

Ελληνικό Μεσογειακό Πανεπιστήμιο



Τμήμα Ηλεκτρονικών Μηχανικών

Αρχές Γλωσσών Προγρ/σμού & Μεταγλωττιστές Κεφ. 5 Γλώσσες Προγραμματισμού

Δρ. Εμμανουήλ Σκουνάκης

Διδάκτωρ του Πανεπιστημίου Brunel του Λονδίνου
M.Sc. στις Τηλεπικοινωνίες και Δίκτυα Η/Υ
M.Sc. Ηλεκτρονικού Μηχανικού και Μηχανικού Η/Υ

Χανιά, Κρήτη

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Οι γλώσσες προγραμματισμού είναι τα μέσα επικοινωνίας ανάμεσα στον προγραμματιστή εφαρμογών και τον υπολογιστή και αποτελούν εργαλεία δημιουργίας προγραμμάτων.

Ο τρόπος με τον οποίο σκέφτεται ο άνθρωπος είναι πολύ διαφορετικός και πιο πολύπλοκος από τις εντολές που μπορεί να εκτελέσει ένας ηλεκτρονικός υπολογιστής.

Ο ηλεκτρονικός υπολογιστής στηρίζεται σε πολύ απλές ιδέες, και οι πράξεις που μπορεί να κάνει και οι εντολές που έχει τη δυνατότητα να εκτελέσει είναι στοιχειώδεις σε σχέση με την πολυπλοκότητα του ανθρώπινου εγκεφάλου.

Οι εντολές αυτές καλούνται γλώσσα μηχανής (machine language) και είναι η γλώσσα με την οποία προγραμματίζεται ο ηλεκτρονικός υπολογιστής.

Η γλώσσα, αυτή αποτελείται από στοιχειώδεις εντολές, κάτι που κάνει τον προγραμματισμό σε αυτή δύσκολο, επίπονο και μακριά από τον ανθρώπινο τρόπο σκέψης (π.χ. αριθμητικές πράξεις).

Ένα πρόγραμμα το οποίο πρόκειται να εκτελεστεί φορτώνεται αρχικά στην κύρια μνήμη (RAM) του υπολογιστή.

Μια περιγραφή ενός προγράμματος σε φυσική γλώσσα είναι η ακόλουθη:

- Μετάφερε στην Κεντρική Μονάδα Επεξεργασίας τα περιεχόμενα της θέσης μνήμης M1.
- Πρόσθεσε στο προηγούμενο το περιεχόμενο της θέσης μνήμης M2.
- Αποθήκευσε το αποτέλεσμα στη θέση μνήμης M3.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Το παραπάνω πρόγραμμα αποτελείται από τρεις εντολές και αποθηκεύεται στη μνήμη σε δυαδική μορφή. Η μορφή αυτή ονομάζεται γλώσσα μηχανής.

Η γλώσσα μηχανής αντιστοιχίζεται σε μια μορφή πιο κατανοητή από τον άνθρωπο η οποία ονομάζεται συμβολική γλώσσα. Το παραπάνω πρόγραμμα εκφρασμένο σε συμβολική γλώσσα είναι δυνατόν να γραφεί ως:

- LDA M1
- ADA M2
- STA M3

Η ανάπτυξη εφαρμογών σε γλώσσα μηχανής θα ήταν υπερβολικά χρονοβόρα και το οικονομικό κόστος ανάπτυξης υπερβολικά υψηλό.

Επίσης μειονέκτημα είναι ότι υπάρχει εξάρτηση από τον τύπο του Η/Υ (π.χ. διαφορετικές λέξεις για 32bits ή 64bits συστήματα).

Το πρόβλημα αυτό έρχονται να λύσουν οι γλώσσες υψηλού επιπέδου, που είναι οι γνωστές γλώσσες προγραμματισμού.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Οι γλώσσες προγραμματισμού υψηλού επιπέδου (high level programming languages) αποτελούν ένα ενδιάμεσο στρώμα ανάμεσα στον προγραμματιστή εφαρμογών και στη γλώσσα μηχανής.

Οι μεταβλητές, οι εντολές και οι δομές που χρησιμοποιούν είναι πιο εύχρηστες από τις αντίστοιχες της γλώσσας μηχανής και κάνουν τον προγραμματισμό ευκολότερο, πιο ευχάριστο και αποδοτικό.

Έχουν προταθεί και χρησιμοποιηθεί κατά καιρούς πολλές συμβολικές γλώσσες προγραμματισμού, όπου εδώ έγινε αντικατάσταση του δυαδικού κώδικα με σύμβολα όπως Load, Add, Store, κα

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32 PROC      ; procedure begins here
  CMP AX,97     ; compare AX to 97
  JL  DONE     ; if less, jump to DONE
  CMP AX,122   ; compare AX to 122
  JG  DONE     ; if greater, jump to DONE
  SUB AX,32    ; subtract 32 from AX
DONE: RET      ; return to main program
SUB32 ENDP     ; procedure ends here
```

FIGURE 17. Assembly language

Ο Συμβολομεταφραστής αναλαμβάνει τη μετάφραση του κώδικα σε γλώσσα μηχανής.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Τα μειονεκτήματα της χρήσης των συμβολικών γλωσσών προγραμματισμού είναι

- Εξαρτώνται από τον τύπο του Η/Υ
- Είναι πολύ δύσκολη και χρονοβόρα η δημιουργία των προγραμμάτων.

Η πιο σύγχρονη τεχνολογία δημιουργίας προγραμμάτων στους Η/Υ αναφέρεται στις γλώσσες προγραμματισμού υψηλού επιπέδου:

- Τα προγράμματα μπορούν να μεταφερθούν από έναν Η/Υ σε έναν άλλο.
- Ο προγραμματιστής δεν χρειάζεται να γνωρίζει την αρχιτεκτονική του Η/Υ.
- Οι εντολές είναι πιο κατανοητές στον άνθρωπο, έχουν νόημα όταν διαβάζονται από μη πολύ εξειδικευμένους χρήστες.

Ο μεταγλωττιστής ή ο διερμηνευτής μεταφράζει το πρόγραμμα σε γλώσσα μηχανής.

Γλώσσα Μηχανής vs Συμβολική Γλώσσα vs Γλώσσα Υψηλού Επιπέδου

Γλώσσα Μηχανής	Συμβολική Γλώσσα	Γλώσσα Υψηλού Επιπέδου
10101000	INDEX=\$01	sum = 0
10001100	SUM=\$02	For index=1 to 10
00111100	LDA #10	sum=sum+index
01010001	STA INDEX	Next index
01000011	CLA	END
11000000	LOOP ADD INDEX	
10001100	DEC INDEX	
11111111	BNE LOOP	
	STA SUM	
	BRK	

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Στάδια Διαδικασίας Μετάφρασης

1. Πηγαίος Κώδικας (Αρχικό Πρόγραμμα Εντολών)
2. Λεκτικός Αναλυτής
3. Συντακτικός Αναλυτής
4. Σημασιολογικός Αναλυτής
5. Γεννήτρια Κώδικα
6. Εκτελέσιμο Αρχείο

Η τεχνολογία αυτή ξεκίνησε από τις σχετικά παλαιότερες FORTRAN (1957), BASIC (1964) και PASCAL (1970).

Η γλώσσα C (1969) αποτέλεσε έναν σταθμό στην ιστορία των γλωσσών προγραμματισμού υψηλού επιπέδου, έγινε ιδιαίτερα δημοφιλής και παραμένει μέχρι και σήμερα.

Εξελίξεις της όπως η C++ (1980), η C# (2000) και η Java (1995) έχουν κατακτήσει σημαντικό μέρος της αγοράς και εντάσσονται στις πιο δημοφιλείς και περισσότερο χρησιμοποιούμενες στην ανάπτυξη εφαρμογών σήμερα γλώσσες.

Κάθε γλώσσα προγραμματισμού δεν είναι τίποτε άλλο παρά ένα πρόγραμμα το οποίο δέχεται σαν είσοδο το δικό μας πρόγραμμα και το μετασχηματίζει σε ένα πρόγραμμα σε γλώσσα μηχανής ώστε να μπορεί να το εκτελέσει ο υπολογιστής.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Η επιστήμη των μεταφραστών έχει προχωρήσει τόσο, που τα αποτελέσματα της μεταγλώττισης και η βελτιστοποίηση που επιτυγχάνουν στον τελικά παραγόμενο κώδικα μηχανής είναι εντυπωσιακά (π.χ. προτάσεις βελτιστοποίησης του κώδικα από το Visual Studio).

Επίσης, η ευκολία που παρέχεται στον προγραμματιστή μέσω της χρήσης γλωσσών υψηλού επιπέδου είναι πολύ σημαντική, αφού οι εντολές της γλώσσας μηχανής είναι τόσο στοιχειώδεις σε σχέση με τις δομές μιας γλώσσας υψηλού επιπέδου που θα ήταν άσκοπο ή και αδύνατον από άποψη χρόνου και κόστους να κατασκευαστούν με αυτόν τον τρόπο μεγάλες εφαρμογές, και η συντήρησή τους μεταγενέστερα πολύ δύσκολες και χρονοβόρες.

Σε γλώσσα μηχανής γράφονται συνήθως μόνο τμήματα κώδικα που είναι σχετικά μικρά σε μέγεθος και είναι καίριο να εκτελούνται γρήγορα (π.χ. κάποια τμήματα ενός λειτουργικού συστήματος).

Ο χρόνος και ο τρόπος που επιλέγεται για τον μετασχηματισμό ενός προγράμματος από γλώσσα υψηλού επιπέδου σε γλώσσα μηχανής διακρίνουν τις γλώσσες προγραμματισμού σε δύο κύριες κατηγορίες:

Τους **διερμηνευτές (interpreters)** και τους **μεταγλωττιστές (compilers)**.

Οι **διερμηνευτές** μετασχηματίζουν τα προγράμματα μία προς μία τις εντολές σε γλώσσα μηχανής κατά τη διάρκεια της εκτέλεσής τους, ενώ παρέχουν και το περιβάλλον μέσα στο οποίο θα εκτελεστεί το πρόγραμμα που παράχθηκε.

Αυτό σημαίνει ότι ο διερμηνευτής θα μετασχηματίσει την πρώτη εντολή του προγράμματος και θα την εκτελέσει, στη συνέχεια θα κάνει το ίδιο και με τη δεύτερη και με την τρίτη έως ότου τελειώσει το πρόγραμμα.

Οι **μεταγλωττιστές** ακολουθούν διαφορετική διαδικασία. Το πρόγραμμα μεταγλωττίζεται ολόκληρο και αποθηκεύεται στον δίσκο. Στη συνέχεια μπορεί να εκτελεστεί κατευθείαν από τον υπολογιστή.

Καθεμία από τις δύο τεχνικές έχει τα πλεονεκτήματά της.

Περισσότερο διαδεδομένοι είναι οι μεταγλωττιστές, αφού έχουν σημαντικά πλεονεκτήματα σε σχέση με τους διερμηνευτές.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Το σημαντικότερο από αυτά είναι ότι τα προγράμματα που έχουν μεταφραστεί από μεταγλωττιστές τρέχουν γρηγορότερα από τα αντίστοιχα των διερμηνευτών, αφού

- η μεταγλώττιση δεν γίνεται κατά την εκτέλεση της εφαρμογής ώστε να την καθυστερεί και μάλιστα σημαντικά.
- η μεταγλώττιση γίνεται μόνο μία φορά και αποθηκεύεται στον δίσκο και όχι κάθε φορά που θα τρέξει το πρόγραμμα όπως γίνεται με τους διερμηνευτές.

Ακόμα, το πρόγραμμα που μετασχηματίζεται από έναν μεταγλωττιστή είναι ανεξάρτητο από αυτόν στο τέλος της μεταγλώττισης, ενώ ο διερμηνευτής πρέπει να παρέχει και το περιβάλλον για την εκτέλεση του προγράμματος.

Ένα πλεονέκτημα των διερμηνευτών είναι ότι υπάρχει μία περισσότερο διαδραστική σχέση ανάμεσα στον προγραμματιστή και στην εκτέλεση, αφού ο προγραμματιστής μπορεί να εκτελέσει εντολές οι οποίες δεν εντάσσονται σε ένα πρόγραμμα, αλλά κατά την εκτέλεση του προγράμματος μπορεί να διακόψει την εφαρμογή, να εκτελέσει πάλι εντολές, να ελέγξει τιμές μεταβλητών και στη συνέχεια να συνεχίσει την εκτέλεση.

Ένας από τους λόγους που η Java έγινε τόσο δημοφιλής είναι ότι στηρίζεται σε διερμηνευτή, αλλά η φιλοσοφία διαφέρει από αυτήν των υπόλοιπων διερμηνευόμενων γλωσσών.

Η Java μεταφράζεται σε χρόνο μετάφρασης σε μία γλώσσα η οποία δεν είναι κατανοητή στον υπολογιστή (δεν είναι γλώσσα μηχανής, αλλά δεν έχει σύνθετες δομές) και λέγεται Java byte code.

Για να εκτελεστεί πρέπει σε χρόνο εκτέλεσης να μεταφραστεί σε κώδικα μηχανής με τη βοήθεια της εικονικής μηχανής της Java (Java Virtual Machine), ένα λογισμικό που τρέχει στη μηχανή που τελικά εκτελείται το πρόγραμμα. Πρόκειται για μια ενδιάμεση λύση η οποία έχει το μειονέκτημα της μετάφρασης σε χρόνο εκτέλεσης, αλλά ο κώδικας ο οποίος παράγεται είναι ανεξάρτητος από το υλικό (τον υπολογιστή).

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Αντιπροσωπευτικές γλώσσες που χρησιμοποιούν την τεχνική των διερμηνευτών είναι η BASIC, η Java, η Prolog και η Python, ενώ η τεχνική των μεταγλωττιστών ακολουθείται μεταξύ άλλων από τις C, C++, C#, Pascal και FORTRAN.

Ένας άλλος διαχωρισμός των γλωσσών προγραμματισμού μπορεί να γίνει με βάση τη φιλοσοφία που ακολουθείται στον προγραμματισμό.

Η GWBasic είναι από τις πρώτες γλώσσες που προτάθηκαν και χρησιμοποιούν απλές δομές (γραμμικός κώδικας).

Οι γλώσσες αυτές βρίσκονται κοντύτερα στη γλώσσα μηχανής από ότι οι γλώσσες που βρίσκονται σε άλλες κατηγορίες, οι εντολές τους δηλαδή αποσυντίθενται σχετικά εύκολα σε εντολές γλώσσας μηχανής.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Στη συνέχεια εμφανίστηκε ο δομημένος προγραμματισμός (structured programming) ή διαδικασιακός προγραμματισμός με κύριους εκπροσώπους τη C και την Pascal. Στον δομημένο προγραμματισμό τόσο η διάρθρωση του προγράμματος όσο και οι δομές των δεδομένων είναι πιο πολύπλοκες και πλησιάζουν περισσότερο στον τρόπο σκέψης και έκφρασης του ανθρώπου, κάτι που έκανε τον τρόπο αυτόν προγραμματισμού να επικρατήσει.

Κάποια στιγμή εμφανίστηκαν οι γλώσσες που βασίζονται σε αντικείμενα (προσομοιάζοντας τον πραγματικό κόσμο) και έτσι προέκυψαν οι αντικειμενοστραφείς γλώσσες (OOP - Object Oriented Programming).

Για παράδειγμα, σε ένα παιχνίδι με τράπουλα θα δημιουργηθούν 2 ειδών αντικείμενα, για τα τραπουλόχαρτα και τους παίκτες. Τα αντικείμενα αποτελούνται από δεδομένα με ιδιότητες (π.χ. ένα αντικείμενο - φύλλο μπορεί να έχει για δεδομένα το είδος του, τον αριθμό και το χρώμα του) και μεθόδους για να διαχειρίζονται αυτά τα δεδομένα (π.χ. ένας παίκτης μπορεί να έχει τη μέθοδο μέτρησε τα σπαθιά που έχεις).

Κάθε αντικείμενο ανήκει σε μία κλάση (class) που είναι κάτι σαν ο τύπος δεδομένων του αντικειμένου ή μια περιγραφή του. Π.χ. η κλάση car (περιέχει μεταβλητές του τύπου αριθμός τροχών, αριθμός θυρών, υποδύναμη, χρώμα, τιμή, κλπ).

Το βασικότερο χαρακτηριστικό των αντικειμενοστραφών γλωσσών προγραμματισμού είναι η κληρονομικότητα (inheritance).

Η κληρονομικότητα είναι η δυνατότητα να φτιάχνεις κλάσεις οι οποίες κληρονομούν χαρακτηριστικά από άλλες κλάσεις. Έτσι, μπορείς να φτιάξεις μία νέα κλάση επεκτείνοντας μία άλλη, χωρίς φυσικά να καταστρέψεις την αρχική κλάση. Αυτό έχει πολλά πλεονεκτήματά και συνέβαλε καθοριστικά στην καθιέρωση των αντικειμενοστραφών γλωσσών προγραμματισμού.

Υπάρχουν ακόμη γλώσσες προσανατολισμένες σε προγραμματισμό στο διαδίκτυο, όπως η html5, η javascript και η php.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Στάδια Δημιουργίας Εφαρμογών

1. Ανάλυση
2. Σχεδίαση
3. Κατασκευή
4. Συντήρηση

Η γλώσσα προγραμματισμού Python

Η Python είναι μία γλώσσα προγραμματισμού

- απλή για την ανάπτυξη εφαρμογών
- έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες
- διακρίνεται για την ευκολία και την ταχύτητα εκμάθησής της

και σήμερα βρίσκεται στην κορυφή προτίμησης στις γλώσσες προγραμματισμού.

Αρχικά, ήταν μια γλώσσα σεναρίων που χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba (Αμοιβάδας), ένα λειτουργικό σύστημα για δίκτυα υπολογιστών που παρουσιάζει το δίκτυο στον χρήστη σαν να ήταν ένα μόνο μηχάνημα.

Ένα δίκτυο Amoeba αποτελείται από έναν αριθμό σταθμών εργασίας που είναι συνδεδεμένοι σε μια "ομάδα" επεξεργαστών και η εκτέλεση ενός προγράμματος από ένα τερματικό το κάνει να εκτελείται σε οποιονδήποτε από τους διαθέσιμους επεξεργαστές, με το λειτουργικό σύστημα να παρέχει εξισορρόπηση φορτίου.

Η Python είναι γλώσσα προγραμματισμού ανοικτού κώδικα. Αυτό σημαίνει ότι οι χρήστες μπορούν ελεύθερα να την χρησιμοποιούν, να την αντιγράψουν, να την διανέμουν και να την τροποποιούν ανάλογα με τις ανάγκες τους.

Σήμερα ένα παγκόσμιο δίκτυο προγραμματιστών ανοικτού κώδικα αναπτύσσουν και διορθώνουν κώδικες προγραμμάτων, κυκλοφορώντας ταχύτατα νέες βελτιωμένες εκδόσεις τους.

Γλώσσες Προγραμματισμού

Βασικές Γνώσεις

Απόρροια των πλεονεκτημάτων του ανοικτού λογισμικού είναι και ο μεγάλος αριθμός βιβλιοθηκών της Python, ελεύθερων στο διαδίκτυο προς χρήση.

Οι διαθέσιμες βιβλιοθήκες καλύπτουν ένα πολύ μεγάλο εύρος αναγκών. Η Python στηρίζει αλλά και στηρίζεται από την ανάπτυξη των τεχνολογιών του διαδικτύου νέας γενιάς.

Η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL (General Public Licence).

Κατά την GPL οι χρήστες μπορούν να τρέξουν ένα πρόγραμμα για οποιονδήποτε λόγο, να μελετήσουν τη λειτουργία ενός προγράμματος και να το τροποποιήσουν,

να διανείμουν αντίγραφα του προγράμματος, να βελτιώσουν το πρόγραμμα και να προσφέρουν τις βελτιώσεις στους άλλους χρήστες.

Και κάτι που δεν είναι προφανές.

Ενώ η Python 3 έχει κυκλοφορήσει από το 2008, μέχρι και σήμερα δεν έχει αντικαταστήσει την Python 2 (κυκλοφόρησε το 2000).

Ο λόγος είναι ότι υπάρχει ένας μεγάλος όγκος από βιβλιοθήκες διαθέσιμες για την Python 2, ο οποίος δεν έχει μεταφερθεί ακόμη στην Python 3.

Έτσι, η Python 2 παραμένει ζωντανή και ίσως περισσότερο χρησιμοποιούμενη από την Python 3.

Γενικά η Python μειονεκτεί και στο γεγονός ότι επειδή είναι διερμηνευόμενη, είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++.

Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή εφαρμογών που χρειάζονται άμεση απόκριση, όπως π.χ. τα λειτουργικά συστήματα.