

# A Program for the Statistical Control of the Listserv Operation

## Ένα Πρόγραμμα για τον Στατιστικό Έλεγχο της Λειτουργίας του Listserv

A.S. DRIGAS  
Applied Technologies Department  
NCSR "DEMOKRITOS"  
Ag. Paraskevi  
GREECE  
[dr@imm.demokritos.gr](mailto:dr@imm.demokritos.gr)  
<http://imm.demokritos.gr>

### 1. Γενικά

Συνήθως για να αξιολογήσουμε τη λειτουργία μίας υπηρεσίας θα πρέπει να είμαστε σε θέση να προσδιορίσουμε τις τιμές εκείνων των στατιστικών της μεγεθών που καθορίζουν την παραγωγή των προϊόντων της καθώς και τη λειτουργικότητά της και χρησιμότητά της. Έτσι θα μπορέσουμε να εξάγουμε συμπεράσματα για την κατάστασή της και να εργαστούμε για τη βελτίωση και ανάπτυξή της. Η διαδικασία αυτή ονομάζεται στατιστική ανάλυση και εξετάζει τα φυσικά μεγέθη μιας "παραγωγικής" διαδικασίας σε συνάρτηση με το χρόνο. Στην συνέχεια και εφ'όσον ήδη έχει προσδιοριστεί η χρονική μεταβολή ενός μεγέθους για να προσδώσουμε μία περιεκτική και συνοπτική εικόνα της μεταβολής του σε συνάρτηση με το χρόνο σχεδιάζουμε μία κατάλληλη γραφική απεικόνιση. Η γραφική απεικόνιση ενός μεγέθους σε συνάρτηση με ένα άλλο εκφράζει με τον πιο παραστατικό τρόπο τη σχέση των δύο μεγεθών. Κατά τη διαδικασία της γραφικής απεικόνισης διαφαίνεται ένα σημαντικό μέγεθος της σχέσης των δύο μεγεθών που είναι ο ρυθμός μεταβολής του πρώτου μεγέθους σε σχέση με το δεύτερο. Πολλές φορές ο ρυθμός μεταβολής είναι πιο σημαντικός και μας ενδιαφέρει ίσως περισσότερο για την εξαγωγή ποικίλων συμπερασμάτων από την ίδια την στατική τιμή του μεγέθους, κάποια συγκεκριμένη χρονική στιγμή.

Στο φυλλάδιο αυτό περιγράφεται η διαδικασία λήψης-συλλογής στατιστικών στοιχείων από την υπηρεσία Listserv [35] του δικτύου Αριάδνη και η παρουσίαση τους σε μορφή γραφικής απεικόνισης.

### 2. Εισαγωγή

Στο δίκτυο Αριάδνη είναι εγκατεστημένη μία υπηρεσία με την ονομασία Listserv η οποία εκτός άλλων είναι και υπηρεσία διακίνησης μηνυμάτων με τη μέθοδο της λειτουργίας "Λιστών Ηλεκτρονικού Ταχυδρομείου". Οι λίστες αυτές είναι οι γνωστές mailing lists και προσφέρουν υπηρεσίες mail conference στους χρήστες δικτύων που

γίνονται μέλη σε αυτές. Εφόσον κάποιος χρήστης έχει γίνει συνδρομητής (μέλος) σε μία ή περισσότερες mailing lists έχει τη δυνατότητα να στέλνει mail στην λίστα (λίστες). Στη συνέχεια, το σύστημα Listserv αναλαμβάνει να διανείμει αυτόματα τα μηνύματα αυτά (mails) που έχουν σταλεί στη λίστα, στους υπόλοιπους συνδρομητές αυτής.

Η υπηρεσία Listserv στο δίκτυο Αριάδνη υποστηρίζει περισσότερες από 30 λίστες. Γι' αυτό το λόγο είναι δύσκολο να έχουμε κάθε στιγμή μία συνολική εικόνα της κίνησης των λιστών, εξαιτίας της μορφής και της δομής των στατιστικών πληροφοριών που παρέχει το σύστημα.

Το σύστημα Listserv με τη δομή που έχει κατασκευαστεί παρέχει πληροφορίες, ειδικές για κάθε λίστα, και κυρίως υπό τη μορφή αθροιστικών τιμών από την αρχή λειτουργίας του για κάποια μεγέθη ή στιγμιαίων τιμών κατά την παρούσα χρονική στιγμή, για άλλα μεγέθη. Οι πληροφορίες αυτές είναι δομημένες σε αρχεία στην περιοχή που καταλαμβάνει κάθε λίστα στο σύστημα.

Εξαιτίας των προαναφερόμενων λόγων (πολλές λίστες και όχι κατάλληλη μορφή στατιστικής πληροφορίας) δημιουργήθηκε η αναγκαιότητα ύπαρξης προγράμματος στατιστικών που η έξοδος του θα δίνει μία ολοκληρωμένη εικόνα της κίνησης των λιστών στην υπηρεσία Listserv του δικτύου Αριάδνη.

### 3. Περιγραφή του συστήματος Listserv

Το σύστημα Listserv γενικότερα βασίζεται σε mail aliases, το καθένα από τα οποία αναφέρεται σε μία λίστα (Mailing List) ή στο Listserv.

Όλες οι λίστες (Λίστες Ηλεκτρονικού Ταχυδρομείου) βρίσκονται κάτω από την περιοχή **/.../lists/LISTNAME** όπου **LISTNAME** είναι κάθε φορά η λίστα Ηλεκτρονικού Ταχυδρομείου.

Το σύστημα Listserv όσον αφορά τις λίστες ηλεκτρονικού ταχυδρομείου αναλαμβάνει τα εξής:

**α.** Να πραγματοποιήσει μία συνδρομή, εφ' όσον έχει λάβει συγκεκριμένη εντολή - αίτησης από κάποιο χρήστη, για κάποιες λίστες. Μετά την εκτέλεση της εντολής το σύστημα ενημερώνει τα κατάλληλα αρχεία της συγκεκριμένης λίστας (ή λιστών) όπως το αρχείο **/.../lists/LISTNAME/.subscribers** καθώς και αρχεία που αφορούν τον Server. Επίσης αναλαμβάνει να στείλει mail επιβεβαίωσης εγγραφής στον αποστολέα της εντολής - αίτησης και ενημερώνει σε περίπτωση λάθους και τον αποστολέα και τον owner της λίστας. Για όλα αυτά το σύστημα Listserv κρατάει reports στα κατάλληλα αρχεία.

**β.** Να διανείμει τα μηνύματα που διακινούνται σε κάποια λίστα από κάποιο χρήστη (συνδρομητή) στους υπόλοιπους συνδρομητές της λίστας. Εφόσον το σύστημα έχει πραγματοποιήσει τη διανομή του μηνύματος με επιτυχία ενημερώνει τα κατάλληλα αρχεία της συγκεκριμένης λίστας.

### 3.1. Δομή του συστήματος Listserv

Κάθε λίστα Ηλεκτρονικού Ταχυδρομείου, όπως αναφέραμε προηγουμένως έχει τα δικά της αρχεία τα οποία βρίσκονται κάτω από την περιοχή: `/.../lists/LISTNAME/`. Τα σημαντικότερα από αυτά είναι:

**.headers** : Αρχείο μοναδικό για κάθε λίστα, το οποίο ενημερώνεται κάθε φορά που κάποιος χρήστης - συνδρομητής στέλνει mail σε αυτή. Συγκεκριμένα το αρχείο `.headers` ενημερώνεται με τις e-Mail addresses των συνδρομητών που έχουν στείλει μηνύματα στη λίστα. Δηλαδή αν κάποιος συνδρομητής μιλήσει για παράδειγμα τρεις φορές σε μία συγκεκριμένη λίστα μέσα στο αρχείο `.headers` θα υπάρχουν τρεις εισαγωγές για αυτόν. Από τα προαναφερόμενα γίνεται αντιληπτό ότι από το αρχείο `.headers` μπορούν να εξαχθούν πληροφορίες για τα ενεργά μέλη της λίστας.

**.message.ids** : Database με πρόσφατα message ids από μηνύματα που έχουν ήδη διανεμηθεί για αποφυγή των mail loops. Δηλαδή όταν κάποιο mail διανέμεται σε κάποια λίστα ενημερώνεται το αρχείο `.message.ids` και στην συνέχεια το σύστημα κάθε φορά που μπαίνει σε διαδικασία διανομής συμβουλευεται το αρχείο `.message.ids` για να μην επαναλάβει την ίδια εργασία.

**.msgno** : Το αρχείο `.msgno`, το οποίο ενημερώνεται συνεχώς, είναι αρχείο του οποίου το περιεχόμενο είναι δύο αριθμητικές τιμές τοποθετημένες στην ίδια σειρά (line). Η πρώτη αριθμητική τιμή αναφέρεται στον αριθμό των μηνυμάτων που έχουν διανεμηθεί στη λίστα και η δεύτερη στον αριθμό των μηνυμάτων λάθους που έχουν σταλεί στη λίστα. Τα μηνύματα λάθους μπορεί να είναι error messages από το σύστημα Listserv ή από mailer daemons από το δίκτυο.

**.rep.list.acc** : Αρχείο με όλα τα reports των μηνυμάτων που φτάσανε στη λίστα.

**.subscribers** : Αρχείο που περιέχει τις e-Mail addresses, τα option του mail (ack, noack, postpone, digest) και τα πλήρη ονόματα των συνδρομητών. Κάθε συνδρομητής στο αρχείο `.subscribers` καταλαμβάνει μία σειρά. Στο αρχείο αυτό το σύστημα δεν επιτρέπει δύο συνδρομές της ίδιας μορφής. Δηλαδή κάθε σειρά του αρχείου `.subscribers` είναι μοναδική.

**mbox** : Αρχείο το οποίο κρατάει όλα τα μηνύματα τα οποία έχουν φτάσει μέχρι σήμερα στη λίστα. Αυτά περιλαμβάνουν και error messages από mailer daemon κ.λ.π..

**archive** : Αρχείο με όλα τα μηνύματα τα οποία έχουν γίνει αποδεκτά από τη λίστα και έχουν διανεμηθεί.

### 3.2. Περιγραφή του προγράμματος Στατιστικών

Έχοντας υπ' όψη τη δομή του Listserv και τις πληροφορίες που παρέχει για κάθε λίστα προέκυψε η ανάγκη δημιουργίας Προγράμματος Στατιστικών που να συμβουλευεται τα αρχεία που διαθέτει το σύστημα, για κάθε λίστα, και να δημιουργεί την επιθυμητή έξοδο. Η έξοδος αυτή δίνει συνολικά στατιστικά για όλες τις λίστες ανά τακτά χρονικά διαστήματα (κάθε 15 ημέρες). Το πρόγραμμα αυτό είναι ένα C\_Shell Script. Τα στοιχεία που μας ενδιαφέρουν για τα στατιστικά είναι τα εξής:

**α.** Ο ακριβής αριθμός των χρηστών που είναι συνδρομητές σε κάθε λίστα, την παρούσα χρονική στιγμή.

**β.** Το πλήθος των μηνυμάτων που έχουν διακινηθεί στη λίστα, για μία συγκεκριμένη χρονική περίοδο.

**γ.** Το πλήθος των μηνυμάτων λάθους (error messages), που έχουν περάσει σε κάθε λίστα. Τα μηνύματα αυτά μπορεί να είναι μηνύματα που έχουν σταλεί στη λίστα από χρήστες που δεν είναι συνδρομητές ή μηνύματα που επιστρέφουν στο σύστημα γιατί δε βρέθηκε στο δίκτυο κάποιος συνδρομητής (δηλ. user unknown messages), ή γιατί κάποιο host domain έχει αλλάξει (δηλ. host unknown messages) κ.ά..

**δ.** Το πλήθος των συνδρομητών που έχουν ενεργό συμμετοχή στις λίστες, δηλ. οι συνδρομητές που έχουν στείλει mail στις λίστες, για μία συγκεκριμένη χρονική περίοδο καθώς και πόσα mail έχει στείλει ο κάθε ένας.

Οι τιμές των προαναφερόμενων μεγεθών δίνονται από το σύστημα Listserv στη περιοχή `/.../lists/*` και στα ακόλουθα αρχεία (όπου \* το directory που αντιστοιχεί σε κάθε λίστα).

**.subscribers**

**.msgno**

**.headers**

Το πρόγραμμα που αναπτύχθηκε ενημερώνεται από αυτά τα αρχεία και εκτελεί τις ακόλουθες εργασίες:

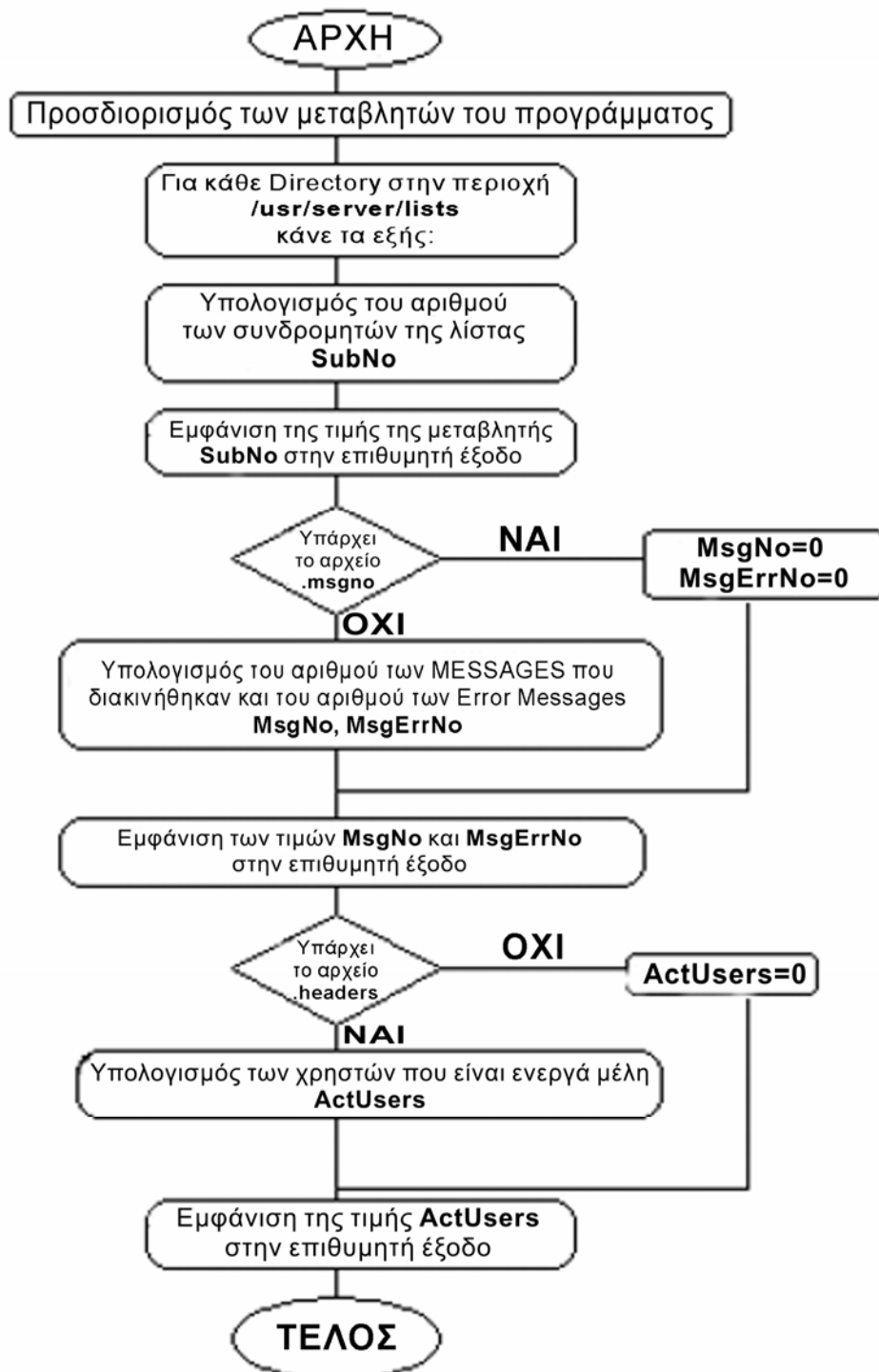
1. Μετράει τις γραμμές του αρχείου `.subscribers` και βάζει το αποτέλεσμα στην μεταβλητή `xxx` η τιμή της οποίας αντιπροσωπεύει το σύνολο των συνδρομητών της λίστας.

2. Από το αρχείο `.msgno` διαβάζει και τοποθετεί σε δύο μεταβλητές `yyy` και `zzz` το σύνολο των μηνυμάτων που έχουν διακινηθεί στη λίστα και το σύνολο των μηνυμάτων λάθους που έχουν σταλεί στη λίστα.

3. Στη συνέχεια ταξινομεί το περιεχόμενο του αρχείου .headers. Η ταξινόμηση γίνεται με κάποιο option το οποίο παράλληλα διώχνει τις κοινές γραμμές. Κατόπιν μετράει τις γραμμές του αρχείου και βάζει σε μία μεταβλητή το αποτέλεσμα. Η τιμή της μεταβλητής αυτής περιέχει τελικά το σύνολο των συνδρομητών που ήταν ενεργό, τη συγκεκριμένη χρονική περίοδο.

Στη συνέχεια παρουσιάζεται το λογικό διάγραμμα του προγράμματος στατιστικών, καθώς και ο κώδικας του προγράμματος.

### **3.2.1. Λογικό διάγραμμα προγράμματος Στατιστικών**



### 3.2.2. Έξοδος του προγράμματος Στατιστικών

Η έξοδος του προγράμματος Στατιστικών και η οποία αποτελεί είσοδο στο πρόγραμμα Γραφικών, έχει την ακόλουθη δομή:

1 JUNE 1993

<b>LIST CATALOG Date</b>	<b>SubNo</b>	<b>MsgNo</b>	<b>MsgErNo</b>	<b>ActUsers</b>	
ARIA_USER	86	17	16	9	01-06-93
AR_BIOLOGY	8	2	4	2	01-06-93
AR_CHEMISTRY	9	3	3	2	01-06-93
AR_COMPUTER.SCIENCE	85	50	30	18	01-06-93
AR_ECONOMY	6	5	4	2	01-06-93
AR_GOSSIP	16	19	14	8	01-06-93
AR_MATERIAL.SCIENCE	6	6	2	3	01-06-93
AR_MEDICAL	10	1	4	1	01-06-93
AR_MESSAGES	49	8	5	3	01-06-93
AR_METEOROLOGY	3	5	1	2	01-06-93
AR_MICROELECTRONICS	27	15	3	6	01-06-93
AR_NUCLEAR.TECHNOLOG Y	2	1	5	1	01-06-93
AR_PHYSICS	18	1	4	1	01-06-93
GR_AGRICULTURE	7	1	5	1	01-06-93
GR_ASTRO	13	22	9	2	01-06-93
GR_CHEMISTRY	25	3	7	2	01-06-93
GR_COMPUTER	166	81	314	38	01-06-93
GR_EARTH.SPACE	24	2	8	2	01-06-93
GR_ECONOMICS	19	3	5	2	01-06-93
GR_EPAFI	6	0	4	0	01-06-93
GR_HISTORY	8	2	5	2	01-06-93
GR_LIFE.SCIENCE	20	7	9	3	01-06-93
GR_LINGUISTICS	11	1	4	1	01-06-93
GR_MATHEMATICS	43	9	5	5	01-06-93
GR_MEDICAL	18	8	344	3	01-06-93
GR_PHILOSOPHY	10	1	4	1	01-06-93
GR_PHYSICS	57	19	77	8	01-06-93

GR_POLITICAL	11	6	1	3	01-06-93
GR_PSYCHOLOGY	7	0	5	0	01-06-93
GR_SCIENT	126	26	33	15	01-06-93
GR_TECHNOLOGY	111	87	159	35	01-06-93
SCIENTISTS	538	28	1317	20	01-06-93
SPEAKERS_CORNER	16	6	3	3	01-06-93
TELEMED	40	30	18	17	01-06-93

#### 4. Το πρόγραμμα Γραφικών

Το πρόγραμμα γραφικών δημιουργήθηκε για να αποδώσει σε μορφή γραφικής απεικόνισης ( γραμμική απεικόνιση όπως επίσης και σε μορφή ιστογραμμάτων ) τα στατιστικά δεδομένα που συλλέγονται κατά τακτά χρονικά διαστήματα από τη λειτουργία των λιστών του Listserv που έχει εγκατασταθεί και λειτουργεί σε μία από τις μηχανές του Δικτύου Αριάδνη δίνοντας τη δυνατότητα στατιστικής ανάλυσης της χρήσης του Listserv. Για τον σκοπό αυτό σχεδιάστηκε και αναπτύχθηκε το πρόγραμμα του οποίου ο τρόπος λειτουργίας και η δομή του περιγράφονται στη συνέχεια αυτού του φυλλαδίου. Το πρόγραμμα σχεδιάστηκε να λειτουργεί στο περιβάλλον γραφικών του Sun workstation με λειτουργικό SunOS 4.1.2 και σε περιβάλλον των sunview. Ο κώδικας του προγράμματος είναι γραμμένος στην C που παρέχεται από την Sun. Έγινε επιλογή για ανάπτυξη του προγράμματος με τη χρήση του μεταγλωττιστή της C γιατί δίνεται η δυνατότητα για το σχεδιασμό σε περιβάλλον γραφικών με τη χρησιμοποίηση πολλών ρουτινών που παρέχονται έτοιμες από την Sun. Επιπλέον η ανάπτυξη του προγράμματος έγινε με σκοπό οι συναρτήσεις που θα αναπτυχθούν να έχουν συμβατότητα με τις μηχανές της Sun και να μπορούν να χρησιμοποιηθούν και σε άλλα παρόμοια προγράμματα που ίσως χρειαστεί να αναπτυχθούν στο μέλλον. Η επιλογή του Sun workstation για την ανάπτυξη του προγράμματος έγινε λόγω της διάδοσης που έχουν αυτά τα workstation στον επιστημονικό κυρίως χώρο και έτσι οι συναρτήσεις που κατασκευάστηκαν αλλά και το πρόγραμμα θα έχουν μεγαλύτερη φορητότητα (portability) και σαν αυτοτελή προγράμματα σε παρόμοιες εφαρμογές αλλά και σαν μέρη άλλων προγραμμάτων.

Το πρόγραμμα αυτό χρησιμοποιεί το παράθυρο γραφικών (canvas) που παρέχουν τα Sunview για να σχεδιάσει τις γραφικές παραστάσεις και τα ιστογράμματα. Επιλέχτηκε να σχεδιάζονται τέσσερις συνολικά παραστάσεις κάθε φορά που γίνεται η εκτέλεση του προγράμματος. Οι παραστάσεις σχεδιάζονται σε δύο άξονες συντεταγμένων οι θέσεις των οποίων πάνω στο παράθυρο γραφικών κάθε φορά είναι συνάρτηση των διαστάσεων του παραθύρου. Σε κάθε άξονα συντεταγμένων εμφανίζονται δύο γραφικές απεικονίσεις. Η κλίμακα των στατιστικών μεγεθών αποδίδεται στους δύο κάθετους άξονες αριστερά και δεξιά του άξονα χρόνων (βλέπε 4.2.7, 4.2.8).



Στο ένα διάγραμμα συντεταγμένων έχουμε απεικόνιση των Msg και Msgser και στο άλλο διάγραμμα συντεταγμένων έχουμε απεικόνιση των Sub και Actuser.

Όπου:

**Msg** : Ο αριθμός των μηνυμάτων που διακινήθηκαν μέσω του **Listserv**.

**Msgser** : Ο αριθμός των λαθών που σημειώθηκαν κατά τη μετάδοση των μηνυμάτων.

**Sub** : Ο αριθμός των καινούργιων αιτήσεων συμμετοχής στις λίστες.

**Actuser** : Ο αριθμός των χρηστών που χρησιμοποίησαν τον listserv για να στείλουν μηνύματα, τη συγκεκριμένη χρονική περίοδο.

#### 4.1. Περιγραφή του προγράμματος Γραφικών

Στην αρχή το πρόγραμμα ζητάει να κάνουμε εισαγωγή στοιχείων όπως:

**α.** Το όνομα του αρχείου δεδομένων που θέλουμε να ανοίξουμε για να διαβάσουμε τα δεδομένα που είναι συγκεντρωμένα σε αυτό.

**β.** Το είδος της απεικόνισης που θέλουμε το πρόγραμμα να δημιουργήσει, ιστόγραμμα ή γραφική παράσταση.

**γ.** Τη χρονική περίοδο χρήσης του listserv που επιλέγουμε για την απεικόνιση.

Μετά την εισαγωγή των στοιχείων γίνεται έλεγχος της ορθότητας τους, αν δηλαδή υπάρχει το αρχείο το οποίο επιθυμούμε να ανοίξουμε για να το διαβάσουμε και αν η χρονική περίοδος είναι έγκυρη, που σημαίνει ότι δεν έχουμε επιλέξει σαν μήνα κάποιον με αριθμό μεγαλύτερο του 12 ή σαν ημέρα κάποιου μήνα μεγαλύτερη από τη τελευταία ημέρα του συγκεκριμένου μήνα.

Η κλίμακα των στατιστικών μεγεθών υπολογίζεται αφού πρώτα ληφθούν υπόψη οι μέγιστες και οι ελάχιστες τιμές (που διαβάστηκαν από τα αντίστοιχα files) του κάθε στατιστικού μεγέθους κατά τη χρονική περίοδο που επιλέχθηκε. Η κλίμακα του χρόνου υπολογίζεται σε σχέση με την αρχική και τελική ημερομηνία που επιλέξαμε.

Οι κλίμακες τοποθετούνται πάνω στο φυσικό μήκος των αξόνων που εξαρτάται από το μέγεθος του παράθυρου. Κάθε φορά που μεταβάλλουμε τις διαστάσεις του παράθυρου γραφικών, τα διαγράμματα σχεδιάζονται από την αρχή λαμβάνοντας υπόψη τις καινούργιες διαστάσεις του παράθυρου.

Μετά τη λήψη των στοιχείων το αρχείο δεδομένων ανοίγεται και προσδιορίζονται τα μέγιστα και ελάχιστα των τεσσάρων στατιστικών μεγεθών κατά τη χρονική περίοδο που έχει προεπιλεγεί. Μετά από αυτό ξεκινά η διαδικασία των μαθηματικών και λογικών πράξεων για να υπολογιστούν οι συντεταγμένες που αντιστοιχούν σε κάθε σημείο και χρησιμοποιώντας συναρτήσεις γραφικών να αποδοθούν αυτά στην οθόνη. Η όλη διαδικασία δομείται σε μία βασική συνάρτηση και σε τρεις ρουτίνες που θα περιγραφούν αναλυτικά στη συνέχεια.

## 4.2. Κορμός του προγράμματος

Ο κορμός του προγράμματος έχει επιφορτιστεί με δύο κυρίως λειτουργίες, την εισαγωγή στοιχείων και το άνοιγμα παραθύρου γραφικών. Μετά από τον έλεγχο των στοιχείων πραγματοποιούνται ορισμένες μαθηματικές πράξεις για τον υπολογισμό της χρονικής περιόδου σε δεκαπενθήμερα, γιατί η μονάδα χρόνου την οποία επιλέξαμε για τη γραφική απεικόνιση είναι ίση με 15 μέρες (την πρώτη και τη δεκάτη πέμπτη μέρα κάθε μήνα). Η μονάδα χρόνου στο μέλλον μπορεί να μεταβληθεί ανάλογα με τον όγκο της κίνησης του **Listserv**.

Στη συνέχεια ανοίγουμε το αρχείο στατιστικών - δεδομένων για να υπολογιστούν τα μέγιστα και τα ελάχιστα των στατιστικών μεγεθών. Αυτά θα χρησιμοποιηθούν για τον υπολογισμό της κλίμακας κάθε στατιστικού μεγέθους.

Επιπλέον υπολογίζεται ο αριθμός των έγκυρων εγγραφών δηλαδή των εγγραφών που ανήκουν μέσα στην χρονική περίοδο που επιλέξαμε. Ο αριθμός αυτός των εγγραφών θα χρησιμοποιηθεί αργότερα κατά το σχεδιασμό των ιστογραμμάτων για την επιλογή του πλάτους που πρέπει να έχουν αυτά ώστε να μην υπερκαλύπτονται μεταξύ τους.

Στο τέλος καλείται η συνάρτηση `canvas` για το σχεδιασμό του παραθύρου γραφικών και μέσω αυτής η συνάρτηση `repaint canvas` που ελέγχει τις διαστάσεις του παραθύρου και καλεί τις συναρτήσεις που σχεδιάζουν τα διαγράμματα. Η εισαγωγή στοιχείων γίνεται από το πληκτρολόγιο με ταυτόχρονη εμφάνιση των στοιχείων στο κοινό παράθυρο του Sun. Κατά την εισαγωγή στοιχείων το πρόγραμμα μας ζητάει τα ακόλουθα:

```
tiger % grafer  
give filename :  
give type of draw (isto or curve) :  
give start time ( dd-mm-yy ) :  
give finish time ( dd-mm-yy ) :
```

Αν κάποιο από τα εισαγόμενα στοιχεία δεν είναι σωστό ζητείται να το επαναεισάγουμε. Το πρόγραμμα τελειώνει όταν κλείσουμε εμείς το παράθυρο γραφικών μέσω της διαδικασίας `quit` των `suntools`.

### 4.2.1. Συνάρτηση Canvas

Η συνάρτηση αυτή αποτελείται από ένα σύνολο από ρουτίνες που παρέχονται έτοιμες από τα `suntools` και δημιουργεί ένα περιβάλλον γραφικών. Η συνάρτηση παρέχεται έτοιμη από τα `suntools` και όταν καλείται μέσα από το πρόγραμμα ανοίγει ένα παράθυρο γραφικών που η Sun ονομάζει `canvas`. Το παράθυρο γραφικών είναι ένα επιπλέον παράθυρο που εμφανίζεται πάνω από το παράθυρο εργασίας `ascii`. Ο `canvas` ελέγχεται μέσω του `mouse` επιλέγοντας τις εντολές `move`, `resize`, `close`, `open`, `quit`.

Στον `canvas` δεν έχουμε τη δυνατότητα να εξάγουμε αριθμούς σε μορφή `integer` ή `float`. Αν θέλουμε να κάνουμε κάτι τέτοιο θα πρέπει πρώτα να μετατρέψουμε τους αριθμούς σε μορφή `char`. Η εισαγωγή αριθμητικών αλλά και των αλφαριθμητικών στοιχείων γίνεται μέσω του παραθύρου εργασίας. Μπορούμε να εισάγουμε δεδομένα σε ένα πρόγραμμα μέσω του

παράθυρου του canvas μόνο αν αυτά είναι τύπου boolean. Αν θέλουμε κάτι τέτοιο πρέπει να ορίσουμε μία περιοχή στο περιβάλλον γραφικών που θα έχει μια αρχική τιμή boolean, όταν αυτή η περιοχή επιλεγεί μέσω του mouse υπάρχει η δυνατότητα η τιμή boolean να αλλάξει την τιμή της. Η συνάρτηση που ανοίγει τον canvas ορίζεται ως εξής.

```

frame window_create( NULL, FRAME, 0 )
canvas window_create( frame, CANVAS,
CANVAS_RETAINED, FALSE,
CANVAS_REPAINT_PROC, repaint_canvas, 0 );
pw canvas_pixwin(canvas)
window_main_loop(frame)

```

#### 4.2.2. Συνάρτηση repaint canvas

Η συνάρτηση αυτή μας επιστρέφει τις διαστάσεις του παράθυρου γραφικών και κάθε φορά που θα εντοπιστεί μία μεταβολή στο μέγεθος του παραθύρου των γραφικών η συνάρτηση αυτή διαβάζει τις τιμές των καινούργιων διαστάσεων. Η ρουτίνα τερματίζει όταν κλείσουμε τον canvas με την εντολή quit. Γνωρίζοντας τις διαστάσεις του canvas υπολογίζουμε τα μήκη των αξόνων των συντεταγμένων των γραφικών παραστάσεων καθώς και τις συντεταγμένες τους. Η διαδικασία αυτή μας δίνει τη δυνατότητα να σχεδιάζουμε κάθε φορά τις γραφικές παραστάσεις προσαρμοσμένες στις διαστάσεις του παραθύρου γραφικών. Τέλος η συνάρτηση υπολογίζει τις συντεταγμένες των αξόνων και καλεί τη συνάρτηση drawaxes για το σχεδιασμό των γραφικών.

#### 4.2.3. Συνάρτηση drawaxes

Η συνάρτηση υπολογίζει τις συντεταγμένες που αντιστοιχούν σε κάθε μέτρηση ή σε κάθε στατιστικό δεδομένο και σχεδιάζει τα γραφήματα. Η συνάρτηση αυτή λαμβάνει ως δεδομένο τις αρχικές και τις τελικές συντεταγμένες του άξονα των χρόνων και τις αρχικές και τις τελικές συντεταγμένες του άξονα των στατιστικών και υπολογίζει τις συντεταγμένες των στατιστικών στοιχείων που αντιστοιχούν σε κάθε μέτρηση. Η διαδικασία υπολογισμού των συντεταγμένων περιγράφεται στη συνέχεια.

Γνωρίζοντας τον αρχικό και τον τελικό χρόνο που ο χρήστης επιλέγει για την απεικόνιση του γραφήματος η συνάρτηση διαβάζει κάθε φορά από το αρχείο στατιστικών - δεδομένων το χρόνο που συλλέχθηκε η συγκεκριμένη εγγραφή. Αν ο χρόνος που συλλέχθηκε η εγγραφή ανήκει στο χρονικό διάστημα μεταξύ του αρχικού και του τελικού χρόνου (τους οποίους ορίσαμε στην αρχή τρέχοντας το πρόγραμμα) τότε η εγγραφή θεωρείται έγκυρη και γίνεται η επιπλέον επεξεργασία της.

Μετά από αυτό υπολογίζονται οι συντεταγμένες που αντιστοιχούν στη συγκεκριμένη εγγραφή και τέλος γίνεται ο σχεδιασμός ενός μέρους της γραφικής παράστασης ή ενός μέρους του ιστογράμματος δηλαδή αυτού του μέρους που αντιστοιχεί στη συγκεκριμένη μέτρηση. Οι συντεταγμένες που αντιστοιχούν στις μέγιστες και ελάχιστες τιμές των στατιστικών μεγεθών καθορίζονται σε σχέση με τις διαστάσεις του παράθυρου. Ο υπολογισμός των συντεταγμένων στις ενδιάμεσες τιμές των

στατιστικών μεγεθών γίνεται σε σχέση με τις συντεταγμένες των μέγιστων και ελάχιστων.

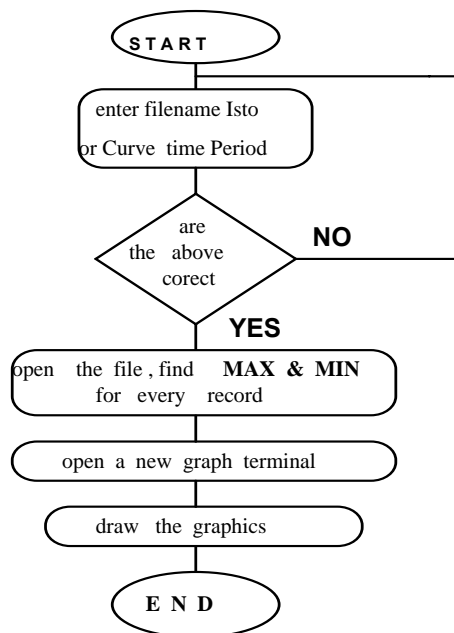
Για το σχεδιασμό της γραφικής παράστασης χρειάζεται να κρατούνται και οι συντεταγμένες για το προηγούμενο στατιστικό δεδομένο ώστε έχοντας δύο σημεία στο επίπεδο να χαράζουμε το μέρος της γραφικής παράστασης που προσδιορίζεται από τα σημεία αυτά. Στην περίπτωση αυτή αποθηκεύονται σε ένα προσωρινό καταχωρητή οι συντεταγμένες κάθε στατιστικού μεγέθους έτσι ώστε να υπάρχουν στην μέτρηση της επόμενης εγγραφής.

#### 4.2.4. Συνάρτηση drawisto

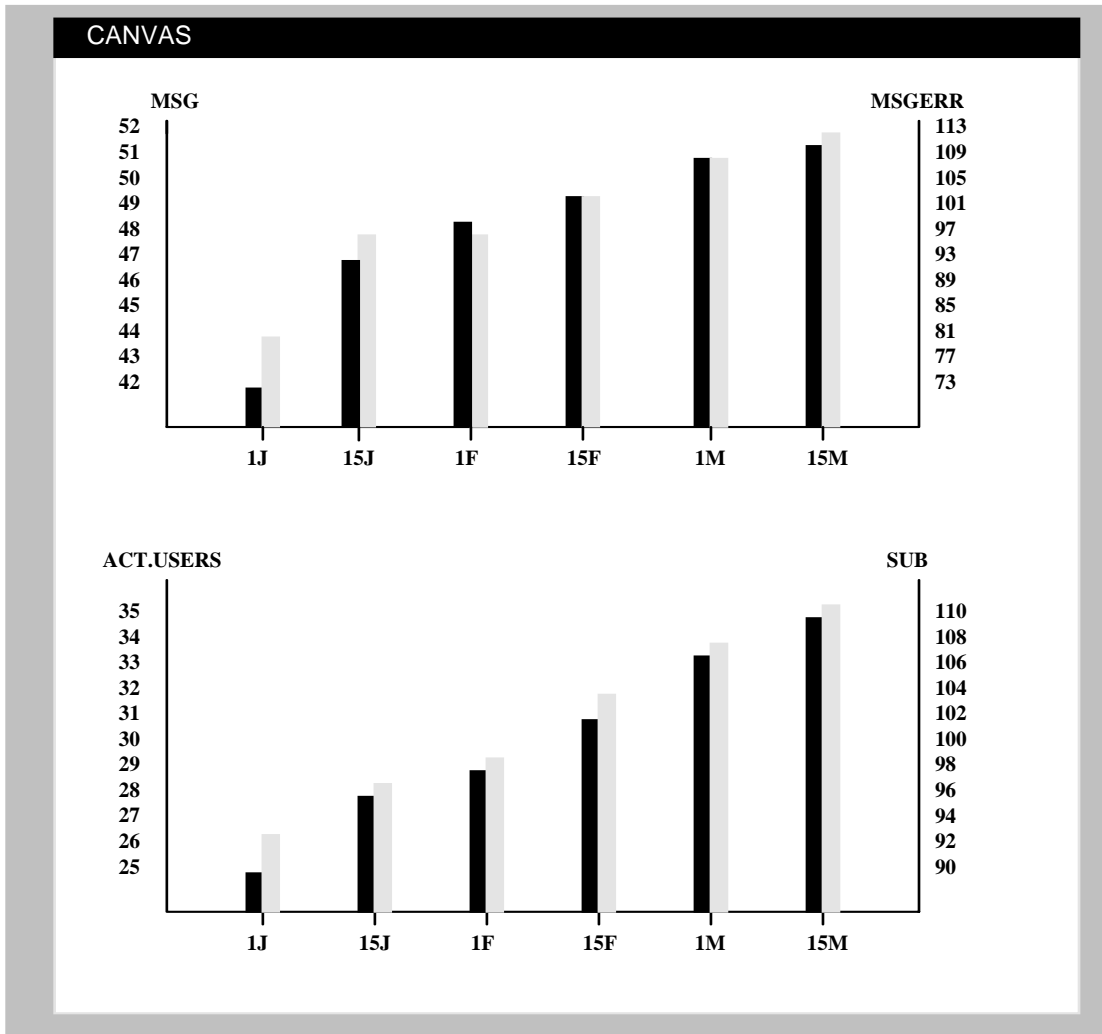
Η συνάρτηση αυτή έχει ως μοναδικό σκοπό να σχεδιάσει τα ιστογράμματα παίρνοντας τις συντεταγμένες από τη συνάρτηση drawaxes. Τα ιστογράμματα εμφανίζονται με διαφορά πάχους των γραμμών σχεδίασης έτσι ώστε να είναι δυνατή και εύκολη η διάκριση μεταξύ τους. Το πλάτος του κάθε ιστογράμματος υπολογίζεται έτσι ώστε να μη συμπίπτουν ποτέ δύο ιστογράμματα.

Για τον καθορισμό του πλάτους που θα έχει το ιστόγραμμα κάθε φορά που σχεδιάζεται, πρέπει να είναι γνωστός ο αριθμός των μετρήσεων που έγιναν κατά το χρονικό διάστημα που εξετάζεται καθώς και η μικρότερη χρονική διάρκεια που εμφανίζεται μεταξύ δύο μετρήσεων.

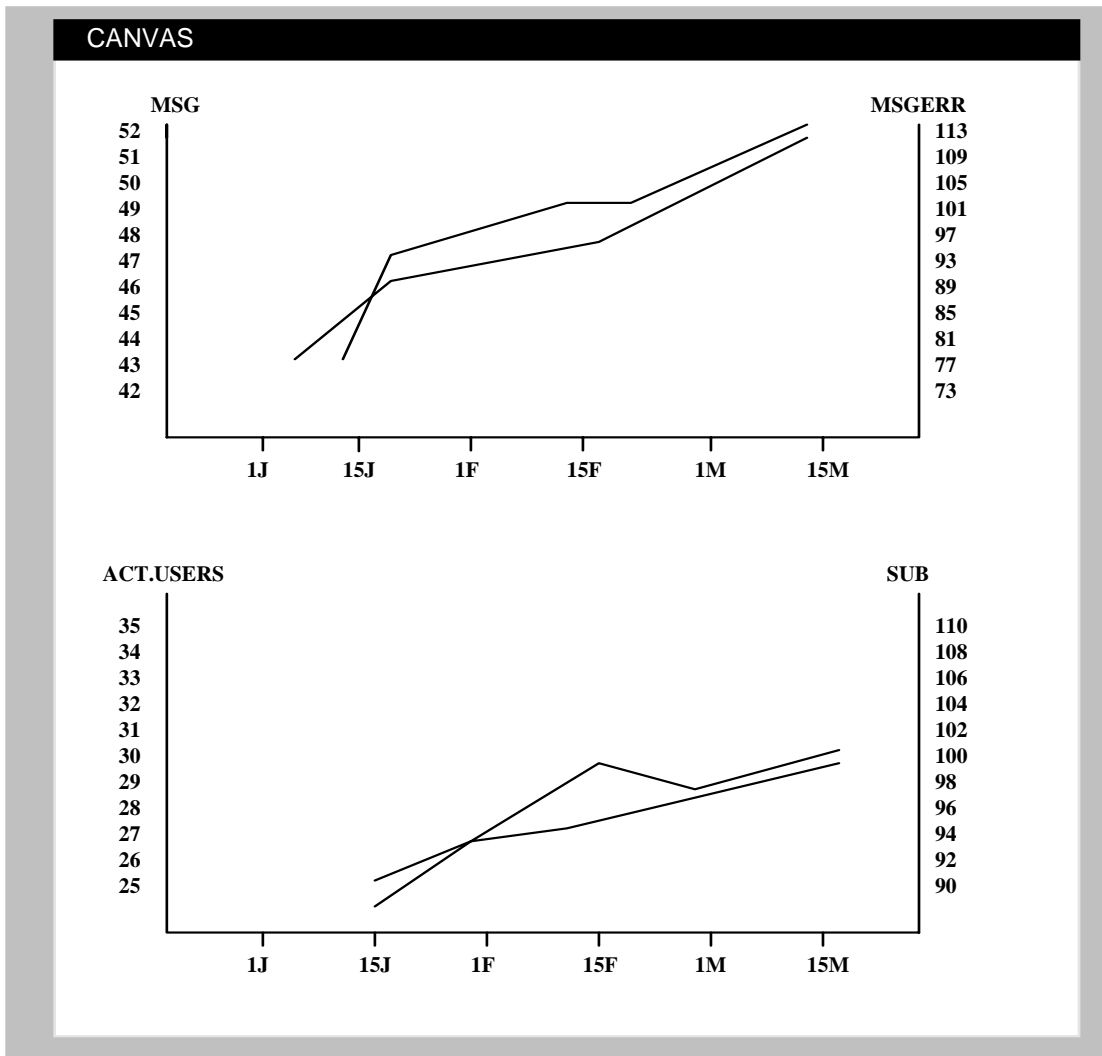
#### 4.2.5. Λογικό Διάγραμμα Προγράμματος Γραφικών



## 4.2.6 Έξοδος του προγράμματος Γραφικών σε μορφή Ιστογραμμάτων



#### 4.2.7 Έξοδος του προγράμματος Γραφικών σε μορφή γραφικής απεικόνισης



## 5. Παράρτημα

### 5.1 Κώδικας του προγράμματος Στατιστικών

```
#!/bin/csh
```

```
#
```

```
# ΠΡΟΓΡΑΜΜΑ ΣΤΑΤΙΣΤΙΚΩΝ ΓΙΑ ΤΟ ΣΥΣΤΗΜΑ  
LISTSERV.
```

```
#
```

```
# ΟΡΙΣΜΟΣ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
#
```

```
set env = /usr/server/Statistics/stat  
set env1 = /usr/server/Statistics/statistics  
set Date = `date +%d- %m- %y`  
set date = `date`  
set date3 = $date[3]  
set date2 = $date[2]  
set date7 = $date[7]  
unalias cd  
cd /usr/server/Statistics  
/bin/rm stat  
touch stat  
foreach dir (~/lists/*)  
#
```

```
# ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΠΛΗΘΟΥΣ ΤΩΝ ΣΥΝΔΡΟΜΗΤΩΝ
```

```
#
```

```
echo -n $dir:t >> $env  
cd $dir  
set w cres = `wc -l .subscribers`  
set SubNo = $w cres[1]  
echo -n " $SubNo " >> $env
```

```
# ΥΠΟΛΟΓΙΣΜΟΣ ΤΩΝ ΜΗΝΥΜΑΤΩΝ ΠΟΥ ΔΙΑΚΙΝΗΘΗΚΑΝ  
ΣΤΗ ΛΙΣΤΑ ΚΑΙ ΤΩΝ ΜΗΝΥΜΑΤΩΝ ΛΑΘΟΥΣ
```

```
#
```

```
if (-e .msgno) then  
set a = `cat .msgno`  
set MsgNo = $a[1]  
set MsgErNo = $a[2]  
echo -n " $MsgNo $MsgErNo ">> $env  
else  
set MsgNo = 0  
set MsgErNo = 0  
echo -n " $MsgNo $MsgErNo " >> $env  
endif
```

```
#
```

```
# ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΠΛΗΘΟΥΣ ΤΩΝ  
ΣΥΝΔΡΟΜΗΤΩΝ ΜΕ ΕΝΕΡΓΟ ΣΥΜΜΕΤΟΧΗ
```

```
#
```

```
if (-e .headers) then  
/bin/rm sort
```

```
sort -u .headers > sort
set wres1 = `wc -l sort`
set ActUsers = $wres1[1]
echo -n " $ActUsers " >> $env
echo " $Date " >> $env
else
set ActUsers = 0
echo -n " $ActUsers " >> $env
echo " $Date " >> $env
endif
end
#
# ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΤΕΛΙΚΟΥ ΑΡΧΕΙΟΥ ΣΤΑΤΙΣΤΙΚΩΝ
#
echo " " >> $env1
echo " " >> $env1
echo -n " $date3 " >> $env1
echo -n " $date2 " >> $env1
echo " $date7 " >> $env1
echo " " >> $env1
echo -n " LIST CATALOG " >> $env1
echo -n " SubNo MsgNo MsgErNo ActUsers " >> $env1
echo " Date " >> $env1
echo " " >> $env1
awk -f ~/Statistics/form ~/Statistics/stat >> $env1
```



## 5.2 Κώδικας του προγράμματος Γραφικών

```

/*****
/
/**** This program created to make a graph representation *****/
/**** of data for utilization of Listserv. *****/
/*****
/

#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <stdio.h>

static void repaint_canvas();
static void drawaxes();
static void drawisto();

/* extern */ int msgmax, msgmin, msgermax, msgermin, submax, submin;
/* extern */ int actusermax, actusermin;
/* extern */ char filename[20];
/* extern */ char type[10];
/* extern */ int startcounter, finishcounter;
/* extern */ int fpfin, fpstart, Xstart, Xfin;
/* extern */ int mounths, years, howmanyisto;
main( argc, argv)
    int argc;
    char *argv [];
/*****
/* MAIN PROCEDURE */
/*****
{
    int subno, msgno, msgerno, actuserno;
    float i;
    char sub[11], msg[11], msger[11], actuser[11];
    int msgpoint, msgerpoint, subpoint, actuserpoint;
    int correct;
    extern int msgmin, msgmax;
    extern int msgermin, msgermax;
    extern int actusermin, actusermax;
    extern int submax, submin;
    extern int startcounter, finishcounter;
    extern int fpstart, fpfin;
    extern int howmanyisto;
    int boulean, fpend;
    int counter, fpdate, fpdateb;
    char ch[9];
    char starttime[10], finishtime[10];
    extern char filename[20];
    int msgwide, subwide, msgerwide, actuserwide;

```

```

int msgdist, subdist, msggerdist, actuserdist;
char daystart[4], dayfinish[4], mounthstart[4], mounthfinish[4];
char yearstart[4], yearfinish[4];
char dayl[4],mounthl[4], yearl[4];
char date[11];
int day, mounth, year;
int days,dayf,mounthf, yearf;
extern int mounths, years;
extern char type[10];
char str[3];
FILE *fp, *fopen();
int m, k;
Canvas canvas;
Pixwin *pw;
struct pr_brush *brush;
struct pr_texture *tex;
Rectlist *repaint_area;
Frame frame;
msgmax = msggermax = submax = actusermax = 0;
msgmin = msggermin = submin = actusermin = 10000;
printf("give filename   :");
gets(filename);

if (( fp = fopen( filename, "r")) == 0)
{
    printf("I cannot open %s\n",filename);
    printf(" Try again!\n");
    exit(0);
}
k = 0;
while ( k == 0 )
{
    printf("give type of draw ( isto or curve ) :");
    gets(type);
    if (( strcmp( type, "isto")== 0 ) || (strcmp( type, "curve" ) == 0) )
        k=1;
    else
        k = 0 ;
}
k = 1;
while ( k == 1 )
{
    printf("give starting time ( dd-mm-yy ) :");
    gets(starttime);
    sprintf(daystart, "%c%c\n",starttime[0],starttime[1]);
    sprintf(mounthstart, "%c%c\n", starttime[3],starttime[4]);
    sprintf(yearstart, "%c%c%c\n", starttime[6],starttime[7]);
    days = atoi(daystart);
    mounths = atoi(mounthstart);
    years = atoi(yearstart);
}

```

```

if ( days > 31 || mounths > 12 || years < 92 )
{
    k = 1;
    printf(" Retype the date \n");
}
else k =0;
}
k = 1;
while ( k == 1 )
{
    printf("give finishing time ( dd-mm-yy) :");
    gets(finishtime);

    sprintf(dayfinish, "%c%c\n", finishtime[0], finishtime[1]);
    sprintf(mounthfinish, "%c%c\n", finishtime[3] , finishtime[4]);
    sprintf(yearfinish, "%c%c\n", finishtime[6], finishtime[7]);

    dayf = atoi(dayfinish);
    mounthf = atoi(mounthfinish);
    yearf = atoi(yearfinish);

```

```

if ( dayf > 31 || mounthf > 12 || yearf < 92 )
{
    k =1;
    printf( "Retype the date \n");
}
else k = 0;
}
k =0;
fpstart = 30*mounths+365*years + days-1;
if (days > 15) k=1;
startcounter=2*mounths + k;
k =0;
fpfin = 30*mounthf+365*yearf+ dayf+1;
if (dayf > 15) k=1;
if (yearf > years && mounthf < mounths)
    finishcounter = 2*(12+mounthf) + k+ 24*(yearf-1-years);
else finishcounter = 2*mounthf +k + 24*(yearf-years);
counter=0;

```

```

/*****
***** /
/** OPEN THE GIVEN FILE AND FIND THE MAX AND THE ***/
/** MIN VALUES IN THE GIVEN TIME DURATION ***/
/*****
***** /

```

```

howmanyisto = 0;
fp = fopen( filename, "r" );

```

```

while(getc(fp) != EOF )
{
    fgets( sub, 10, fp );
    fgets( msg, 10, fp );
    fgets( msger, 10, fp );
    fgets( actuser, 10, fp );
    fgets( date, 10, fp );
    sprintf( dayl, "%c%c\n", date[0], date[1]);
    sprintf( mounthl, "%c%c\n", date[3], date[4]);
    sprintf( yearl, "%c%c\n", date[6], date[7]);

    /***** converting char to integer *****/
    day = atoi(dayl);
    mounth = atoi(mounthl);
    year = atoi(yearl);

    fpdate = 30* mounth + 365* year + day;
    /**** find the max. and min. numbers in the vilidate ****/
    /**** period of time ****/
    if ( fpdate >= fpstart && fpdate <= fpfin && fpdateb != fpdate )
    {
        subno = atoi(sub);
        msgno = atoi(msg);
        msgerno = atoi(msger);
        actuserno = atoi(actuser);

        if (submin > subno) submin = subno;
        if (submax < subno ) submax = subno;
        if (msgmin > msgno ) msgmin = msgno;
        if (msgmax < msgno ) msgmax = msgno;
        if (msggermin > msgerno ) msggermin = msgerno;
        if (msggermax < msgerno ) msggermax = msgerno;
        if (actusermin > actuserno ) actusermin = actuserno;
        if (actusermax < actuserno ) actusermax = actuserno;
        howmanyisto ++;
    } /* this procedure finds the max and min values of the validate records */

    fpdateb = fpdate;
}
fclose(fp);
msgmax = msgmax + 1;
msgmin = msgmin -1;
msggermax = msggermax +1;
msggermin = msggermin -1;

```

```

submax = submax + 1;
submin = submin - 1;
actusermax = actusermax + 1;
actusermin = actusermin -1;

```

```

/*****/
/**   Open a graphics environment   ***/
/*****/
frame = window_create(NULL, FRAME, 0);
canvas =window_create(frame, CANVAS,
                      CANVAS_RETAINED, FALSE,
                      CANVAS_FIXED_IMAGE, FALSE,
                      CANVAS_REPAINT_PROC, repaint_canvas,
                      0);
pw = canvas_pixwin(canvas);

window_main_loop(frame);
exit(0);
}

static void
repaint_canvas(canvas, pw, repaint_area)
    Canvas canvas;
    Pixwin *pw;
    Rectlist *repaint_area;
{

extern int msgmax, msgmin, msgermax, msgermin, actusermax,
actusermin;
extern int submin, submax;
int boolean;
int width = (int>window_get(canvas, CANVAS_WIDTH );
int height = (int>window_get(canvas, CANVAS_HEIGHT );
int margin = 40;
int xleft = margin;
int xright = width - margin;
int ytop = margin;
int ybottom = (height/2)-(margin/2);
int ybottom2 = height - margin;
int ytop2 = (height /2)+(margin/2);
extern char filename[20];
extern years, mounths;
char month[3], year[3], day[3];
int monthfp, yearfp, dayfp;
boolean = 0;
drawaxes(canvas, pw, repaint_area, xleft, ytop, xright, ybottom, boolean);

```

```

    boolean = 1;
    drawaxes(canvas, pw, repaint_area, xleft, ytop2, xright, ybottom2,
boolean);
}

```

```

/*****
/* THIS PROCEDURE DRAW THE AXES*/
*****/
static void drawaxes(canvas, pw, repaint_area, x1, y1, x2, y2, boolean )
int x1,y1,x2,y2;
int boolean;
Canvas canvas;
Pixwin *pw;
Rectlist *repaint_area;
{

int spaceY, spaceX, distX, distY;
int *pvalue1, value1, value2, value3, value4;
int counter;
FILE *fp, *fopen();
int ch;
int msgno, msgerno, subno, actuserno;
char msg[10], msger[10], sub[10], actuser[10];
int msgnob, msgernob, subnob, actusernob;
extern int msgmax, msgmin, msgermax, msgermin, submax, submin;
extern int actusermax, actusermin;
extern int finishcounter, startcounter;
extern int fpstart, fpfin;
extern int mounths, years;
extern char type[10];
extern Xstart, Xfin;
int counter2;
int fdateb;
int defX, defY;
int counter1;
char value1ch[6],value2ch[6], value3ch[6], value4ch[6];
static char array[][3]={ {' ','1','J'},{'1','5','J'}, {' ','1','F'},{'1','5','F'},{'
','1','M'},{'1','5','M'},{' ','1','A'},{'1','5','A'},{' ','1','M'},{'1','5','M'},{' ','1','J'},{'1','5','J'},{'
','1','J'},{'1','5','J'},{' ','1','A'},{'1','5','A'},{' ','1','S'},{'1','5','S'},{' ','1','O'},{'1','5','O'},{'
','1','N'},{'1','5','N'},{' ','1','D'},{'1','5','D'}, '\0'};
float a;
float b, c, d, e, f, g, h, i, k, w;
char date[9], str[3];
char mounth[4], year[4], day[4];
static char what1[10]='{c','u','r','v','e'};
static char what2[10]='{i','s','t','o'};
int mounthfp, yearfp, dayfp;
int fdate, z;
msgnob= msgernob= subnob= actusernob=1;

```

```

pw = canvas_pixwin(canvas);
pw_vector(pw, x1, y1, x1, y2, PIX_SRC, 1);
pw_vector(pw, x1, y2, x2, y2, PIX_SRC, 1);
pw_vector(pw, x2, y1, x2, y2, PIX_SRC, 1);
spaceY = ( ( y2 - y1 ) / 20)+10;
spaceX = ( x2 - x1 ) / 20;
distX =( x2 - x1 - 2*spaceX ) / (finishcounter - startcounter);
distY = ( y2 - y1 - 2*spaceY ) / 10;
defX = x2 - x1 - 2*spaceX;
defY = y2 - y1 - 2*spaceY;
Xstart = x1 + spaceX;
Xfin = x2 - spaceX;
value1 = (msgmax - msgmin)/10;
value2 = (msggermax - msggermin)/10;
value3 = (submax- submin)/10;
value4 = (actusermax - actusermin)/10;
counter = 0;

if (boulean == 0 )
{
for ( counter = 0; counter < 11; counter++ )
{
pw_vector(pw, x1-5, y2-spaceY - counter*distY, x1+5, y2-spaceY -
counter*distY, PIX_SRC, 1);
printf( value1ch, "%d\n",msgmin + value1*counter);
pw_text(pw, x1 -25, y2-spaceY - counter*distY, PIX_SRC, 0,
value1ch);
pw_vector(pw, x2-5, y2-spaceY-counter*distY, x2+5, y2-spaceY-
counter*distY, PIX_SRC, 1 );
printf( value2ch, "%d\n",msggermin + value2*counter);
pw_text(pw, x2 +5, y2-spaceY - counter*distY, PIX_SRC, 0,
value2ch);
}
if ( strcmp( type, what1 )== 0 )
{
pw_text(pw, x1-12, y1+spaceY-20, PIX_SRC, 0, "MSG (.");
pw_text(pw, x2-22, y1+spaceY-30, PIX_SRC, 0, "MSGERR (*)");
}
if ( strcmp( type, what2 ) == 0)
{
pw_text(pw, x1-12, y1+spaceY-20, PIX_SRC, 0, "MSG");
pw_text(pw, x2-22, y1+spaceY-30, PIX_SRC, 0, "MSGERR");
}
}

/*****/
/**** this routine puts date label on X axe***/

```

```

/*****
counter2 = 0;
counter1 = startcounter-2;
for ( counter = startcounter; counter <= finishcounter; counter++ )
{
    if (counter1 == 24) counter1 = 0;
    pw_vector(pw, x1+spaceX+counter2*distX, y2-5, x1+spaceX +
counter2*distX, y2+5, PIX_SRC, 1);
    sprintf( date,
"%c%c%c",array[counter1][0],array[counter1][1],array[counter1][2]);
    pw_text(pw, x1+spaceX+counter2*distX-10, y2+15, PIX_SRC, 0,
date);

    counter1 ++;
    counter2 ++;

} /* end of routine */

/
if (boulean == 1 )
{
for ( counter = 0; counter < 11; counter++ )
{
    sprintf( value3ch, "%d\n",actusermin+value3*counter);
    pw_text(pw, x1-25, y2-spaceY - counter*distY, PIX_SRC, 0, value3ch);
    pw_vector(pw, x1-5, y2-spaceY-counter*distY, x1+5, y2-spaceY-
counter*distY, PIX_SRC, 1 );
    pw_vector(pw, x2-5, y2-spaceY-counter*distY, x2+5, y2-spaceY-
counter*distY, PIX_SRC, 1 );
    sprintf( value4ch, "%d\n",submin+value4*counter );
    pw_text(pw, x2+5, y2-spaceY - counter*distY, PIX_SRC, 0, value4ch);
}
if ( strcmp( type, what1 )== 0 )
{
    pw_text(pw, x1-12, y1+spaceY-20, PIX_SRC, 0, "SUB (.");
    pw_text(pw, x2-22, y1+spaceY-30, PIX_SRC, 0, "ACTUSER(*)");
}
if ( strcmp( type, what2 ) == 0 )
{
    pw_text(pw, x1-12, y1+spaceY-20, PIX_SRC, 0, "SUB");
    pw_text(pw, x2-22, y1+spaceY-30, PIX_SRC, 0, "ACTUSER");
}
}

counter = 0;
fp = fopen( filename, "r" );
msgnob =0;
fpdateb = fpstart+1;
while(getc(fp) != EOF )

```



```

{
    fgets( sub, 10, fp );
    fgets( msg, 10, fp );
    fgets( msgger, 10, fp );
    fgets( actuser, 10, fp );
    fgets( date, 10, fp );

    sprintf( day,"%c%c\n",date[0],date[1] );
    sprintf( mounth, "%c%c\n",date[3],date[4]);
    sprintf( year, "%c%c\n",date[6],date[7]);

    dayfp = atoi(day);
    mounthfp = atoi(mounth);

    yearfp = atoi(year);

    fpdate = 30*mounthfp + 365*yearfp + dayfp;

    /*****
    /*** Keep and exploit only the validates records ***/
    /*****

    if ( fpdate >= fpstart && fpdate <= fpfin && fpdateb != fpdate )
    {

        subno = atoi(sub);
        msgno = atoi(msg);
        msggerno = atoi(msgger);
        actuserno = atoi(actuser);

        a = ((float)(msgmax - msgmin ))/((float)(msgno-msgmin));
        b = ((float)(msgmax - msgmin ))/((float)(msgnob-msgmin));
        c = ((float)(msggermax -msggermin))/((float)(msggerno - msggermin));
        d = ((float)(msggermax - msggermin ))/((float)( msggernob - msggermin));
        e = ((float)(submax - submin ))/((float)( subno - submin));
        f = ((float)(submax - submin ))/((float)(subnob - submin));
        g = ((float)(actusermax - actusermin ))/((float)(actuserno - actusermin));
        h = ((float)(actusermax -actusermin )) /((float)( actusernob - actusermin));

        i = ((float)(fpfin - fpstart))/((float)(fpdate - fpstart));
        k = ((float)(fpfin - fpstart))/((float)(fpdateb - fpstart));

        counter ++;

    /**** draw the curve *****/
        if (boulean == 0 && strcmp(type, what2))

        {
            pw_text(pw, (int)(x1+spaceX+((x2 -x1-2*spaceX )/i)), y2-spaceY-
            (int)(distY*10/a), PIX_SRC, 0, "");

```

```

    pw_text(pw, (int)(x1+spaceX+((x2 -x1-2*spaceX )/i)), y2-spaceY-
(int)(distY*10/c), PIX_SRC,
0, ".");

    pw_vector(pw, (int)(x1+spaceX+((x2 -x1-2*spaceX )/k)), y2-spaceY-
(int)(distY*10/b)
, (int)(x1+spaceX+((x2-x1-2*spaceX)/i)), y2-spaceY-(int)(distY*10/a),
PIX_SRC, 1);
    pw_vector(pw, (int)(x1+spaceX+((x2 -x1-2*spaceX )/k)), y2-spaceY-
(int)(distY*10/d)
, (int)(x1+spaceX+((x2-x1-2*spaceX)/i)), y2-spaceY-(int)(distY*10/c),
PIX_SRC, 1);
}

/**      draw istogramme          ****/
    if (boulean == 0 && strcmp( type, what1) )

    {
        pw_text(pw, (int)(x1+spaceX+(x2 -x1-2*spaceX )/i), y2-spaceY-
(int)(distY*10/a), PIX_SRC, 0, "");

        pw_text(pw, (int)(x1+spaceX+(x2 -x1-2*spaceX )/i), y2-spaceY-
(int)(distY*10/c), PIX_SRC,
0, ".");

        drawisto(canvas, pw, repaint_area, y2, (int)(x1+spaceX+(x2-x1-
2*spaceX)/i), y2-spaceY-(int)(distY*10/a), y2-spaceY-(int)(distY*10/c));

    }

    msgnob = msgno;
    msgernob = msgerno;

/**      draw curve          ***/
    if (boulean == 1 && strcmp( type, what2) )
    {
        pw_text(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-
(int)(distY*10/e), PIX_SRC,
0, "");
        pw_text(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-
(int)(distY*10/g), PIX_SRC,
0, ".");
    }

```

```

    pw_vector(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/k), y2-spaceY-
(int)(distY*10/f)
, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-(int)(distY*10/e), PIX_SRC,
1);

```

```

    pw_vector(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/k), y2-spaceY-
(int)(distY*10/h)
, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-(int)(distY*10/g), PIX_SRC,
1);

```

```

}

```

```

/**** draw istogramme ****/
if (boulean == 1 && strcmp( type, what1) )

```

```

{
    pw_text(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-
(int)(distY*10/e), PIX_SRC,
0, "");
    pw_text(pw, (int)(x1+spaceX+(x2-x1-2*spaceX)/i), y2-spaceY-
(int)(distY*10/g), PIX_SRC,
0, ".");

```

```

    drawisto(canvas, pw, repaint_area, y2, (int)(x1+spaceX+(x2-x1-
2*spaceX)/i), y2-spaceY-(int)(distY*10/e)
, y2-spaceY-(int)(distY*10/g));

```

```

}
actusernob = actuserno;
subnob = subno;

```

```

}
fpdateb = fpdate;
}
fclose(fp);
return;
}

```

```

/*****
/**** This procedure draws the istogram onto canva ****/
/**** terminal ****/
/*****

```

```

static void drawisto( canvas, pw, repaint_area, y1, x2, yisto1, yisto2 )
int x2, y1, yisto1, yisto2;
Canvas canvas;
Pixwin *pw;
Rectlist *repaint_area;

```

```
{
extern int howmanyisto, startcounter, finishcounter;
extern int Xstart, Xfin;
int dististo;
int maxperiod;
maxperiod = finishcounter - startcounter;
if ( maxperiod > howmanyisto )
dististo = (int)(( Xfin - Xstart )/(4* maxperiod));
else dististo = (int)(( Xfin - Xstart )/(4* howmanyisto ));
pw_vector( pw, x2-dististo, y1, x2-dististo, yisto1, PIX_SRC, 1 );
pw_vector( pw, x2, y1, x2, yisto1, PIX_SRC, 1 );
pw_vector( pw, x2-dististo, yisto1, x2, yisto1, PIX_SRC, 1 );
pw_vector( pw, x2, y1, x2, yisto2, PIX_SRC, 1 );
pw_vector( pw, x2+1, y1, x2+1, yisto2, PIX_SRC, 1 );
pw_vector( pw, x2, yisto2, x2+dististo, yisto2, PIX_SRC, 1 );
pw_vector( pw, x2, yisto2+1, x2+dististo, yisto2+1, PIX_SRC, 1 );
pw_vector( pw, x2+dististo, yisto2, x2+dististo, y1, PIX_SRC, 1 );
pw_vector( pw, x2+dististo+1, yisto2, x2+dististo+1, y1, PIX_SRC, 1 );
}
```