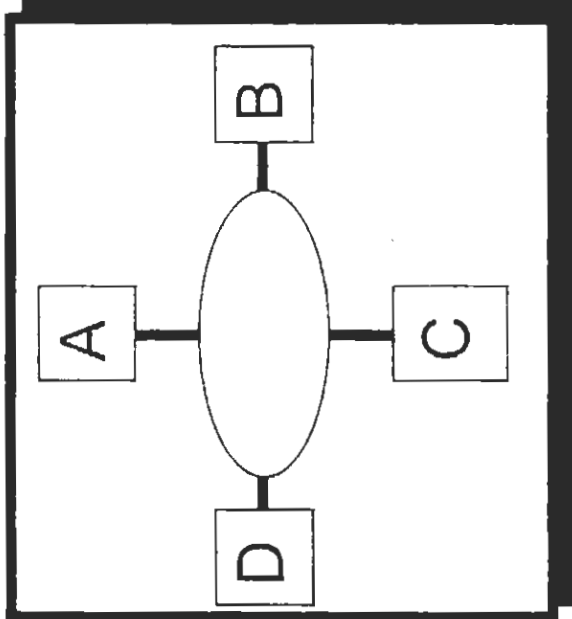
Η επικοινωνία επιτυγχάνεται μέσω υπολογιστικών κόμβων όπου εκτελείται η επεξεργασία δεδομένων και επικοινωνιακούς δίαυλους που τους διασυνδέουν. Στο σχήμα () αναπαρίστανται μια βασική δομή δικτύου.

Η εγκατάσταση ενός δικτύου σε μία επιχείρηση πρέπει να πραγματοποιείται μετά από ανάλυση των εναλλακτικών λύσεων που υπάρχουν . Το κυριότερο όμως που πρέπει να διευκρινίζεται είναι τι απαιτήσεις υπάρχουν από το δίκτυο , αλλά και τι δυνατότητες έχει το δίκτυο. Αναλυτικότερα θα πρέπει να εξετάσουμε :

- ❶ Τον αριθμό των σταθμών που μπορούν να συνδεθούν
- ❷ Τον αριθμό των περιφερειακών συσκευών που μπορούν να υπάρξουν
- ❸ Την δυνατότητα ενσωμάτωσης του σε μεγαλύτερο δίκτυο
- ❹ Την ευκολία χρήσης
- ❺ Τις προδιαγραφές ασφάλειας
- ❻ Την υποστήριξη από τον προμηθευτή
- ❼ Την δυνατότητα χρήσης λογισμικού πολλών χρήσεων
- ❽ Τα προβλήματα των φυσικών συνδέσεων

Δίκτυα

Τοπολογίες Δικτύων



Δακτύλιος

Αστέρας

Δίαυλος

3.1 Ανάλυση Χαρακτηριστικών

Γενικά τα δίκτυα μπορούν να ταξινομηθούν ως προς τα ακόλουθα χαρακτηριστικά :

- ♦ *τοπολογία*
- ♦ *ιεραρχία ελέγχου και ροής δεδομένων*
- ♦ *περιοχή κάλυψης*
- ♦ *κυριότητα και χρήση .*

3.1.1.Τοπολογίες Δικτύων

Τοπολογία Δικτύου: Ονομάζεται η καλωδίωση που ακολουθεί μια προκαθορισμένη διαμόρφωση , η οποία καθορίζει τη ροή των δεδομένων και των πληροφοριών . Ανάλογα με το σχήμα σύνδεσης οι τοπολογίες διακρίνονται στις ακόλουθες κατηγορίες .

⇒ Τοπολογία αστέρα

Όλα τα τερματικά συνδέονται απευθείας και με ξεχωριστή γραμμή με ένα κεντρικό υπολογιστή . Από την φύση της δεν απαιτεί πολύπλοκο λογισμικό επικοινωνιών , χαρακτηρίζεται από υψηλή αξιοπιστία γιατί η βλάβη σε ένα τερματικό αφήνει ανηπερέαστα τα άλλα , καλή απόδοση λόγω απευθείας σύνδεσης αλλά και υψηλό κόστος λόγω του ότι για κάθε τερματικό απαιτείται ξεχωριστή γραμμη .

⇒ Τοπολογία δένδρου

Σε κάθε γραμμή μπορεί να συνδέονται περρισότερα από ένα τερματικά . Ο κεντρικός υπολογιστής χρησιμοποιώντας ορισμένες τεχνικές κλήσης και επιλογής καθορίζει ποιο τερματικό λαμβάνει ή στέλνει δεδομένα . Χρησιμοποιείται ευρύτατα γιατί στοιχίζει φθηνά αλλά παρουσιάζει μεγάλη πολυπλοκότητα στην σύνδεση και μπορεί να έχει χειρότερη απόδοση . Στην

τοπολογία αυτή η χρήση πολυπλεκτών συσκευών για την πολλαπλή μετάδοση πληροφοριών ταυτόχρονα από ένα μέσο βοηθούν στην μείωση του κόστους .

⇒ Τοπολογία δακτυλίου

Όλοι οι υπολογιστές του δικτύου και τα τερματικά συνδέονται σε δακτύλιο . Τα μηνύματα ρέουν στο δακτύλιο με την ίδια φορά και περιέχουν την διεύθυνση του τερματικού για τον οποίο προορίζεται. Τα τερματικά βλέπουν όλα τα μηνύματα και αν η διεύθυνση είναι δική τους τα επεξεργάζονται αλλιώς τα αγνοούν . Η τοπολογία αυτή χρησιμοποιείται κυρίως στα τοπικά δίκτυα και μειονεκτεί κατά το ότι μία βλάβη στη γραμμή σταματά τη λειτουργία του δικτύου .

⇒ Τοπολογία βρόγχων

Χαρακτηριστικό της είναι η ύπαρξη επικοινωνιακών κόμβων οι οποίοι διασυνδέουν σταθμούς δηλαδή τερματικά ή υπολογιστές εφαρμογών και δεν έχουν ως αντικείμενο την δημιουργία των επικοινωνιακών συνδέσεων μεταξύ σταθμών .

Οι σταθμοί συνδέονται με κόμβους και επικοινωνούν μέσω αυτών . Οι κόμβοι ανάλογα με το είδος της σύνδεσης μπορεί να είναι διαφορετικοί .

3.1.2 Ιεραρχία Ελέγχου και ροής Δεδομένων

Τα δίκτυα μπορούν να ταξινομηθούν με κριτήριο την ύπαρξη ή όχι βαθμίδων συγκέντρωσης κατά την πορεία της ροής από τα τερματικά προς τον υπολογιστή και αντίστροφα .

Στα **μη ιεραρχικά δίκτυα** (non-hierarchical) δεν μεσολαβεί βαθμίδα συγκέντρωσης και υπάρχει απευθείας σύνδεση μεταξύ του κεντρικού υπολογιστή (ή της μονάδας ελέγχου επικοινωνιών που θεωρείται μέρος του υπολογιστή εξειδικευμένο στην προσαρμογή των διαφορετικών ταχυτήτων

μετάδοσης (τερματικών και υπολογιστή) . Σε ένα τέτοιο δίκτυο μπορεί να περιλαμβάνονται και πολυπλέκτες για την μείωση του αριθμού των απαιτούμενων γραμμών επικοινωνίας .

Στα **ιεραρχικά δίκτυα** (hierarchical) τα τερματικά συνδέονται κατά ομάδες σε μονάδες ελέγχου και επικοινωνιών , οι οποίες συνδέονται σε συγκεντρωτές οι οποίοι με την σειρά τους συνδέονται στον κεντρικό υπολογιστή από τον οποίο και ελέγχονται .

3.1.3 Περιοχή κάλυψης του δικτύου

Η ευρύτητα της περιοχής στην οποία αναπτύσσεται ένα δίκτυο καθορίζει και τις τεχνικές μετάδοσης που πρέπει να χρησιμοποιηθούν για κάθε περίπτωση . Διακρίνουμε τρεις κατηγορίες δικτύων

⇒ **Δίκτυα ευρείας περιοχής** .

Οι υπολογιστές και τα τερματικά βρίσκονται σε μεγάλη απόσταση μεταξύ τους και οι συνδέσεις τους γίνονται με γραμμές όλων των ειδών : επίγειες , δορυφορικές που διατείνονται από τις εταιρείες τηλεπικοινωνιών

Η ταχύτητα μετάδοσης ποικίλει κατά περίπτωση και φθάνει μέχρι 19.200 bps με συνηθέστερες τιμές τα 9.000 bps (ή και τα 2.400 bps όπου τα δίκτυα δεν είναι σε καλή κατάσταση) .

⇒ **Δίκτυα τοπικής περιοχής ή τοπικά δίκτυα** .

Αναπτύσσονται συνήθως μέσα σε ένα κτίριο ή συγκρότημα κτιρίων όπως , Πανεπιστήμια , γραφεία επιχειρήσεων . Οι γραμμές είναι ιδιόκτητες και λόγω περιορισμένου μήκους γραμμών οι ταχύτητες μπορούν να φθάσουν μέχρι και 1Mbps . Τα τοπικά δίκτυα βρήκαν μεγάλη εφαρμογή με τους μικρουπολογιστές στον αυτοματισμό γραφείου . Η εγκατάστασή τους αποκτεί την ύπαρξη ειδικού λογισμικού όπως π.χ. το Novel .

⇒ **Μεικτά δίκτυα .**

Η διασύνδεση διαφόρων τοπικών δικτύων σε ένα ή περισσό τερα δίκτυα ευρείας περιοχής δίνει την δυνατότητα στο μεμονωμένο χρήστη να επεξεργάζεται τα δεδομένα του σε τοπικό επίπεδο και στη συνέχεια να μπορεί να έχει προσπέλαση και σε πληροφορίες που βρίσκονται αποθηκευμένες σε απομακρυσμένους υπολογιστές .

3.1.4. Ταξινόμηση ως προς την κυριότητα

Διακρίνουμε δύο κατηγορίες ποιός είναι ο ιδιοκτήτης και ποιό το χρησιμοποιούν .

⇒ **Ιδιωτικά δίκτυα .**

Όταν ο εξοπλισμός και το λογισμικό του δικτύου ανήκουν στην επιχείρηση ιδιοκτήτη ανεξάρτητα τοπυ ποιός διέθεσε τις γραμμές .

⇒ **Δημόσια δίκτυα .**

Η μεγάλη ζήτηση τηλεφωνικών γραμμών λόγω της ραγδαίας ανάπτυξης των εφαρμογών τηλεπληροφορικής οδήγησε δημόσιους και ιδιωτικούς οργανισμούς στη δημιουργία δημόσιων δικτύων κοινής χρήσης από τους ενδιαφερόμενους . Ένα δημόσιο δίκτυο αποτελείται από επικοινωνιακούς κόμβους και γραμμές . Μέσω τοπικών γραμμών οι υπολογιστές και τα τερματικά συνδέονται με τους κόμβους του δικτύου και ο κάθε χρήστης χρεώνεται ανάλογα με τον αριθμό και το είδος των συνδέσεων καθώς και τον όγκο των δεδομένων που διακινούν .

3.2 Αναφορά στα συστατικά μέρη του δικτύου

Είναι σκόπιμο για την σωστότερη κατανόηση των θεμάτων περί δικτύων να γίνει μια μικρή ανάλυση στα συστατικά μέρη ενός δικτύου .

3.2.1 Μονάδες εξυπηρέτησης του δικτύου

Μονάδα εξυπηρέτησης του δικτύου: Καλείται ο συνδυασμός μηχανικού και λογισμικού εξοπλισμού . Ο μηχανικός εξοπλισμός μπορεί να είναι ένας προσωπικός υπολογιστής ή ένας υπολογιστής που έχει σχεδιαστεί ειδικά για να εξυπηρετεί το δίκτυο . Συχνά παρατηρείται ότι ένας κεντρικός υπολογιστής χρησιμοποιείται ως διανομέας και περιλαμβάνει ολόκληρο το λογισμικό της επικοινωνίας δεδομένων που ελέγχει το δίκτυο , δηλαδή λογισμικό επικοινωνίας είναι το πρόγραμμα που καθορίζει ποιές εργασίες θα εκτελεστούν ώστε να λειτουργήσει το δίκτυο.

3.2.2 Διαύλοι επικοινωνίας

Μονάδες -σύνδεσμοι με τις οποίες μεταδίδονται και λαμβάνονται δεδομένα . Για τον σκοπό αυτό χρησιμοποιούνται τα ομοαξονικά καλώδια , οι τηλεφωνικές γραμμές , οπτικές ίνες , συστήματα μικροκυμάτων , και τέλος δορυφόροι .

3.2.3 Ενδιάμεσες μονάδες επικοινωνίας δεδομένων

Μονάδες όπως δι-αποδιαμορφωτές (modems) , ηλεκτές επικοινωνιών μετωπικοί επεξεργαστές που καθιστούν δυνατή την μετάδοση δεδομένων μεταξύ τερματικών και υπολογιστών . Ρόλος τους είναι , γενικά η μετατροπή των δεδομένων από ψηφιακή σε αναλογική μορφή και αντιστρόφως , η κωδικοποίηση και αποκωδικοποίηση των δεδομένων , ο έλεγχος της ροής

επικοινωνιών μεταξύ υπολογιστών και τερματικών σε ένα δίκτυο επικοινωνίας δεδομένων .

3.2.4 Λογισμικό επικοινωνιών

Προγράμματα ελέγχου που βρίσκονται άλλα εγκατεστημένα στον κεντρικό υπολογιστή και άλλα σε άλλες μονάδες του δικτύου .

Τα προγράμματα αυτά ελέγχουν τις δραστηριότητες εισόδου - εξόδου μέσα σε ένα σύστημα επικοινωνίας δεδομένων και διαχειρίζονται γενικά το δίκτυο .

3.2.5 Τερματικά

Μονάδες εισόδου - εξόδου όπως οθόνες , εκτυπωτές , ή συνδυασμός αυτών και δίαυλοι επικοινωνίας για μετάδοση ή λήψη δεδομένων, όπως καλώδια .

3.3 Περιήγηση στο Novell Netware

Όπως αναφέραμε και πιο πάνω το συστημά μας θα λειτουργήσει στο δίκτυο Novell Netware . Το Novell Netware είναι σήμερα το πρώτο σε πωλήσεις λογισμικό δίκτυο της αγοράς συνδυαζόμενο με τον κατάλληλο εξοπλισμό μπορεί να προσφέρει μια πολύ προχωρημένη λύση στον καταμερισμό πόρων και δεδομένων .

3.3.1. Δομικά στοιχεία

Ένα τοπικό δίκτυο Novell αποτελείται από ένα συνδυασμό υλικού και λογισμικού , που επιτρέπει τον καταμερισμό πληροφοριών και πόρων . Ενώ υπάρχει μεγάλη ποικιλία στο μέγεθος , στη μορφή και στις εφαρμογές του τοπικού δικτύου Novell , τα βασικά δομικά στοιχεία είναι τα εξής :

- ❶ Το λογισμικό που επιλέγεται
- ❷ Οι κάρτες διασύνδεσης με το δίκτυο
- ❸ Ένας ή περισσότεροι διακομιστές αρχείων
- ❹ Ένας ή περισσότεροι Σταθμοί εργασίας
- ❺ Το σύστημα καλωδίωσης
- ❻ Το λογισμικό εφαρμογών
- ❼ Οι περιφερειακές Συσκευές

Οι διάφορες εκδόσεις του Novell Netware διαφέρουν στο επιτρεπόμενο αριθμό διακομιστών αρχείων και χρηστών , στη ταχύτητα μεταφοράς δεδομένων , στη δυνατότητα προστασίας από απώλεια ή καταστροφής δεδομένων , και φυσικά στην τιμή .

Κάθε υπολογιστής στο δίκτυο πρέπει να περιλαμβάνει μια κάρτα διασύνδεσης με το δίκτυο . Αυτή αποτελεί τη σύνδεση μέσα από την οποία κάθε υπολογιστής μοιράζεται πληροφορίες (μέσω του συστήματος καλωδίωσης) με τους άλλους . Σε μερικές διευθετήσεις Novell ο διακομιστής αρχείων διαθέτει ένα ή περισσότερους σκληρού δίσκους . Οι σταθμοί εργασίας δεν είναι απαραίτητο να περιλαμβάνουν σκληρούς δίσκους και σε μερικές περιπτώσεις ακόμα και οι οδηγοί δισκέτας είναι προαιρετικοί .

Οι περισσότερες από τις εφαρμογές για DOS εκτελούνται σε δίκτυο Novell , όπως είναι και η δική μας εφαρμογή . Μία εφαρμογή σχεδιασμένη για χρήση σε δίκτυο , επιτρέπει συνήθως σε διάφορους χρήστες να έχουν πρόσβαση στο ίδιο πρόγραμμα και στα ίδια δεδομένα , την ίδια χρονική στιγμή , καθώς και τη σύνδεση με εκτυπωτές και άλλα περιφερειακά που χρησιμοποιούνται από κοινού .

Λογισμικό

Το λογισμικό του Novell περιλαμβάνει ένα λειτουργικό σύστημα που λειτουργεί στους διακομιστές αρχείων του δικτύου " κέλυφη " λογισμικού που εκτελούνται στους σταθμούς εργασίας και ένα πλήθος διαταγών και βοηθητικών προγραμμάτων που βρίσκονται στο διακομιστή ή στους διακομιστές αρχείων του δικτύου .

Οι σημερινές εκδόσεις του Novell προσφέρουν αρκετά πλεονεκτήματα όπως ανοχή σε σφάλματα , δηλαδή το δίκτυο έχει την δυνατότητα να δουλεύει στην περίπτωση βλάβης .Επίσης έχει την δυνατότητα να διατηρεί διπλά αντίγραφα ασφαλείας του καταλόγου του δίσκου και των πινάκων κατανομής αρχείων έτσι ώστε αν κάποιος καταστραφεί να υπάρχει αντίγραφο .

Διακομιστές αρχείων

Ο διακομιστής αρχείων είναι μία συσκευή μαζικής αποθήκευσης που αποτελεί την καρδιά του δικτύου . Περιλαμβάνει έναν ή περισσότερους

σκληρούς δίσκους και μοιράζεται τα προγράμματα και τα δεδομένα που περιέχουν με τους προσαρτημένους σταθμούς εργασίας . Σε ένα διακομιστή αρχείων μπορεί να συνδεθούν διάφορες συσκευές εξόδου όπως εκτυπωτές , σχεδιογράφοι . Κι αυτές οι συσκευές χρησιμοποιούνται από κοινού με τους σταθμούς εργασίας .

Σταθμοί εργασίας

Οι σταθμοί εργασίας είναι οι ιδιαίτεροι υπολογιστές μέσω των οποίων έχετε πρόσβαση στο δίκτυο , οι τύποι υπολογιστών που μπορούν να χρησιμοποιηθούν είναι οποιοδήποτε υπολογιστές με λειτουργικό σύστημα DOS . Σε κάθε σταθμό εργασίας αποδίδεται μια μοναδική διεύθυνση η οποία καθορίζεται από την κάρτα διασύνδεσης . Σε ποιά μέρη του προγράμματος μπορεί να εργαστεί ο κάθε σταθμός εργασίας εξαρτάται από τον χρήστη .

Κάρτες διασύνδεσης δικτύου

Οι διακομιστές αρχείων και οι σταθμοί εργασίας του δικτύου επικοινωνούν μεταξύ τους μέσω των καρτών σύνδεσης με το δίκτυο (Nic). Κάθε υπολογιστής στο δίκτυο πρέπει να περιέχει μια κάρτα .

Η κάρτα Nic καταλαμβάνει μια υποδοχή επέκτασης στον υπολογιστή και παρέχει μια εξωτερική σύνδεση , στην οποία συνδέεται το καλώδιο του δικτύου . Υπάρχουν διάφορες μορφές καλωδίωσης . Πάντως με κάποιο τρόπο το καλώδιο του δικτύου συνδέει όλους τους υπολογιστές στο δίκτυο. Υπάρχει ένας αριθμός χαρακτηριστικών που διαφοροποιούν τις κάρτες Nic που μπορεί κανείς να χρησιμοποιήσει για την δημιουργία δικτύου της Novell Netware . Τα κυριότερα από αυτά θα αναλυθούν στις παρακάτω παραγράφους.

Τύπος καλωδίωσης Ο τύπος καλωδίωσης είναι ένα χαρακτηριστικό που πρέπει να εξεταστεί γιατί κάθε τύπος κάρτας Nic έχει ιδιαίτερες απαιτήσεις

καλωδίωσης .Στις οποίες περιλαμβάνονται το ομοαξονικό , το συστρόφου ζεύγους , το καλώδιο πολλών αγωγών ή ακόμα και με οπτικές ίνες .

Ταχύτητα μεταφοράς : Η ταχύτητα με την οποία τα δεδομένα μεταφέρονται μέσω του δικτύου καθορίζεται από την κάρτα Nic και μετριέται σε δεκαδικά ψηφία άνα δευτερόλεπτο . Το δυαδικό ψηφίο είναι η μικρότερη μονάδα πληροφορίας και αναπαρίσταται με 0 ή 1 . Στις κάρτες η ταχύτητα μεταφοράς μετριέται σε χιλιοψηφία ή μεγαψηφία άνα δευτερόλεπτο . Ανάλογα με την κάρτα Nic που θα επιλέξετε η ταχύτητα επικοινωνίας μπορεί να είναι από 500 χιλιοψηφία άνα δευτερόλεπτο ως 20 μεγαψηφία άνα δευτερόλεπτο .

Μέθοδος ελέγχου συγκρούσεων : Σε ένα περιβάλλον δικτύου οι σταθμοί εργασίας και οι διακομιστές αρχείων επικοινωνούν μεταξύ τους μέσω ενός συστήματος καλωδίωσης .Επειδή όμως την ίδια στιγμή είναι πιθανόν να χρειάζονται πρόσβαση περισσότεροι από ένας υπολογιστές στο δίκτυο .

Αποστάσεις δικτύου : Για να γίνει η σωστή επιλογή μιας κάρτας Nic χρειάζεται να ληφθούν υπόψη και οι αποστάσεις του δικτύου , έτσι ώστε κάθε φορά να επιλέγεται ο κατάλληλος τύπος καλωδίου ώστε να καλύπτει τις αποστάσεις του δικτύου .

Μέγεθος προσωρινής αποθήκευσης : Οι κάρτες Nic διαθέτουν μνήμη που παίζει το ρόλο της περιοχής προσωρινής αποθήκευσης μεταξύ της κάρτας και του δικτύου . Αυτό τους επιτρέπει να στέλνουν και να λαμβάνουν δεδομένα , ενώ ο υπολογιστής να ασχολείται με άλλες εργασίες . Όσο πιο μεγάλη είναι η περιοχή προσωρινής αποθήκευσης που διαθέτει μια κάρτα Nic τόσο πιο γρήγορη μπορεί να είναι η συνολική λειτουργία .

Συνολική ταχύτητα : Οι επιδόσεις μιας κάρτας Nic εξαρτώνται από την ταχύτητα μετάδοσης , τον τρόπο που χειρίζεται τις συγκρούσεις , την ποσότητα

μνήμης πάνω στην κάρτα που χρησιμοποιείται σαν περιοχή προσωρινής αποθήκευσης . Η ταχύτητα ενός δικτύου μπορεί να εξαρτάται από πολλούς παράγοντες στους οποίους περιλαμβάνονται η ταχύτητα του διακομιστή αρχείων , ή των σταθμών εργασίας , ο αριθμός των χρηστών και η ταχύτητα των καρτών Nic .

"Μέρος Β"

Κεφάλαιο 4ο

Ανάλυση-Σχεδιασμός του Συστήματος Τιμολόγησης



Εισαγωγή

Στην ενότητα αυτή εξηγούμε την διαδικασία που ακολουθήσαμε ώστε να δημιουργήσουμε ένα πρόγραμμα Τιμολόγησης .

Ξεκινώντας την εργασία ορίσαμε τις λειτουργίες που θα καλύψει το σύστημα Τιμολόγησης, δηλαδή ποιός είναι ο σκοπός δημιουργίας του. Δώσαμε μεγάλη προσοχή στην ανάλυση των εργασιών που θέλουμε να πραγματοποιεί το σύστημα γιατί ο ακριβής ορισμός των λειτουργιών διευκολύνει τον σχεδιασμό του.

Επόμενο βήμα είναι η φάση του Σχεδιασμού όπου περιλαμβάνει ανάλυση των προδιαγραφών του συστήματος σε βασικά στοιχεία (γενικός σχεδιασμός) και διαχωρισμό των βασικών στοιχείων σε αυτόνομες ενότητες (λεπτομερής σχεδιασμός) .

Ο Γενικός σχεδιασμός μας βοήθησε στο να καθορίσουμε τα κυριότερα τμήματα του συστήματος και την μεταξύ τους επικοινωνία . Επίσης ο διαχωρισμός του συστήματος στα μέρη που το αποτελούν έχει ως στόχο να προσδιορίσουμε και να αξιολογήσουμε τα προβλήματα και τα τμήματα που χρειάζονται ιδιαίτερη προσοχή .

Ο λεπτομερής σχεδιασμός είναι η διαδικασία όπου από τις αυτόνομες ενότητες προσπαθήσαμε να δημιουργήσουμε κάποια σχέδια των προγραμμάτων και ορίσαμε τις δομές των δεδομένων .

Τελευταία ενέργεια είναι η Υλοποίηση του συστήματος , δηλαδή είναι η αντιστοιχία του σχεδιασμού στη γλώσσα προγραμματισμού και παράλληλα ελέγχουμε το τμήμα που φτιάχνουμε εάν λειτουργεί .

Θεωρήσαμε σκόπιμο πριν προχωρήσουμε στην διαδικασία προγραμματισμού να αναφέρουμε λίγα λόγια για τα Αρχεία και για τις γλώσσες Προγραμματισμού δηλαδή για τα στοιχεία που απαιτούνται για την δημιουργία ενός προγράμματος

Ανάλυση Αρχείων

Για τη δημιουργία ενός προγράμματος είναι απαραίτητο να δημιουργηθούν πριν κάποια **αρχεία (files)** . **Αρχείο** ονομάζεται το σύνολο των ομοειδών λογικών εγγραφών που έχουν καταχωρηθεί σε ένα μέσο καταχώρησης στοιχείων , με τρόπο ώστε να μπορεί να γίνει επεξεργασία των στοιχείων αυτών ,δηλαδή εντοπισμός λογικών εγγραφών για εκμετάλλευση , διαγραφή ενημέρωση των περιεχομένων των πεδίων , προσθήκη νέων λογικών εγγραφών , κτλ.

Φυσικό αρχείο ονομάζεται το μέσο στο οποίο έχουν καταχωρηθεί οι εγγραφές ενός αρχείου . **Λογικό μέγεθος** ενός αρχείου ονομάζεται ο αριθμός των λογικών εγγραφών που περιλαμβάνονται στο αρχείο αυτό . **Φυσικό μέγεθος** ενός αρχείου ονομάζεται ο αριθμός των χαρακτήρων που περιλαμβάνεται στο αρχείο αυτό .

Οι βασικοί τύποι των εγγραφών που μπορεί να υπάρχουν σε ένα αρχείο είναι :

- ⇒ Εγγραφές στοιχείων (data records)
- ⇒ Είναι οι εγγραφές που περιέχουν τα στοιχεία των λογικών ενοτήτων που έχουν καταχωρηθεί στο αρχείο .
- ⇒ Εγγραφές ετικέτας (label records)

Είναι οι εγγραφές που υπάρχουν μόνο στα μαγνητικά αρχεία και περιέχουν στοιχεία για την αναγνώριση και ασφάλεια των αρχείων και διακρίνονται σε ετικέτες αρχής και σε ετικέτες τέλους .

Κάθε αρχείο αποτελείται από ορισμένα πεδία και εγγραφές .

Πεδίο (field) ονομάζεται κάθε λογική ενότητα στοιχείων , στην οποία έχει νόημα η επεξεργασία από τον Η/Υ . **Μήκος πεδίου** ονομάζεται ο αριθμός των χαρακτήρων του πεδίου .

Λογική εγγραφή ονομάζεται το σύνολο των πεδίων που ανήκουν στην ίδια λογική ενότητα . Η λογική εγγραφή αποτελεί τη βασική μονάδα για την επεξεργασία στοιχείων στον Η/Υ . Με κάθε εντολή αναγνώσεως (read) του προγράμματος , μια λογική εγγραφή μεταφέρεται στη μνήμη του Η/Υ ή γίνεται διαθέσιμη . στην περίπτωση ομαδοποίησης , για επεξεργασία . **Μήκος λογικής εγγραφής** ονομάζεται ο αριθμός των χαρακτήρων , από τον οποίο αποτελείται αυτή και είναι ίσο με το άθροισμα των μηκών των πεδίων , που περιέχονται στη λογική εγγραφή . **Γραμμογράφηση λογικής εγγραφής** ονομάζεται ο τρόπος διατάξεως των στοιχείων μέσα σε αυτή . Αν υπάρχει διαφορά στη σειρά των πεδίων ή στα μήκη τους τότε πρόκειται για διαφορετική γραμμογράφηση .

Φυσική εγγραφή : Στα μαγνητικά μέσα καταχωρήσεως στοιχείων οι λογικές εγγραφές καταχωρούνται η μια μετά την άλλη και μεταξύ τους υπάρχει ένα κενό που ονομάζεται Inter Record Gap (IRG) . Τις περισσότερες φορές μεταξύ δύο IRG περιλαμβάνονται περισσότερες από μια λογικές εγγραφές . Στην περίπτωση αυτή οι λογικές εγγραφές είναι **ομαδοποιημένες** .

Το τμήμα του μέσου καταχωρήσεως , που βρίσκεται μεταξύ δύο IRG ονομάζεται **φυσική εγγραφή** ή block . Ο αριθμός των λογικών εγγραφών , που περιλαμβάνονται στη φυσική εγγραφή , ονομάζεται **συντελεστής ομαδοποίησης** . Όταν η φυσική εγγραφή συμπίπτει με τη λογική και δεν υπάρχει ομαδοποίηση η φυσική εγγραφή είναι η βασική μονάδα μεταφοράς στοιχείων από τις περιφερειακές μονάδες στη μνήμη του Η/Υ και από τη μνήμη στις περιφερειακές μονάδες .

Μήκος φυσικής εγγραφής ονομάζεται ο αριθμός των χαρακτήρων που υπάρχουν σε αυτή , δηλαδή μεταξύ δύο IRG και είναι ίσο με το άθροισμα των μηκών των λογικών εγγραφών που περιλαμβάνονται στη φυσική εγγραφή .

Ονομασία αρχείων : Το όνομα του αρχείου αποτελείται υποχρεωτικά από το κυρίως όνομα που μπορεί να είναι μέχρι 8 χαρακτήρες και από την επέκταση που μπορεί να είναι μέχρι 3 χαρακτήρες . Μεταξύ του ονόματος και της επέκτασης πρέπει υποχρεωτικά να μπει τελεία . (π.χ. timo.prg)

Γλώσσες προγραμματισμού

Εκτός από τα αρχεία που αναφέραμε παραπάνω για να γίνει ένα πρόγραμμα χρειάζεται απαραίτητως να χρησιμοποιηθεί και μια γλώσσα προγραμματισμού . Γλώσσα προγραμματισμού είναι ένα σύνολο στοιχείων με τα οποία σχηματίζονται σύμβολα και λέξεις , έτσι ώστε τα διάφορα βήματα επίλυσης ενός προβλήματος να μπορούν να γίνουν αντιληπτά από έναν υπολογιστή . Για την επικοινωνία με τον υπολογιστή έχουν βρεθεί πολλές γλώσσες π.χ. FORTRAN , BASIC , CLIPPER , PASCAL , COBOL , κτλ.

Τα προγράμματα προκειμένου να γίνουν αποδεκτά από τον υπολογιστή θα πρέπει να εκφράζονται σε γλώσσα μηχανής . Προγράμματα όμως γραμμένα σε γλώσσα μηχανής είναι μεγάλα σε έκταση , επίπονα στην γραφή και ενέχουν πολλές πιθανότητες λαθών από την μεριά του προγραμματιστή . Για την ανακούφιση μερικών από τα παραπάνω προβλήματα και για την απλούστευση της εργασίας των προγραμματιστών δημιουργήθηκαν γλώσσες προγραμματισμού ασέμπλυ και μνημοτεχνικές . Ένας μνημοτεχνικός κώδικας είναι έτσι σχεδιασμένος ώστε να βοηθά την ανθρώπινη μνήμη μέσω της χρησιμοποίησης λέξεων της ομιλουμένης γλώσσας για την διατύπωση εντολών του προγράμματος π.χ. sum σημαίνει άθροισε , browse εμφάνισε , κτλ. Υπάρχουν γλώσσες προγραμματισμού κατωτέρου και ανωτέρου επιπέδου . Οι

ανωτέρου επιπέδου γλώσσες είναι ακόμη ευκολότερες στην χρήση , γιατί κάθε επιμέρους δομή μιας ανωτέρου επιπέδου γλώσσας συνήθως αντιστοιχεί σε πολλές εντολές της χαμηλού επιπέδου γλώσσας. Με τον τρόπο αυτό τα προγράμματα συντομεύονται και μπορούν να είναι πιο φυσικά και κατανοητά στην διατύπωσή τους . Οι ποικιλία των γλωσσών προγραμματισμού ανωτέρου επιπέδου συνεχίζει να εμπλουτίζεται , καθώς αναπτύσσονται νέες εφαρμογές και γίνεται ολοένα ευρύτερη ή χρήση των υπολογιστών . Οι γλώσσες χωρίζονται στις εξής κατηγορίες :

- ❶ Βασικές αλγοριθμικές και διαδικαστικές γλώσσες .
- ❷ Παραλλαγές γλωσσών για αλληλεπιδραστική χρήση .
- ❸ Μη αριθμητικές γλώσσες
- ❹ Γλώσσες προσωμοίωσης και
- ❺ Γλώσσες ελέγχου διεργασιών και αριθμητικού ελέγχου .

Γενικά

Σ' αυτό το σημείο είμαστε πλέον έτοιμοι να μπούμε στην συγκεκριμένη διαδικασία προγραμματισμού για την δημιουργία του προγράμματος Τιμολόγησης . Τιμολόγηση είναι η πράξη και το αποτέλεσμα του τιμολογώ . Πιο συγκεκριμένα καθορίζω την τιμή του εμπορεύματος που προορίζεται για πώληση , πολλαπλασιάζω με την ποσότητα υπολογίζω τον Φ.Π.Α και την έκπτωση εάν υπάρχει για όλα τα είδη .Η βασική δηλαδή λειτουργία του προγράμματος είναι η Τιμολόγηση και η μηχανογραφική έκδοση παραστατικών. Επίσης το πρόγραμμα έχει την δυνατότητα να ενημερώνει Αρχεία πελατών,προμηθευτών ,προϊόντων και μπορεί να λειτουργήσει και σε περιβάλλον δικτύου Novell .

4.1 Ανάλυση προδιαγραφών του Συστήματος

Το πρόγραμμα έχει την δυνατότητα της Μηχανογραφικής Έκδοσης παραστατικών .Απευθύνεται και σε χρήστες όπου δεν έχουν μεγάλη εμπειρία στην χρήση πληροφοριακών συστημάτων . Έχει σχεδιαστεί και ένα βοηθητικό τμήμα όπου γίνονται οι εργασίες αναδιοργάνωσης ,σελιδοποίησης , και καταχώρησης των στοιχείων της επιχείρησης .Επίσης το πρόγραμμα μας δίνει όλες τις απαραίτητες πληροφορίες για τα παραστατικά που εκδόσαμε , μέσα στο αρχείο παραστατικών . Επίσης πρέπει να αναφέρουμε και τα αρχεία πελατών-προμηθευτών και προϊόντων που ενημερώνονται κατά την Έκδοση και δίνουν την δυνατότητα στους χρήστες να έχουν χρήσιμες πληροφορίες .

Τέλος σας πληροφορούμε ότι το πρόγραμμα έχει την δυνατότητα να λειτουργήσει σε περιβάλλον δικτύου Novell. Πιστεύουμε ότι το πρόγραμμα ανταποκρίνεται πλήρως στις απαιτήσεις της Μηχανογραφικής έκδοσης των παραστατικών και είναι εύκολο στην χρήση .

4.2 Σχεδιασμός του Συστήματος Τιμολόγησης .

Όπως αναφέραμε και πιο πάνω στην ενότητα αυτή δημιουργώντας κάποια σχεδιαγράμματα θα προσπαθήσουμε να αναλύσουμε τα τμήματα του προγράμματος ώστε να εξηγήσουμε πως δημιουργήθηκε . Αρχικά σχεδιάσαμε σε ένα χαρτί τη μορφή που έχει κάποιο παραστατικό και αναλύσαμε τα στοιχεία που περιλαμβάνει .

Θεωρήσαμε σκόπιμο να χρησιμοποιήσουμε μια Μάσκα έκδοσης παραστατικού η οποία θα χρησιμοποιείται και για όλα τα παραστατικά .Η μάσκα χωρίζεται σε τέσσερα βασικά μέρη .

Το πρώτο μέρος περιλαμβάνει τα στοιχεία της ταυτότητας της επιχείρησης όπως(την επωνυμία,επάγγελμα, διεύθυνση,τηλέφωνο.).

Το επόμενο περιλαμβάνει τα στοιχεία του παραστατικού τον κωδικό του είδους την περιγραφή του παραστατικού ,την σειρά εκδόσής του και τον αριθμό του και τα στοιχεία των πελατών/προμηθευτών. Το βασικότερο στοιχείο πελατών - προμηθευτών είναι ο κωδικός , που είναι το (30) για τους πελάτες και το (50) για τους προμηθευτές και ακολουθεί ο προσωπικός κωδικός , η επωνυμία και το Φορολογικό μητρώο του κάθε πελάτη ή του κάθε προμηθευτή.

Το τρίτο μέρος περιλαμβάνει τα στοιχεία του προϊόντος / προϊόντων τα οποία αναλύονται στον κωδικό , την περιγραφή του είδους , την ποσότητα, την τιμή , την αξία και την έκπτωση που θα χορηγήσει η εταιρεία .

Το τέταρτο μέρος περιλαμβάνει τα γενικά σύνολα του κάθε παραστατικού όπως την συνολική αξία, το σύνολο έκπτωσης , την καθαρή αξία , την αξία Φ.Π.Α , και τέλος το καθαρό πληρωτέο ποσό. (Διάγραμμα Α)

Διάγραμμα Α

Μορφή Παραστατικού

Είδος		Κωδ.Πελ./Προμ.		Α.Φ.Μ.	
Σειρά	Αριθμός	Επωνυμία			
Κωδικός	Περιγραφή	Ποσότη.	Τιμή	Εκπτ.	Αξία
Σύνολα					

Όσον αφορά το αρχείο παραστατικών σχεδιάσαμε τη μάσκα που θέλαμε να έχει και η οποία χωρίζεται σε πέντε στήλες : Στις δύο πρώτες καταχωρούμε τον κωδικό και την περιγραφή του παραστατικού.Οι επόμενες τρεις στήλες αφορούν τις κινήσεις της αποθήκης ,με την πρώτη (Κ) ορίζουμε ότι εισάγουμε προϊόντα με τον αριθμό **(1)** (εισαγωγές) και εξάγουμε με το **(2)** (εξαγωγές) προϊόντων από την αποθήκη. Οι επόμενες στήλες της ποσότητας και της αξίας δηλώνουν εάν το παραστατικό ενημερώνει την αποθήκη με θετική εισαγωγή ή εξαγωγή προϊόντων με το πλήκτρο (+) και αρνητική εισαγωγή ή εξαγωγή προϊόντων από την αποθήκη με το πλήκτρο (-) . Υπάρχει περίπτωση ένα παραστατικό να ενημερώνει μόνο ποσότητα και όχι αξία ή το αντίθετο . Οπότε

όποια στήλη δεν ενημερώνετε το δηλώνουμε με ένα κενό (). Για παράδειγμα με το Δελτίο Αποστολής εμπορευμάτων γίνεται μια εξαγωγή οπότε δηλώνουμε στη στήλη (Κ) το (2) που σημαίνει εξαγωγή και ενημερώνετε η αποθήκη μόνο με ποσότητα και όχι με αξία . Αυτό το δηλώνουμε στο αρχείο παραστατικών βάζοντας ένα κενό () στην στήλη ποσότητα και ένα (+) στη στήλη αξία που δηλώνει θετική εξαγωγή . Όταν έρθει το τιμολόγιο ενημερώνεται η αποθήκη με αξία και όχι με ποσότητα , αφού έχει ενημερωθεί ήδη από το Δελτίο Αποστολής. Σε αυτή την περίπτωση βάζουμε ένα (+) στη στήλη αξία και ένα κενό () στην ποσότητα . Η μάσκα αυτή παρουσιάζεται στο διάγραμμα Β .

Διάγραμμα Β

Κωδ.	Περιγραφή	Κ	Π	Α
01	Τιμολ.Πώλησης -Δ.Α.	2	+	+
02	Πιστωτικό Τιμολόγιο	2		-
03	Δελτίο Αποστολής	2	+	
04	Τιμολόγιο Πώλησης	2		+

Μια τρίτη μάσκα που σχεδιάσαμε είναι για την εμφάνιση παραστατικών , η οποία αναλύεται σε πέντε πεδία : Τον κωδικό , την περιγραφή του παραστατικού , τη σειρά του , τον αριθμό του και την ημερομηνία .

(Διάγραμμα Γ). Στην μάσκα αυτή καταχωρούνται αυτόματα μέσω του προγράμματος τα παραστατικά που εκδόσαμε . Όπως αναφέραμε οι στήλες αυτές περιέχουν τα στοιχεία ταυτότητας του παραστατικού.

Διάγραμμα Γ

Κωδ.	Περιγραφή Παραστατικού	Σειρά	Αριθμός	Ημερομην
05	Τιμολόγιο Πώλησης	A	15	20/07/97
04	Δελτίο Αποστολής	B	12	23/07/97

Επόμενο βήμα μας ήταν να βρούμε από που θα παίρνουμε αυτά τα στοιχεία για να κάνουμε τις λειτουργίες που αναφέρθηκαν παραπάνω . Ο καλύτερος τρόπος ήταν να φτιάξουμε κάποιους πίνακες που θα περιλαμβάνουν αυτά τα στοιχεία . Έτσι καθορίσαμε τους πίνακες που μπορούμε να φτιάξουμε μέσα από τα στοιχεία του παραστατικού . Οπότε έχουμε έναν πίνακα για τα στοιχεία της επιχείρησης , έναν για τα στοιχεία των παραστατικών , έναν για τα στοιχεία πελατών / προμηθευτών και έναν για τα στοιχεία των διαφόρων προϊόντων . Κατά τη διαδικασία του προγραμματισμού εκτός από αυτούς τους κυρίως πίνακες , φτιάξαμε και κάποιους δευτερεύοντες που ήταν απαραίτητοι για τη λειτουργία του προγράμματος . Μέσα σε αυτούς τους πίνακες έχουμε αποθηκευμένα όλα τα στοιχεία που χρειαζόμαστε και ανά πάσα στιγμή μπορούμε να τα καλέσουμε για επεξεργασία . Αυτοί οι πίνακες είναι τα λεγόμενα **αρχεία (DBF)**. Τα βασικά αρχεία που χρησιμοποιήσαμε περιλαμβάνονται στον (πίνακα 1) της επόμενης σελίδας .

Οι ονομασίες τους δόθηκαν ανάλογα με το αντικείμενο στο οποίο αναφέρονται. Από τα βασικά αρχεία δημιουργήσαμε και κάποια ταξινομημένα (Βλέπε πίνακα 2).

Πίνακας Α
Βασικά Αρχεία

Α/Α	Ονομασία	Περιγραφή
1	Arh_prd	Αρχείο Προϊόντων
2	B_Tim	Βοηθητικό Τιμολόγησης
3	Tim	Αρχείο Τιμολόγησης
4	Tmaster	Βασικών στοιχείων Παραστατικών
5	Boss	Στοιχείων Επιχείρησης
6	Par_ika	Στοιχείων Παραστατικών
7	Ari_par	Αρίθμησης Παραστατικών
8	Pel_tes	Στοιχείων Πελατών
9	Pro_tes	Στοιχείων Προμηθευτών
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		

Πίνακας Β

Ταξινομημένα Αρχεία

Α/Α	Ταξινομημένα Αρχεία		Ταξινομημένα Αρχεία	
	Βασικά Αρχεία	ARX1.NTX	ARX2.NTX	
1	ARX PRD	ARX1.NTX	ARX2.NTX	
2	TIM	TIM1.NTX		
3	TMASTER	TIM2.NTX		
4	PAR_IKA	PAR1.NTX		
5	ARI_PAR	ARI1.NTX		
6	PEL_TES	PEL1.NTX	PEL2.NTX	
7	PRO_TES	PRO1.NTX	PRO2.NTX	
8				
9				
10				
11				
12				
13				
14				
15				

Εδώ θέλουμε να τονίσουμε ότι τα βασικά αρχεία τα δημιουργούμε όλα στην αρχή ,πριν μπούμε στον κώδικα , ενώ κάθε ταξινομημένο το φτιάχνουμε εκείνη τη συγκεκριμένη στιγμή που το χρειαζόμαστε . Όλα τα αρχεία τα δημιουργήσαμε με το εργαλείο ED μέσω της γλώσσας **clipper** , αλλά μπορούν να δημιουργηθούν και με τη γνωστή σε όλους μας **dbase** .

Ένα ακόμα πολύ βασικό μέλημα πριν μπούμε στον κώδικα του προγράμματος είναι να προσδιορίσουμε πως θα συνδέονται τα αρχεία μεταξύ τους . Η σύνδεση των αρχείων γίνεται προσδιορίζοντας κάποιο πεδίο του αρχείου ως **πεδίο κλειδί** . Το πεδίο κλειδί είναι κοινό σε δύο τουλάχιστον αρχεία. (Βλέπε Πίνακα 3).

Πίνακας Γ
Πεδία Κλειδιά

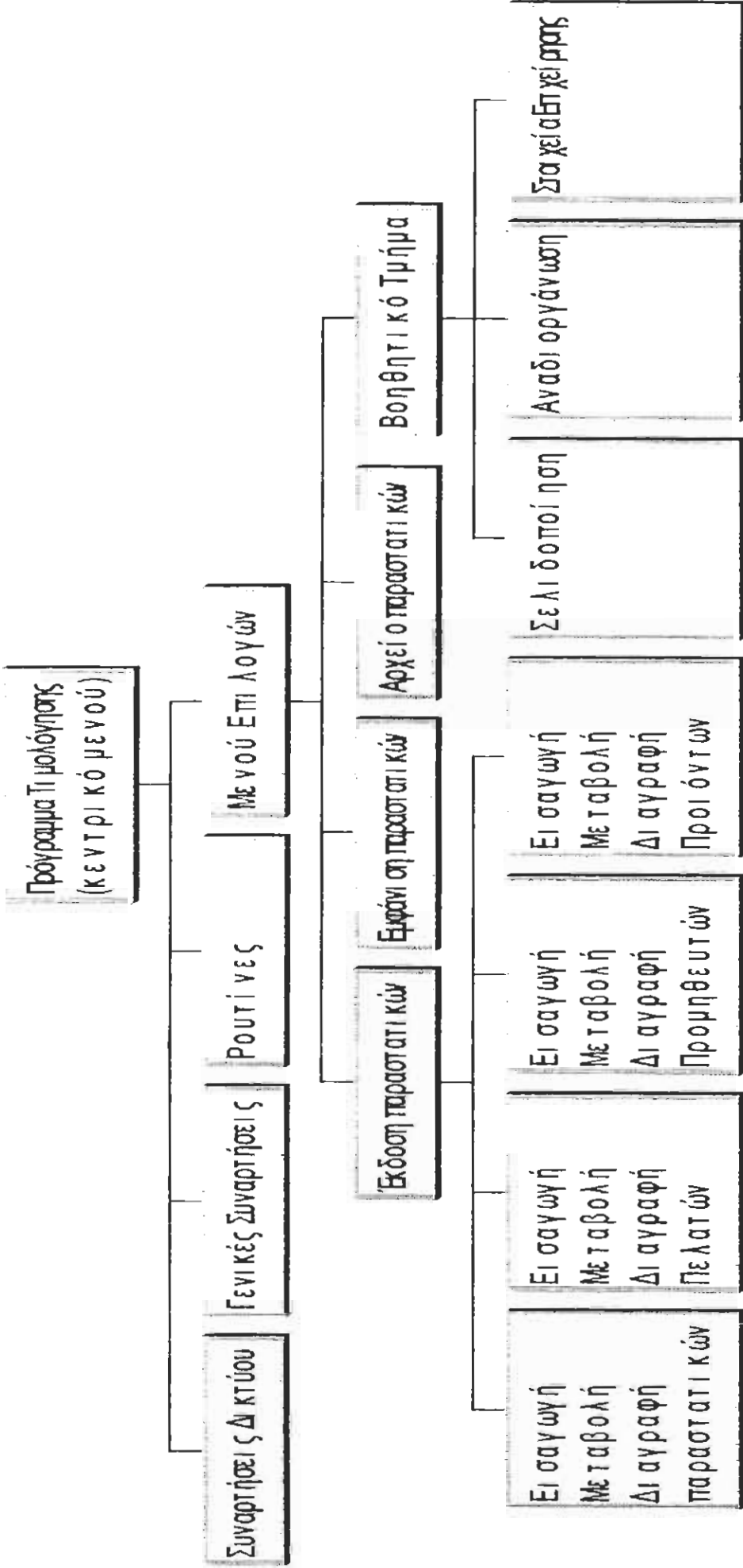
Α/Α	Αρχεία NTX	Πεδία κλειδιά
1	ARI1.NTX	KOD_PAR+SEI_PAR
2	ARX1.NTX	KOD_PRD
3	ARX2.NTX	PER_PRD
4	PEL1.NTX	KOD_PP
5	PEL2.NTX	EPO_PP
6	PRO1.NTX	KOD_PP
7	PRO2.NTX	EPO_PP
8	PAR1.NTX	KOD_PAR
9	TIM1.NTX	KOD_PAR+SEI_PAR+STR(ARI_PAR,6)
10	TIM2.NTX	KOD_PAR+SEI_PAR+STR(ARI_PAR,6)
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

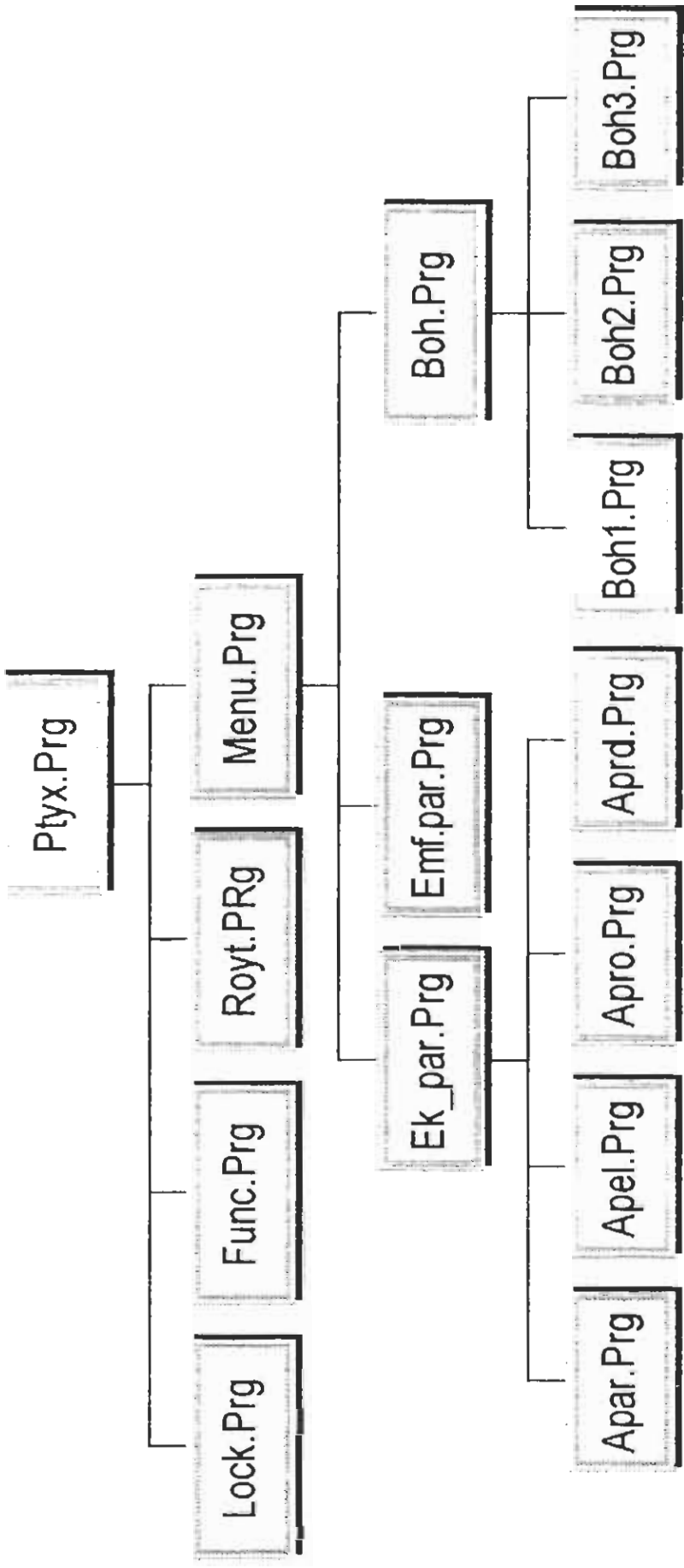
Μετά τη δημιουργία των αρχείων σχεδιάσαμε το "σκελετό" του προγράμματος. Αναλύσαμε τις συγκεκριμένες ενότητες που θα περιλαμβάνει και έτσι χωρίσαμε το πρόγραμμα σε δύο μέρη . Το **κυρίως πρόγραμμα** που περιλαμβάνει την έκδοση και το αρχείο παραστατικών και το **βοηθητικό πρόγραμμα** που περιλαμβάνει τα στοιχεία της επιχείρησης , την αναδιοργάνωση και την σελιδοποίηση . Σχεδιάσαμε πάνω στο χαρτί όλες τις λειτουργίες του προγράμματος αναλυτικά . (Διάγραμμα στην πίσω σελίδα).

Τελειώνοντας τους παραπάνω σχεδιασμούς και παίρνοντας μια ολοκληρωμένη εικόνα για το πρόγραμμα συνεχίζουμε στον κώδικα του προγράμματος . Παίρνουμε τις λειτουργίες του προγράμματος και τις κωδικοποιούμε μία - μία . Για κάθε λειτουργία φτιάχνουμε και ένα μικρό πρόγραμμα , το λεγόμενο **PRG**. (Διάγραμμα στην πίσω σελίδα).

Οι ονομασίες των PRG και των DBF είναι προτιμότερο να δίνονται ανάλογα με το αντικείμενο στο οποίο απευθύνονται π.χ. το αρχείο τιμολόγησης λέγεται TIM.DBF και το βοηθητικό πρόγραμμα BOH.PRG . Αυτό είναι απαραίτητο για την καλύτερη αναγνώριση και χρησιμοποίησή τους. Για τη δημιουργία καθενός PRG φτιάχνουμε και ένα ξεχωριστό πλάνο.

Όλα τα PRG συνδέονται μεταξύ τους με τις κατάλληλες εντολές , ρουτίνες και συναρτήσεις και υπάρχει και το κεντρικό PRG πάνω στο οποίο συνδέονται όλα τα άλλα και με το οποίο τρέχει το πρόγραμμα . Όλα αυτά τα PRG γίνανε με το εργαλείο ED μέσω τις γλώσσας **clipper** . Αναλυτικά DBF και PRG βλέπε στο παράρτημα .





4.4 Ανάλυση των εντολών Προγραμματισμού (clipper).

Εντολές προγραμματισμού

DO WHILE .T.

(κώδικας προγράμματος)

ENDO

Δημιουργεί μια κυκλική διαδικασία κατά την οποία εκτελούνται όλες οι εντολές που περικλείονται μεταξύ DO WHILE .T. ΚΑΙ ENDO , η μια μετά την άλλη συνεχώς. Αυτή η διαδικασία τερματίζεται από την συνθήκη που υπάρχει μεταξύ των εντολών .

IF LASTKEY () = 27

RETURN

ENDIF

Η εντολή αυτή δημιουργεί μια συνθήκη που στη συγκεκριμένη περίπτωση σημαίνει : εάν το τελευταίο πλήκτρο που πατήθηκε από τον χρήστη (η συνάρτηση IF LASTKEY() μας δίνει την ASCII τιμή του τελευταίου χαρακτήρα που πληκτρολογήθηκε) ήταν το ESC (η ASCII τιμή για το ESC είναι το 27) τότε ο έλεγχος περνάει από το πρόγραμμα στο DOS με την εντολή RETURN .

mEP =0

Με αυτή την εντολή δημιουργείται στη μνήμη η μεταβλητή mEP και τοποθετείται σε αυτή η τιμή 0 .

@ 08,24 PROMPT "1... Έκδοση παραστατικών "

-

-

@ 14,24 PROMPT "4... Βοηθητικό τμήμα "

Δημιουργεί το μενού φωτεινής μπάρας σε συγκεκριμένες θέσεις στην οθόνη (στο παράδειγμα η πρώτη φωτεινή φράση θα αρχίζει από τη θέση που ορίζεται από τη γραμμή 8 και τη στήλη 24) και αναγράφει τη συγκεκριμένη φράση που ο προγραμματιστής δίνει στη θέση αυτή .

MENU TO mEP

Είναι η απαραίτητη εντολή για να ενεργεί η εντολή @ ...PROMPT . Καθώς ο χρήστης μετακινείται μεταξύ των επιλογών στο μενού , η εντολή MENU αποθηκεύει μία αριθμητική τιμή στην μεταβλητή mEP η οποία βρίσκεται σε αντιστοιχία με την σειρά των επιλογών του μενού (όταν π.χ. ο χρήστης φωτίζει την 3η κατά σειρά επιλογή η εντολή MENU τοποθετεί στη μεταβλητή mEP τον αριθμό 3 .

DO CASE

CASE (συνθήκη)
(κώδικας προγράμματος)
CASE (συνθήκη)
(κώδικας προγράμματος)

ENDACSE

Η εντολή αυτή δίνει τη δυνατότητα εκτέλεσης συγκεκριμένου κώδικα προγράμματος ανάλογα με τη συνθήκη που κάθε φορά ισχύει . Έτσι στο παράδειγμά μας εάν η εντολή μενού MENU TO mEP έχει τοποθετήσει στην μεταβλητή mEP τον αριθμό 1 τότε θα εκτελεσθεί η εντολή DO EK_PAR (με την οποία καλείται άλλο τμήμα της εφαρμογής , το EK_PAR.PRG) . Εάν έχει τοποθετήσει τον αριθμό 2 θα εκτελεσθεί η εντολή DO EM_PAR , κ.ο.κ

DECLARE mPED [5], mTIT [5]

Η εντολή αυτή δημιουργεί δύο μονοδιάστατους πίνακες . Ο ένας έχει όνομα mPED και ο άλλος mTIT και το πλήθος των στοιχείων και των δύο πινάκων είναι (5).

DBEDIT (04,05,18,74,mPED, "FEMPAR" , .T. , mTIT)

Πρόκειται για μια συνάρτηση που σχεδιάζει την οθόνη , εμφανίζει σε αυτή ένα συγκεκριμένο αρχείο δεδομένων (αυτό που ήδη έχει επιλεγεί) , δίνει την δυνατότητα στο χρήστη να βλέπει τις υπάρχουσες εγγραφές του αρχείου και να κάνει τροποποιήσεις . Αναλυτικά χαρακτηρίζεται από τα εξής στοιχεία :

04,05,18,74 : σχεδιάζεται η οθόνη με βάση αυτές τις συντεταγμένες από την γραμμή 04 και στήλη 05 έως τη γραμμή 18 και στήλη 74 .

mPED : χρησιμοποιείται ο πίνακας ο οποίος περιέχει τα προς επεξεργασία στοιχεία και τα οποία εμφανίζονται στη σχεδιασμένη οθόνη (όπως έχει ήδη οριστεί στις προηγούμενες γραμμές του κώδικα του προγράμματος ο πίνακας αυτός είναι ένα υποσύνολο των πεδίων του τρέχοντος αρχείου δεδομένων TMASTER.DBF) .

fEMPAR : ορίζεται η συνάρτηση fEMPAR και η οποία θα εκτελήται κάθε φορά που πατιέται ένα μη αναμενόμενο πλήκτρο (αναμενόμενα από την DBEDIT πλήκτρα είναι τα βέλη , PgUp , PgDn , Home , End ,Ctrl w ,Ctrl PgUp, Ctrl PgDn , Ctrl E , Ctrl X , Ctrl S , Ctrl D , Ctrl A , Ctrl F , Ctrl R , Ctrl C , ESC , Ctrl Home , Ctrl End) .

.T. : χρησιμοποιείται για να δεχθούμε τις εξορισμού τιμές των παραμέτρων που χρησιμοποιεί η DBEDIT για τη σχεδίαση της οθόνης (π.χ. πάχος και φωτεινότητα των γραμμών) .

mTIT : χρησιμοποιείται ο πίνακας ο οποίος περιέχει τις επικεφαλίδες των στηλών .

RETURN

a) RETURN 1

Με την εντολή αυτή συνεχίζεται η λειτουργία της DBEDIT .

b) RETURN 2

Με την εντολή αυτή επανασχεδιάζεται η οθόνη και συνεχίζεται η λειτουργία της DBEDIT

c) RETURN 0

Με την εντολή αυτή εγκαταλείπεται η DBEDIT και τερματίζεται η λειτουργία της .

4.3 Υλοποίηση του Συστήματος τιμολόγησης

Έκδοση Παραστατικών

Η έκδοση των παραστατικών πραγματοποιείται από το πρόγραμμα Ek_par.prg και τα αρχεία που ανοίγουμε είναι τα εξής :

Par_ika.dbf , Ari_par.dbf , Tmaster. Dbf , B_tim.dbf , Tim.dbf , Pel_tes , Pro_tes και το Arh_prd.dbf . Κάθε φορά επιλέγουμε από τα ήδη ανοιχτά αρχεία το συγκεκριμένο που χρειαζόμαστε . Τα αρχεία συνδέονται τόσο μεταξύ τους όσο και με τα ταξινομημένα αρχεία με τα πεδία κλειδιά (πχ .το Par_ika.dbf και το ταξινομημένο Par1.ntx συνδέονται μεταξύ τους με το πεδίο κλειδί Kod_par). Έτσι μας δίνεται η δυνατότητα να ελέγχουμε εάν οι μεταβλητές που εισάγουμε είναι σωστές .

Η Έκδοση των παραστατικών γίνεται μέσα από μία επαναληπτική διαδικασία που επιβεβαιώνεται με το πλήκτρο Enter και διακόπτεται με το πλήκτρο Esc.

Τα πρώτα στοιχεία που καταχωρούμε σε ένα παραστατικό είναι ο κωδικός , η σειρά , η ημερομηνία και ο αριθμός του. Δίνοντας τον κωδικό ανοίγει το αρχείο Par_ika.dbf μέσω του προγράμματος Apar.prg χρησιμοποιώντας την συνάρτηση (fPARAST) , γίνεται έλεγχος εάν υπάρχει το παραστατικό και εάν υπάρχει εισάγεται αυτόματα η περιγραφή του . Στο σημείο αυτό ανοίγει το Ari_par.Dbf το οποίο λειτουργεί ταξινομημένο ως προς τον κωδικό και την σειρά του παραστατικού και αριθμεί αυτόματα τα παραστατικά .Επίσης γίνεται έλεγχος στο Tmaster για την αποφυγή καταχώρισης διπλής εγγραφής .Μέσα στο σύστημα δημιουργήσαμε δικλίδες ελέγχου για τον εντοπισμό πιθανών λαθών και ορίσαμε να βγαίνουν μηνύματα έτσι ώστε να ειδοποιούν τον χρήστη σε περίπτωση που κάνει λάθος και να τον εμποδίζουν να συνεχίσει αν δεν το διορθώσει . Για παράδειγμα όταν ο χρήστης πάει να κάνει διπλή καταχώριση

παραστατικού το πρόγραμμα κάνει εσωτερικό έλεγχο και αφού εντοπίσει ότι η εγγραφή υπάρχει ήδη βγάζει μήνυμα στη γραμμή κατάστασης που λέει : Η εγγραφή ήδη υπάρχει . Επανάλαβε .

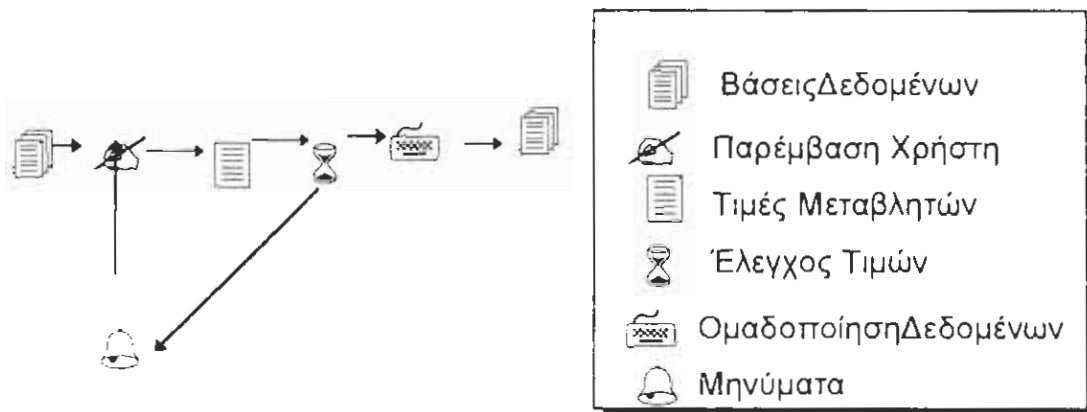
Επόμενη καταχώρηση είναι ο κωδικός πελάτη ή προμηθευτή χρησιμοποιώντας τα αρχεία Pel_tes , Pro_tes αντίστοιχα , οπότε εμφανίζεται μέσα από το αντίστοιχο αρχείο αυτόματα η περιγραφή και ο ΑΦΜ .Λέγοντας αυτόματα εννοούμε ότι το πρόγραμμα κάνει κάποιες εσωτερικές διαδικασίες που ο προγραμματιστής έχει ορίσει και εισάγει τα δεδομένα χωρίς την παρέμβαση του χρήστη . Αυτό μπορεί να γίνει και αντίστροφα , δίνοντας την περιγραφή και εμφανίζονται αυτόματα τα άλλα δύο δεδομένα .

Με τα αρχεία Apel.prg και Apro.prg μεταφερόμαστε στην διαδικασία "εισαγωγή -μεταβολή -διαγραφή πελατών ή προμηθευτών " .Μέσω των dbedit εμφανίζονται στην οθόνη τα δεδομένα των αρχείων Pel_tes.dbf και pro_tes.dbf και με τις συναρτήσεις (fPEI) και (fPRO) γίνονται οι εργασίες που θέλουμε να εκτελέσουμε και οι αλλαγές αυτές αποθηκεύονται στα προηγούμενα αρχεία .

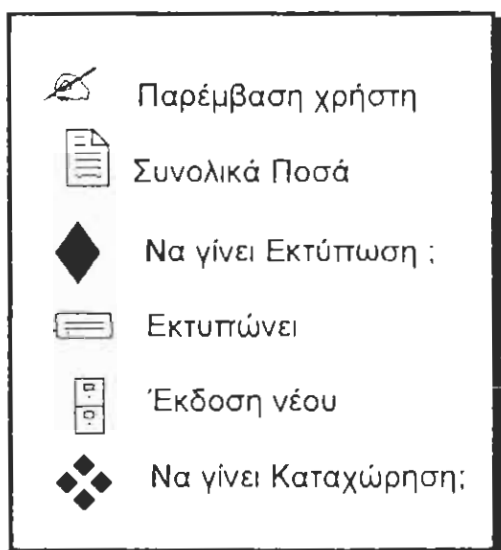
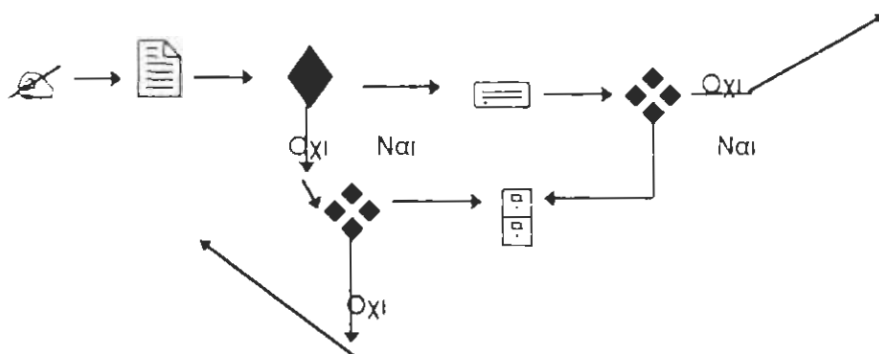
Στην συνέχεια καταχωρούμε τα στοιχεία των προϊόντων στο παραστατικό . Η διαδικασία έχει ως εξής : Δίνοντας τον κωδικό ανοίγει το αρχείο Arh_prd.dbf και εμφανίζεται αυτόματα η περιγραφή του παραστατικού αφού ελεχθεί προηγουμένως εάν υπάρχει .

Με το αρχείο προϊόντων Aprgd.prg μεταφερόμαστε στην διαδικασία " Εισαγωγή - μεταβολή - διαγραφή προϊόντων " .Μέσω της dbedit εμφανίζονται στην οθόνη τα δεδομένα του αρχείου Arh_prd.dbf και με την συνάρτηση (fPRD)γίνονται οι εργασίες που θέλουμε και οι αλλαγές αποθηκεύονται στο ίδιο αρχείο .Όλα αυτά τα δεδομένα κατά τη διάρκεια έκδοσης παραστατικών αποθηκεύονται στο B_Tim.dbf προσωρινά με την συνάρτηση (fA)για να μεταφερθούν με το κλείσιμο του παραστατικού στα κύρια αρχεία και το B_Tim.dbf να αδειάσει για να χρησιμοποιηθεί για την επόμενη

έκδοση ή εμφάνιση παραστατικού . Με απλά λόγια θα λέγαμε ότι το B_Tim.dbf είναι ένα βοηθητικό αρχείο που χρησιμοποιείται σαν "καροτσάκι " για να παίρνει τα στοιχεία που του δίνει ο χρήστης να τα κρατάει προσωρινά και μετά την επιβεβαίωση της καταχώρισης να τα στέλνει στα κύρια αρχεία . Τα παρακάτω σχήματα είναι κατατοπιστικά της όλης διαδικασίας.

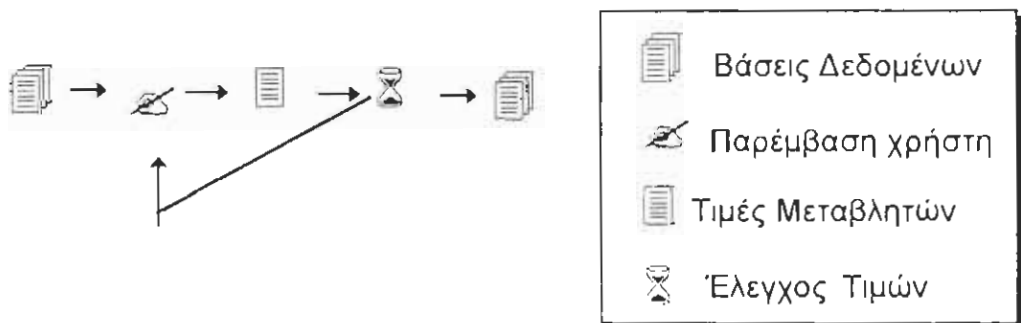


Αφού ολοκληρώσουμε την διαδικασία εισαγωγής των δεδομένων, πρέπει να γίνει οριστικοποίηση αυτών. Περνούμε σε μία διαδικασία όπου πατώντας το πλήκτρο **ctrl+f1** μέσω των συναρτήσεων που δίνονται στο **Par.Prg** γίνονται οι υπολογισμοί και εμφανίζεται στην οθόνη πίνακας με τα γενικά σύνολα αξίας ποσότητας, έκπτωσης και πληρωτέου ποσού. Σε αυτό το σημείο μπορεί να γίνει η εκτύπωση του παραστατικού επιβεβαιώνοντας το σχετικό μήνυμα. Επόμενη εργασία είναι η καταχώρησή του όπου το **B_tim.Dbf** που περιέχει τα αναλυτικά στοιχεία του κάθε παραστατικού αποθηκεύεται στο **Tim.Dbf** και μένει κενό. Επίσης τα γενικά στοιχεία και σύνολα του παραστατικού αποθηκεύονται στο **Tmaster.Dbf**. Η σχηματική παράσταση δείχνει τη διαδικασία.

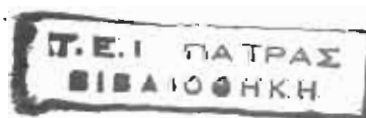


Αρχείο Παραστατικών :

Η επεξεργασία του Αρχείου Παραστατικών γίνεται μέσα από το αρχείο Apar.Prg. Ανοίγουμε το αρχείο Par_ika.dbf το οποίο περιέχει τον κωδικό του παραστατικού , την περιγραφή του , μία ανάλυση της περιγραφής του , τις κινήσεις της αποθήκης , την ποσότητα και την αξία. Με την Dbedit που έχουμε ορίσει εμφανίζονται τα πεδία του αρχείου αυτού στην οθόνη και με την συνάρτηση (fPARAST) μπορούμε να μεταβάλουμε, να διαγράψουμε , να εκτυπώσουμε δεδομένα και να εμφανίσουμε την ανάλυση της περιγραφής του παραστατικού . Μετά το τέλος των εργασιών που θέλουμε οι αλλαγές αποθηκεύονται στο Αρχείο Par_ika.Dbf . Το παρακάτω σχήμα βοηθά στην κατανόηση των όσων αναφέρθηκαν πιο πάνω.



Μία άλλη λειτουργία του Αρχείου Παραστατικών είναι ότι με τα τρία τελευταία πεδία του προσδιορίζει αν η κίνηση στην αποθήκη θα είναι εισαγωγή ή εξαγωγή . και με τα (+) και (-) ενημερώνεται η αποθήκη θετικά ή αρνητικά . Το επόμενο σχήμα περιγράφει επακριβώς τις κινήσεις .



Αρχείο παραστατικών.

Par_ika.Dbf

Πεδίο	1	2	3
kod_par	01	04	06
Per_par	ΤΠ/ΔΑ	ΤΑ/ΔΠ	Δελτ.παραλαβής
Per	πώληση	για αγορά	για αποστολή
	εμπορευμάτων	προϊόντων	Εμπορευμάτων
kod	2	1	1
Posotita	+	+	+
ajla	+	+	



3... Εισαγωγή-Ηεταβολή-Παράστικων Παραστατικών

Κωδ.	Περιγραφή	Κ	Π	Α
81	Τιμολόγιο Πόλησης-Δελτίο Αποστολής	3	+	+
82	Τιμολόγιο Πόλησης-Δελτίο Αποστολής	3	+	+
83	Τιμολόγιο Πόλησης	3	+	+
84	Τιμολόγιο Αγοράς-Δελτίο Παραλαβής	3	+	+
85	Πιστωτικό Τιμολόγιο	1	+	+
86	Δελτίο Αποστολής	2	+	-

Enter-Ηεταβολή ins: εισαγωγή del: διαγραφή F3: Εκτύπωση F10: Αναμ. Περιγρ.

Enter--del-- ins

Πεδίο	1	2
kod_par	01	04
Per_par	ΤΠ/ΔΑ	ΤΑ/ΔΠ
Per	πώληση	για αγορά
	εμπορευμάτων	προϊόντων
kod	2	1
Posotita	+	+
ajla	+	+



3... Εισαγωγή-Ηεταβολή-Παράστικων Παραστατικών

Κωδ.	Περιγραφή	Κ	Π	Α
81	Τιμολόγιο Πόλησης-Δελτίο Αποστολής	3	+	+
82	Τιμολόγιο Πόλησης-Δελτίο Αποστολής	3	+	+
83	Τιμολόγιο Πόλησης	3	+	+
84	Τιμολόγιο Αγοράς-Δελτίο Παραλαβής	3	+	+
85	Πιστωτικό Τιμολόγιο	1	+	+

Enter-Ηεταβολή ins: εισαγωγή del: διαγραφή F3: Εκτύπωση F10: Αναμ. Περιγρ.

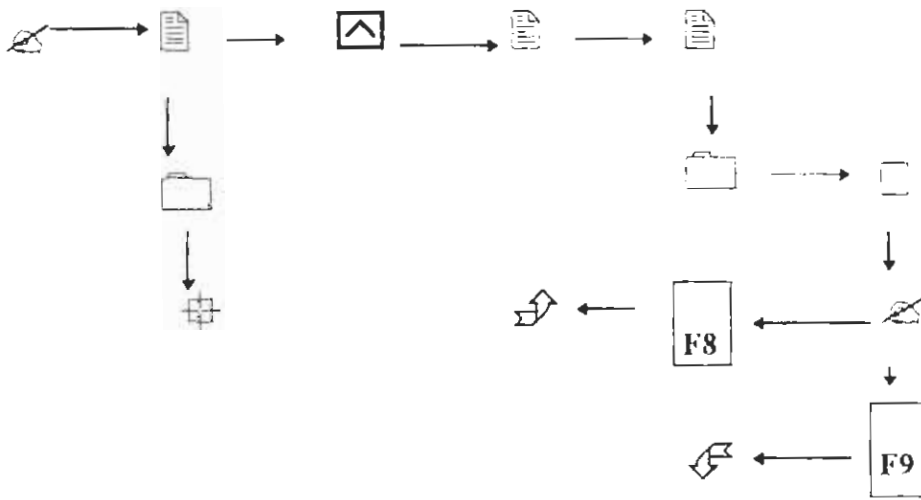
Εμφάνιση Παραστατικών









Η εμφάνιση Παραστατικών γίνεται μέσα από το αρχείο EM-PAR.PRG . Ανοίγουμε πρώτα το αρχείο δεδομένων TMASTER.DBF, μετά το TIM.DBF και κατόπιν το βοηθητικό αρχείο B-TIM.DBF , το οποίο είναι κενό . Σε αυτό μεταφέρουμε συγκεκριμένα δεδομένα από το κύριο αρχείο TIM.DBF και αφού τελειώσει η διαδικασία ,δηλαδή εμφανισθούν από το B-TIM.DBF στην οθόνη για να τα δούμε , φεύγοντας από αυτό σβήνονται όλα τα δεδομένα από το βοηθητικό αρχείο B-TIM.DBF ώστε να μείνει κενό .

Ανάλυση της διαδικασίας που γίνεται μέσω της εντολής DBEDIT στο αρχείο EM_PAR.PRG .

Με την πρώτη DBEDIT έχουν εμφανισθεί στην οθόνη του υπολογιστή πεδία του αρχείου δεδομένων TMASTER.DBF , στο οποίο υπάρχουν καταχωρημένα όλα τα παραστατικά με τα σύνολά τους . Σε περίπτωση που ο χρήστης επιλέξει ένα συγκεκριμένο παραστατικό (μέσω της συνάρτησης fEMPAR) για να δει , τότε ανοίγει το αρχείο δεδομένων TIM.DBF (στο οποίο έχουν καταχωρηθεί τα αναλυτικά στοιχεία κάθε παραστατικού) και εντοπίζονται μέσα σε αυτό οι εγγραφές που αφορούν στο συγκεκριμένο παραστατικό .

Αυτές οι εγγραφές αντιγράφονται στο αρχείο B_TIM.DBF και μέσω μιας νέας DBEDIT εμφανίζεται στην οθόνη το B_TIM.DBF . Επίσης μέσω της συνάρτησης fPAR1 της νέας αυτής DBEDIT πραγματοποιούνται τα εξής : εάν ο χρήστης πατήσει το πλήκτρο F8 ή F9 ο κώδικας προγράμματος επιστρέφει στην προηγούμενη DBEDIT- μέσω της οποίας εμφανίζεται το αρχείο TMASTER.DBF - μεταφέρει τον ενδείκτη εγγραφών στην προηγούμενη εγγραφή του αρχείου (ή στην επόμενη) και επιλέγει αυτόματα αυτή την εγγραφή χωρίς να χρειαστεί να πληκτρολογήσει κάτι ο χρήστης . Τα επόμενα σχήματα δείχνουν την παραπάνω διαδικασία .



-  Παρέμβαση Χρήστη
-  Αρχεία
-  DBEDIT
-  Εμφάνιση συνόλου παρ/κών
-  Εμφάνιση επιλεγμένου παρ/κου
-  Συνάρτηση fMPAR
-  Εμφάνιση προηγούμενου παρ/κου
-  Εμφάνιση επόμενου παρ/κου

Πεδίο	1	2	3
Kod par	01	01	04
Per par	Τιμ.Π/ΔΑ	Τιμ.Π/Δ	Τιμ.Αγ/ΔΠ
Sel par	A	A	B
Art par	1	1	1
Kod PP	012423	011111	001-04678
Afm pp	89977097	789797990707087	
Epo pp	Μάντακας	Αγρίου	Τερζόπ.
Hme par	15/08/97	15/08/97	15/08/97
S-pos	1	2	3
S-alla	23000	26450	46800
S-ekp	0	350	350
S k.AJ	23000	26100	46450
S_fpa	4140	4698	8361
P_syn	27140	30798	54811

Κωδ.	Περιγραφή παραστατικού	Σειρ	Αριθμός	Ημερομηνία
01	Τιμολόγιο Πώλησης-Ποσότητας	A	1	15/08/97
01	Τιμολόγιο Πώλησης-Ποσότητας	A	1	15/08/97
01	Τιμολόγιο Πώλησης-Ποσότητας	A	1	15/08/97
02	Τιμολόγιο Πώλησης-Ποσότητας	A	2	15/08/97
04	Τιμολόγιο Αγοράς-Ποσότητας	B	1	15/08/97
06	Ποσότητα Αποστολής	A	1	15/08/97

Πεδίο	1	1	2
kod par	01	01	06
sel par	A	A	A
art par	1	1	1
kod-prd	001-002	006-870	009-008
per_prd	Παντελόνι	Μπλούζα	Παλτ
pos	1	1	1
tim	17500	8950	5280
pek%	0	0	10
k_aj	17150	8950	4752
			0

DBEDIT

B_TIM.DBF

Κωδικός	Περιγραφή	Είδος	Ποσότητα	Τιμή	Εκπ.	Αξία
001-002	Παντελόνι Lee	1	1	17500	2	17150
006-678	Μπλούζα Βαμβακερή	1	1	8950	0	8950
Σύνολα						26100

Πεδίο	1	1
kod prd	001-002	006-670
Per prd	Παντελ.	Μπλούζ
Pos	1	1
Tim	17500	8950
Pek%	2	0
K_aj	17150	8950
Pfpa	8	8
Ekp	350	0
Alia	17500	8950
FPA		

DBEDIT

Πεδίο	1	2	3
Kod_par	01	01	04
Per_par	Τιμ.Π/ΔΑ	Τιμ.Π/ΔΑ	Τιμ.Αγ/ΔΠ
Sel_par	-	A	B
Arl_par	1	1	1
Kod_PP	012423	011111	001-04678
Afm_pp	89977097	78979791	90707087
Epo_pp	Μάντακας Αγρίου	Τερζόπ.	
Hme_par	15/08/97	15/08/97	15/08/97
S-pos	1	2	3
S-ajla	23000	26450	46800
S_ekp	0	350	350
S_kAJ	23000	26100	46450
S_fpa	4140	4698	8361
P_syn	27140	30798	54811

F8

2... Επεξεργασία Παραστατικών			
Κωδ.	Περιγραφή παραστατικού	Σειρά	Ημερομηνία
01	Τιμολόγιο Πλήρους-Θελίτο Αποστολής	A	15/08/97
01	Τιμολόγιο Πλήρους-Θελίτο Αποστολής	A	15/08/97
02	Τιμολόγιο Πλήρους-Θελίτο Αποστολής	B	15/08/97
04	Τιμολόγιο Ανορθο-Θελίτο Παραστέλλο	B	15/08/97
06	Θελίτο Αποστολής	A	15/08/97
Επιλεγμένο Παραστατικό			

2... Επεξεργασία Παραστατικών				Τιμολόγιο Πλήρους-Θελίτο Αποστολής			
Υπόδος :	01	Ημ/νύα :	15/08/97	Κωδικός :	012423	Α.Φ.Μ. :	89977897
Σειρά :		Αριθμός :	1	Επωνυμία :	ΜΑΝΤΑΚΑΣ		
Κωδικός	Περιγραφή	Είδους	Ποσότητα	Τιμή	Εκπ.	Αξία	
015-899	Μηκίτσι		1	23000	0	23000	
Σύνολο			1			23000	
F8: Επεξεργασία παραστ.				F10: Ανάθεση παραστ. (ESC)			

B_T1M.DBF T1M.DBF

Πεδίο	1	2	2
kod_par	01	01	01
sel_par	-	A	A
arl_par	1	1	1
kod-prd	015-099	001-002	006-670
per_prd	μηκίτσι	Παντελόνι	Μπλουζα
pos	1	1	1
tlm	23000	17500	8950
pek%	0	2	0
k_aj	23000	17150	8950

Πεδίο	1
kod_prd	015-099
Per_prd	Μηκίτσι
Pos	1
Tlm	23000
Pek%	0
K_aj	23000
Pfpa	18
Ekp	
Ajla	27140
FPA	4140

Μια ακόμη ενέργεια που γίνεται στην λειτουργία εμφάνισης παραστατικών είναι η ανάλυση του παραστατικού με το πλήκτρο F10 . Πατώντας ο χρήστης το F10 το πρόγραμμα ανοίγει και χρησιμοποιεί το αρχείο Tmaster.dbf όπου μέσω της Dbedit εμφανίζει στην οθόνη τη μάσκα του παραστατικού , μέσα στην οποία εμφανίζεται ένα πινακάκι που περιέχει τα τελικά στοιχεία του παραστατικού (αξία , έκπτωση , ΦΠΑ , και πληρωτέο ποσό . Αυτή η ενέργεια χρησιμοποιείται όταν ο χρήστης θέλει να δει τα τελικά στοιχεία για κάθε ένα από τα παραστατικά που έχει εκδώσει . Διακόπτεται με το πλήκτρο Esc . Το σχήμα που ακολουθεί δείχνει τον τρόπο λειτουργίας του F10 .

F10

TMMASTER.DBF

Πεδίο	1	2	3
Kod par	01	01	04
Per par	Τιμ.Π/ΔΑ	Τιμ.Π/ΔΑ	Τιμ.Αν/ΔΠ
Sel par	A	A	B
Art par	1	1	1
Kod PP	012423	011111	001-04678
Afm pp	89977097	78979791	90707087
Epo pp	Μάντακας	Αγρίου	Τερζόπ.
Hme par	15/08/97	15/08/97	15/08/97
S-pos	1	2	3
S-alla	23000	26450	46800
S-ekp	0	350	350
S-k.AI	23000	26100	46450
S-tpa	4140	4688	8361
P_syn	27140	30798	54811



2... Εμφάνιση Παραστατικών		Τιμολόγιο Πέλησης-Θερίτο Αποστολής	
Τίτλος : 01	Ημ/ν(α) : 15/08/97	Κωδικός : 011111	Α.Φ.Η. : 78979791
Σειρά : A	Αριθμός : 1	Επωνυμία : ΑΓΡΙΟΥ	
Κωδικός	Περιγραφή	Ποσότητα	Μηνιαία
801-002	Παυτ		
006-678	Μηνια		
	Ευνοϊκή Αξ(α) : 26450	2	17158
	Σύνολο Εκπτώσης : 350	0	8938
	Καθαρή Αξ(α) : 26100		
	Αξ(α Φ.Π.Α. : 4698		
	Πληρωτέο Ποσό : 30798		
		Σύνολο	26100
			(F50) Εξόφλ.

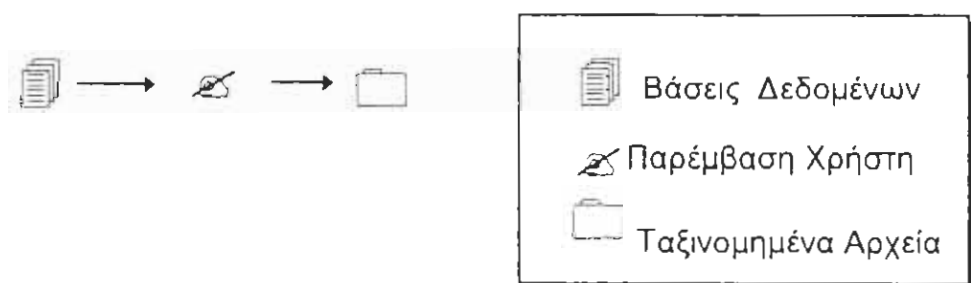
Ανάλυση παραστατικού με F10

Βοηθητικό Τμήμα

Το βοηθητικό τμήμα λειτουργεί μέσα από το αρχείο Boh.Prg. Σ' αυτό δεν χρειάζεται κανένα αρχείο Δεδομένων . Περιέχεται ένα μενού επιλογών "Αναδιοργάνωση--Σελιδοποίηση--Στοιχεία επιχείρησης " .

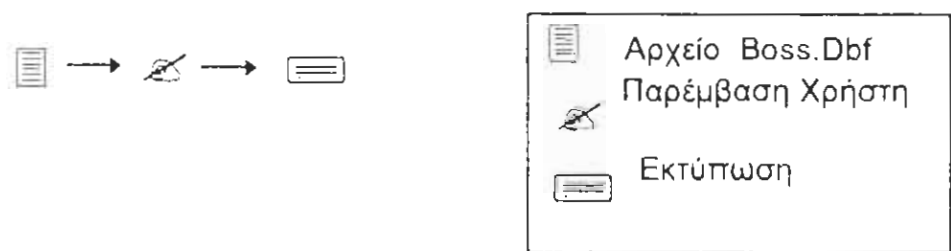
Αναδιοργάνωση :

Γίνεται μέσα από το αρχείο Boh1.Prg Ανοίγουμε όλα τα αρχεία δεδομένων που έχουμε ταξινομήσει και τα αναδιοργανώνουμε , δημιουργούμε δηλαδή εκ νέου ταξινομημένα αρχεία (*.ntx). Συγκεκριμένα ανοίγουμε τα εξής αρχεία δεδομένων : Ari-Par.Dbf , Arh-prd.Dbf, Pel-Tes.Dbf , Pro-Tes.Dbf , Par-ika.Dbf , Tim.Dbf , και Tmaster.Dbf και αφού ταξινομηθούν κλείνουν αυτόματα από το πρόγραμμα .



Σελιδοποίηση :

Η Σελιδοποίηση γίνεται μέσα από το αρχείο Προγράμματος Boh2.Prg . Ανοίγουμε το αρχείο Boss.Dbf γιατί θέλουμε να εκτυπώσουμε στοιχεία από αυτό . Στην διαδικασία αυτή δεν αποθηκεύουμε στοιχεία σε κανένα αρχείο Δεδομένων .



boss.dbf

Πεδίο	1
Epo	Αλφα-Βήτα Ε.Π.Ε
Epa	Έμπορος
Dle	Γούναρη 55 Πάτρα
Doy	Α' Πατρών
Afm	98493223

Επωνυμία : Αλφα-Βήτα Ε.Π.Ε
 Επάγγελμα: Έμπορος
 Διεύθυνση: ΓΟΥΝΑΡΗ 55 ΠΑΤΡΑ
 Δ.Ο.Υ. : Α' ΠΑΤΡΩΝ
 Α.Φ.Μ. : 98493223



Boss.Dbf

4/4... ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ	
Επωνυμία :	Μανδέλιος Σπύρ Α.Ε
Επάγγελμα :	Έμπορος
Διεύθυνση :	Παπαστράτου 30 Αγρίνιο
Δ.Ο.Υ. :	Α' ΑΓΡΙΝΙΟΥ
Α.Φ.Μ. :	98493223
(Enter) : Αποδοχή στοιχείου (ESC) : Εξόδος	



Πεδίο	1
Epo	Μανδέλιος Α.Ε
Epa	Έμπορος
Dle	Παπαστράτου 30 Αγρίνιο
Doy	Α' Αγρινίου
Afm	98493223

4.5 Έλεγχος και δοκιμή του προγράμματος

Κατά τη διαδικασία της κωδικοποίησης των PRG τρέχαμε το πρόγραμμα σταδιακά για να δούμε τι πετύχαμε και τι εκκρεμότητες υπάρχουν . Βλέποντας τις εκκρεμότητες κάναμε τις ανάλογες διορθώσεις και ξανατρέχαμε το πρόγραμμα έως ότου πετύχουμε αυτό που θέλουμε . Εδώ θέλουμε να τονίσουμε ότι απαιτείται ιδιαίτερη προσοχή , υπομονή και φαντασία . Φαντασία όμως απαιτείται περισσότερο κατά το σχεδιασμό του προγράμματος σε χαρτί .

Κατόπιν κάνουμε έναν έλεγχο του προγράμματος να προσθέσουμε τυχόν ελλείψεις , να διορθώσουμε κάποιες εκκρεμότητες , να αφαιρέσουμε περιττά πράγματα και να κάνουμε την τελική διαμόρφωση η οποία πρέπει να αρέσει στο χρήστη και να μην του δημιουργεί πλήξη.

Τέλος κάνουμε τη δοκιμή και το πρόγραμμα είναι έτοιμο για χρήση συνοδευόμενο και από ένα εγχειρίδιο χρήσης ,το οποίο είναι απαραίτητο για την εκμάθηση του προγράμματος .

"Παράρτημα"

Δομή Αρχείου (DBF)

Όνομασία Αρχείου		PAR_IKA (SELE 1)		
------------------	--	--------------------	--	--

A/A	Όνομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD-PAR	C	2	Κωδικός παραστατικού
2	PER_PAR	C	40	Περιγραφή παραστατικού
3	PER	C	40	Ανάλυση περιγραφής παραστατικού
4	KOD	C	1	Κίνηση 1 - 2
5	POSOTHTA	C	1	(+), (-), ()
6	AJIA	C	1	(+), (-), ()
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Ονομασία Αρχείου		ARI_PAR (SELE 2)		
A/A	Ονομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	ARI_PAR	N	6	Αριθμός παραστατικού
2	KOD_PAR	C	2	Κωδικός παραστατικού
3	SEI_PAR	C	2	Σειρά παραστατικού
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Όνομασία Αρχείου		TMASTER (SELE 3)		
A/A	Όνομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD_PAR	C	2	Κωδικός παραστατικού
2	PER_PAR	C	40	Περιγραφή παραστατικού
3	SEI_PAR	C	2	Σειρά παραστατικού
4	ARI_PAR	N	6	Αριθμός παραστατικού
5	KOD_PP	C	13	Κωδικός πελάτη - προμηθευτή
6	AFM_PP	C	8	Α.Φ.Μ.
7	EPO_PP	C	30	Επωνυμία
8	HME_PAR	D	8	Ημερομηνία παραστατικού
9	S_POS	N	5	Σύνολο ποσότητας
10	S_AJIA	N	11	Σύνολο Αξίας
11	S_EKP	N	11	Σύνολο έκπτωσης
12	S_K_AJ	N	11	Σύνολο καθαρής αξίας
13	S_FPA	N	11	Σύνολο Φ.Π.Α.
14	P_SYN	N	12	Πληρωτέο σύνολο
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Ονομασία Αρχείου		B_TIM (SELE 4)		
A/A	Ονομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD_PRD	C	11	Κωδικός προϊόντος
2	PER_PRD	C	28	Περιγραφή προϊόντος
3	POS	N	3	Ποσότητα
4	TIM	N	6	Τιμή
5	PEK	N	2	Ποσοστό έκπτωσης
6	K_AJ	N	9	Καθαρή αξία(ποσότητα x τιμή - έκπτωση)
7	PFPA	N	2	Ποσοστό Φ.Π.Α.
8	EKP	N	9	Έκπτωση
9	AJIA	N	9	Αξία (ποσότητα x τιμή)
10	FPA	N	9	Φ.Π.Α.
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Ονομασία Αρχείου		TIM (SELE 5)		
A/A	Ονομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD_PAR	C	2	Κωδικός παραστατικού
2	SEI_PAR	C	2	Σειρά παραστατικού
3	ARI_PAR	N	6	Αριθμός παραστατικού
4	KOD_PRD	C	11	Κωδικός προϊόντος
5	PER_PRD	C	28	Περιγραφή προϊόντος
6	POS	N	3	Ποσότητα
7	TIM	N	6	Τιμή
8	PEK	N	2	Ποσοστό έκπτωσης
9	K_AJ	N	9	Καθαρή αξία(ποσότητα x τιμή - έκπτωση)
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Ονομασία Αρχείου		BOSS (SELE 6)		
A/A	Ονομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	EPO	C	30	Επωνυμία επιχείρησης
2	EPA	C	30	Επάγγελμα
3	DIE	C	30	Διεύθυνση
4	DOY	C	20	Δ.Ο.Υ.
5	AFM	C	8	Α.Φ.Μ.
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

	Όνομασία Αρχείου	PEL-ΤΕΣ (SELE 7) / PRO_ΤΕΣ (SELE 8)		
--	------------------	---	--	--

A/A	Όνομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD	C	13	Κωδικός πελάτη / προμηθευτή
2	EPO	C	30	Επωνυμία πελάτη / προμηθευτή
3	AFM	C	8	Α.Φ.Μ. πελάτη /προμηθευτή
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

Δομή Αρχείου (DBF)

Ονομασία Αρχείου		ARX_PRD (SELE 9)		
A/A	Ονομασία	Τύπος	Μήκος	Περιγραφή πεδίου
1	KOD_PRD	C	11	Κωδικός προϊόντος
2	PER_PRD	C	28	Περιγραφή προϊόντος
3	TIM	N	6	Τιμή
4	PFPA	N	2	Ποσοστό Φ.Π.Α.
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

"Κώδικας προγράμματος - PRG"

* **PTYX.PRG** *

SET DATE BRITISH
 SET CONFIRM ON
 SET SCOREBOARD OFF
 SET WRAP ON

PUBLIC

Q1,Q2,Q3,Q4,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13,Q14,Q15,Q16,Q17,Q18,mHELP,m
 HELP1

Q1="Επιλογή με (" +CHR(24)+"),(" +CHR(25)+"),(Home),(End),(Enter) "

Q2=" Enter:Μεταβολή Ins:Εισαγωγή Del:Διαγραφή F5:Εκτύπωση F10:Ανάλ Περιγρ. "

Q3=" (" +CHR(24)+"),(" +CHR(25)+"): Enter :Μετακίνηση Αποδοχή στοιχείων

Ins:Εισαγωγή (Esc) Έξοδος"

Q4=" Αποδοχή Στοιχείων με (Enter). (Esc) Έξοδος "

Q6="Μετακίνηση στην Επωνυμία με (Enter). Πίνακας Προμ-Πελατ με (Esc) Έξοδος"

Q7="Μετακίνηση στο κωδ με (Enter). Πίνακ προμ-πελατ με (+). (Esc) Έξοδος "

Q8="Ο κωδικός δεν υπάρχει Δώστε (Ins) για εισαγωγή, (Enter) για συνέχεια."

Q9=" Η επωνυμία δεν υπάρχει Δώστε :Ins για εισαγωγή, Enter για συνέχεια."

Q10=" (" +CHR(24)+"),(" +CHR(25)+"):Μετακίνηση (Enter):Αποδοχή στοιχείων ,
 (Esc) Έξοδος "

Q11=" Μετακίνηση στην περιγραφή με (Enter). Πίνακας ειδών (+). (Esc) Έξοδος "

Q12="Μετακίνηση στον κωδικό με (Enter). Πίνακας ειδών (+). (Esc) Έξοδος "

Q13=" Enter:Μεταβολή Ins:Εισαγωγή Del:Διαγραφή F5:Εκτύπωση (Esc) Έξοδος "

Q14="Ηπεριγραφή δεν υπάρχει Δώστε (Ins) για εισαγωγή. (Enter) για συνέχεια"

Q15="Δώστε (30) για πελάτες, (50) για προμηθευτές. (Esc) Έξοδος "

Q16=" Enter:Μεταβολή Ins/Ctrl+N: Εισαγωγή Del: Διαγραφή Ctrl+F1:Κλείσιμο "

Q17="Μετακίνηση με " +CHR(24)+"),(" +CHR(25)+"),(PgDn),(PgUp). Επιλογή με
 (Enter). (Esc) Έξοδος "

Q18=" F8:Προηγούμενο παραστατικό F9: Επόμενο παραστατικό.

F10:Ανάλυση παραστατικού. (Esc) "

mHELP="77"

mHELP1=0

SET PROCEDURE TO LOCK
 SET PROCEDURE TO FUNC
 SET PROCEDURE TO ROYT
 DO MENU

```

*-----*
*  LOCK . PRG  *
*-----*
FUNCTION NET_USE
*-----*
PARAMETERS mFILE,mEXCLU
mTIME=5
DO WHILE .T.
  IF mEXCLU
    USE &mFILE EXCLUSIVE
  ELSE
    USE &mFILE
  ENDIF
  IF .NOT. NETERR()
    RETURN (.T.)
  ENDIF
  IF mTIME > 0
    INKEY(1)
    mTIME=mTIME-1
  LOOP
  ENDIF
  SAVE SCREEN
  SET COLOR TO I
  @ 08,16 SAY " "
  @ 09,16 SAY " Η προσπάθεια ανοίγματος αρχείου δεν πέτυχε. "
  @ 10,16 SAY " "
  @ 11,16 SAY " Το αρχείο δεν είναι διαθέσιμο αυτή τη στιγμή. "
  @ 12,16 SAY " "
  @ 13,16 SAY " Βρίσκεται σε χρήση από άλλο χρήστη....."
  @ 14,16 SAY " "
  SET CURS ON
  SET COLOR TO
  mYN=YesNo(" Να προσπαθήσω πάλι για λίγο .....(N/O) ")
  SET CURS OFF
  RESTORE SCREEN
  IF mYN=1
    mTIME=5
  LOOP
  ENDIF
  SET CURS ON
  RETURN (.F.)
ENDDO
*-----*
FUNCTION APP_LOC
*-----*
mTIME=5
DO WHILE .T.
  APPEND BLANK
  IF .NOT. NETERR()
    RETURN (.T.)

```

```

ENDIF
IF mTIME > 0
  INKEY(1)
  mTIME=mTIME-1
  LOOP
ENDIF
SAVE SCREEN
SET COLOR TO I
@ 08,16 SAY " "
@ 09,16 SAY " Η προσπάθεια ανοίγματος αρχείου δεν πέτυχε. "
@ 10,16 SAY " "
@ 11,16 SAY " Το αρχείο δεν είναι διαθέσιμο αυτή την στιγμή. "
@ 12,16 SAY " "
@ 13,16 SAY " Βρίσκεται σε χρήση από άλλο χρήστη..... "
@ 14,16 SAY " "
SET CURS ON
SET COLOR TO
mYN=YesNo(" Να προσπαθήσω πάλι για λίγο..... (N/O) ")
SET CURS OFF
RESTORE SCREEN
IF mYN=1
  mTIME=5
  LOOP
ENDIF
SET CURS ON
RETURN (.F.)
ENDDO
*-----*
FUNCTION REC_LOC
*-----*
mTIME=5
DO WHILE .T.
  IF RLOCK()
    RETURN (.T.)
  ENDIF
  IF mTIME > 0
    INKEY(1)
    mTIME=mTIME-1
  LOOP
ENDIF

```

```
SAVE SCREEN
SET COLOR TO I
@ 08,16 SAY "
@ 09,16 SAY " Η προσπάθεια ανοίγματος αρχείου δεν πέτυχε. "
@ 10,16 SAY "
@ 11,16 SAY " Το αρχείο δεν είναι διαθέσιμο αυτή τη στιγμή. "
@ 12,16 SAY "
@ 13,16 SAY " Βρίσκεται σε χρήση από άλλο χρήστη..... "
@ 14,16 SAY "
SET CURS ON
SET COLOR TO
mYN=YesNo(" Να προσπαθήσω πάλι για λίγο.... (N/O) ")
SET CURS OFF
RESTORE SCREEN
IF mYN=1
  mTIME=5
  LOOP
ENDIF
SET CURS ON
RETURN (.F.)
ENDDO
*-----*
```



```

*-----*
*      FUNC . PRG      *
*-----*
*-----*
FUNCTION YesNo
*-----*
PARAMETERS mMHNYMA,mTIMH
IF PCOUNT()=1
  mTIMH="0"
ENDIF
TONE(499,3)
TONE(700,3)
@ 22,02 SAY SPACE(76)
SET COLOR TO 0/7
SET CURS ON
DO WHILE .T.
  mN_O=mTIMH
  @ 22,02 SAY SUBS(mMHNYMA,1,76) GET mN_O
  READ
  IF mN_O $ "NnNvOo0o" .AND. .NOT. LASTKEY()=27
    EXIT
  ENDIF
ENDDO
SET CURS OFF
SET COLOR TO 7
@ 22,02 SAY SPACE(76)
RETURN IF(mN_O $ "NnNv",1,0)
*-----*
*-----*
FUNCTION dSCREEN
*-----*
PARAMETERS name
fname=name+".mem"
IF FILE(fname)
  RESTORE FROM &name ADDITIVE
  RESTORE SCREEN FROM &name
  RELEASE &name
ELSE
  @ 00,00 SAY " Το αρχείο "+fname+" δεν υπάρχει "
  WAIT
ENDIF
RETURN 0
*-----*
*-----*
FUNCTION PRINT_TEST
*-----*
DO MHNHMATA WITH " Ετοιμάσε Εκτυπωτή ...Πίεσε ένα πλήκτρο "
DO WHILE .T.
  IF LASTKEY()=27
    RETURN 0

```

```

ENDIF
IF ISPRINTER()
  DO MHNHYMA WITH "Εκτελείται εκτύπωση ..... Παρακαλώ περιμένετε "
  RETURN 1
ENDIF
DO MHNHMATA WITH " Ο  Εκτυπωτής δεν είναι σε κατάσταση
  λειτουργίας...Επανάλαβε "
ENDDO
*-----*
*-----*
FUNCTION INSERT
*-----*
mAP=FCOUNT()
DECLARE mKENO[mAP]
mRECINS=RECNO()
APPEND BLANK
FOR I=1 TO mAP
  mOP=FIELDNAME(I)
  mKENO[I]=&mOP
NEXT I
SKIP -1
DO WHILE RECNO()>=mRECINS .AND. .NOT. BOF()
  FOR I=1 TO mAP
    mOP=FIELDNAME(I)
    mPP=&mOP
    SKIP
    REPLACE &mOP WITH mPP
    SKIP -1
  NEXT I
  SKIP -1
ENDDO
IF .NOT. BOF()
  SKIP
ENDIF
FOR I=1 TO mAP
  mOP=FIELDNAME(I)
  REPLACE &mOP WITH mKENO[I]
NEXT I
RETURN .T.
*-----*
*-----*
FUNCTION BOX_ONE
*-----*
* Παράθυρο με σκιά
PARAMETERS ROW_1,COL_1,ROW_2,COL_2
@ ROW_1+1,COL_1+1,ROW_2+1,COL_2+1 BOX "  "
@ ROW_1,COL_1,ROW_2,COL_2 BOX "  "

RETURN (.T.)
*-----*

```

```
*-----*  
*   ROYT . PRG   *  
*-----*
```

```
*-----*  
PROCEDURE MHNHYMA  
*-----*
```

```
PARAMETERS mMHNHYM  
SET COLOR TO 0/7  
@ 22,02 SAY SPACE(76)  
@ 22,02 SAY mMHNHYM  
SET COLOR TO 7  
RETURN
```

```
*-----*  
PROCEDURE MHNHMATA  
*-----*
```

```
PARAMETERS mMHNHYMA
```

```
TONE(700,5)  
SET COLOR TO 0/7  
@ 22,02 SAY SPACE(76)  
@ 22,02 SAY mMHNHYMA  
SET COLOR TO 7  
SET CONSOLE OFF  
WAIT ""  
SET CONSOLE ON  
@ 22,02 SAY SPACE(76)  
RETURN
```

```
*-----*
*   MENU . PRG   *
*-----*
DO WHILE .T.
  *-----*
  dSCREEN("S_MENU")
  *-----*
  SET COLOR TO 0/7
  @ 22,69 SAY DATE()
  SET COLOR TO 7
  SET CURS OFF
  *-----*
  mEP=0
  @ 08,24 PROMPT " 1...Έκδοση Παραστατικών  "
  @ 10,24 PROMPT " 2...Εμφάνιση Παραστατικών  "
  @ 12,24 PROMPT " 3...Αρχείο Παραστατικών  "
  @ 14,24 PROMPT " 4...Βοηθητικό Τμήμα      "
  MENU TO mEP
  *-----*
  IF LASTKEY()=27
    SET CURS ON
    RETURN
  ENDIF
  *-----*
  DO CASE
    CASE mEP=1
      DO EK_PAR
    CASE mEP=2
      DO EM_PAR
    CASE mEP=3
      DO APAR
    CASE mEP=4
      DO BOH
  ENDCASE
  *-----*
ENDDO
*-----*
```

```

*-----*
*      EK_PAR.PRG      *
*-----*
SELE 1
IF .NOT. NET_USE("PAR_IKA",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PAR1
*-----*
SELE 2
IF .NOT. NET_USE("ARI_PAR",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO ARI1
*-----*
SELE 3
IF .NOT. NET_USE("TMASTER",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO TIM2
*-----*
SELE 4
IF .NOT. NET_USE("B_TIM",.T.)
  CLOSE ALL
  RETURN
ENDIF
*-----*
SELE 5
IF .NOT. NET_USE("TIM",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO TIM1
*-----*
SELE 7
IF .NOT. NET_USE("PEL_TES",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PEL1,PEL2
*-----*

```

```

SELE 8
IF .NOT. NET_USE("PRO_TES",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PRO1,PRO2
*-----*
SELE 9
IF .NOT. NET_USE("ARX_PRD",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO ARX1,ARX2
*-----*
DO WHILE .T.
  dSCREEN("S_TIM")
  SAVE SCREEN TO mA
  *-----*
  SELE 1
  SET ORDER TO 1
  mKOD_PAR=KOD_PAR
  SET CURS ON
  @ 03,10 GET mKOD_PAR
  READ
  @ 03,10 SAY mKOD_PAR
  IF LASTKEY()=27
    IF YesNo("Επιβεβαίωση Διαγραφής (N/O)")=1
      RETURN
    ENDIF
  LOOP
ENDIF
IF mKOD_PAR="+"
  mHELP1=1
  DO MHNHYMA WITH Q3
  BOX_ONE(03,10,15,60)
  @ 04,17 SAY " "
  SET COLOR TO 0/7
  @ 04,11 SAY " Κωδ. "
  @ 04,18 SAY " Περιγραφή "
  STORE SAVESCREEN(03,11,04,59) TO mS1
  SET COLOR TO 7
  SELE 1
  SET ORDER TO 1
  GO TOP
  DECLARE mPED[2],mTIT[2]
  mPED[1]="KOD_PAR"
  mPED[2]="PER_PAR"
  mTIT[1]=" "
  mTIT[2]=" "

```

```
DBEDIT(04,11,14,59,mPED,"fA",.T.,mTIT)
RESTORE SCREEN FROM mA
IF LASTKEY()=27
  LOOP
ENDIF
ENDIF
SELE 1
SET ORDER TO 1
SEEK mKOD_PAR
IF .NOT. FOUND()
  DO MHNYMA WITH Q8
  TONE(200,5)
  INKEY(0)
  IF LASTKEY()=22
    SAVE SCREEN TO mALAG
    DO APAR
  ENDIF
  LOOP
ENDIF
mPER_PAR=ALLTRIM(PER_PAR)
mPER=ALLTRIM(PER)
mKOD1=KOD
mPP=IIF(mKOD1="1","50","30")
@ 03,10 SAY mKOD_PAR
@ 01,36 SAY mPER_PAR
*-----*
mHME_PAR=DATE()
DO MHNYMA WITH Q4
@ 03,25 GET mHME_PAR
READ
@ 03,25 SAY mHME_PAR
IF LASTKEY()=27 .OR. EMPTY(mHME_PAR)
  LOOP
ENDIF
*-----*
mSEI_PAR=SPACE(2)
DO MHNYMA WITH Q4
@ 04,10 GET mSEI_PAR
READ
@ 04,10 SAY mSEI_PAR
IF LASTKEY()=27
  LOOP
ENDIF
mKEY=mKOD_PAR+UPPER(mSEI_PAR)
SELE 2
SET ORDER TO 1
SEEK mKEY
IF .NOT. FOUND()
  ARI_PAR=0
ENDIF
```

```

mARI_PAR=mARI_PAR+1
@ 04,25 SAY mARI_PAR PICTURE "999999"
*-----*
DO MHNHYMA WITH Q4
@ 04,25 GET mARI_PAR PICTURE "999999" RANGE 1,999999
READ
@ 04,25 SAY mARI_PAR PICTURE "999999"
IF LASTKEY()=27
  LOOP
ENDIF
mKEY8=mKOD_PAR+UPPER(mSEI_PAR)+STR(mARI_PAR,6)
SELE 3
SET ORDER TO 1
SEEK mKEY8
IF FOUND()
  DO MHNHYMA WITH "Το παραστατικό αυτό ήδη υπάρχει . Επαναλάβετε"
  TONE(300.7)
  INKEY(1.5)
  LOOP
ENDIF
*-----*
mKOD_PP=SPACE(13)
DO WHILE EMPTY(mKOD_PP)
  DO MHNHYMA WITH Q15
  @ 03,47 GET mPP
  READ
  IF LASTKEY()=27
    mKOD_PP=mHELP
    LOOP
  ENDIF
*-----*
IF mPP="30"
  SELE 7
ELSEIF mPP="50"
  SELE 8
ELSE
  TONE(300.7)
  LOOP
ENDIF
*-----*
DO MHNHYMA WITH Q6
SAVE SCREEN TO mB
@ 03,47 GET mKOD_PP
READ
@ 03,47 SAY mKOD_PP
*-----*
IF LASTKEY()=27
  mKOD_PP=mHELP
  LOOP
ENDIF
*-----*

```



```

IF EMPTY(mKOD_PP) .AND. LASTKEY()=13
  mEPO_PP=SPACE(30)
  DO WHILE EMPTY(mEPO_PP)
    DO MHNYMA WITH Q7
    SAVE SCREEN TO mK
    @ 04,47 GET mEPO_PP
    READ
    @ 04,47 SAY mEPO_PP
    *-----*
  IF LASTKEY()=27
    mEPO_PP=mHELP
    mKOD_PP=mHELP
    LOOP
  ENDIF
  *-----*
  IF EMPTY(mEPO_PP) .AND. LASTKEY()=13
    mEPO_PP=mHELP
    LOOP
  ENDIF
  *-----*
  IF mPP="30"
    SELE 7
  ELSE
    SELE 8
  ENDIF
  *-----*
  SET ORDER TO 2
  GO TOP
  mKEY=ALLTRIM(mEPO_PP)
  X=LEN(mKEY)
  SET SOFTSEEK ON
  SEEK mKEY
  SET SOFTSEEK OFF
  mEPO=ALLTRIM(EPO)
  mKEY1=LEFT(mEPO,X)
  *-----*
  IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mEPO_PP="+"
    DO MHNYMA WITH Q9
    TONE(200,5)
    INKEY(0)
    IF LASTKEY()=22
      SAVE SCREEN TO mALAG
      IF mPP="30"
        DO APEL
      ELSE
        DO APRO
      ENDIF
    ENDIF
    mEPO_PP=SPACE(30)
    RESTORE SCREEN FROM mK

```

```

    LOOP
ENDIF
*-----*
SKIP
mEPO=ALLTRIM(EPO)
mKEY1=LEFT(mEPO,X)
*-----*
IF .NOT. EOF() .AND. mKEY=mKEY1 .OR. mEPO_PP="+ "
    mHELP1=2
    SKIP -1
    IF mEPO_PP="+ "
        GO TOP
    ENDIF
    DO MHNHYMA WITH Q3
    BOX_ONE(04,25,16,74)
    @ 05,58 SAY " "
    SET COLOR TO 0/7
    @ 05,26 SAY "Επωνυμία          "
    @ 05,59 SAY "Κωδικός          "
    STORE SAVESCREEN(04,25,05,74) TO mS2
    SET COLOR TO 7
    DECLARE mPED[2],mTIT[2]
    mPED[1]="EPO"
    mPED[2]="KOD"
    mTIT[1]="   "
    mTIT[2]="   "
    DBEDIT(05,26,15,73,mPED,"fA"..T..mTIT)
    RESTORE SCREEN FROM mK
    IF LASTKEY()=27
        mEPO_PP=SPACE(30)
        LOOP
    ENDIF
ENDIF
ENDIF
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKEY="+ "
    SKIP -1
ENDIF
mKOD_PP=KOD
mEPO_PP=EPO
mAFM_PP=AFM
@ 03,47 SAY mKOD_PP
@ 03,69 SAY mAFM_PP
@ 04,47 SAY mEPO_PP
ENDDO
LOOP
ENDIF
*-----*

```

```

SET ORDER TO 1
GO TOP
mKEY=ALLTRIM(mKOD_PP)
X=LEN(mKEY)
SET SOFTSEEK ON
SEEK mKEY
SET SOFTSEEK OFF
mKOD=ALLTRIM(KOD)
mKEY1=LEFT(mKOD,X)
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKOD_PP="+
  DO MHNHYMA WITH Q8
  TONE(200,5)
  INKEY(0)
  IF LASTKEY()=22
    SAVE SCREEN TO mALAG
    IF mPP="30"
      DO APEL
    ELSE
      DO APRO
    ENDIF
  ENDIF
  mKOD_PP=SPACE(13)
  RESTORE SCREEN FROM mB
  LOOP
ENDIF
*-----*
SKIP
mKOD=ALLTRIM(KOD)
mKEY1=LEFT(mKOD,X)
*-----*
IF .NOT. EOF() .AND. mKEY=mKEY1 .OR. mKOD_PP="+
  mHELP1=3
  SKIP -1
  IF mKOD_PP="+
    GO TOP
  ENDIF
  DO MHNHYMA WITH Q3
  BOX_ONE(04,25,16,74)
  @ 05,41 SAY " "
  SET COLOR TO 0/7
  @ 05,26 SAY "Κωδικός      "
  @ 05,42 SAY " Επωνυμία          "
  STORE SAVESCREEN(04,25,05,74) TO mS3
  SET COLOR TO 7
  DECLARE mPED[2],mTIT[2]
  mPED[1]="KOD"
  mPED[2]="EPO"
  mTIT[1]="      "
  mTIT[2]="      "

```

```

DBEDIT(05,26,15,73,mPED,"fA",.T.,mTIT)
RESTORE SCREEN FROM mB
IF LASTKEY()=27
    mKOD_PP=SPACE(13)
    LOOP
ENDIF
ENDIF
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKEY="+ "
    SKIP -1
ENDIF
mKOD_PP=KOD
mEPO_PP=EPO
mAFM_PP=AFM
@ 03,47 SAY mKOD_PP
@ 03,69 SAY mAFM_PP
@ 04,47 SAY mEPO_PP
ENDDO
IF mKOD_PP=mHELP
    LOOP
ENDIF
*-----*
KEYBOARD CHR(13)
STORE SAVESCREEN(04,00,07,80) TO mS
*-----*
SELE 4
GO TOP
IF EOF()
    APPEND BLANK
ENDIF
*-----*
DECLARE mPED[6],mTIT[6]
mPED[1]="KOD_PRD"
mPED[2]="PER_PRD"
mPED[3]="POS"
mPED[4]="TIM"
mPED[5]="PEK"
mPED[6]="K_AJ"
*-----*
mTIT[1]="  "
mTIT[2]="  "
mTIT[3]="  "
mTIT[4]="  "
mTIT[5]="  "
mTIT[6]="  "
*-----*
DBEDIT(06,01,18,78,mPED,"fPAR",.T.,mTIT)
*-----*
*-----*
SELE 4

```

```

ZAP
*-----*
ENDDO
*-----*
*-----*
*-----*
*-----*
FUNCTION fPAR
*-----*
RESTSCREEN(04,00,07,80,mS)
PARAMETERS mMODE,mPED
mROW=ROW()
mCOL=COL()
DO CASE
  CASE mMODE=0
    RETURN 1
  CASE mMODE=1
    RETURN 1
  CASE mMODE=2
    RETURN 1
  CASE mMODE=4
    DO CASE
      CASE LASTKEY()=27
        RETURN 0
      CASE LASTKEY()=13
        *-----*
        IF mPED=1
          *-----*
          mKOD_PRD=KOD_PRD
          IF EMPTY(mKOD_PRD)
            DO MHNHYMA WITH Q11
          ELSE
            DO MHNHYMA WITH " Πίνακας Ειδών με (+).    (ESC) Έξοδος"
          ENDIF
          *-----*
          SET CURS ON
          @ mROW,mCOL GET mKOD_PRD
          READ
          SET CURS OFF
          *-----*
        IF LASTKEY()=27
          IF EMPTY(mKOD_PRD)
            mMAR=RECNO()
            DELETE RECORD RECNO()
            PACK
            GO mMAR
          ENDIF
          DO MHNHYMA WITH Q16
          RETURN 2
        ENDIF
        *-----*

```

```

SAVE SCREEN TO mZ
IF EMPTY(mKOD_PRD) .AND. LASTKEY()=13
  KEYBOARD CHR(4)+CHR(13)
  RETURN 1
ENDIF
*-----*
SELE 9
SET ORDER TO 1
mKEY=ALLTRIM(mKOD_PRD)
X=LEN(mKEY)
SET SOFTSEEK ON
SEEK mKEY
SET SOFTSEEK OFF
mKOD=ALLTRIM(KOD_PRD)
mKEY1=LEFT(mKOD,X)
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKOD_PRD="+"
  SELE 4
  mREC=RECNO()
  DO MHNYMA WITH Q8
  TONE(200,5)
  INKEY(0)
  IF LASTKEY()=22
    SAVE SCREEN TO mALAG
    DO APRD
  ENDIF
  SELE 4
  GO mREC
  KEYBOARD CHR(13)
  RETURN 1
ENDIF
SELE 9
SET ORDER TO 1
SKIP
mKOD=ALLTRIM(KOD_PRD)
mKEY1=LEFT(mKOD,X)
*-----*
IF .NOT. EOF() .AND. mKEY=mKEY1 .OR. mKOD_PRD="+"
  mHELP1=4
  SELE 9
  SET ORDER TO 1
  SKIP -1
  IF mKOD_PRD="+"
    GO TOP
  ENDIF
  DO MHNYMA WITH Q3
  BOX_ONE(06,10,18,60)
  @ 07,26 SAY " "
  SET COLOR TO 0/7
  @ 07,11 SAY " Κωδικός "

```

```

@ 07,27 SAY " Περιγραφή Είδους          "
STORE SAVESCREEN(06,10,07,60) TO mS4
SET COLOR TO 7
DECLARE mPEDIO{2},mTIT{2}
mPEDIO{1}="KOD_PRD"
mPEDIO{2}="PER_PRD"
mTIT{1}="          "
mTIT{2}="          "
DBEDIT(07,11,17,59,mPEDIO,"fA",.T.,mTIT)
RESTORE SCREEN FROM mZ
IF LASTKEY()=27
    SELE 4
    KEYBOARD CHR(13)
    RETURN 1
ENDIF
ENDIF
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKEY="+ "
    SELE 9
    SET ORDER TO 1
    SKIP -1
ENDIF
mKOD_PRD=KOD_PRD
mPER_PRD=PER_PRD
mTIM=TIM
mPFPA=PFPA
SELE 4
mR=RECNO()
GO TOP
DO WHILE .NOT. EOF()
    IF KOD_PRD=mKOD_PRD .AND. .NOT. RECNO()=mR
        DO MHNHYMA WITH " Το Είδος υπάρχει .      Επανάλαβε."
        TONE(300.7)
        INKEY(1)
        GO mR
        KEYBOARD CHR(13)
        RETURN 1
    ENDIF
    SKIP
ENDDO
*-----*
SELE 4
GO mR
REPL KOD_PRD WITH mKOD_PRD
REPL PER_PRD WITH mPER_PRD
REPL TIM    WITH mTIM
REPL PFPA   WITH mPFPA
KEYBOARD CHR(4)+CHR(4)+CHR(13)
ENDIF
*-----*

```

```

IF mPED=2
*-----*
  mPER_PRD=PER_PRD
  IF EMPTY(mPER_PRD)
    DO MHNHYMA WITH Q12
  ELSE
    DO MHNHYMA WITH "Πίνακας Ειδών με (+). (ESC) Έξοδος "
  ENDIF
*-----*
  SET CURS ON
  @ mROW,mCOL GET mPER_PRD
  READ
  SET CURS OFF
*-----*
  IF LASTKEY()=27
    IF EMPTY(mPER_PRD)
      mMAR=RECNO()
      DELETE RECORD RECNO()
      PACK
      GO mMAR
    ENDIF
    DO MHNHYMA WITH Q16
    RETURN 2
  ENDIF
*-----*
  SAVE SCREEN TO mE
  IF EMPTY(mPER_PRD) .AND. LASTKEY()=13
    KEYBOARD CHR(19)+CHR(13)
    RETURN 1
  ENDIF
  SELE 9
  SET ORDER TO 2
  mKEY=ALLTRIM(mPER_PRD)
  X=LEN(mKEY)
  SET SOFTSEEK ON
  SEEK mKEY
  SET SOFTSEEK OFF
  mPER=ALLTRIM(PER_PRD)
  mKEY1=LEFT(mPER,X)
*-----*
  IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mPER_PRD="+"
    SELE 4
    mREC=RECNO()
    DO MHNHYMA WITH Q14
    TONE(200,5)
    INKEY(0)
    IF LASTKEY()=22
      SAVE SCREEN TO mALAG
      DO APRD
    ENDIF

```



```

    SELE 4
    GO mREC
    KEYBOARD CHR(13)
    RETURN 1
ENDIF
*-----*
SELE 9
SET ORDER TO 2
SKIP
mPER=ALLTRIM(PER_PRD)
mKEY1=LEFT(mPER,X)
*-----*
IF .NOT. EOF() .AND. mKEY=mKEY1 .OR. mPER_PRD="+
    mHELP1=5
    SELE 9
    SET ORDER TO 2
    SKIP -1
    IF mPER_PRD="+
        GO TOP
    ENDIF
    DO MHNMYA WITH Q3
    BOX_ONE(06,10,18,60)
    @ 07,43 SAY " "
    SET COLOR TO 0/7
    @ 07,11 SAY " Περιγραφή Είδους      "
    @ 07,44 SAY " Κωδικός      "
    STORE SAVESCREEN(06,10,07,60) TO mS5
    SET COLOR TO 7
    DECLARE mPEDIO[2],mTIT[2]
    mPEDIO[1]="PER_PRD"
    mPEDIO[2]="KOD_PRD"
    mTIT[1]="      "
    mTIT[2]="      "
    DBEDIT(07,11,17,59,mPEDIO,"fA"..T..mTIT)
    RESTORE SCREEN FROM mE
    IF LASTKEY()=27
        SELE 4
        KEYBOARD CHR(13)
        RETURN 1
    ENDIF
ENDIF
*-----*
IF EOF() .OR. .NOT. mKEY=mKEY1 .AND. .NOT. mKEY="+
    SELE 9
    SET ORDER TO 2
    SKIP -1
ENDIF
mKOD_PRD=KOD_PRD
mPER_PRD=PER_PRD
mTIM=TIM
mPFPA=PFPA

```

```

SELE 4
mR=RECNO()
GO TOP
DO WHILE .NOT. EOF()
  IF KOD_PRD=mKOD_PRD .AND. .NOT. RECNO()=mR
    DO MHNHYMA WITH " Το Είδος υπάρχει .      Επανελάβε."
    TONE(300.7)
    INKEY(1)
    GO mR
    KEYBOARD CHR(13)
    RETURN 1
  ENDIF
SKIP
ENDDO
*-----*
SELE 4
GO mR
REPL KOD_PRD WITH mKOD_PRD
REPL PER_PRD WITH mPER_PRD
REPL TIM WITH mTIM
REPL PFPA WITH mPFPA
KEYBOARD CHR(4)+CHR(13)
ENDIF
*-----*
IF mPED=3
*-----*
  mPOS=POS
  SET CURS ON
  DO MHNHYMA WITH " Δώστε ποσότητα. "
  @ mROW,mCOL GET mPOS PICTURE "999"
  READ
  SET CURS OFF
  *-----*
  IF EMPTY(mPOS)
    DO MHNHYMA WITH " Ποσότητα μηδενική.      Επαναλάβετε."
    TONE(300.7)
    INKEY(1)
    KEYBOARD CHR(13)
    RETURN 1
  ENDIF
  *-----*
  REPL POS WITH mPOS
  mMAR=RECNO()
  SUM POS TO mS_POS
  IF mS_POS>9999
    DO MHNHYMA WITH " Πρέπει να εκδώσετε άλλο παραστατικό για
    αυτό το προϊόν. "
    TONE(300.7)
    INKEY(1.5)
    GO mMAR

```

```

DELETE RECORD RECNO()
PACK
SUM POS TO mS_POS
SUM K_AJ TO mS_K_AJ
@ 20,46 SAY mS_POS PICTURE "9999"
@ 20,68 SAY mS_K_AJ PICTURE "9999999999"
GO mMAR
DO MHNHYMA WITH Q16
RETURN 2
ENDIF
@ 20,46 SAY mS_POS PICTURE "9999"
GO mMAR
KEYBOARD CHR(4)+CHR(13)
ENDIF
*-----*
IF mPED=4
*-----*
    mTIM=TIM
    SET CURS ON
    DO MHNHYMA WITH " Δώστε Τιμή  ."
    @ mROW,mCOL GET mTIM PICTURE "999999"
    READ
    SET CURS OFF
    *-----*
    IF EMPTY(mTIM)
        DO MHNHYMA WITH " Τιμή μηδενική .           Επαναλάβετε."
        TONE(300.7)
        INKEY(1)
        KEYBOARD CHR(13)
        RETURN 1
    ENDIF
    *-----*
    REPL TIM WITH mTIM
    KEYBOARD CHR(4)+CHR(13)
ENDIF
*-----*
IF mPED=5
*-----*
    mPEK=PEK
    SET CURS ON
    DO MHNHYMA WITH " Δώστε ποσοστό έκπτωσης.           "
    @ mROW,mCOL GET mPEK PICTURE "99"
    READ
    SET CURS OFF
    *-----*
    REPL PEK WITH mPEK
    KEYBOARD CHR(4)+CHR(13)
ENDIF
*-----*
IF mPED=6
*-----*

```

```

SET CURS OFF
mAJIA=POS*TIM
mEKP=mAJIA*(PEK/100)
mEKP1=ROUND(mEKP,0)
mK_AJ=mAJIA-mEKP
mK_AJ1=ROUND(mK_AJ,0)
mFPA=mK_AJ1*(PFPA/100)
mFPA1=ROUND(mFPA,0)
REPL AJIA WITH mAJIA
REPL EKP WITH mEKP1
REPL K_AJ WITH mK_AJ1
REPL FPA WITH mFPA1
mMAR=RECNO()
SUM K_AJ TO mS_K_AJ
IF mS_K_AJ>9999999999
    DO MHNHYMA WITH " Πρέπει να εκδώσετε άλλο παραστατικό για
    αυτό το προϊόν . "
    TONE(300.7)
    INKEY(1.5)
    GO mMAR
    DELETE RECORD RECNO()
    PACK
    SUM POS TO mS_POS
    SUM K_AJ TO mS_K_AJ
    @ 20,46 SAY mS_POS PICTURE "9999"
ENDIF
@ 20,68 SAY mS_K_AJ PICTURE "9999999999"
GO mMAR
KEYBOARD CHR(1)
DO MHNHYMA WITH Q16
ENDIF
RETURN 2
CASE LASTKEY()=7
    *
    SELE 4
    mD=" "+KOD_PRD+" | "+PER_PRD+" | "+STR(POS)+" | "+STR(TIM)+"
    | "+STR(PEK)+" | "+STR(K_AJ)+" "
    SET COLOR TO 0/7
    @ mROW,01 SAY mD
    SET COLOR TO 7
    *-----*
    IF YesNo(" Επιβεβαίωση Διαγραφής (N/O) ")=1
        mMAR=RECNO()
        DELETE RECORD RECNO()
        PACK
        SUM POS TO mS_POS
        SUM K_AJ TO mS_K_AJ
        @ 20,46 SAY mS_POS PICTURE "9999"
        @ 20,68 SAY mS_K_AJ PICTURE "9999999999"
        GO mMAR
    ENDIF

```

```

*-----*
DO MHNHYMA WITH Q16
RETURN 2
CASE LASTKEY()=22
APPEND BLANK
KEYBOARD CHR(30)+CHR(29)+CHR(13)
RETURN 2
CASE LASTKEY()=14
INSERT()
KEYBOARD CHR(29)+CHR(13)
RETURN 2
CASE LASTKEY()=-20
SELE 4
mMAR=RECNO()
*-----*
SUM POS TO mS_POS
SUM AJIA TO mS_AJIA
SUM EKP TO mS_EKP
SUM FPA TO mS_FPA
*-----*
GO mMAR
mS_K_AJ=mS_AJIA-mS_EKP
mP_SYN=mS_K_AJ+mS_FPA
*-----*
BOX_ONE(07,20,16,54)
@ 09,23 SAY "Συνολική Αξία : "
@ 09,41 SAY mS_AJIA PICTURE "9999999999"
@ 10,23 SAY "Σύνολο Έκπτωσης:"
@ 10,41 SAY mS_EKP PICTURE "9999999999"
@ 11,23 SAY "καθαρή Αξία : "
@ 11,41 SAY mS_K_AJ PICTURE "9999999999"
@ 12,23 SAY "Αξία ΦΠΑ : "
@ 12,41 SAY mS_FPA PICTURE "9999999999"
@ 14,23 SAY "Πληρωτέο Ποσό : "
SET COLOR TO 0/7
@ 14,40 SAY mP_SYN PICTURE "9999999999"
SET COLOR TO 7
SAVE SCREEN TO mO
*-----*
IF YesNo(" Να γίνει Εκτύπωση ;N/O")=1
IF PRINT_TEST()=0
DO MHNHYMA WITH Q16
RETURN 2
ENDIF
SET DEVICE TO PRINT
mEKT10=mPER_PAR+" "+mSEI_PAR+" "+STR(mARI_PAR)+"
"+DTC(mHME_PAR)
@ 05,02 SAY mEKT10
IF .NOT. EMPTY(mPER)
@ 06,02 SAY mPER
ENDIF

```

```

@ 08,02 SAY mEPO_PP
@ 09,02 SAY mAFM_PP
GO TOP
X=11
DO WHILE .NOT. EOF()
  mKOD_PRD=KOD_PRD
  mPER_PRD=ALLTRIM(PER_PRD)
  mPOS=POS
  mTIM=TIM
  mPEK=PEK
  mK_AJ=K_AJ
  mPFPA=PFPA
  @ X,02 SAY mKOD_PRD+" "+mPER_PRD
  @ X,44 SAY mPOS
  @ X,48 SAY mTIM
  @ X,55 SAY mPEK
  @ X,58 SAY mK_AJ
  @ X,68 SAY mPFPA
  SKIP
  X=X+1
  IF X=21 .AND. .NOT. EOF()
    EJECT
    X=11
  ENDIF
ENDDO
@ 23,59 SAY mS_AJIA
@ 24,59 SAY mS_EKP
@ 25,59 SAY mS_K_AJ
@ 26,59 SAY mS_FPA
@ 27,59 SAY mP_SYN
EJECT
SET DEVICE TO SCREEN
ENDIF
*-----*
RESTORE SCREEN FROM mO
*-----*
IF YesNo("Να γίνει καταχώρηση ; (N/O)")=1
GO TOP
*-----*
DO WHILE .NOT. EOF()
  mKOD_PRD=KOD_PRD
  mPER_PRD=PER_PRD
  mPOS=POS
  mTIM=TIM
  mPEK=PEK
  mK_AJ=K_AJ
  mPFPA=PFPA
  mFPA=FPA

```

```

    SELE 5
    APP_LOC()
    REPLACE KOD_PAR WITH mKOD_PAR
    REPLACE SEI_PAR WITH UPPER(mSEI_PAR)
    REPLACE ARI_PAR WITH mARI_PAR
    REPLACE KOD_PRD WITH mKOD_PRD
    REPLACE PER_PRD WITH mPER_PRD
    REPLACE POS    WITH mPOS
    REPLACE TIM    WITH mTIM
    REPLACE PEK    WITH mPEK
    REPLACE K_AJ   WITH mK_AJ
    UNLOCK
    SELE 4
    SKIP
ENDDO
*-----*
SELE 3
APP_LOC()
REPLACE KOD_PAR WITH mKOD_PAR
REPLACE PER_PAR WITH mPER_PAR
REPLACE SEI_PAR WITH UPPER(mSEI_PAR)
REPLACE ARI_PAR WITH mARI_PAR
REPLACE KOD_PP  WITH mKOD_PP
REPLACE AFM_PP  WITH mAFM_PP
REPLACE EPO_PP  WITH mEPO_PP
REPLACE HME_PAR WITH mHME_PAR
REPLACE S_POS   WITH mS_POS
REPLACE S_AJIA  WITH mS_AJIA
REPLACE S_EKP   WITH mS_EKP
REPLACE S_K_AJ  WITH mS_K_AJ
REPLACE S_FPA   WITH mS_FPA
REPLACE P_SYN   WITH mP_SYN
UNLOCK
*-----*
mKEY=mKOD_PAR+mSEI_PAR
SELE 2
SET ORDER TO 1
GO TOP
SEEK mKEY
IF .NOT. FOUND()
    APP_LOC()
    REPL KOD_PAR WITH mKOD_PAR
    REPL SEI_PAR WITH UPPER(mSEI_PAR)
    UNLOCK
ENDIF
REC_LOC()
REPL ARI_PAR WITH mARI_PAR
UNLOCK
RETURN 0
ENDIF

```

```

        GO mMAR
        *-----*
        DO MHNHYMA WITH Q16
        RETURN 2
    ENDCASE
    RETURN 1
ENDCASE
*-----*
*-----*
*-----*
*-----*
FUNCTION fA
*-----*
DO CASE
    CASE mHELP1=1
        RESTSCREEN(03,11,04,59,mS1)
    CASE mHELP1=2
        RESTSCREEN(04,25,05,74,mS2)
    CASE mHELP1=3
        RESTSCREEN(04,25,05,74,mS3)
    CASE mHELP1=4
        RESTSCREEN(06,10,07,60,mS4)
    CASE mHELP1=5
        RESTSCREEN(06,10,07,60,mS5)
ENDCASE
PARAMETERS mMODE
DO CASE
    CASE mMODE=0
        RETURN 1
    CASE mMODE=1
        RETURN 1
    CASE mMODE=2
        RETURN 1
    CASE mMODE=4
        DO CASE
            CASE LASTKEY()=13
                IF mHELP1=1
                    mKOD_PAR=KOD_PAR
                ELSEIF mHELP1=2
                    mEPO_PP=EPO
                ELSEIF mHELP1=3
                    mKOD_PP=KOD
                ELSEIF mHELP1=4
                    mKOD_PRD=KOD_PRD
                ELSEIF mHELP1=5
                    mPER_PRD=PER_PRD
                ENDIF
            RETURN 0
        
```



```
CASE LASTKEY()=22
  SAVE SCREEN TO mALAG
  IF mHELP1=1
    DO APAR
  ELSEIF mHELP1=2 .OR. mHELP1=3
    IF mPP="30"
      DO APEL
    ELSE
      DO APRO
    ENDIF
  ELSEIF mHELP1=4 .OR. mHELP1=5
    DO APRD
  ENDIF
  RETURN 2
CASE LASTKEY()=27
  RETURN 0
ENDCASE
RETURN 1
ENDCASE
*-----*
```

```

*-----*
*   APAR . PRG   *
*-----*
IF mEP=1
  dSCREEN("S_APAR2")
ELSEIF mEP=3
  dSCREEN("S_APAR1")
ENDIF
STORE SAVESCREEN(05,08,06,70) TO mS6
*-----*
SELE 1
IF .NOT. NET_USE("PAR_IKA"..F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PAR1
*-----*
SET ORDER TO 1
GO TOP
IF EOF()
  APP_LOC()
  UNLOCK
  KEYBOARD CHR(13)
ENDIF
*-----*
DECLARE mPED[5],mTIT[5]
mPED[1]="KOD_PAR"
mPED[2]="PER_PAR"
mPED[3]="KOD"
mPED[4]="POSOTHTA"
mPED[5]="AJIA"
*-----*
mTIT[1]="  "
mTIT[2]="    "
mTIT[3]="  "
mTIT[4]="  "
mTIT[5]="  "
*-----*
DBEDIT(05,09,16,69,mPED,"fPARAST",.T.,mTIT)
IF mEP=1
  RESTORE SCREEN FROM mALAG
ENDIF
*-----*
*-----*
*-----*
*-----*

```

```

FUNCTION fPARAST
RESTSCREEN(05,08,06,70,mS6)
PARAMETERS mMODE,mPED
mROW=ROW()
mCOL=COL()
IF LASTKEY()=22
  KEYBOARD CHR(22)
ENDIF
DO CASE
  CASE mMODE=0
    RETURN 1
  CASE mMODE=1
    @ 01,62 SAY " Αρχή Γραμμών "
    TONE(200.5)
    INKEY(.5)
    SET COLOR TO 0/7
    @ 01,62 SAY "          "
    SET COLOR TO 7
    RETURN 1
  CASE mMODE=2
    @ 01,62 SAY " Τέλος Γραμμών "
    TONE(200.5)
    INKEY(.5)
    SET COLOR TO 0/7
    @ 01,62 SAY "          "
    SET COLOR TO 7
    RETURN 1
  CASE mMODE=4
    DO CASE
      CASE LASTKEY()=13
        IF mPED=1
          DO MHNHYMA WITH " Το πεδίο αυτό παίρνει αριθμούς
          από 01 έως 99.          (Esc) Έξοδος "
          *-----*
          mKOD_PAR=KOD_PAR
          SET CURS ON
          @ mROW,mCOL GET mKOD_PAR PICTURE "99"
          READ
          SET CURS OFF
          *-----*
          mKEY=ALLTRIM(mKOD_PAR)
          DO CASE
            CASE EMPTY(mKOD_PAR)
              IF LASTKEY()=27
                DELETE RECORD RECNO()
                PACK
                DO MHNHYMA WITH Q2
                RETURN 2
            OTHERWISE
              RETURN 1
          ENDIF
        ENDIF
      OTHERWISE
        RETURN 1
    END CASE
  END CASE
END FUNCTION

```

```

DO MHNHYMA WITH " Δεν δώσατε κωδικό . Επαναλάβετε "
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
CASE mKOD_PAR="00" .OR. LEN(mKEY)=1
DO MHNHYMA WITH "Δώσατε λανθασμένο κωδικό.Επανάλαβε"
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
CASE LASTKEY()=27
DO MHNHYMA WITH Q2
RETURN 1
ENDCASE
*-----*
mR=RECNO()
SEEK mKOD_PAR
*-----*
IF FOUND() .AND. .NOT. RECNO()=mR
DO MHNHYMA WITH "Ο κωδικός υπάρχει. Επανάλαβε "
TONE(300.7)
INKEY(1)
GO mR
KEYBOARD CHR(13)
RETURN 1
ENDIF
*-----*
GO mR
REC_LOC()
REPL KOD_PAR WITH mKOD_PAR
UNLOCK
KEYBOARD CHR(4)
ENDIF
IF mPED=2
DO MHNHYMA WITH " Περιγραφή Παραστατικού (ESC) Έξοδος
"
*-----*
mPER_PAR=PER_PAR
SET CURS ON
@ mROW,mCOL GET mPER_PAR
READ
SET CURS OFF
*-----*
IF LASTKEY()=27 .AND. .NOT. EMPTY(mPER_PAR)
DO MHNHYMA WITH Q2
RETURN 1
ENDIF

```

```

    IF EMPTY(mPER_PAR)
    DO MHNHYMA WITH "Δεν δώσατε περιγραφή . Επαναλάβετε "
    TONE(300.7)
    INKEY(1)
    KEYBOARD CHR(13)
    RETURN 1
ENDIF
*-----*
REC_LOC()
REPL PER_PAR WITH mPER_PAR
UNLOCK
KEYBOARD CHR(4)+CHR(13)
ENDIF
IF mPED=3
DO MHNHYMA WITH " (1)για Εισαγωγή, (2) για Εξαγωγή .
(ESC) 'Εξοδος "
*-----*
mKOD=KOD
SET CURS ON
@ mROW,mCOL GET mKOD
READ
SET CURS OFF
*-----*
IF LASTKEY()=27 .AND. .NOT. EMPTY(mKOD)
DO MHNHYMA WITH Q2
RETURN 1
ENDIF
IF mKOD="1" .OR. mKOD="2"
REC_LOC()
REPL KOD WITH mKOD
UNLOCK
KEYBOARD CHR(4)+CHR(13)
ELSE
DO MHNHYMA WITH "Δώσατε λανθασμένο κωδικό.Επαναλάβετε"
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
ENDIF
*-----*
ENDIF
IF mPED=4
DO MHNHYMA WITH "Μεταβολές ποσότητας :(+)για Θετικές ,
(-)για αρνητικές. (Esc) 'Εξοδος "
*-----*
mPOSOTHTA=POSOTHTA
SET CURS ON
@ mROW,mCOL GET mPOSOTHTA
READ
SET CURS OFF

```

```

*-----*
IF LASTKEY()=27
  DO MHNHYMA WITH Q2
  RETURN 1
ENDIF
IF mPOSOTHTA="+" .OR. mPOSOTHTA="-" .OR. mPOSOTHTA=""
  REC_LOC()
  REPL POSOTHTA WITH mPOSOTHTA
  UNLOCK
  KEYBOARD CHR(4)+CHR(13)
ELSE
  DO MHNHYMA WITH "Δώσατε λανθασμένη ποσότητα.
  Επανάλαβετε "
  TONE(300.7)
  INKEY(1)
  KEYBOARD CHR(13)
  RETURN 1
ENDIF
*-----*
ENDIF
IF mPED=5
  DO MHNHYMA WITH "Μεταβολές αξίας(+)για Θετικές,
  (-)για Αρνητικές .      (Esc) Έξοδος "
  *-----*
  mAJIA=AJIA
  SET CURS ON
  @ mROW,mCOL GET mAJIA
  READ
  SET CURS OFF
  *-----*
  IF LASTKEY()=27
    DO MHNHYMA WITH Q2
    RETURN 1
  ENDIF
  IF mAJIA="+" .OR. mAJIA="-" .OR. mAJIA=" "
    REC_LOC()
    REPL AJIA WITH mAJIA
    UNLOCK
    KEYBOARD CHR(1)
  ELSE
    DO MHNHYMA WITH "Δώσατε λανθασμένη αξία . Επαναλάβετε . "
    TONE(300.7)
    INKEY(1)
    KEYBOARD CHR(13)
    RETURN 1
  ENDIF
  *-----*
ENDIF
DO MHNHYMA WITH Q2
RETURN 2

```

```
CASE LASTKEY()=7
  mD=" "+KOD_PAR+" | "+PER_PAR+" | "+KOD+" | "+POSOTHTA+" |
  "+AJIA+ " "
  SET COLOR TO 0/7
  @ mROW,09 SAY mD
  SET COLOR TO 7
  *-----*
  IF YesNo("Επιβεβαίωση Διαγραφής(N/O) ")=1
    mKEY=KOD_PAR
    DELETE RECORD RECNO()
    PACK
    SET SOFTSEEK ON
    SEEK mKEY
    SET SOFTSEEK OFF
  ENDIF
  *-----*
  DO MHNHYMA WITH Q2
  RETURN 2
CASE LASTKEY()=22
  APP_LOC()
  UNLOCK
  KEYBOARD CHR(31)+CHR(29)+CHR(13)
  RETURN 1
CASE LASTKEY()=-4
  IF PRINT_TEST()=0
    DO MHNHYMA WITH Q2
    RETURN 1
  ENDIF
  SET DEVICE TO PRINT
  *-----*
  mEKT1="Κωδικός      Περιγραφή                ΚΠΑ "
  @ 01.00 SAY mEKT1
  @ 02.00 SAY REPLICATE("=",LEN(mEKT1))
  *-----*
  mR=RECNO()
  GO TOP
  X=4
  *-----*
  DO WHILE .NOT. EOF()
    mEKT2=KOD_PAR+" "+PER_PAR+" "+KOD+" "+POSOTHTA+"
    "+AJIA
    @ X,00 SAY mEKT2
    IF .NOT. EMPTY(PER)
      X=X+1
      @ X,05 SAY PER
    ENDIF
    @ X+1,00 SAY REPLICATE("-",LEN(mEKT2))
    SKIP
    X=X+2
```

```

    IF X=54 .AND. .NOT. EOF()
      EJECT
      @ 01,00 SAY mEKT1
      @ 02,00 SAY REPLICATE("=",LEN(mEKT1))
      X=4
    ENDIF
  ENDDO
  *-----*
  EJECT
  SET DEVICE TO SCREEN
  GO mR
  RETURN 1
CASE LASTKEY()=-9
  SAVE SCREEN TO ml
  *-----*
  mD=" "+KOD_PAR+" | "+PER_PAR+" | "+KOD+" | "+POSOTHTA+"|
  "+AJIA+ " "
  SET COLOR TO 0/7
  @ mROW,09 SAY mD
  SET COLOR TO 7
  DO MHNHYMA WITH "Ανάλυση περιγραφής Παραστατικού.
  (Esc) Έξοδος "
  *-----*
  mPER=PER
  SET CURS ON
  @ 20,18 GET mPER
  READ
  SET CURS OFF
  *-----*
  IF LASTKEY()=27
    RESTORE SCREEN FROM ml
    RETURN 1
  ENDIF
  *-----*
  RESTORE SCREEN FROM ml
  REC_LOC()
  REPL PER WITH mPER
  UNLOCK
  RETURN 1
CASE LASTKEY()=27
  RETURN 0
ENDCASE
RETURN 1
ENDCASE
*-----*
```



```

*-----*
*  APEL . PRG  *
*-----*
dSCREEN("S_APEL")
STORE SAVESCREEN(05,08,06,70) TO mS7
*-----*
SELE 7
GO TOP
IF EOF()
  APP_LOC()
  UNLOCK
  KEYBOARD CHR(13)
ENDIF
*-----*
DECLARE mPED[3],mTIT[3]
mPED[1]="KOD"
mPED[2]="EPO"
mPED[3]="AFM"
*-----*
mTIT[1]="  "
mTIT[2]="  "
mTIT[3]="  "
*-----*
DBEDIT(05,09,16,69,mPED,"fPEL"..T.,mTIT)
RESTORE SCREEN FROM mALAG
*-----*
*-----*
*-----*
*-----*
FUNCTION fPEL
RESTSCREEN(05,08,06,70,mS7)
PARAMETERS mMODE,mPED
mROW=ROW()
mCOL=COL()
IF LASTKEY()=22
  KEYBOARD CHR(22)
ENDIF
DO CASE
  CASE mMODE=0
    RETURN 1
CASE mMODE=1
  @ 01,62 SAY " Αρχή Γραμμών "
  TONE(200.5)
  INKEY(.5)
  SET COLOR TO 0/7
  @ 01,62 SAY "          "
  SET COLOR TO 7
  RETURN 1

```

```

GO mR
REC_LOC()
REPL KOD WITH mKOD
UNLOCK
KEYBOARD CHR(4)
ENDIF
IF mPED=2
DO MHNHYMA WITH " Δώστε την επωνυμία του πελάτη
(ESC) Έξοδος "
*-----*
mEPO=EPO
SET CURS ON
@ mROW,mCOL GET mEPO
READ
SET CURS OFF
*-----*
IF LASTKEY()=27 .AND. .NOT. EMPTY(mEPO)
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
IF EMPTY(mEPO)
DO MHNHYMA WITH " Δεν δώσατε επωνυμία. Επαναλάβετε . "
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
ENDIF
*-----*
REC_LOC()
REPL EPO WITH mEPO
UNLOCK
KEYBOARD CHR(4)+CHR(13)
ENDIF
IF mPED=3
DO MHNHYMA WITH " Δώστε τον ΑΦΜ του πελάτη
(ESC) Έξοδος "
*-----*
mAFM=AFM
SET CURS ON
@ mROW,mCOL GET mAFM
READ
SET CURS OFF
*-----*
IF LASTKEY()=27
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
*-----*

```

```

CASE mMODE=2
  @ 01,62 SAY "Τέλος Γραμμών "
  TONE(200.5)
  INKEY(.5)
  SET COLOR TO 0/7
  @ 01,62 SAY "          "
  SET COLOR TO 7
  RETURN 1
CASE mMODE=4
  DO CASE
    CASE LASTKEY()=13
      IF mPED=1
        DO MΗΝΥΜΑ WITH " Δώστε τον Κωδικό του Πελάτη
          (ESC) Έξοδος "
          *-----*
          mKOD=KOD
          SET CURS ON
          @ mROW,mCOL GET mKOD
          READ
          SET CURS OFF
          *-----*
        DO CASE
          CASE EMPTY(mKOD)
            IF LASTKEY()=27
              DELETE RECORD RECNO()
              PACK
              DO MΗΝΥΜΑ WITH Q13
              RETURN 2
            ENDIF
            DO MΗΝΥΜΑ WITH " Δεν Δώσατε κωδικό Επαναλάβετε "
            TONE(300.7)
            INKEY(1)
            KEYBOARD CHR(13)
            RETURN 1
          CASE LASTKEY()=27
            DO MΗΝΥΜΑ WITH Q13
            RETURN 1
        ENDCASE
        *-----*
        mR=RECNO()
        SEEK mKOD
        *-----*
        IF FOUND() .AND. .NOT. RECNO()=mR
          DO MΗΝΥΜΑ WITH " Ο Κωδικός υπάρχει .Επαναλάβετε "
          TONE(300.7)
          INKEY(1)
          GO mR
          KEYBOARD CHR(13)
          RETURN 1
        ENDIF
        *-----*

```

```

mR=RECNO()
GO TOP
DO WHILE .NOT. EOF()
  IF AFM=mAFM .AND. .NOT. RECNO()=mR
    DO MHNHYMA WITH " Ο Συγκεκριμένος ΑΦΜ υπάρχει
      Επαναλάβετε "
    TONE(300.7)
    INKEY(1.5)
    GO mR
    KEYBOARD CHR(13)
    RETURN 1
  ENDIF
  SKIP
ENDDO
*-----*
GO mR
REC_LOC()
REPL AFM WITH mAFM
UNLOCK
KEYBOARD CHR(1)
ENDIF
DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=7
mD=" | "+KOD+" | "+EPO+" | "+AFM+" | "
SET COLOR TO 0/7
@mROW,09 SAY mD
SET COLOR TO 7
*-----*
IF YesNo(" Επιβεβαίωση Διαγραφής (N/O)")=1
  mKEY=KOD
  DELETE RECORD RECNO()
  PACK
  SET SOFTSEEK ON
  SEEK mKEY
  SET SOFTSEEK OFF
ENDIF
*-----*

DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=22
APP_LOC()
UNLOCK
KEYBOARD CHR(31)+CHR(29)+CHR(13)
RETURN 1
CASE LASTKEY()=-4
IF PRINT_TEST()=0
DO MHNHYMA WITH Q13
RETURN 1
ENDIF

```



```

*-----*
*   APRO . PRG   *
*-----*
dSCREEN ("S_APRO")
STORE SAVESCREEN(05,08,06,70) TO mS8
*-----*

SELE 8
GO TOP
IF EOF()
  APP_LOC()
  UNLOCK
  KEYBOARD CHR(13)
ENDIF
*-----*
DECLARE mPED[3],mTIT[3]
mPED[1]="KOD"
mPED[2]="EPO"
mPED[3]="AFM"
*-----*
mTIT[1]="  "
mTIT[2]="  "
mTIT[3]="  "
*-----*
DBEDIT(05,09,16,69,mPED,"fPRO",.T.,mTIT)
RESTORE SCREEN FROM mALAG
*-----*
*-----*
*-----*
*-----*

FUNCTION fPRO
RESTSCREEN(05,08,06,70,mS8)
PARAMETERS mMODE,mPED
mROW=ROW()
mCOL=COL()
IF LASTKEY()=22
  KEYBOARD CHR(22)
ENDIF
DO CASE
  CASE mMODE=0
    RETURN 1
  CASE mMODE=1
    @ 01,62 SAY " Αρχή Γραμμών  "
    TONE(200.5)
    INKEY(.5)
    SET COLOR TO 0/7
    @ 01,62 SAY "          "
    SET COLOR TO 7
    RETURN 1

```

```

CASE mMODE=2
  @ 01,62 SAY "Τέλος Γραμμών "
  TONE(200.5)
  INKEY(.5)
  SET COLOR TO 0/7
  @ 01,62 SAY "      "
  SET COLOR TO 7
  RETURN 1
CASE mMODE=4
  DO CASE
    CASE LASTKEY()=13
      IF mPED=1
        DO MHNHYMA WITH " Δώστε τον κωδικό του προμηθευτή
          (ESC) Έξοδος"
          *-----*
          mKOD=KOD
          SET CURS ON
          @ mROW,mCOL GET mKOD
          READ
          SET CURS OFF
          *-----*
        DO CASE
          CASE EMPTY(mKOD)
            IF LASTKEY()=27
              DELETE RECORD RECNO()
              PACK
              DO MHNHYMA WITH Q13
              RETURN 2
            ENDIF
            DO MHNHYMA WITH " Δεν δώσατε κωδικό . Επαναλάβετε "
            TONE(300.7)
            INKEY(1)
            KEYBOARD CHR(13)
            RETURN 1
          CASE LASTKEY()=27
            DO MHNHYMA WITH Q13
            RETURN 1
          ENDCASE
          *-----*
          mR=RECNO()
          SEEK mKOD
          *-----*
          IF FOUND() .AND. .NOT. RECNO()=mR
            DO MHNHYMA WITH "Ο κωδικός υπάρχει. Επανάλαβε  "
            TONE(300.7)
            INKEY(1)
            GO mR
            KEYBOARD CHR(13)
            RETURN 1

```

```

ENDIF
*-----*
GO mR
REC_LOC()
REPL KOD WITH mKOD
UNLOCK
KEYBOARD CHR(4)
ENDIF
IF mPED=2
DO MHNHYMA WITH " Δώστε την επωνυμία του προμηθευτή .
(ESC) Έξοδος "
*-----*
mEPO=EPO
SET CURS ON
@mROW,mCOL GET mEPO
READ
SET CURS OFF
*-----*
IF LASTKEY()=27 .AND. .NOT. EMPTY(mEPO)
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
IF EMPTY(mEPO)
DO MHNHYMA WITH " Δεν δώσατε επωνυμία . Επαναλάβετε "
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
ENDIF
*-----*
REC_LOC()
REPL EPO WITH mEPO
UNLOCK
KEYBOARD CHR(4)+CHR(13)
ENDIF
IF mPED=3
DO MHNHYMA WITH " Δώστε τον ΑΦΜ του προμηθευτή
(ESC) Έξοδος "
*-----*
mAFM=AFM
SET CURS ON
@mROW,mCOL GET mAFM
READ
SET CURS OFF
*-----*
IF LASTKEY()=27
DO MHNHYMA WITH Q13
RETURN 1
ENDIF

```



```

*-----*
mR=RECNO()
GO TOP
DO WHILE .NOT. EOF()
  IF AFM=mAFM .AND. .NOT. RECNO()=mR
    DO MHNHYMA WITH "Ο Συγκεκριμένος ΑΦΜ υπάρχει
      Επανάλαβε "
      TONE(300.7)
      INKEY(1.5)
      GO mR
      KEYBOARD CHR(13)
      RETURN 1
    ENDIF
  SKIP
ENDDO
*-----*

GO mR
REC_LOC()
REPL AFM WITH mAFM
UNLOCK
KEYBOARD CHR(1)
ENDIF
DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=7
  mD=" | "+KOD+" | "+EPO+" | "+AFM+" | "
  SET COLOR TO 0/7
  @ mROW,09 SAY mD
  SET COLOR TO 7
*-----*

IF YesNo(" Επιβεβαίωση Διαγραφής (N/O) ")=1
  mKEY=KOD
  DELETE RECORD RECNO()
  PACK
  SET SOFTSEEK ON
  SEEK mKEY
  SET SOFTSEEK OFF
ENDIF
*-----*

DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=22
  APPEND BLANK
  UNLOCK
  KEYBOARD CHR(31)+CHR(29)+CHR(13)
  RETURN 1
CASE LASTKEY()=-4
  IF PRINT_TEST()=0
    DO MHNHYMA WITH Q13

```

```

RETURN 1
ENDIF
SET DEVICE TO PRINT
*-----*
mEKT5="Κωδικός      Επωνυμία           ΑΦΜ "
@ 01,00 SAY mEKT5
@ 02,00 SAY REPLICATE("=",LEN(mEKT5))
*-----*

mR=RECNO()
GO TOP
X=4
*-----*

DO WHILE .NOT. EOF()
  mEKT6=KOD+" "+EPO+" "+AFM
  @ X,00 SAY mEKT6
  @ X+1,00 SAY REPLICATE("-",LEN(mEKT6))
  SKIP
  X=X+2
  IF X=54 .AND. .NOT. EOF()
    EJECT
    @ 01,00 SAY mEKT5
    @ 02,00 SAY REPLICATE("=",LEN(mEKT5))
    X=4
  ENDIF
ENDDO
*-----*

EJECT
SET DEVICE TO SCREEN
GO mR
RETURN 1
CASE LASTKEY()=27
  RETURN 0
ENDCASE
RETURN 1
ENDCASE
*-----*
```

```

*-----*
*   APRD . PRG   *
*-----*
dSCREEN("S_APRD")
STORE SAVESCREEN(05,08,06,70) TO mS9
*-----*
SELE 9
SET ORDER TO 1
GO TOP
IF EOF()
  APP_LOC()
  UNLOCK
  KEYBOARD CHR(13)
ENDIF
*-----*
DECLARE mPED[4],mTIT[4]
mPED[1]="KOD_PRD"
mPED[2]="PER_PRD"
mPED[3]="TIM"
mPED[4]="PFPA"
*-----*
mTIT[1]="  "
mTIT[2]="  "
mTIT[3]="  "
mTIT[4]="  "
*-----*
DBEDIT(05,09,16,69,mPED,"fPRD"..T.,mTIT)
RESTORE SCREEN FROM mALAG
*-----*
*-----*
*-----*
*-----*
FUNCTION fPRD
RESTSCREEN(05,08,06,70,mS9)
PARAMETERS mMODE,mPED
mROW=ROW()
mCOL=COL()
IF LASTKEY()=22
  KEYBOARD CHR(22)
ENDIF
DO CASE
  CASE mMODE=0
    RETURN 1
  CASE mMODE=1
    @ 01,62 SAY "Αρχή Γραμμών  "
    TONE(200.5)
    INKEY(.5)
    SET COLOR TO 0/7
    @ 01,62 SAY "      "
    SET COLOR TO 7

```

```

RETURN 1
CASE mMODE=2
  @ 01,62 SAY " Τέλος Γραμμών "
  TONE(200.5)
  INKEY(.5)
  SET COLOR TO 0/7
  @ 01,62 SAY " "
  SET COLOR TO 7
  RETURN 1
CASE mMODE=4
  DO CASE
    CASE LASTKEY()=13
      IF mPED=1
        DO MHNHYMA WITH " Δώστε τον κωδικό του προϊόντος .
          (ESC) "Εξοδος "
          *-----*
          mKOD_PRD=KOD_PRD
          SET CURS ON
          @ mROW,mCOL GET mKOD_PRD
          READ
          SET CURS OFF
          *-----*
        DO CASE
          CASE EMPTY(mKOD_PRD)
            IF LASTKEY()=27
              DELETE RECORD RECNO()
              PACK
              DO MHNHYMA WITH Q13
              RETURN 2
            ENDIF
            DO MHNHYMA WITH "Δέν δώσατε κωδικό .Επαναλάβετε "
            TONE(300.7)
            INKEY(1)
            KEYBOARD CHR(13)
            RETURN 1
          CASE LASTKEY()=27
            DO MHNHYMA WITH Q13
            RETURN 1
        ENDCASE

        *-----*
        mR=RECNO()
        SEEK mKOD_PRD
        *-----*
        IF FOUND() .AND. .NOT. RECNO()=mR
          DO MHNHYMA WITH " Ο κωδικός υπάρχει . Επαναλάβε "
          TONE(300.7)
          INKEY(1)
          GO mR
          KEYBOARD CHR(13)
          RETURN 1

```

```

ENDIF
*-----*
GO mR
REC_LOC()
REPL KOD_PRD WITH mKOD_PRD
UNLOCK
KEYBOARD CHR(4)
ENDIF
IF mPED=2
DO MHNHYMA WITH "Δώστε την περιγραφή του προϊόντος
(Esc) Έξοδος "
*-----*
mPER_PRD=PER_PRD
SET CURS ON
@ mROW,mCOL GET mPER_PRD
READ
SET CURS OFF
*-----*
IF LASTKEY()=27 .AND. .NOT. EMPTY(mPER_PRD)
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
IF EMPTY(mPER_PRD)
DO MHNHYMA WITH "Δεν δώσατε περιγραφή . Επαναλάβετε "
TONE(300.7)
INKEY(1)
KEYBOARD CHR(13)
RETURN 1
ENDIF
*-----*
REC_LOC()
REPL PER_PRD WITH mPER_PRD
UNLOCK
KEYBOARD CHR(4)+CHR(13)
ENDIF
IF mPED=3
DO MHNHYMA WITH "Δώστε την τιμή του προϊόντος .
(ESC)"Έξοδος"
*-----*
mTIM=TIM
SET CURS ON
@ mROW,mCOL GET mTIM
READ
SET CURS OFF
*-----*
IF LASTKEY()=27
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
*-----*

```

```

    REC_LOC()
    REPL TIM WITH mTIM
    UNLOCK
    KEYBOARD CHR(4)+CHR(13)
ENDIF
IF mPED=4
    DO MHNHYMA WITH " Δώστε το ΦΠΑ του προϊόντος.
    (ESC) Έξοδος"
    *-----*
    mPFPA=PFPA
    SET CURS ON
    @ mROW,mCOL GET mPFPA
    READ
    SET CURS OFF
    *-----*
    IF LASTKEY()=27 .AND. .NOT. EMPTY(mPFPA)
        DO MHNHYMA WITH Q13
        RETURN 1
    ENDIF
    IF mPFPA=8 .OR. mPFPA=18
        REC_LOC()
        REPL PFPA WITH mPFPA
        UNLOCK
        KEYBOARD CHR(1)
    ELSE
        DO MHNHYMA WITH "Δώσατε λανθασμένο ποσοστό ΦΠΑ.
        Επαναλάβετε "
        TONE(300.7)
        INKEY(1)
        KEYBOARD CHR(13)
        RETURN 1
    ENDIF
    *-----*
ENDIF
DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=7
    mD="|"+KOD_PRD+"|"+PER_PRD+"|"+STR(TIM)+"|"+STR(PFPA)+" "
    SET COLOR TO 0/7
    @ mROW,09 SAY mD
    SET COLOR TO 7
    *-----*
    IF YesNo("Επιβεβαίωση διαγραφής(N/O ")=1
        mKEY=KOD_PRD
        DELETE RECORD RECNO()
        PACK
        SET SOFTSEEK ON
        SEEK mKEY
        SET SOFTSEEK OFF
    ENDIF

```

```

*-----*

DO MHNHYMA WITH Q13
RETURN 2
CASE LASTKEY()=22
APP_LOC()
UNLOCK
KEYBOARD CHR(31)+CHR(29)+CHR(13)
RETURN 1
CASE LASTKEY()=-4
IF PRINT_TEST()=0
DO MHNHYMA WITH Q13
RETURN 1
ENDIF
SET DEVICE TO PRINT
*-----*
mEKT7="Κωδικός Περιγραφή Προϊόντος Τιμή ΦΠΑ "
@ 01,00 SAY mEKT7
@ 02,00 SAY REPLICATE("=",LEN(mEKT7))
*-----*
mR=RECNO()
GO TOP
X=4
*-----*
DO WHILE .NOT. EOF()
mEKT8=KOD_PRD+" "+PER_PRD+" "+STR(TIM)+" "+STR(PFPA)
@ X,00 SAY mEKT8
@ X+1,00 SAY REPLICATE("-",LEN(mEKT7))
SKIP
X=X+2
IF X=54 .AND. .NOT. EOF()
EJECT
@ 01,00 SAY mEKT7
@ 02,00 SAY REPLICATE("=",LEN(mEKT7))
X=4
ENDIF
ENDDO
*-----*
EJECT
SET DEVICE TO SCREEN
GO mR
RETURN 1
CASE LASTKEY()=27
RETURN 0
ENDCASE
RETURN 1
ENDCASE
*-----*

```

* **EM_PAR.PRG** *

```
SELE 3
IF .NOT. NET_USE("TMASTER",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO TIM2
```

```
SELE 4
IF .NOT. NET_USE("B_TIM",.T.)
  CLOSE ALL
  RETURN
ENDIF
```

```
SELE 5
IF .NOT. NET_USE("TIM",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO TIM1
```

```
DO WHILE .T.
  *-----*
  dSCREEN("S_TIM1")
  STORE SAVESCREEN(03,03,05,76) TO mS9
```

```
mKOD_PAR=SPACE(2)
mSEI_PAR=SPACE(2)
SET COLOR TO 0/7
SET CURS ON
@ 22,61 GET mKOD_PAR
@ 22,73 GET mSEI_PAR
READ
SET CURS OFF
SET COLOR TO 7
SAVE SCREEN TO mT
```

```
IF LASTKEY()=27
  RETURN
```

```
ENDIF
```

```
mKEY7=mKOD_PAR+UPPER(mSEI_PAR)
SELE 3
SEEK mKEY7
```

```

IF .NOT. FOUND()
  DO MHNHYMA WITH "Το παραστατικό δεν υπάρχει .Επαναλάβετε ."
  TONE(300.7)
  INKEY(1.5)
  RESTORE SCREEN FROM mT
  LOOP
ENDIF
*-----*
DECLARE mPED[5],mTIT[5]
mPED[1]="KOD_PAR"
mPED[2]="PER_PAR"
mPED[3]="SEI_PAR"
mPED[4]="ARI_PAR"
mPED[5]="HME_PAR"
*-----*
mTIT[1]=" "
mTIT[2]=" "
mTIT[3]=" "
mTIT[4]=" "
mTIT[5]=" "
*-----*
DBEDIT(04,05,18,74,mPED,"fEMPAR",.T.,mTIT)
*-----*
ENDDO
*-----*
*-----*
*-----*
FUNCTION fEMPAR
*-----*
RESTSCREEN(03,03,05,76,mS9)
IF mHELP1=9
  mROW=ROW()
  mD=" "+KOD_PAR+" | "+PER_PAR+" | "+SEI_PAR+" |
"+STR(ARI_PAR)+" | "+DLOC(HME_PAR)
  SET COLOR TO 0/7
  @ mROW,03 SAY mD
  SET COLOR TO 7
  DO MHNHYMA WITH " Επιλεγμένο Παραστατικό "
  INKEY(1)
  KEYBOARD CHR(13)
ENDIF
PARAMETERS mMODE
DO CASE
CASE mMODE=0
  DO MHNHYMA WITH Q17
  RETURN 1
CASE mMODE=1
  RETURN 1
CASE mMODE=2
  RETURN 1

```

```
CASE mMODE=4
DO CASE
CASE LASTKEY()=27
RETURN 0
CASE LASTKEY()=13
mR=RECNO()
mKOD_PAR=KOD_PAR
mPER_PAR=ALLTRIM(PER_PAR)
mSEI_PAR=SEI_PAR
mARI_PAR=ARI_PAR
mHME_PAR=HME_PAR
mKOD_PP=KOD_PP
mEPO_PP=EPO_PP
mAFM_PP=AFM_PP
mS_AJIA=S_AJIA
mS_EKP=S_EKP
mS_POS=S_POS
mS_K_AJ=S_K_AJ
mS_FPA=S_FPA
mP_SYN=P_SYN
*-----*
mKEY8=mKOD_PAR+mSEI_PAR+STR(mARI_PAR,6)
SELE 5
SEEK mKEY8
DO WHILE KOD_PAR+SEI_PAR+STR(ARI_PAR,6)=mKEY8
AND. .NOT. EOF()
mKOD_PRD=KOD_PRD
mPER_PRD=PER_PRD
mPOS=POS
mTIM=TIM
mPEK=PEK
mK_AJ=K_AJ
SELE 4
APPEND BLANK
REPL KOD_PRD WITH mKOD_PRD
REPL PER_PRD WITH mPER_PRD
REPL POS WITH mPOS
REPL TIM WITH mTIM
REPL PEK WITH mPEK
REPL K_AJ WITH mK_AJ
SELE 5
SKIP
ENDDO
*-----*
dSCREEN("S_TIM2")
STORE SAVESCREEN(05,00,07,80) TO mS
*-----*
```

```
@ 01,36 SAY mPER_PAR
@ 03,10 SAY mKOD_PAR
@ 03,25 SAY mHME_PAR
@ 04,10 SAY mSEI_PAR
@ 04,25 SAY mARI_PAR
@ 03,47 SAY mKOD_PP
@ 03,69 SAY mAFM_PP
@ 04,47 SAY mEPO_PP
@ 20,46 SAY mS_POS
@ 20,68 SAY mS_K_AJ PICTURE "9999999999"
```

```
*-----*
```

```
SAVE SCREEN TO mN
SELE 4
GO TOP
```

```
*-----*
```

```
DECLARE mPED[6],mTIT[6]
mPED{1}="KOD_PRD"
mPED{2}="PER_PRD"
mPED{3}="POS"
mPED{4}="TIM"
mPED{5}="PEK"
mPED{6}="K_AJ"
```

```
*-----*
```

```
mTIT{1}="  "
mTIT{2}="  "
mTIT{3}="  "
mTIT{4}="  "
mTIT{5}="  "
mTIT{6}="  "
```

```
*-----*
```

```
DBEDIT(06,01,18,78,mPED,"fPAR1",.T.,mTIT)
SELE 4
ZAP
SELE 3
IF LASTKEY()=27
    GO mR
ENDIF
RESTORE SCREEN FROM mT
RETURN 2
```

```
*-----*
```

```
ENDCASE
RETURN 1
```

```
ENDCASE
```

```
*-----*
```

```
*-----*
```

```
FUNCTION fPAR1
```

```
*-----*
```

```
RESTSCREEN(05,00,07,80,mS)
PARAMETERS mMODE
mROW=ROW()
mCOL=COL()
```

```

DO CASE
CASE mMODE=0
RETURN 1
CASE mMODE=1
RETURN 1
CASE mMODE=2
RETURN 1
CASE mMODE=4
DO CASE
CASE LASTKEY()=27
mHELP1=0
RETURN 0
CASE LASTKEY()=-9
BOX_ONE(07,20,16,54)
@ 09,23 SAY ""Συνολική Αξία :"  

@ 09,41 SAY mS_AJIA PICTURE "9999999999"  

@ 10,23 SAY "Σύνολο Εκπτώσης:"  

@ 10,41 SAY mS_EKP PICTURE "9999999999"  

@ 11,23 SAY "Καθαρή Αξία  :"  

@ 11,41 SAY mS_K_AJ PICTURE "9999999999"  

@ 12,23 SAY "Αξία Φ.Π.Α  :"  

@ 12,41 SAY mS_FPA PICTURE "9999999999"  

@ 14,23 SAY "Πληρωτέο ποσό  :"  

SET COLOR TO 0/7  

@ 14,40 SAY mP_SYN PICTURE "9999999999"  

SET COLOR TO 7  

DO ΜΗΝΥΜΑ WITH " Ανάλυση Παραστατικού      (ESC) Έξοδος"  

INKEY(0)  

RESTORE SCREEN FROM mN  

RETURN 2
CASE LASTKEY()=-7
SELE 3  

SKIP -1  

IF BOF()  

DO ΜΗΝΥΜΑ WITH " Δεν υπάρχει προηγούμενο Παραστατικό."  

TONE(300.7)  

INKEY(1)  

DO ΜΗΝΥΜΑ WITH Q18  

SELE 4  

RETURN 2
ENDIF  

mHELP1=9  

RETURN 0

```

```
CASE LASTKEY()=-8
  SELE 3
  SKIP 1
  IF EOF()
    SKIP -1
    DO ΜΗΝΥΜΑ WITH " Δεν υπάρχει επόμενο παραστατικό "
    TONE(300.7)
    INKEY(1)
    DO ΜΗΝΥΜΑ WITH Q18
    SELE 4
    RETURN 2
  ENDIF
  mHELP1=9
  RETURN 0
ENDCASE
RETURN 1
ENDCASE
*-----*
*-----*
```

```
*-----*
*   BOH . PRG   *
*-----*
DO WHILE .T.
  *-----*
  dSCREEN("S_BOH")
  SET COLOR TO 0/7
  @ 22,69 SAY DATE()
  SET COLOR TO 7
  *-----*
  SET CURS OFF
  mEPI=0
  @ 09,25 PROMPT " 1...Αναδιοργάνωση  "
  @ 11,25 PROMPT " 2...Σελιδοποίηση    "
  @ 13,25 PROMPT " 3...Στοιχεία Επιχείρησης  "
  MENU TO mEPI
  SET CURS ON
  *-----*
  IF LASTKEY()=27
    RETURN
  ENDIF
  *-----*
  DO CASE
    CASE mEPI=1
      DO BOH1
    CASE mEPI=2
      DO BOH2
    CASE mEPI=3
      DO BOH3
  ENDCASE
  *-----*
ENDDO
*-----*
```

```

*-----*
*      BOH1 . PRG      *
*-----*
SET CURS OFF
dSCREEN("S_BOH1")
*-----*
IF .NOT. NET_USE("ARI_PAR",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO ARI1
REINDEX
*-----*
IF .NOT. NET_USE("ARX_PRD",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO ARX1,ARX2
REINDEX
*-----*
IF .NOT. NET_USE("PEL_TES",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PEL1,PEL2
REINDEX
*-----*
IF .NOT. NET_USE("PRO_TES",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PRO1,PRO2
REINDEX
*-----*
IF .NOT. NET_USE("PAR_IKA",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO PAR1
REINDEX
*-----*
IF .NOT. NET_USE("TIM",.F.)
  CLOSE ALL
  RETURN
ENDIF
SET INDEX TO TIM1

```

REINDEX

-----*

IF .NOT. NET_USE("TMASTER"..F.)

 CLOSE ALL

 RETURN

ENDIF

SET INDEX TO TIM2

REINDEX

CLOSE ALL

-----*

RETURN


```

*-----*
*   BOH2 . PRG   *
*-----*
SELE 6
IF .NOT. NET_USE("BOSS",.F.)
  CLOSE ALL
  RETURN
ENDIF
*-----*
DO WHILE .T.
  *-----*
  dSCREEN("S_BOH2")
  mSEL1=0
  mSEL2=0
  *-----*
  SET CURS ON
  @ 09,10 SAY "Από Σελίδα:" GET mSEL1 PICTURE "99999"
  @ 11,10 SAY "Εώς Σελίδα:" GET mSEL2 PICTURE "99999"
  READ
  SET CURS OFF
  *-----*
  DO CASE
    CASE LASTKEY()=27
      RETURN
    CASE mSEL1>mSEL2 .OR. EMPTY(mSEL1)
      DO MHNHYMA WITH " Τα στοιχεία είναι ανεπαρκή . Επαναλάβετε. "
      TONE(300.5)
      INKEY(1)
      LOOP
  ENDCASE
  *-----*
  IF PRINT_TEST()=0
    LOOP
  ENDIF
  SET DEVICE TO PRINT
  X=mSEL1
  Y=mSEL2

```

```
*-----*  
DO WHILE .NOT. X=Y+1  
  @ 01,02 SAY EPO  
  @ 02,02 SAY EPA  
  @ 03,02 SAY DIE  
  @ 04,02 SAY "f...".  
  @ 04,10 SAY DOY  
  @ 05,02 SAY "c...c."  
  @ 05,10 SAY AFM  
  @ 55,39 SAY X  
  EJECT  
  X=X+1  
ENDDO
```

```
*-----*  
  SET DEVICE TO SCREEN  
ENDDO  
*-----*
```

```
*-----*
*  BOH3 . PRG  *
*-----*
dSCREEN("S_BOH3")
*-----*
SELE 6
IF .NOT. NET_USE("BOSS"..F.)
  CLOSE ALL
  RETURN
ENDIF
IF EOF()
  APP_LOC()
  UNLOCK
ENDIF
*-----*
mEPO=EPO
mEPA=EPA
mDIE=DIE
mDOY=DOY
mAFM=AFM
*-----*
SET CURS ON
@ 08,07 SAY "Επωνυμία: GET mEPO
@ 10,07 SAY "Επάγγελμα:" GET mEPA
@ 12,07 SAY "Διεύθυνση:" GET mDIE
@ 14,07 SAY "Δ.Ο.Υ  :" GET mDOY
@ 16,07 SAY "Α.Φ.Μ  :" GET mAFM
READ
SET CURS OFF
*-----*
REC_LOC()
REPL EPO WITH mEPO
REPL EPA WITH mEPA
REPL DIE WITH mDIE
REPL DOY WITH mDOY
REPL AFM WITH mAFM
UNLOCK
*-----*
IF LASTKEY()=27
  RETURN
ENDIF
*-----*
```

ΚΑΤΣΩΛΑ ΠΑΝΑΓΙΩΤΑ
33564769

001-01-0001	Παντελόνι λινό D & C	3	56000	0	168000	18
002-02-0002	Πουκάμισο βαμβακερό D & C	4	24500	0	98000	18
003-01-0002	Φούστα λινή KENZO	6	47800	20	229440	18
001-01-0002	Παντελόνι λινό KENZO	2	45000	0	90000	18
002-02-0001	Πουκάμισο βαμβακερό KENZO	2	21800	15	37050	18
004-02-0001	Μπλούζα βαμβακερή KENZO	1	22800	0	22800	18
004-01-0001	Μπλούζα λινή KENZO	1	36700	0	36700	18
003-01-0001	Φούστα λινή D & C	2	52000	30	72800	18
001-02-0001	Παντελόνι βαμβακερό D & C	1	52600	0	52600	18

902500
95100
807400
145332
952732

Αλφα-Βητα Ε.Π.Ε.
Εμπορος
Γούναρη 55, Πάτρα
Δ.Ο.Υ. Α Πατοών
Α.Φ.Μ. 98493223

ΚΩΔΙΚΟΣ	ΕΠΩΝΥΜΙΑ ΠΕΛΑΤΗ	Α.Φ.Μ
01-01-1003	ΚΑΤΣΩΛΑ ΠΑΝΑΓΙΩΤΑ	33564769
01-01-5647	ΑΛΕΒΙΖΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ	36587899
01-01-5678	ΧΑΡΤΙΟΣ ΜΑΡΙΝΟΣ	46467989
01-01-6769	ΤΣΟΜΠΑΝΙΔΗΣ ΣΩΤΗΡΗΣ	45476988
01-01-7845	ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ	57658709
01-01-7856	ΚΥΡΙΑΚΙΔΗΣ ΝΕΑΡΧΟΣ	57577980
01-02-0345	ΑΝΤΩΝΙΟΥ ΝΙΚΟΛΑΟΣ	68768090
01-03-4532	ΚΑΛΟΓΕΡΟΠΟΥΛΟΣ ΑΝΤΩΝΗΣ	46538708
01-03-5767	ΧΡΥΣΑΦΙΔΗΣ ΠΕΤΡΟΣ	58909090
01-03-7800	ΒΑΣΙΛΑΚΗΣ ΓΕΩΡΓΙΟΣ	26758778
01-03-7856	ΠΑΡΑΣΧΟΣ ΣΤΕΦΑΝΟΣ	67904566
01-04-1245	ΒΑΛΑΒΑΝΗ ΑΝΑΣΤΑΣΙΑ	65587832
01-04-6878	ΜΑΝΤΑΚΑΣ ΙΩΑΝΝΗΣ	56758799
02-01-1000	ΠΑΠΑΠΕΤΡΟΥ ΑΦΟΙ Ο.Ε.	34576577
02-01-5767	ΚΑΜΑΡΓΙΑΡΗΣ & ΣΙΑ Ο.Ε.	65876537
02-02-5789	ΠΥΡΑΜΙΔΑ Ε.Π.Ε.	96709045
02-04-6787	ΦΑΣΜΑ Α.Ε.	98435765

ΚΩΔΙΚΟΣ	ΠΕΡΙΓΡΑΦΗ ΠΡΟΙΟΝΤΟΣ	ΤΙΜΗ	ΦΠΑ
001-01-0001	Παντελόνι λινό D & C	56000	18
001-01-0002	Παντελόνι λινό KENZO	45000	18
001-02-0001	Παντελόνι βαμβακερό D & C	52600	18
001-03-0001	Παντελόνι μάλλινο D & C	63200	18
002-02-0001	Πουκάμισο βαμβακερό KENZO	21800	18
002-02-0002	Πουκάμισο βαμβακερό D & C	24500	18
002-03-0001	Πουκάμισο μάλλινο D & C	32000	18
003-01-0001	Φούστα λινή D & C	52000	18
003-01-0002	Φούστα λινή KENZO	47800	18
004-01-0001	Μπλούζα λινή KENZO	36700	18
004-02-0001	Μπλούζα βαμβακερή KENZO	22800	18
005-03-0001	Παλτό μακρού μάλλινο D & C	123500	18
005-03-0002	Παλτό κοντό μάλλινο D & C	96300	18
005-03-0003	Παλτό μακρύ μάλλινο KENZO	114900	18

ΚΩΔ. ΠΕΡΙΓΡΑΦΗ

Κ Π Α

ΚΩΔ.	ΠΕΡΙΓΡΑΦΗ	Κ	Π	Α
01	Τιμολόγιο Πώλησης-Δελτίο Αποστολής για την πώληση εμπορευμάτων	2	+	+
02	Τιμολόγιο Πώλησης	2		+
03	Δελτίο Αποστολής	2	+	
04	Πιστωτικό Τιμολόγιο	2		-
05	Πιστωτικό Τιμολόγιο-Δελτίο Αποστολής	2	-	-
06	Τιμολόγιο Πώλησης-Δελτίο Αποστολής για την πώληση προϊόντων	2	+	+
07	Τιμολόγιο Αγοράς-Δελτίο Παραλαβής	1	+	+
08	Τιμολόγιο Αγοράς	1		+
09	Δελτίο Παραλαβής	1	+	

Βιβλιογραφία

Δίκτυα Υπολογιστών IBM-PC και Συμβατούς ---Μ Γκιούρδας

Ανάπτυξη Πληροφοριακών Συστημάτων -- Ανάλυση και Σχεδιασμός Συστημάτων . ---Βασίλης Λαοπόδης

Ο Εξοπλισμός και η λειτουργία του PC -- Επεκτάσεις Βελτιώσεις στους IBM PC και τους συμβατούς . --- Clive Smith

Novell-Netware Εγχειρίδιο Χρήσης--- Timothy K. McDonald.

Εγχειρίδιο Basic --- Ανδρέα Βούλγαρη .

Μηχανογραφικά Αρχεία --Δημιουργία , Οργάνωση , Επεξεργασία Data Base---Αντώνη Βατικιώτη.

Πληροφοριακά Συστήματα Διοίκησης--Ανάλυση και σχεδιασμός
Α. Σταμούλη

Ο Computer με απλά λόγια --Βασικές Αρχές πληροφορικής ---
Εκδόσεις Personal

Ο οδηγός της Microsoft για το MS-Dos Έκδοση 6.2--- Van Wolverton

Πληροφορική --Ηλεκτρονικοί Υπολογιστές ---

Μηχανογραφική Τήρηση του Ημερολογίου Ταμείου (Εισπράξεων Πληρωμών)

Μηχανογραφική Εκκαθάριση Φόρου Προστιθέμενης αξίας (ΦΠΑ)

Μηχανογραφική Παρακολούθηση Αναλυτικού κ' Γενικού Καθολικού

Μηχανογράφηση Ισολογισμού και Αποτελεσμάτων Χρήσεως.

Βάση 95'

Οι ανάγκες των Μικρομεσαίων Επιχειρήσεων σε στελέχη
Μηχανογράφησης.

