

Το περιβάλλον-X: ένας μικρόκοσμος για μια εισαγωγή στον προγραμματισμό¹

Δαγδιλέλης Βασίλειος, Λέκτορας, Παιδαγωγικής Σχολής Φλώρινας, Αριστοτέλειο Παν. Θεσσαλονίκης, dagdil@uom.gr

Ευαγγελίδης Γεώργιος, Επικ. Καθηγητής, Τμ. Εφαρμ. Πληροφορικής, Πανεπιστήμιο Μακεδονίας, gevan@uom.gr

Σατρατζέμη Μαρία, Επικ. Καθηγητής, Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, maya@uom.gr

Εφόπουλος Βασίλειος, Καθηγητής Πληροφορικής Μέσης Εκπαίδευσης, efor@uom.gr

Κολοβός Φίλιππος, Τμ. Εφαρμ. Πληροφορικής, Πανεπιστήμιο Μακεδονίας, elp99fk@sheffield.ac.uk

Λέξεις Κλειδιά: Εκπαιδευτικό προγραμματιστικό περιβάλλον, σύστημα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων, προγραμματιστικός μικρόκοσμος, διδασκαλία του προγραμματισμού

Περίληψη

Στην εργασία αυτή παρουσιάζουμε μια απλή γλώσσα προγραμματισμού και ένα εκπαιδευτικό προγραμματιστικό περιβάλλον, τον X-Compiler, για μια εισαγωγή στον προγραμματισμό. Από τεχνική άποψη, το προγραμματιστικό περιβάλλον που αναπτύξαμε επιτρέπει την έκδοση, εκσφαλμάτωση και εκτέλεση προγραμμάτων γραμμένων στη γλώσσα X – ένα υποσύνολο της Pascal - δικής μας επινόησης. Ωστόσο, ο X-Compiler ενσωματώνει μια σειρά χαρακτηριστικών που παρουσιάζουν διδακτικό ενδιαφέρον: Ο χρήστης μπορεί να παρακολουθήσει όλα τα ενδιάμεσα στάδια μιας εκτέλεσης προγράμματος: τη μεταγλώττιση, τον assembly κώδικα, τα περιεχόμενα των καταχωρητών, τις ενδιάμεσες τιμές όλων των μεταβλητών. Επιπλέον, ο χρήστης μπορεί να παρέμβει απευθείας στον assembly κώδικα, να τον μεταβάλλει και στη συνέχεια να τον εκτελέσει. Η ύπαρξη πολλών μηνυμάτων, όσο γίνεται ακριβέστερων και επεξηγηματικών, βοηθούν τον αρχάριο προγραμματιστή στην εκσφαλμάτωση, αλλά και την γενικότερη βελτίωση του προγράμματός του.

Abstract

This paper presents a simple programming language and an educational programming environment, called X-Compiler, designed to introduce students to programming. X-Compiler can be used to edit, debug and run programs written in X, a subset of Pascal designed by our team. We believe that X-Compiler is didactically interesting because of the following features:

- The user can watch the intermediate steps of the execution of a program: compilation, assembly code, register contents and intermediate values of the user and temporary system variables. Also, the user can edit the produced assembly code and re-execute it.
- There are many detailed and explanatory messages that can guide the novice user when debugging his programs and, in general, help him write better programs.

¹ Ο X-Compiler αποτελεί ένα τμήμα του λογισμικού ΔΕΛΥΣ, ενός προγράμματος για την υποστήριξη της διδασκαλίας της Πληροφορικής στη Δευτεροβάθμια εκπαίδευση. Το ΔΕΛΥΣ υλοποιήθηκε στα πλαίσια της “Ναυσικάς” που αποτελεί τμήμα του έργου της “Οδύσσειας”.

1. Εισαγωγή

Ακόμη και σήμερα οι αρχές του προγραμματισμού συνεχίζουν, πολύ συχνά, να διδάσκονται με βάση την ίδια μέθοδο [3]. Αυτή η μέθοδος βασίζεται στη χρήση: μιας γλώσσας προγραμματισμού γενικού σκοπού, όπως Pascal, C κτλ, και ενός επαγγελματικού περιβάλλοντος προγραμματισμού. Ωστόσο, η μέθοδος αυτή δε φαίνεται να είναι διδακτικά αποτελεσματική, καθώς οι αρχάριοι πρέπει να αφομοιώσουν μια σειρά από έννοιες σχετικές με τα πληροφορικά συστήματα και τον προγραμματισμό και η ανωτέρω μέθοδος δεν αντιμετωπίζει συστηματικά το διδακτικό αυτό πρόβλημα. Ο Du Boulay [1], για παράδειγμα, επισημαίνει τις εξής γενικές κατηγορίες που προκαλούν δυσκολίες στην εκμάθηση του προγραμματισμού:

Η νοητή μηχανή: δυσκολίες που οφείλονται στην κατανόηση των γενικών ιδιοτήτων της «μηχανής» που μαθαίνει να ελέγχει ο σπουδαστής και της σχέσης της με τη φυσική μηχανή.

Οι κανόνες της γλώσσας προγραμματισμού: η σύνταξη και η σημασιολογία ως επέκταση των ιδιοτήτων και της συμπεριφοράς της νοητής μηχανής.

Η εκμάθηση των καθιερωμένων δομών

Η λύση προβλήματος: απόκτηση της ικανότητας καθορισμού, ανάπτυξης, ελέγχου και εκσφαλμάτωσης ενός προγράμματος με τα διαθέσιμα εργαλεία.

Στη συνέχεια αναλύονται, κατά το δυνατόν, οι παραπάνω κατηγορίες δυσκολιών.

- Η εκτέλεση ενός προγράμματος είναι ένα είδος μηχανισμού και ο σπουδαστής χρειάζεται πολύ χρόνο για να εντοπίσει τη σχέση ανάμεσα στο πρόγραμμα και στο μηχανισμό που περιγράφει.
- Ο υπολογιστής μαζί με τον προγραμματιστικό περιβάλλον μπορεί να θεωρηθεί ως ένας μηχανισμός, που χρησιμοποιείται στη συνέχεια για τη δημιουργία άλλων μηχανισμών, δηλαδή των προγραμμάτων. Η περιγραφή του υπολογιστικού συστήματος ως μιας μηχανής θα πρέπει να πραγματοποιηθεί στο κατάλληλο επίπεδο, έτσι ώστε ο σπουδαστής να μπορεί να αντιλαμβάνεται τι συμβαίνει στο εσωτερικό του ΗΥ, γιατί διαφορετικά μπορεί να αναπτύξει δική του θεωρία που τις περισσότερες φορές είναι ανεπαρκής και ίσως λανθασμένη.
- Οι σπουδαστές δυσκολεύονται να κατανοήσουν τι παρουσιάζεται κάθε φορά στην οθόνη. Στην οθόνη του υπολογιστή μπορεί να εμφανίζονται πληροφορίες που αφορούν σε προηγούμενη αλληλεπίδραση του χρήστη – υπολογιστή, ή εμφάνιση δεδομένων εισόδου που πληκτρολόγησε ο χρήστης ή εμφάνιση αποτελεσμάτων συγκεκριμένων εντολών.
- Η διαχείριση της νοητής μηχανής, όπως η ανάκληση, αποθήκευση και έκδοση προγραμμάτων.
- Πολλές σημαντικές αλλαγές στην εσωτερική κατάσταση της μηχανής δεν παρουσιάζονται.

- Το πρόγραμμα γίνεται κατανοητό από το σύστημα βάσει πολύ αυστηρών κανόνων πράγμα που τους προκαλεί δυσκολίες μια και προσδίδουν στο σύστημα ανθρωπομορφικά χαρακτηριστικά.
- Η χρήση αγγλικών λέξεων στις γλώσσες προγραμματισμού δημιουργεί προβλήματα, αφού:
 - προσδίδει στο σύστημα μεγαλύτερη νοημοσύνη
 - στη γλώσσα προγραμματισμού έχουν διαφορετικό νόημα απ' ό τι στην Αγγλική γλώσσα.
- Μικρή στήριξη κατά την εκσφαλμάτωση, αφού συνήθως τα μηνύματα λάθους δεν είναι και τόσο φιλικά προς το χρήστη.

Η δημιουργία λοιπόν εναλλακτικών διδακτικών μεθόδων για την εισαγωγή στον προγραμματισμό, με την ενδεχόμενη υποστήριξη ιδιαίτερων προγραμματιστικών περιβαλλόντων, αποτελεί ένα ζητούμενο. Στην παρούσα εργασία παρουσιάζουμε έναν μικρόκοσμο για μια εισαγωγή στον προγραμματισμό. Πρόκειται για μια μικρή γλώσσα προγραμματισμού, τη γλώσσα X δικής μας επινόησης, και ένα αντίστοιχο προγραμματιστικό περιβάλλον. Η εργασία οργανώνεται ως εξής: στην 2η ενότητα παρουσιάζουμε τις βασικές αρχές σχεδίασης του περιβάλλοντος X-Compiler, στην 3η ενότητα γίνεται μια σύντομη περιγραφή της γλώσσας X και του προγραμματιστικού περιβάλλοντος που αναπτύξαμε. Η 4η ενότητα παρουσιάζει χρήσεις του προγραμματιστικού περιβάλλοντος και η εργασία ολοκληρώνεται με τον επίλογο.

2. Αρχές σχεδίασης του προγραμματιστικού περιβάλλοντος X-Compiler

Η ανάπτυξη προγραμματιστικών περιβαλλόντων για αρχάριους προγραμματιστές πρέπει να γίνεται κατά τέτοιο τρόπο ώστε να αποτελούν ένα αποτελεσματικό εργαλείο για την ανάπτυξη συγκεκριμένου διδακτικού στόχου. Στη συνέχεια παραθέτουμε ένα σύνολο αρχών πάνω στις οποίες στηρίχθηκε το περιβάλλον προγραμματισμού που αναπτύξαμε.

Μινιμαλιστικός σχεδιασμός: Το σύστημα δεν πρέπει να παρουσιάζει πληροφορίες που παρουσιάζονται σπάνια. Κάθε επιπλέον πληροφορία έχει ως αποτέλεσμα την αύξηση του όγκου των πληροφοριών και κατά συνέπεια τη μείωση της σωστής εκμετάλλευσής τους. Σύμφωνα με την γνωστή *αρχή της περιεκτικότητας*, πρέπει να αποφεύγονται τα περιττά σύμβολα, όπως επικεφαλίδες προγραμμάτων, η αναλυτική, ρητή δήλωση των μεταβλητών κτλ.

Απλότητα του συστήματος: Η απλότητα και η ορατότητα του συστήματος είναι δυο βασικά χαρακτηριστικά των γλωσσών προγραμματισμού για αρχάριους [2], [6]. Οι αρχάριοι ξεκινούν να προγραμματίζουν έχοντας μόνο μια μικρή γνώση για τις ιδιότητες της νοητής μηχανής, του νοητού δηλαδή υπολογιστή του οποίου οι ιδιότητες καθορίζονται ή καλύτερα υπονοούνται από τις δομές

της γλώσσας προγραμματισμού. Για τη στήριξη των σπουδαστών στην εκμάθηση αυτών των ιδιοτήτων, η νοητή μηχανή πρέπει να είναι απλή, πρέπει, δηλαδή, να αποτελείται από ένα μικρό αριθμό τμημάτων που αλληλεπιδρούν με τρόπο που μπορεί να γίνει εύκολα κατανοητή.

Τμηματική εκτέλεση και έλεγχος με άμεση ανάδραση: Η άμεση ανάδραση βοηθάει σε μεγάλο βαθμό τον αρχάριο προγραμματιστή στη διαδικασία επίλυσης προβλημάτων και κυρίως στην εκσφαλμάτωση των προγραμμάτων τους. Ένας «γραφικός» εκσφαλματωτής μπορεί να χρησιμοποιηθεί από τον αρχάριο όχι μόνο για την εκσφαλμάτωση, αλλά και για την κατανόηση ενός προγράμματος ακόμη και αν αυτό δεν έχει λάθη. Το σύστημα πρέπει να στηρίζει το χρήστη στη διαδικασία ελέγχου, εκσφαλμάτωσης και εκτέλεσης των προγραμμάτων, διαδικασίες δύσκολες για τους αρχάριους προγραμματιστές. Υπάρχει λοιπόν ανάγκη για ένα χαμηλού επιπέδου εκσφαλματωτή και ένα «ανιχνευτή» (tracer) εκτέλεσης ενός προγράμματος με ταυτόχρονη οπτικοποίηση των δεδομένων.

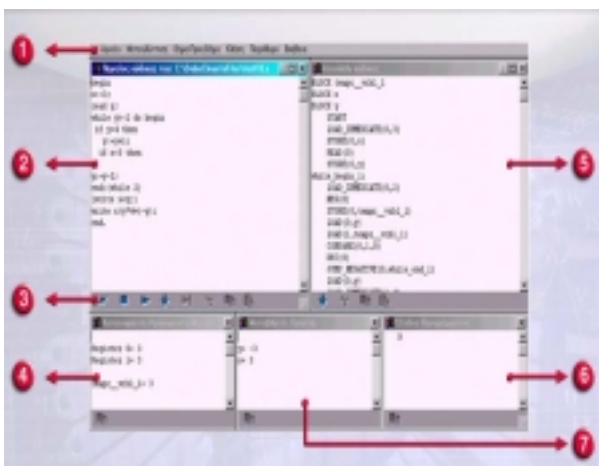
3. Περιγραφή του περιβάλλοντος X-Compiler

Η γλώσσα X είναι ένα υποσύνολο μιας Pascal-like γλώσσας και υποστηρίζει:

Ακέραια αναγνωριστικά και αριθμητικές σταθερές, τις τέσσερις ακέραιες πράξεις ανάμεσα σε ακέραια αναγνωριστικά και αριθμούς, εμφωλευμένες αριθμητικές παραστάσεις οποιουδήποτε μήκους με τη χρήση παρενθέσεων, αρχικοποίηση αναγνωριστικών με την πράξη της εκχώρησης, σχεσιακές παραστάσεις με τη χρήση των τελεστών >, =, και <>, εντολή ελέγχου if rel-exp then ..., εντολή βρόγχου while rel-exp do ..., εντολές write και read, υποστήριξη block εντολών με τη χρήση των begin ... end, σχόλια με τη χρήση των ειδικών χαρακτήρων { και }.

Το περιβάλλον προγραμματισμού της γλώσσας X επιτρέπει την έκδοση, εκσφαλμάτωση και εκτέλεση προγραμμάτων, αποτελείται από 7 συνολικά περιοχές, όπως φαίνεται στην εικόνα 1, και λειτουργεί σε δυο καταστάσεις. Στην κατάσταση «Κλάση αρχαρίων» που αποτελείται από τα παράθυρα (2) και (6) και στην κατάσταση «Κλάση προχωρημένων» που αποτελείται από τα πέντε παράθυρα (2), (4), (5), (6), (7). Στην περιοχή (1) εμφανίζεται το «Μενού Επιλογών» από όπου εκτελούνται λειτουργίες που σχετίζονται με το προγραμματιστικό περιβάλλον της γλώσσας X. Στο παράθυρο «Πηγαίος κώδικας» (περιοχή 2) συντάσσεται ή τροποποιείται ένα πρόγραμμα σε γλώσσα X. Στην περιοχή (3) εμφανίζεται η βοηθητική μπάρα των εργαλείων για την εκτέλεση λειτουργιών που σχετίζονται με τον πηγαίο και τον Assembly κώδικα, όπως μεταγλώττιση, εκτέλεση, βηματική εκτέλεση, συνέχεια, απαλοιφή, αντιγραφή, επικόλληση. Στο παράθυρο «Μεταβλητές συστήματος -

Καταχωρητές» (περιοχή 4) εμφανίζονται οι τιμές των μεταβλητών συστήματος & των καταχωρητών (ορατό μόνο στην κλάση των προχωρημένων). Στο παράθυρο «Assembly Κώδικας» (περιοχή 5, ορατό μόνο στην κλάση των προχωρημένων), εμφανίζονται οι εντολές του Assembly κώδικα που αντιστοιχούν στον πηγαίο κώδικα του παραθύρου (1). Ακόμη στο παράθυρο αυτό ο χρήστης μπορεί να γράψει και το δικό του Assembly κώδικα. Στο παράθυρο «Εξόδος Προγράμματος» (περιοχή 6) εμφανίζονται τα αποτελέσματα του προγράμματος. Στο παράθυρο «Μεταβλητές Χρήστη» (περιοχή 7, ορατό μόνο στην κλάση των προχωρημένων) εμφανίζονται οι τιμές των μεταβλητών που έχει ορίσει ο χρήστης στον πηγαίο κώδικα κατά τη διάρκεια της εκτέλεσης του προγράμματος.



Εικόνα 1



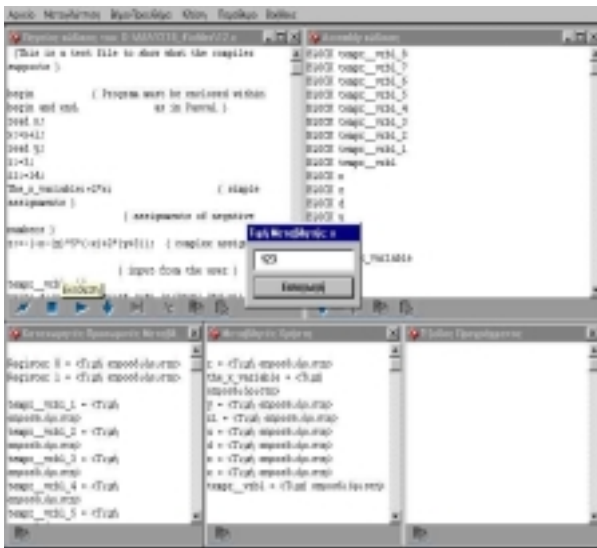
Εικόνα 2

Στη συνέχεια παρουσιάζουμε τις βασικές λειτουργίες του X-Compiler.

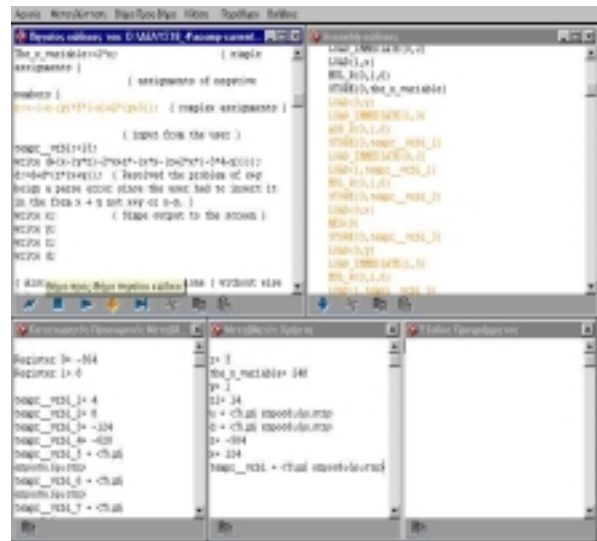
Από το μενού «Μεταγλώττιση» ο χρήστης ζητά τη μεταγλώττιση ενός προγράμματος. Η διαδικασία της μεταγλώττισης ελέγχει τον κώδικα X του προγράμματος για τυπογραφικά και συντακτικά λάθη και παράγει τον ισοδύναμο κώδικα assembly τον οποίο και εμφανίζει στο αντίστοιχο παράθυρο. Στην περίπτωση που η μεταγλώττιση είναι ανεπιτυχής, εμφανίζεται το παράθυρο της εικόνας 2, στο οποίο αναφέρονται τα λάθη καθώς επίσης και κάποιες προειδοποιήσεις. Τα λάθη και οι προειδοποιήσεις εμφανίζονται σε δυο διαφορετικούς χώρους που αποτελούνται από μία αναδιπλωμένη προς τα κάτω λίστα και από ένα πλαίσιο κειμένου. Στην αναδιπλωμένη προς τα κάτω λίστα εμφανίζονται τα λάθη ακολουθούμενα από τη γραμμή στην οποία έγιναν, ή οι προειδοποιήσεις (στον χώρο των προειδοποιήσεων). Μέσα από τις λίστες ο χρήστης μπορεί να επιλέξει κάποιο από τα αναφερόμενα λάθη και αντίστοιχα θα μαυρίσει η περιοχή εκείνη στον

πηγαίο κώδικα όπου βρίσκεται το λάθος. Παράλληλα στο αντίστοιχο πλαίσιο κειμένου εμφανίζεται μια επεξήγηση του λάθους ή της προειδοποίησης.

Από τη στιγμή που ο πηγαίος κώδικας έχει μεταγλωττιστεί επιτυχώς ο χρήστης μπορεί να ζητήσει να εκτελεστεί. Επιλέγοντας «Εκτέλεση Προγράμματος», από το μενού «Μεταγλώττιση», προκαλείται είτε απλή εκτέλεση του κώδικα, είτε βήμα-προς-βήμα. Μια τυπική εκτέλεση ενός προγράμματος φαίνεται στην εικόνα 3, όπου η εκτέλεση του προγράμματος έχει σταματήσει και περιμένει να πληκτρολογηθεί τιμή για τη μεταβλητή x (εκτελεί την εντολή εισόδου read).



Εικόνα 3



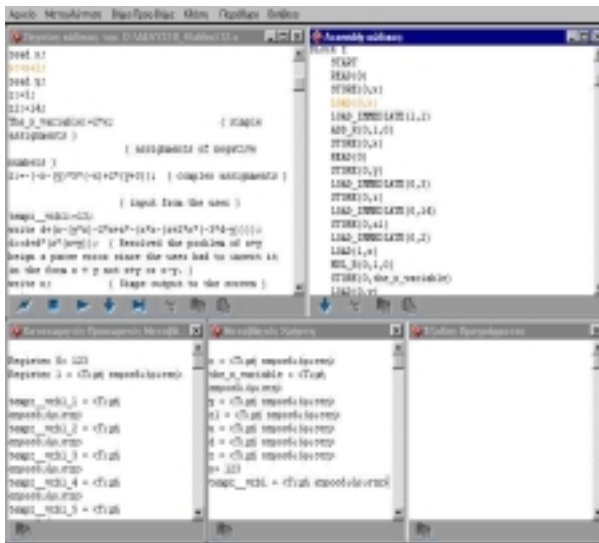
Εικόνα 4

Το μενού «Βήμα-Προς-Βήμα» παρέχει τη δυνατότητα της εκτέλεσης βήμα προς βήμα είτε του πηγαίου κώδικα είτε του assembly κώδικα. Στην εικόνα 4 φαίνεται η εκτέλεση του προγράμματος Βήμα-προς-Βήμα *ως προς τον πηγαίο κώδικα*. Αν επιλεγεί «Εκτέλεση Μέχρι Τέλους» από το μενού «Βήμα-προς-Βήμα», τότε ο κώδικας θα εκτελεστεί βήμα-προς-βήμα αυτόματα μέχρι το τέλος. Στην εικόνα 5 φαίνεται η Βήμα-προς-Βήμα εκτέλεση του Assembly κώδικα με αντιστοίχιση στον πηγαίο κώδικα.

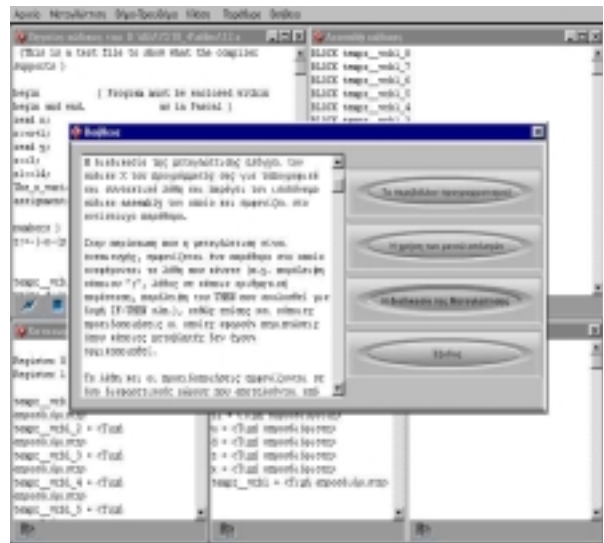
Ακόμη το περιβάλλον προγραμματισμού της γλώσσας X δίνει τη δυνατότητα στο χρήστη να γράψει τον δικό του Assembly κώδικα (εφόσον έχει επιλέξει την κλάση προχωρημένων) και να τον εκτελέσει. Στη συνέχεια αφού γράψει τον Assembly κώδικα στο αντίστοιχο παράθυρο επιλέγει «Εκτέλεση Assembly Κώδικα» από το μενού «Μεταγλώττιση».

Για το προγραμματιστικό περιβάλλον της γλώσσας X υπάρχουν δύο είδη βοήθειας. Η βοήθεια που εμφανίζεται σε περίπτωση που ο χρήστης επιλέξει «Περιεχόμενα» από το μενού «Βοήθεια» (εικόνα

6) και η βοήθεια που εμφανίζεται όταν ο χρήστης κάνει διπλό κλικ είτε σε μια λέξη κλειδί, είτε σε κάποιον από τους τελεστές του πηγαίου κώδικα.



Εικόνα 5



Εικόνα 6

4. Χρησιμοποιώντας τον X-Compiler

Ο X-Compiler αποτελεί ένα ολοκληρωμένο περιβάλλον για μια εισαγωγή στον προγραμματισμό, σχεδιασμένο αποκλειστικά για εκπαιδευτική χρήση. Όπως ήδη περιγράφηκε στην 3η ενότητα, το περιβάλλον του X-Compiler επιτρέπει την έκδοση, εκσφαλμάτωση και εκτέλεση προγραμμάτων γραμμένων στη γλώσσα X – ένα υποσύνολο της Pascal δικής μας επινόησης. Ωστόσο ο X-Compiler ενσωματώνει μια σειρά χαρακτηριστικών που παρουσιάζουν διδακτικό ενδιαφέρον:

- Ο χρήστης μπορεί να παρακολουθήσει όλα τα ενδιάμεσα στάδια μιας εκτέλεσης προγράμματος: τη μεταγλώττιση, τον assembly κώδικα, τα περιεχόμενα των καταχωρητών, τις ενδιάμεσες τιμές όλων των μεταβλητών. Επί πλέον ο χρήστης μπορεί να παρέμβει απευθείας στον assembly κώδικα, να τον μεταβάλλει και στη συνέχεια να τον εκτελέσει.
- Η ύπαρξη πολλών μηνυμάτων, όσο γίνεται ακριβέστερων και επεξηγηματικών, βοηθούν τον αρχάριο προγραμματιστή στην εκσφαλμάτωση, αλλά και την γενικότερη βελτίωση του προγράμματός του.

Καθώς ο X-Compiler σχεδιάστηκε για εκπαιδευτικούς σκοπούς, συνοδεύεται από δυο εγχειρίδια που προτείνουν μια σειρά δραστηριότητες για μια εισαγωγή στον προγραμματισμό. Οι διεθνείς έρευνες στη Διδακτική της Πληροφορικής, αλλά και οι προσωπικές εμπειρίες όσων διδάσκουν προγραμματισμό, μας επιτρέπουν να γνωρίζουμε σχετικά καλά τις δυσκολίες που παρουσιάζει, για

τους μαθητές, αυτή η εισαγωγή στον προγραμματισμό. Για παράδειγμα, η “κατ’ αναλογία σκέψη” [4], [5] πολλές φορές δυσχεραίνει την επίλυση ενός προβλήματος σε προγραμματιστικό περιβάλλον. Έτσι οι μαθητές θεωρούν ότι το παρακάτω πρόγραμμα υπολογίζει το εμβαδόν ενός ορθογωνίου παραλληλογράμμου:

```
emvadon:=vasis * ypsos;  
readln(vasis);  
readln (ypsos);  
writeln(emvadon);
```

Οι μαθητές, κατά κανόνα, εκπλήσσονται όταν το ανωτέρω πρόγραμμα δε δίνει το σωστό αποτέλεσμα. Ωστόσο, στο περιβάλλον του X-Compiler, η αναγραφή των ενδιάμεσων τιμών δείχνει αμέσως όχι μόνο το γεγονός ότι το πρόγραμμα δε δίνει σωστά αποτελέσματα, αλλά και γιατί είναι λανθασμένο.

Η διαδικασία που μεσολαβεί μέχρι την εκτέλεση του προγράμματος είναι άγνωστη στους αρχάριους προγραμματιστές. Η αναγραφή ωστόσο όλων των ενδιάμεσων τιμών και διεργασιών διευκολύνει την κατανόηση της λειτουργίας των Υπολογιστικών Συστημάτων. Έτσι, ως ένα δεύτερο παράδειγμα, δίνουμε το εξής: επιθυμούμε να αντιμετωπίσουμε το απλό πρόβλημα της αντιμετάθεσης των τιμών δυο μεταβλητών A και B:

```
TEMP:=A  
A:=B;  
B:=TEMP;
```

Ωστόσο, αν οι A και B περιέχουν αριθμητικές τιμές, η αντιμετάθεση θα μπορούσε να πραγματοποιηθεί χωρίς τη χρήση της βοηθητικής μεταβλητής TEMP:

```
A:=A+B;  
B:=A-B;  
A:=A-B;
```

Οι εντολές αυτές όμως, δίνουν πάντοτε το σωστό αποτέλεσμα; Για ποιο λόγο; Η παρατήρηση των ενδιάμεσων τιμών δείχνει, με εμπειρικό τρόπο, την ορθότητα των εντολών και μπορεί να υποστηρίξει μια απλή αλγεβρική απόδειξη της ορθότητας αυτής. Ωστόσο, ο assembly κώδικας αποκαλύπτει δυο εγγενείς αδυναμίες της δεύτερης αυτής λύσης:

- Το αντίστοιχο τμήμα κώδικα έχει περισσότερες πράξεις – άρα είναι πιο χρονοβόρο
- Το εύρος των τιμών που μπορούν να χρησιμοποιηθούν είναι μικρότερο – λόγω κινδύνου υπερχείλισης.

Τα δυο ανωτέρω παραδείγματα δείχνουν, μερικές από τις διδακτικές δυνατότητες που προσφέρει ο X-Compiler. Με ένα γενικό τρόπο, το περιβάλλον του X-Compiler δίνει στο χρήστη τη δυνατότητα όχι μόνο να διαπιστώσει αν το πρόγραμμά του δίνει ορθά αποτελέσματα ή όχι, αλλά και να

εντοπίζει πολύ ευκολότερα την πηγή των λογικών του σφαλμάτων ή των αδυναμιών που παρουσιάζει το πρόγραμμα του και να υποστηρίξει τη διαδικασία διόρθωσης ή βελτίωσης του.

5. Επίλογος

Το περιβάλλον X-Compiler, αν και βρίσκεται ήδη στο στάδιο πειραματικής χρήσης του σε σχολεία, δεν έχει ολοκληρώσει τον κύκλο ανάπτυξης του: σχεδιάζεται ήδη η δεύτερη φάση ανάπτυξης του, η οποία θα περιλαμβάνει:

Επέκταση της γλώσσας X προκειμένου να συμπεριλάβει ένα ακόμη μεγαλύτερο κομμάτι της Pascal,

- Ενσωμάτωση ενός τμήματος διδακτικής βοήθειας προς το διδάσκοντα και
- Ενσωμάτωση ενός τμήματος για μια ουσιαστικότερη βοήθεια προς τον μαθητή, προσαρμοσμένη στο είδος του προβλήματος που επιχειρεί να επιλύσει.

Βιβλιογραφία

1. Du Boulay, B., (1989), Some Difficulties Of Learning To Program, *In Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, 283-300.
2. Du Boulay, B., O'Shea, T. and Monk, J., (1989b), The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices, *In Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, 431-446.
3. Brusilovsky P. et al, (1997), Mini-languages: a way to learn programming principle, *Education and Information Technologies*, 2, 65-83
4. Dagdilelis V., (1986), *Conceptions des eleves a propos des notions fondamentales de la programmation informatique en classe de Troisieme*, Memoire D.E.A., Universite Joseph FOURIER, Grenoble, France.
5. Dagdilelis V., (1989), *La validation en programmation : a propos de conceptions des etudiants*, actes V Ecole d'ete de Didactique des Mathematiques et de l'Informatique, Plestin-les-Greves, France.
6. Mendelsohn, P., Green, T.R.G., Brna, P. (1990), Programming Languages in Education: The Search for an Easy Start, *In Psychology of Programming*, Hoc, J., Green, T., Samurcay, R., Gilmore, D. (Eds.) Academic Press, 175-200.