

# Κεφάλαιο 6

## Υλοποίηση Γλωσσών Προγραμματισμού

Προπτυχιακό μάθημα  
Αρχές Γλωσσών Προγραμματισμού  
Π. Ροντογιάννης



# Μεταγλωττιστής

## ■ Πρόγραμμα

- Διαβάζει προγράμματα δεδομένης γλώσσας (πηγαία γλώσσα) και τα μετατρέπει σε ισοδύναμα προγράμματα άλλης γλώσσας (γλώσσα στόχος).
- Αναφέρει στον προγραμματιστή τυχόν λάθη, που υπάρχουν στο πηγαίο πρόγραμμα



# Βασικές Διεργασίες Μεταγλωττιστή

- Ανάλυση πηγαίου προγράμματος
- Σύνθεση τελικού προγράμματος–στόχου
- Κατευθυνόμενος από το συντακτικό
  - Η διαδικασία μεταγλώττισης ενός πηγαίου προγράμματος καθορίζεται σχεδόν αποκλειστικά από τη συντακτική δομή του



# Λεκτική Ανάλυση

- Λειτουργία κατά την οποία ένα τμήμα του μεταγλωττιστή (λεκτικός αναλυτής) διαβάσει το πηγαίο πρόγραμμα χαρακτήρα προς χαρακτήρα.
- Οι χαρακτήρες ομαδοποιούνται σε ανεξάρτητες οντότητες πχ λέξεις-κλειδιά, μεταβλητές, αριθμοί κλπ.



# Παράδειγμα

Ο λεκτικός αναλυτής διαβάσει την εντολή:

```
height := base+old *10;
```

Η εντολή χωρίζεται στις οντότητες «height», «:=», «base», «+», «old», «\*», «10» και «;».

Ο λεκτικός αναλυτής πραγματοποιεί και άλλες αρχικές διεργασίες, που διευκολύνουν τις επόμενες φάσεις της μεταγλώττισης.



# Συντακτική Ανάλυση

- Η ακολουθία από ανεξάρτητες οντότητες μεταβιβάζεται στον συντακτικό αναλυτή
  - Ελέγχει αν η ακολουθία σχηματίζει συντακτικά ορθές φράσεις της πηγαίας γλώσσας
  - Η πληροφορία, που διαθέτει για αυτό, είναι ένα σύνολο κανόνων, που περιγράφουν το συντακτικό της πηγαίας γλώσσας και έχουν μορφή επεκτεταμένης γραμματικής χωρίς συμφραζόμενα



# Συντακτική Ανάλυση

- Αν εντοπισθεί λάθος στο συντακτικό εξάγεται κατάλληλο διαγνωστικό μήνυμα
- Σε πολλές περιπτώσεις λάθους ο συντακτικός αναλυτής ξεπερνά το λάθος και συνεχίζει τη διαδικασία
- Στο τέλος έχει σχηματισθεί ένα συντακτικό δέντρο, που χρησιμοποιείται στην επόμενη φάση της μεταγλώττισης



# Σημασιολογική Ανάλυση

- Έλεγχος της στατικής σημασιολογίας κάθε φράσης του πηγαίου προγράμματος. Πχ για μία εντολή ανάθεσης ελέγχεται η δήλωση όλων των μεταβλητών, που θα χρησιμοποιηθούν
- Μετάφραση στην περίπτωση, που είναι σωστά
- Παραγωγή ενδιάμεσου κώδικα, που υλοποιεί τη συγκεκριμένη εντολή





# Βελτιστοποίηση

- Ο ενδιάμεσος κώδικας μετασχηματίζεται σε ισοδύναμο και περισσότερο αποδοτικό κώδικα.
- Η φάση αποτελείται από υπο-φάσεις, που επαναλαμβάνονται
- Επιτρέπεται στον προγραμματιστή να απενεργοποιήσει τη φάση των βελτιστοποιήσεων



# Παραγωγή Τελικού Κώδικα

- Μετατροπή ενδιάμεσου κώδικα σε εκτελέσιμο κώδικα μηχανής
  - Απαιτείται λεπτομερής γνώση χαρακτηριστικών μηχανής στην οποία θα εκτελεσθεί ο κώδικας-στόχος
- Απλούστευση της διαδικασίας με συνένωση φάσεων παραγωγής ενδιάμεσου και τελικού κώδικα



# Διαδικασίες (procedures)

- Χαρακτηριστικό σύγχρονων γλωσσών προγραμματισμού
  - Επιτρέπει την ονομασία τμημάτων του κώδικα και «κλήση» τους από άλλα σημεία του προγράμματος
    - Το τμήμα αυτό του κώδικα ονομάζεται σώμα της διαδικασίας
  - Έχει σύνολο από τυπικές παραμέτρους
  - Καλείται με πραγματικές παραμέτρους
    - Το σώμα της εκτελείται (ενεργοποίηση)



# Αναδρομή

Μία διαδικασία ονομάζεται αναδρομική όταν καλεί η ίδια τον εαυτό της ή όταν καλεί κάποια άλλη διαδικασία, που μέσω σειράς κλήσεων ξανακαλεί την αρχική διαδικασία.



# Μέθοδοι Περάσματος Παραμέτρων

Τρόποι αντικατάστασης των τυπικών παραμέτρων μιας διαδικασίας από πραγματικές



# Κλήση με Τιμή

- Οι τυπικές παράμετροι της διαδικασίας που καλείται παίρνουν την τιμή των πραγματικών παραμέτρων
- Βήματα
  - Υπολογισμός της τιμής της έκφρασης  $E$  (έστω  $u$ )
  - Εκτέλεση του σώματος της διαδικασίας  $p$ . Όταν συναντάται η τυπική παράμετρος της  $p$  (έστω  $x$ ), αντικαθίσταται με την προϋπολογισμένη τιμή  $u$ .



# Παράδειγμα 6.1

Έστω η διαδικασία

```
function square(x:integer):integer;  
begin  
    square := x*x;  
end;
```

Αν καλέσουμε τη συνάρτηση square ως `square(2+3)` υπολογίζεται η τιμή  $2+3=5$  και αρχίζει η εκτέλεση του σώματος της διαδικασίας. Όταν συναντηθεί η έκφραση  $x*x$ , τα  $x$  αντικαθίστανται από το 5 και επιστρέφει η τιμή 25.



# Παράδειγμα 6.2

Έστω η διαδικασία

```
procedure change (x,y: integer);
```

```
var z:integer;
```

```
begin
```

```
    z:=x; x:=y; y:=z;
```

```
end;
```

Η κλήση `change (a,b)`, όπου `a,b` μεταβλητές τύπου `integer`, δεν έχει κανένα αποτέλεσμα πάνω στις μεταβλητές `a` και `b`.





# Κλήση με αναφορά

- Οι τυπικές παράμετροι μιας διαδικασίας γίνονται συνώνυμες με τη θέση στη μνήμη αντίστοιχων πραγματικών παραμέτρων

- Pascal

- Η λέξη-κλειδί `var` κάνει την τυπική παράμετρο `x` να φιλοξενεί στη διάρκεια των κλήσεων τη διεύθυνση της αντίστοιχης πραγματικής παραμέτρου



# Παράδειγμα 6.3 - 1

Έστω η διαδικασία

```
procedure swap(var x:integer; var y:integer);  
var z:integer;  
begin  
    z:=x; x:=y; y:=z;  
end;
```



# Παράδειγμα 6.3 - 2

- Κλήση της μορφής `swap(a,b)` με `a` και `b` μεταβλητές τύπου `integer`
  - αποτέλεσμα ανταλλαγή τιμών των μεταβλητών `a` και `b`
- Κλήση με αναφορά πραγματοποιείται
  - C με χρήση δεικτών
  - Java
    - βασική μέθοδος περάσματος παραμέτρων, όταν πρόκειται για πέραςμα αντικειμένων



# Κλήση με Όνομα

- Ιστορικό ενδιαφέρον
  - Βασική μέθοδος κλήσης στην Algol 60
- Σημερινή χρήση
  - Παρόμοια μέθοδος κλήσης στις συναρτησιακές γλώσσες προγραμματισμού



# Συνοπτική Περιγραφή Μεθόδου Κλήσης με Όνομα

- Αντικατάσταση πραγματικών παραμέτρων ως κείμενο στο σώμα της διαδικασίας στα σημεία εμφάνισης αντίστοιχων τυπικών παραμέτρων
  - Αλλαγή ονομάτων των μεταβλητών της διαδικασίας για να μην υπάρχουν μεταβλητές πραγματικών παραμέτρων με το ίδιο όνομα



# Συνοπτική Περιγραφή Μεθόδου Κλήσης με Όνομα

- Αντικατάσταση του σώματος της διαδικασίας στο σημείο αρχικής κλήσης
  - Αλλαγή ονομάτων των μεταβλητών της διαδικασίας για να μην υπάρχουν τοπικές μεταβλητές στο σημείο κλήσης με το ίδιο όνομα



# Εγγραφές ενεργοποίησης

Η δυνατότητα δημιουργίας αναδρομικά οριζόμενων διαδικασιών δημιουργεί πληθώρα προβλημάτων σχετικά με την υλοποίησή τους.



# Παράδειγμα 6.4

Αναδρομικός υπολογισμός του παραγοντικού  
ενός φυσικού αριθμού

```
function f(n:integer): integer;  
begin  
    if n=0 then f:=1 else f:=n*f(n-1);  
end;
```





# Ενεργοποίηση

- Μία αναδρομική διαδικασία μπορεί να έχει περισσότερες από μία ταυτόχρονες ενεργοποιήσεις
  - Δικό της χώρο στη μνήμη, για να εκτελεσθεί
  - Τοπικές μεταβλητές ανεξάρτητες από αυτές των υπόλοιπων ενεργοποιήσεων



# Δέντρο Ενεργοποιήσεων

- Η ροή ελέγχου ανάμεσα στις ενεργοποιήσεις διαδικασιών δημιουργεί ένα νοητό δέντρο
- Όταν η ενεργοποίηση  $p$  καλεί την  $q$ , τότε ο κόμβος του δέντρου για την  $p$  έχει τον κόμβο για την  $q$  ως παιδί του. Αν η  $p$  καλεί την  $q$  πριν την  $r$ , τότε ο κόμβος για την  $r$  εμφανίζεται δεξιά από τον κόμβο για την  $q$ .



# Εγγραφή ενεργοποίησης

- Τα δεδομένα των ενεργοποιήσεων μαζεύονται σε μία δομή, που ονομάζεται εγγραφή ενεργοποίησης.
- Οι εγγραφές τοποθετούνται η μία μετά την άλλη σε μία κατάλληλη μεγαλύτερη δομή, που ονομάζεται στοίβα.

