Αρχές Γλωσσών Προγραμματισμου

Χρήστος Νομικός

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Ιωαννίνων

2016

Περιεχόμενα

1. Γλώσσες Προγραμματισμού
2. Κατηγορίες Γλωσσών Προγραμματισμού
3. Υλοποίηση Γλωσσών Προγραμματισμού
4. Χρόνος Δέσμευσης

Περιεχόμενα

1. Γλώσσες Προγραμματισμού
2. Κατηγορίες Γλωσσών Προγραμματισμού
3. Υλοποίηση Γλωσσών Προγραμματισμού
4. Χρόνος Δέσμευσης

Γλώσσες Προγραμματισμού

Γλώσσα Προγραμματισμού ονομάζεται ένας συστηματικός συμβολισμός για περιγραφή υπολογιστικών διαδικασιών.

Γλώσσες Προγραμματισμού

Μία γλώσσα προγραμματισμού θα πρέπει να έχει

- αυστηρά ορισμένη σύνταξη
- αυστηρά ορισμένη σημασιολογία

Η σύνταξη μιας γλώσσας καθορίζει αν μία ακολουθία συμβόλων αποτελεί πρόγραμμα γραμμένο στη γλώσσα ή όχι.

Η σημασιολογία μιας γλώσσας καθορίζει τη σημασία του προγράμματος, δηλαδή το ποιά υπολογιστική διαδικασία περιγράφει το πρόγραμμα.

Γλώσσες Προγραμματισμού

Μία γλώσσα προγραμματισμού θα πρέπει να είναι κατανοητή

- από τον άνθρωπο, ο οποίος θα πρέπει να μπορεί να περιγράφει μία υπολογιστική διαδικασία
- από τον υπολογιστή, ο οποίος θα πρέπει να μπορεί να εκτελεί την υπολογιστική διαδικασία

Γλώσσες Προγραμματισμού

Ερώτημα: Γιατί δεν επιλέγεται ως γλώσσα προγραμματισμού η φυσική γλώσσα (η οποία είναι κατανοητή από τον άνθρωπο);

Απάντηση:

- Η φυσική γλώσσα είναι πάρα πολύ σύνθετη και είναι δύσκολο να γίνει κατανοητή από τον υπολογιστή.
- Η περιγραφή μιας υπολογιστικής διαδικασίας σε φυσική γλώσσα μπορεί να περιέχει ασάφειες.

Ερώτημα: Γιατί δεν επιλέγεται ως γλώσσα προγραμματισμού η γλώσσα μηχανής (η οποία είναι άμεσα κατανοητή από τον υπολογιστή);

Γλώσσες Προγραμματισμού

Απάντηση: αυτή ήταν η λύση όταν πρωτοεμφανίστηκαν οι υπολογιστές. Ωστόσο, η γλώσσα μηχανής παρουσιάζει πολλά μειονεκτήματα:

- εξαρτάται από τον υπολογιστή, συνεπώς ο προγραμματιστής πρέπει να μαθαίνει μία διαφορετική γλώσσα για κάθε υπολογιστή.
- αν αλλάξει ο υπολογιστής, τότε το σύνολο των προγραμμάτων πρέπει να ξαναγραφτούν στη γλώσσα μηχανής του νέου υπολογιστή.
- περιέχει πολύ απλές εντολές, συνεπώς απαιτούνται μεγάλα σε μήκος προγράμματα ακόμη και για απλούς υπολογισμούς.
- είναι δυσνόητη για τον άνθρωπο και συνεπώς είναι σύνηθες φαινόμενο η εμφάνιση σφαλμάτων στα προγράμματα.
- μικρές αλλαγές στο πρόγραμμα είναι δυνατόν να επιφέρουν πολλές αλλαγές διευθύνσεων.

Γλώσσες Προγραμματισμού

Με την πάροδο του χρόνου, αναπτύχθηκαν οι συμβολικές γλώσσες (assembly).

Η συμβολική γλώσσα assembly αναθέτει συμβολικά ονόματα στις εντολές ενός υπολογιστή που παραπέμπουν στη λειτουργία της κάθε εντολής, ενώ δινει τη δυνατότητα ανάθεση συμβολικών ονομάτων και σε θέσεις μνήμης.

Γλώσσες Προγραμματισμού

Η μετατροπή ενός προγράμματος σε συμβολική γλώσσα στο αντίστοιχο πρόγραμμα σε γλώσσα μηχανής, γίνεται από το συμβολομεταφραστή (assembler).

Ο συμβολομεταφραστής, είναι ένα πρόγραμμα γραμμένο σε γλώσσα μηχανής, το οποίο αντικαθιστά τα συμβολικά ονόματα με τους αντίστοιχους δυαδικούς κωδικούς των έντολών σε γλώσσα μηχανής.

Γλώσσες Προγραμματισμού

Η συμβολική γλώσσα:

- διορθώνει το τελευταίο πρόβλημα της γλώσσας μηχανής.
- είναι κάπως πιο κατανοητή από τον άνθρωπο.
- ωστόσο τα τρία πρώτα προβλήματα παραμένουν.

Γλώσσες Προγραμματισμού

Πιο εξελιγμένοι συμβολομεταφραστές επιτρέπουν τον ορισμό μακροεντολών, δηλαδή την αντιστοίχιση παραμετρικών συντομογραφιών σε ακολουθίες εντολών γλώσσας μηχανή που επαναλαμβάνονται συχνά.

Ωστόσο και σε αυτή την περίπτωση, η αντιστοιχία των μακροεντολών της συμβολικής γλώσσας και των εντολών της γλώσσας μηχανής ήταν προφανής.

Γλώσσες Προγραμματισμού

Οι γλώσσες υψηλού επιπέδου αποτελούν μία ενδιάμεση λύση ανάμεσα στη φυσική γλώσσα και τη γλώσσα μηχανής:

- είναι εύκολα κατανοητές από τον άνθρωπο και μπορούν να γίνουν κατανοητές από τον υπολογιστή.
- είναι σε μεγάλο βαθμό ανεξάρτητες από τον υπολογιστή.
- περιέχουν αρκετά περιεκτικές εντολές και άλλα χαρακτηριστικά που επιτρέπουν την ευκολότερη περιγραφή υπολογιστικών διαδικασιών.

Γλώσσες Προγραμματισμού

Η κατασκευή μίας γλώσσας προγραμματισμού απαιτεί τα παρακάτω:

- σχεδιασμό της γλώσσας: καθορισμό της σύνταξης και της σημασιολογίας.
- υλοποίηση της γλώσσας: κατασκευή ενός προγράμματος (σε γλώσσα μηχανής ή άλλη γλώσσα που είναι ήδη υλοποιημένη) που καθιστά τη γλώσσα κατανοητή για τον υπολογιστή.

Γλώσσες Προγραμματισμού

Ερώτημα: Για ποιο λόγο υπάρχουν πολλές γλώσσες προγραμματισμού;

Απάντηση:

- στην πορεία του χρόνου, η σχεδίαση των γλωσσών προγραμματισμού αναπτύχθηκε, προέκυψαν νέες τάσεις και νέες ανάγκες, ανακαλύφτηκαν καλύτεροι τρόποι για να κάνουμε τα ίδια πράγματα, ενώ το μέγεθος και το είδος των εφαρμογών άλλαξε και η αρχιτεκτονική των υπολογιστών εξελίχθηκε.
- υπάρχουν πολλά διαφορετικά πεδία εφαρμογής, που επιβάλλουν διαφορετικά χαρακτηριστικά στις γλώσσες.
- διαφορετικοί προγραμματιστές έχουν διαφορετική αντίληψη για το τι κάνει μία γλώσσα καλύτερη από μία άλλη.

Γλώσσες Προγραμματισμού

Ερώτημα: Ποιοί παράγοντες συνιστούν στο να είναι μία γλώσσα επιτυχημένη;

Απάντηση:

- ευκολία στη μάθηση
- ευκολία στην συγγραφή και συντήρηση ορθών προγραμμάτων και αναγνωσιμότητα
- εκφραστική δύναμη
- δυνατότητα καλής υλοποίησης
- ύπαρξη υλοποιήσεων και εφαρμογών ανοικτού κώδικα
- οικονομικοί παράγοντες

Γλώσσες Προγραμματισμού

Σκοπιμότητα μελέτης γλωσσών προγραμματισμού:

- καλύτερη κατανόηση των χαρακτηριστικών των γλωσσών προγραμματισμού. Η ενσωμάτωση ενός χαρακτηριστικού μπορεί να προσθέσει ευελιξία σε μία γλώσσα, ωστόσο ενδέχεται να επιβαρύνει τη λειτουργία των προγραμμάτων (π.χ. σε χρόνο ή μνήμη), ακόμη και αυτών που δεν χρησιμοποιούν το χαρακτηριστικό.
- καλύτερη κατανόηση των προβλημάτων και των αλληλοαντικρουόμενων παραγόντων κατά το σχεδιασμό και την υλοποίηση μίας γλώσσας προγραμματισμού.
- επιλογή κατάλληλης γλώσσας προγραμματισμού ανάλογα με το πεδίο εφαρμογής.

Γλώσσες Προγραμματισμού

Σκοπιμότητα μελέτης γλωσσών προγραμματισμού (συνέχεια):

- δυνατότητα εκμάθησης νέων γλωσσών προγραμματισμού.
- δυνατότητα εξομοίωσης χαρακτηριστικών που δεν είναι διαθέσιμα σε μία γλώσσα προγραμματισμού.
- καλύτερη αξιοποίηση των γλωσσών προγραμματισμού.
- δυνατότητα σχεδιασμού νέων γλωσσών προγραμματισμού.

Περιεχόμενα

1 Γλώσσες Προγραμματισμού

2 Κατηγορίες Γλωσσών Προγραμματισμού

3 Υλοποίηση Γλωσσών Προγραμματισμού

4 Χρόνος Δέσμευσης

Κατηγορίες Γλώσσων Προγραμματισμού

Δεν υπάρχει μονοσήμαντος τρόπος διαχωρισμού των γλωσσών προγραμματισμού σε κατηγορίες.

Επιπλέον δεν είναι πάντα δυνατό μιά γλώσσα προγραμματισμού να ενταχθεί με ξεκάθαρο τρόπο σε μία κατηγορία.

Κατηγορίες Γλώσσων Προγραμματισμού

Μία κατηγοριοποίηση των γλωσσών είναι η παρακάτω:

- προστακτικές (imperative): το πρόγραμμα περιγράφει το πώς θέλουμε να γίνει ο υπολογισμός.
  - von Neumann
    - γλώσσες σεναρίων (scripting)
  - αντικειμενοστρεφείς (object oriented)
- δηλωτικές (declarative): το πρόγραμμα περιγράφει το τι θέλουμε να υπολογιστεί.
  - συναρτησιακές (functional)
  - λογικές (logic)
  - γλώσσες ροής δεδομένων (dataflow)

Κατηγορίες Γλώσσων Προγραμματισμού

von Neumann: C, Pascal, Fortran, Algol, ...

Το πρόγραμμα αποτελείται από μια σειρά εντολών τις οποίες ο υπολογιστής εκτελεί ακολουθιακά και οι οποίες επιδρούν στις μεταβλητές του προγράμματος.

Κατηγορίες Γλώσσων Προγραμματισμού

Γλώσσες σεναρίων: Perl, Python, PHP, Javascript, ...

Δίνουν έμφαση στην συγκόλληση προγραμμάτων που έχουν αναπτυχθεί ανεξάρτητα. Χρησιμοποιούνται συχνά για κατασκευή προτοτύπων.

Κατηγορίες Γλώσσων Προγραμματισμού

Αντικειμενοστρεφείς γλώσσες: Simula, C++, Smalltalk, Java, ...

Το πρόγραμμα αποτελείται από ένα σύνολο αντικειμένων τα οποία αλληλεπιδρούν μεταξύ τους μέσω μηνυμάτων. Κάθε αντικείμενο μπορεί να ανταποκριθεί σε ένα συγκεκριμένο σύνολο μηνυμάτων που καθορίζουν τη συμπεριφορά του.

Κατηγορίες Γλώσσων Προγραμματισμού

Συναρτησιακές γλώσσες: Lisp, ML, Scheme, Miranda, Haskell, ...

Το πρόγραμμα αποτελείται από ένα πλήθος συναρτήσεων, που ορίζονται με βάση κάποιες πρωτογενείς συναρτήσεις χρησιμοποιώντας σύνθεση και αναδρομή. Το επιθυμητό αποτέλεσμα επιτυγχάνεται δίνοντας στον υπολογιστή μία παράσταση την οποία του ζητάμε να αποτιμήσει.

Κατηγορίες Γλώσσων Προγραμματισμού

Λογικές γλώσσες: Prolog και οι επεκτάσεις της.

Το πρόγραμμα αποτελείται από ένα σύνολο προτάσεων, οι οποίες θεωρούμε ότι είναι αληθείς και περιγράφουν τον κόσμο. Το επιθυμητό αποτέλεσμα επιτυγχάνεται δίνοντας στον υπολογιστή μία πρόταση και ζητώντας του να βρει κάτω από ποιές συνθήκες η πρόταση αυτή αποτελεί λογικό συμπέρασμα των προτάσεων του προγράμματος.

Κατηγορίες Γλώσσων Προγραμματισμού

Γλώσσες ροής δεδομένων: Id, Val

Ο υπολογισμός γίνεται με ροή πληροφορίας μεταξύ απλών συναρτησιακών κόμβων, οι οποίοι ενεργοποιούνται όταν λάβουν πληροφορία στην είσοδο τους.

Περιεχόμενα

1. Γλώσσες Προγραμματισμού
2. Κατηγορίες Γλωσσών Προγραμματισμού
3. Υλοποίηση Γλωσσών Προγραμματισμού
4. Χρόνος Δέσμευσης

Υλοποίηση Γλωσσών Προγραμματισμού

Η υλοποίηση μίας γλώσσας προγραμματισμού μπορεί να γίνει με έναν από τους παρακάτω τρόπους:

- μετάφραση ή μεταγλώττιση: κατασκευάζεται ένα πρόγραμμα που ονομάζεται μεταφραστής ή μεταγλωττιστής (compiler), το οποίο με είσοδο ένα πρόγραμμα στη γλώσσα προγραμματισμού που υλοποιείται, παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.
- διερμηνεία: σχεδιάζεται ένα πρόγραμμα που ονομάζεται διερμηνέας (interpreter), το οποίο εξομοιώνει έναν ιδεατό υπολογιστή, του οποίου η γλώσσα μηχανής είναι η γλώσσα προγραμματισμού που υλοποιείται. Ο διερμηνέας παίρνει ως είσοδο το πρόγραμμα και την είσοδο σε αυτό, και το εκτελεί.
- υβριδικές μέθοδοι: το πρόγραμμα μεταφράζεται σε μία γλώσσα χαμηλού επιπέδου, που είναι όμως διαφορετική από τη γλώσσα μηχανής, η οποία στη συνέχεια υλοποιείται με διερμηνέα.

Υλοποίηση Γλωσσών Προγραμματισμού

Πλεονεκτήματα υλοποίησης με μετάφραση:

- η μετάφραση του προγράμματος γίνεται μία μόνο φορά (ωστόσο απαιτείται ξεχωριστή μετάφραση για διαφορετικό υπολογιστή ή λειτουργικό).
- δεν χρειάζεται να υπάρχει ο μεταγλωττιστής για να εκτελεστεί το μεταφρασμένο πρόγραμμα.
- ένας καλός μεταφραστής μπορεί να παράγει πολύ αποδοτικό εκτελέσιμο πρόγραμμα, το οποίο πρακτικά είναι γρηγορότερο από αυτό που θα έγραφε ένας προγραμματιστής σε γλώσσα μηχανής ή συμβολική γλώσσα.

Υλοποίηση Γλωσσών Προγραμματισμού

Μειονεκτήματα υλοποίησης με μετάφραση:

- το εκτελέσιμο πρόγραμμα δεν είναι μεταφέρσιμο σε άλλο υπολογιστή / λειτουργικό σύστημα.
- είναι πιο δύσκολη η αποσφαλμάτωση του εκτελέσιμου προγράμματος, επειδή δεν υπάρχει το αρχικό πρόγραμμα.

Υλοποίηση Γλωσσών Προγραμματισμού

Πλεονεκτήματα υλοποίησης με διερμηνεία:

- το πρόγραμμα μπορεί να εκτελείται ακόμη και αν σε κάποια σημεία του υπάρχουν συντακτικά λάθη.
- παρέχονται καλύτερα διαγνωστικά μηνύματα, επειδή κατά την εκτέλεση είναι διαθέσιμο το αρχικό πρόγραμμα.
- μπορουν να υλοποιηθούν χαρακτηριστικά που είναι σχεδόν αδύνατο να υλοποιηθούν με μετάφραση, όπως για παράδειγμα παραγωγή και εκτέλεση κώδικα σε χρόνο εκτέλεσης.

Υλοποίηση Γλωσσών Προγραμματισμού

Μειονεκτήματα υλοποίησης με διερμηνεία:

- ο διερμηνέας χρειάζεται να επεξεργαστεί ένα τμήμα του προγράμματος κάθε φορά που αυτό εκτελείται.
- για την εκτέλεση του προγράμματος απαιτείται ο διερμηνέας.
- ο χρήστης του προγράμματος έχει πρόσβαση στον πηγαίο κώδικα, κατι αδιανόητο σε εμπορικά προγράμματα.

Υλοποίηση Γλωσσών Προγραμματισμού

Πλεονεκτήματα υβριδικής υλοποίησης:

- μπορούν να συνδυάσουν τα πλεονεκτήματα μετάφρασης και διερμηνείας.
- αν αλλάξει το λειτουργικό σύστημα ή ο υπολογιστής, αρκεί να τροποποιηθεί ο διερμηνέας της ενδιάμεσης γλώσσας.

Υλοποίηση Γλωσσών Προγραμματισμού

Προεπεξεργασία: είναι ένα στάδιο επεξεργασίας του προγράμματος, το οποίο προηγείται της μετάφρασης ή διερμηνείας και έχει ως σκοπό την αφαίρεση σχολίων, ανάπτυξη μακροεντολών και συντομογραφιών, αναπαράσταση του προγράμματος που πρόκειται να εκτελεστεί από διερμηνέα σε μία πιο βολική μορφή, ...

Υλοποίηση Γλωσσών Προγραμματισμού

Η μετάφραση είναι δυνατόν να παράγει κώδικα σε συμβολική γλώσσα ή ακόμα και σε άλλη γλώσσα υψηλού επιπέδου (για παράδειγμα ορισμένες από τις πρώτες υλοποιήσεις της C++ παρήγαγαν κώδικα σε C, ο οποίος μεταφραζόταν στη συνέχεια από έναν μεταφραστή της C).

Υλοποίηση Γλωσσών Προγραμματισμού

Όταν χρησιμοποιούνται συναρτήσεις βιβλιοθήκης συνήθως ο μεταφραστής καλέι ένα ξεχωριστό πρόγραμμα που λέγεται συνδέτης (linker) για να ενσωματώσει τις συναρτήσεις βιβλιοθήκης στο μεταφρασμένο πρόγραμμα.

Υλοποίηση Γλωσσών Προγραμματισμού

Μία τυπική μεταγλώττιση περιλαμβάνει τις παρακάτω φάσεις:

1. λεκτική ανάλυση
2. συντακτική ανάλυση
3. σημασιολογική ανάλυση και παραγωγή ενδιάμεσου κώδικα
4. βελτίωση ενδιάμεσου κώδικα
5. παραγωγή τελικού κώδικα
6. βελτίωση τελικού κώδικα

Υλοποίηση Γλωσσών Προγραμματισμού

Η λεκτική ανάλυση διαβάζει το πρόγραμμα ως ακολουθία χαρακτήρων και ομαδοποιεί τους χαρακτήρες σε λεκτικές μονάδες. Οι λεκτικές μονάδες αποτελούν τις μικρότερες συντακτικές μονάδες του προγράμματος στις οποίες μπορεί να αποδοθεί σημασία.

Η συντακτική ανάλυση ελέγχει τη συντακτική ορθότητα του προγράμματος και μετατρέπει την ακολουθία λεκτικών μονάδων που προκύπτει από τη λεκτική ανάλυση σε ένα συντακτικό δέντρο, που περιγράφει πλήρως τη δομή του προγράμματος όπως αυτή περιγράφεται από τους κανόνες σύνταξης (π.χ. σε BNF) της γλώσσας προγραμματισμού.

Υλοποίηση Γλωσσών Προγραμματισμού

Η σημασιολογική ανάλυση και παραγωγή ενδιάμεσου κώδικα εξάγει τη σημασία του προγράμματος από το συντακτικό δέντρο, παράγοντας τον ενδιάμεσο κώδικα. Επίσης πραγματοποιεί σημασιολογικούς ελέγχους και ελέγχει συντακτικές ιδιότητες που δεν αποτυπώνονται στους τυπικούς κανόνες σύνταξης της γλώσσας.

Η βελτίωση ενδιάμεσου κώδικα παράγει ισοδύναμο ενδιάμεσο κώδικα που είναι πιο αποδοτικός (τρέχει γρηγορότερα ή απαιτεί λιγότερη μνήμη).

Υλοποίηση Γλωσσών Προγραμματισμού

Η παραγωγή τελικού κώδικα μετατρέπει τον ενδιάμεσο κώδικα σε ισοδύναμο κώδικα σε γλώσσα μηχανής (ή συμβολική γλώσσα ή οποιαδήποτε άλλη έχει επιλεγεί ως τελική γλώσσα).

Η βελτίωση τελικού κώδικα παράγει ισοδύναμο τελικό κώδικα που είναι πιο αποδοτικός. Σε αυτό το στάδιο λαμβάνεται υπόψη η αρχιτεκτονική του υπολογιστή.

Υλοποίηση Γλωσσών Προγραμματισμού

Είναι σύνηθες δύο ή περισσότερες φάσεις να είναι χρονικά επικαλυπτόμενες, κάτι που σημαίνει ότι μία φάση μπορεί να αρχίσει να εκτελείται πριν ολοκληρωθεί η προηγούμενη.

Για παράδειγμα η συντακτική ανάλυση μπορεί να ξεκινήσει χωρίς να είναι απαραίτητο να έχουν σχηματιστεί όλες οι λεκτικές μονάδες του προγράμματος.

Υλοποίηση Γλωσσών Προγραμματισμού

Ένα σύνολο φάσεων το οποίο πρέπει να ολοκληρωθεί πρωτού ξεκινήσουν οι επόμενες φάσεις ονομάζεται πέρασμα.

Κάθε πέρασμα μπορεί να υλοποιηθεί από ένα ξεχωριστό πρόγραμμα, το οποίο διαβάζει την είσοδο από ένα αρχείο και γράφει την έξοδο σε ένα άλλο αρχείο.

Υλοποίηση Γλωσσών Προγραμματισμού

Οι φάσεις της λεκτικής ανάλυσης, της συντακτικής ανάλυσης, της σημασιολογικής ανάλυσης και παραγωγής ενδιάμεσου κώδικα και της βελτίωσης ενδιάμεσου κώδικα αποτελούν το εμπρόσθιο τμήμα του μεταφραστή.

Οι φάσεις της παραγωγής βελτίωσης τελικού κώδικα αποτελούν το οπίσθιο τμήμα του μεταφραστή.

Υλοποίηση Γλωσσών Προγραμματισμού

Μία συνηθισμένη πρακτική είναι το εμπρόσθιο και το οπίσθιο τμήμα του μεταφραστή να αποτελούν δύο ξεχωριστά περάσματα έτσι ώστε:

- μεταφραστές της ίδιας γλώσσας για διαφορετικά περιβάλλοντα (υπολογιστές ή λειτουργικά συστήματα) να μοιράζονται το ίδιο εμπρόσθιο τμήμα.
- μεταφραστές διαφορετικών γλωσσών ή εκδόσεων μιας γλώσσας να μοιράζονται το ίδιο οπίσθιο τμήμα σε ένα δεδομένο υπολογιστή και λειτουργικό σύστημα.

Περιεχόμενα

1. Γλώσσες Προγραμματισμού
2. Κατηγορίες Γλωσσών Προγραμματισμού
3. Υλοποίηση Γλωσσών Προγραμματισμού
4. Χρόνος Δέσμευσης

Χρόνος Δέσμευσης

Δέσμευση ονομάζεται μία σύνδεση ανάμεσα σε δύο διαφορετικά στοιχεία του προγράμματος ή της γλώσσας προγραμματισμού.
Παραδείγματα δέσμευσης:

- ονόματος με το συστατικό του προγράμματος στο οποίο αναφέρεται
- τελεστή με την πράξη την οποία πραγματοποιεί
- μεταβλητής με τιμή

Χρόνος Δέσμευσης

Χρόνος δέσμευσης είναι ο χρόνος κατά τον οποίο πραγματοποιείται η δέσμευση.

Ο χρόνος δεν καθορίζεται με απόλυτες τιμές (ημερομηνία-ώρα) αλλά σε σχέση με τον κύκλο ζωής της γλώσσας, του προγράμματος και της εκτέλεσής του.

Χρόνος Δέσμευσης

Οι κυριότεροι χρόνοι κατά τους οποίους μπορεί να γίνει μία δέσμευση είναι οι παρακάτω:

- Χρόνος σχεδιασμού της γλώσσας:
  - όνομα - εντολή
  - τελεστής - πράξη
  - τύπος - πεδίο τιμών
- Χρόνος υλοποίησης της γλώσσας
  - βασικοί τύποι - μήκος αναπαράστασης σε bits
  - τιμή - ακολουθία απο bits
  - συντακτικό λάθος - διαγνωστικό μήνυμα
  - στοίβα - μέγεθος
  - σωρός - μέγεθος
  - εξαίρεση - χειρισμός

Χρόνος Δέσμευσης

- Χρόνος συγγραφής του προγράμματος
  - ονομα - υποπρόγραμμα
  - πίνακας - διάσταση και μέγεθος
  - μεταβλητή - τύπος
- Χρόνος μεταγλώττισης
  - συστατικά του προγράμματος - κώδικας σε γλώσσα μηχανής
  - στατική μεταβλητή - σχετική διεύθυνση μνήμης
- Χρόνος σύνδεσης
  - δεσμεύσεις που εμπλέκουν υποπρογράμματα απο βιβλιοθήκες και τμήματα του προγράμματος που μεταγλωττίζονται χωριστά.

Χρόνος Δέσμευσης

- Χρόνος φόρτωσης
  - μεταβλητή - φυσική διεύθυνση μνήμης
- Χρόνος εκτέλεσης
  - μεταβλητή - τιμή
  - μεταβλητή - τύπος
  - μεταβλητή - φυσική διεύθυνση μνήμης
  - πίνακας - διάσταση και μέγεθος

Χρόνος Δέσμευσης

Παρατηρούμε ότι ο χρόνος στον οποίο μπορεί να γίνει μία δέσμευση δεν είναι μονοσήμαντα ορισμένος.

Μια δέσμευση λέγεται στατική αν λαμβάνει χώρα πρίν αρχίσει η εκτέλεση του προγράμματος και παραμένει σταθερή κατά την εκτέλεση.

Μια δέσμευση λέγεται δυναμική αν λαμβάνει χώρα ή αλλάζει κατά την εκτέλεση του προγράμματος.

Οι δυναμικές δεσμεύσεις προσφέρουν μεγαλύτερη ευελιξία σε σχέση με τις στατικές, ωστόσο επιβαρύνουν το χρόνο εκτέλεσης.