

ΚΕΦΑΛΑΙΑ XIII, XIV

Εκσφαλμάτωση Προγράμματος - Κύκλος Ζωής Λογισμικού





ΚΕΦΑΛΑΙΑ XIII, XIV

Εκσφαλμάτωση προγράμματος - Κύκλος Ζωής Λογισμικού

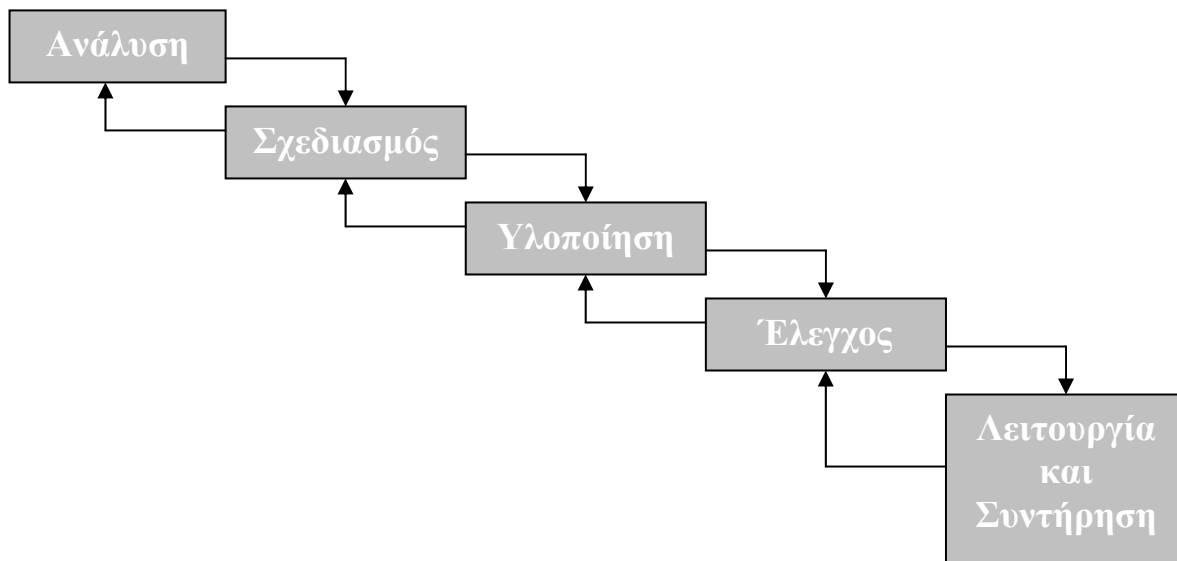
Ένας προγραμματιστής ανεξάρτητα από το πόσο ικανός είναι, όταν δημιουργεί ένα πρόγραμμα, είναι φυσικό να κάνει ορισμένα λάθη. Σε ένα πρόγραμμα είναι δυνατό να παρουσιαστούν διαφορετικής μορφής λάθη τα οποία μπορούν να χωριστούν σε τρεις βασικές κατηγορίες.

- ▶ **Λάθη κατά την υλοποίηση.** Τέτοια λάθη προκαλούνται κυρίως από λανθασμένη σύνταξη εντολών προγράμματος. Η λανθασμένη γραφή μιας δεσμευμένης λέξης ή η χρήση μιας δομής ελέγχου χωρίς την εντολή τερματισμού της αποτελούν τέτοια λάθη (λέγονται και **συντακτικά** λάθη). Η διόρθωση ενός τέτοιου λάθους είναι πολύ εύκολη διαδικασία, αφού οι σύγχρονες γλώσσες συνήθως καταλαβαίνουν που ακριβώς έχει συμβεί το λάθος και μας δίνουν οδηγίες για να το διορθώσουμε. Μερικές φορές αυτό γίνεται ακόμα και κατά την συγγραφή του κώδικα στον editor.
- ▶ **Λάθη κατά την εκτέλεση.** Αυτά τα λάθη προκαλούνται κατά το χρόνο εκτέλεσης του προγράμματος και είναι πιο επώδυνα γιατί συνήθως προκαλούν τον αντικανονικό τερματισμό της εφαρμογής και πολλές φορές το κρέμασμα του συστήματος. Η διαίρεση ενός αριθμού με το μηδέν (0) είναι ένα τέτοιο λάθος που συμβαίνει συχνά. Για τη διόρθωσή αυτών των λαθών, εισάγονται ειδικές εντολές προγράμματος που τα παγιδεύουν και εκτελούν κατάλληλες διαδικασίες χειρισμού τους.
- ▶ **Λογικά λάθη.** Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. Ενώ ο μεταγλωττιστής της γλώσσας προγραμματισμού δεν ανιχνεύει κανένα συντακτικό λάθος και κατά την εκτέλεση του προγράμματος δεν παράγονται τα επιθυμητά αποτελέσματα. Η διόρθωση και η ανίχνευση τέτοιων λαθών είναι πιο δύσκολη.

Εκσφαλμάτωση (debugging) λέγεται η διαδικασία ελέγχου, εντοπισμού και διόρθωσης των σφαλμάτων ενός προγράμματος. Τα σφάλματα που μας ενδιαφέρουν είναι κυρίως των δύο τελευταίων κατηγοριών, αφού τα συντακτικά λάθη ανιχνεύονται από τον ίδιο τον μεταγλωττιστή. Η εργασία αυτή δεν είναι καθόλου εύκολη, απαιτεί βαθιά γνώση της γλώσσας προγραμματισμού και φυσικά αντίστοιχες ικανότητες από τον προγραμματιστή. Τα σύγχρονα προγραμματιστικά περιβάλλοντα προσφέρουν εργαλεία που βοηθούν πολύ τον προγραμματιστή. Πρέπει να σημειώσουμε ότι και η χρήση σχολίων μέσα στις εντολές του προγράμματος υποβοηθά πολύ την εκσφαλμάτωση.

Κύκλος Ζωής Λογισμικού

Κύκλος ζωής ενός λογισμικού (ή ενός μηχανήματος) είναι το η διαδικασία που παρεμβάλλεται από την αρχή της έρευνας μέχρι και τον τερματισμό της χρήσης του προγράμματος που έχει παραχθεί. Το παρακάτω διάγραμμα παρουσιάζει πιο αναλυτικά τον κύκλο ζωής ενός προγράμματος. Τα βέλη δείχνουν την αλληλεπίδραση των διαδοχικών φάσεων. Το τέλος μιας φάσης μπορεί να οδηγήσει στην επόμενη φάση, αλλά μπορεί να οδηγήσει και στην προηγούμενη, αν κάποια στοιχεία πρέπει να επανακαθοριστούν.



Φάση Ανάλυσης και Σχεδίασης

- ▶ Καταγράφονται αναλυτικά τα δεδομένα και τα ζητούμενα του προβλήματος.
- ▶ Ζητούνται οι απαραίτητες διευκρινήσεις από τον πελάτη, σε όσα σημεία οι προδιαγραφές παρουσιάζουν ασάφεια.
- ▶ Καθορίζεται η δομή του προγράμματος.
- ▶ Καθορίζονται οι ενότητες (ρουτίνες, υποπρογράμματα) από τις οποίες θα αποτελείται το πρόγραμμα.
- ▶ Αναζητούνται έτοιμες ενότητες (modules) από παλιότερα προγράμματα που μπορούν να χρησιμοποιηθούν και σε αυτό το πρόγραμμα.
- ▶ Επιλέγονται οι αλγόριθμοι και οι δομές δεδομένων που θα χρησιμοποιηθούν σε κάθε ενότητα.

Φάση Υλοποίησης

- ▶ Επιλέγεται η γλώσσα προγραμματισμού για το συγκεκριμένο πρόγραμμα. Σημειώστε ότι όλες οι γλώσσες δεν είναι κατάλληλες για όλα τα προβλήματα.
- ▶ Εισάγεται το κωδικοποιημένο πρόγραμμα στον υπολογιστή. το πρόγραμμα αυτό είναι το αρχικό πρόγραμμα (source program).
- ▶ Ζητείται η μετάφραση του προγράμματος από ένα μεταγλωττιστή, ώστε αυτό να γίνει κατανοητό από τον υπολογιστή. Το πρόγραμμα αυτό είναι το τελικό πρόγραμμα (object program).
- ▶ Η μετάφραση θα αποκαλύψει λάθη ορθογραφίας και συντακτικού της γλώσσας προγραμματισμού.
- ▶ Διορθώνονται τα λάθη και ακολουθεί ξανά μετάφραση του προγράμματος, εώς την οριστική εξάλειψή τους.

Φάση Ελέγχου

- ▶ το πρόγραμμα ελέγχεται, με τα δεδομένα ελέγχου που είχε ελεγχθεί και ο αλγόριθμος, για να διαπιστωθεί αν παράγει τα επιθυμητά αποτελέσματα.
- ▶ Διαπιστώνονται λάθη που οφείλονται είτε σε λάθη κατά την κωδικοποίηση είτε από λανθασμένη επικοινωνία ενοτήτων.
- ▶ Τα τυχόν λάθη διορθώνονται και οι έλεγχοι συνεχίζονται μέχρι το πρόγραμμα να απαλλαγεί από αυτά.



- ▶ Παρουσιάζονται τα αποτελέσματα στον πελάτη προκειμένου να έχουμε την οριστική επικύρωση εκ μέρους του.
- ▶ Αν παρουσιάζεται απόκλιση από τις προδιαγραφές, θα πρέπει να επιστρέψουμε στην φάση της ανάλυσης και σχεδίασης προκειμένου να γίνουν οι κατάλληλες διορθώσεις.

Στο τέλος της φάσης των ελέγχων έχουμε ένα έτοιμο πρόγραμμα που μπορούμε να δώσουμε στον πελάτη για χρήση. Δεν πρέπει να ξεχνάμε, ότι ο έλεγχος του προγράμματος διαπιστώνει την ύπαρξη ενός λάθους που πρέπει να διορθωθεί, δυστυχώς όμως δεν πιστοποιεί την απουσία λάθους. Έτσι ένα πρόγραμμα ουσιαστικά βρίσκεται κάτω από ένα διαρκή έλεγχο της ορθής λειτουργίας του.

Φάση Συντήρησης

- ▶ Οι προσαρμογές είναι αναπόφευκτες όταν διαφοροποιούνται τα δεδομένα του προβλήματος ή όταν ο χρήστης ζητήσει νέες λειτουργίες.
- ▶ Κάποιες προσαρμογές μπορεί να απαιτήσουν την εκτέλεση της φάσης της ανάλυσης και σχεδίασης και άρα όλων των υπολοίπων φάσεων.
- ▶ Οι βελτιώσεις προκύπτουν από την εμπειρία που αποκτάται με τον καιρό και μας κάνει να βλέπουμε τα ίδια πράγματα με άλλο μάτι.
- ▶ Κάθε προσαρμογή ή βελτίωση θα πρέπει να καταλήγει σε συνολικό έλεγχο του προγράμματος και φυσικά στην καταγραφή των σχετικών σχολίων για την τεκμηρίωση.
- ▶ Το τελευταίο στάδιο αυτής της φάσης έρχεται με την τελειοποίηση του προγράμματος. Τώρα το πρόγραμμά σας δουλεύει υποδειγματικά... μέχρι την επόμενη αλλαγή.



Το παραπάνω γράφημα δείχνει τη κατανομή της προσπάθειας που απαιτείται στις διάφορες φάσεις του κύκλου ζωής. Το μεγαλύτερο ποσοστό έχει η συντήρηση με 60%, η ανάλυση και οι έλεγχοι 16%, ενώ τη μικρότερη προσπάθεια απαιτεί η υλοποίηση με 8%.

