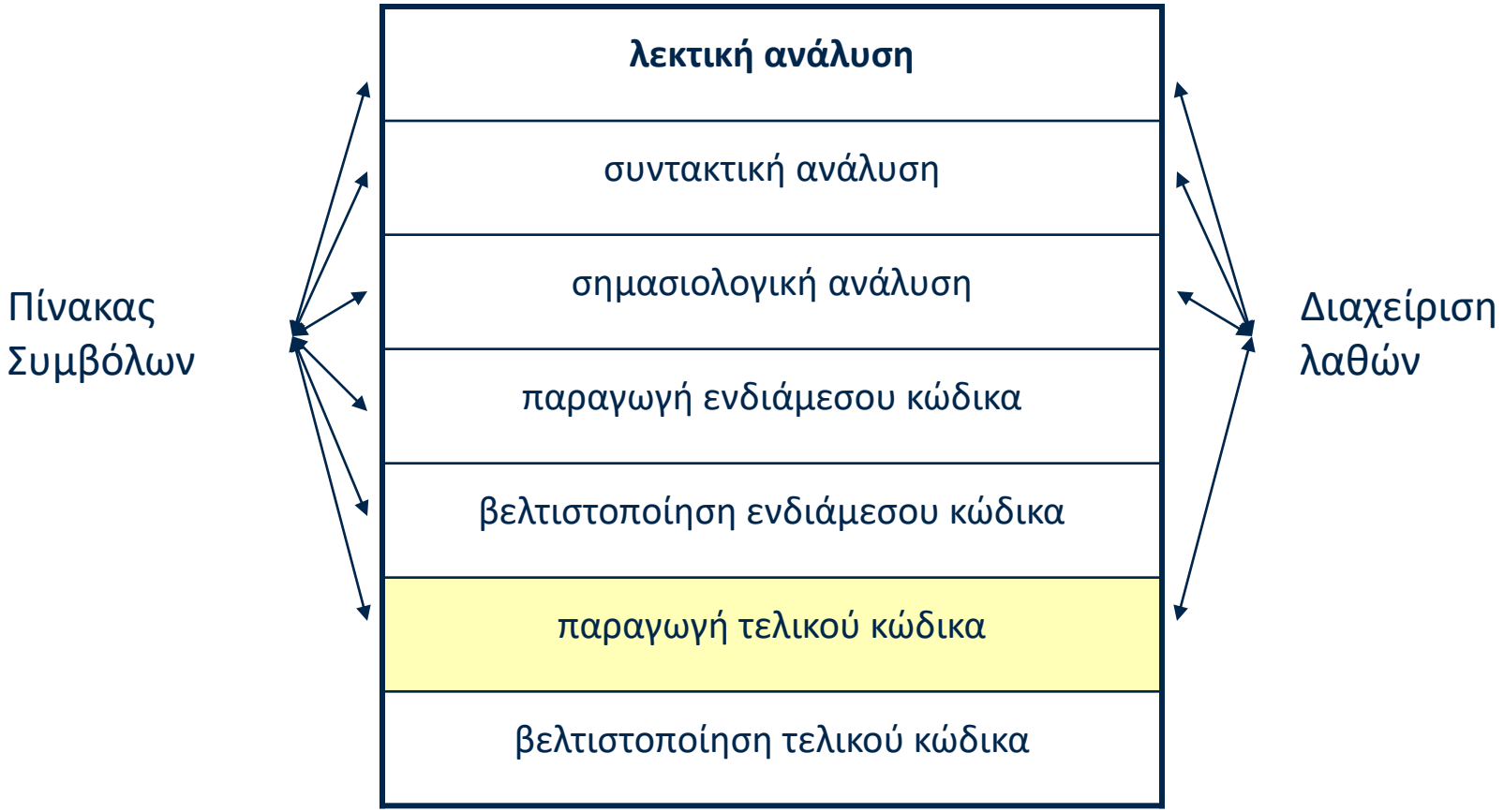
Παραγωγή Τελικού Κώδικα για την Αρχιτεκτονική MIPS

Διαλέξεις στο μάθημα: Μεταφραστές
Γιώργος Μανής

Οι Φάσεις της Μεταγλώττισης



Παραγωγή Τελικού Κώδικα

Παραγωγή Τελικού Κώδικα

- Από κάθε μία εντολή ενδιάμεσου κώδικα παράγουμε τις αντίστοιχες εντολές του τελικού κώδικα
- Κύριες ενέργειες στη φάση αυτή:
  - οι μεταβλητές απεικονίζονται στην μνήμη (στοίβα)
  - το πέρασμα παραμέτρων και η κλήση συναρτήσεων
- θα δημιουργήσουμε κώδικα για τον επεξεργαστή MIPS

Η Αρχιτεκτονική MIPS

- Καταχωρητές που θα μας φανούν χρήσιμοι:
  - καταχωρητές προσωρινών τιμών: $t0…$t7
  - καταχωρητές οι τιμές των οποίων διατηρούνται ανάμεσα σε κλήσεις συναρτήσεων: $s0…$s7
  - καταχωτητές ορισμάτων: $a0…$a3
  - καταχωρητές τιμών: $v0,$v1
  - stack pointer $sp
  - frame pointer $fp
  - return address $ra

Η Αρχιτεκτονική MIPS

- Εντολές που θα μας φανούν χρήσιμες για αριθμητικές πράξεις:

  - add $t0,$t1,$t2 t0=t1+t2
  - sub $t0,$t1,$t2 t0=t1-t2
  - mul $t0,$t1,$t2 t0=t1*t2
  - div $t0,$t1,$t2 t0=t1/t2

Η Αρχιτεκτονική MIPS

- Εντολές που θα μας φανούν χρήσιμες για μετακίνηση δεδομένων:

move $t0,$t1	t0=t1	μεταφορά ανάμεσα σε καταχωρητές
li $t0, value	t0=value	σταθερά σε καταχωρητή
lw $t1,mem	t1=[mem]	περιεχόμενο μνήμης σε καααχωρητή
sw $t1,mem	[mem]=t1	περιεχόμενο καταχωρητή σε μνήμη
lw $t1,($t0)	t1=[t0]	έμμεση αναφορά με καταχωρητή
sw $t1,-4($sp)	t1=[$sp-4]	έμμεση αναφορά με βάση τον $sp

Η Αρχιτεκτονική MIPS

- Εντολές που θα μας φανούν χρήσιμες για άλματα:

b label	branch to label
beq $t1,$t2,label	jump to label if $t1=$t2
blt $t1,$t2,label	jump to label if $t1<$t2
bgt $t1,$t2,label	jump to label if $t1>$t2
ble $t1,$t2,label	jump to label if $t1<=$t2
bge $t1,$t2,label	jump to label if $t1>=$t2
bne $t1,$t2,label	jump to label if $t1<>$t2

Η Αρχιτεκτονική MIPS

- Εντολές που θα μας φανούν χρήσιμες στην κλήση συναρτήσεων:

j label	jump to label
jal label	κλήση συνάρτησης
jr $ra	άλμα στη διεύθυνση που έχει ο καταχωρητής στο παράδειγμα είναι ο $ra που έχει την διεύθυνση επιστροφής συνάρτησης

Βοηθητικές Συναρτήσεις - gnvlcode

- μεταφέρει στον $t0 την διεύθυνση μιας **μη τοπικής** μεταβλητής
- από τον πίνακα συμβόλων βρίσκει πόσα επίπεδα επάνω βρίσκεται η μη τοπική μεταβλητή και μέσα από τον σύνδεσμο προσπέλασης την εντοπίζει

```
lw $t0,-4($sp)              στοίβα του γονέα
όσες φορές χρειαστεί:
        lw $t0,-4($t0)      στοίβα του προγόνου που έχει τη μεταβλητή
add $t0,$t0,-offset         διεύθυνση της μη τοπικής μεταβλητής
```

Βοηθητικές Συναρτήσεις – loadvr

- μεταφορά δεδομένων στον καταχωρητή r
- η μεταφορά μπορεί να γίνει από τη μνήμη (στοίβα)
- ή να εκχωρηθεί στο r μία σταθερά
- η σύνταξη της είναι loadvr(v,r)
- διακρίνουμε περιπτώσεις

Βοηθητικές Συναρτήσεις – loadvr

- αν v είναι σταθερά

  li $tr,v

- αν v είναι καθολική μεταβλητή – δηλαδή ανήκει στο κυρίως πρόγραμμα

  lw $tr,-offset($s0)

Βοηθητικές Συναρτήσεις – loadvr

- αν v είναι τοπική μεταβλητή, ή τυπική παράμετρος που περνάει με τιμή και βάθος φωλιάσματος ίσο με το τρέχον, ή προσωρινή μεταβλητή

```
lw $tr,-offset($sp)
```

- αν v είναι τυπική παράμετρος που περνάει με αναφορά και βάθος φωλιάσματος ίσο με το τρέχον

```
lw $t0,-offset($sp)
lw $tr,($t0)
```

Βοηθητικές Συναρτήσεις – loadvr

- αν v είναι τοπική μεταβλητή, ή τυπική παράμετρος που περνάει με τιμή και βάθος φωλιάσματος μικρότερο από το τρέχον

  gnlvcode()

  lw $tr,($t0)

- αν v είναι τυπική παράμετρος που περνάει με αναφορά και βάθος φωλιάσματος μικρότερο από το τρέχον

  gnlvcode()

  lw $t0,($t0)

  lw $tr,($t0)

Βοηθητικές Συναρτήσεις – storerv

- μεταφορά δεδομένων από τον καταχωρητή r στη μνήμη (μεταβλητή v)
- η σύνταξη της είναι storerv(r,v)
- διακρίνουμε περιπτώσεις

Βοηθητικές Συναρτήσεις – storerv

- αν v είναι καθολική μεταβλητή – δηλαδή ανήκει στο κυρίως πρόγραμμα

  sw $tr,-offset($s0)

Βοηθητικές Συναρτήσεις – storerv

- αν v είναι τοπική μεταβλητή, ή τυπική παράμετρος που περνάει με τιμή και βάθος φωλιάσματος ίσο με το τρέχον, ή προσωρινή μεταβλητή

```
sw $tr,-offset($sp)
```

- αν v είναι τυπική παράμετρος που περνάει με αναφορά και βάθος φωλιάσματος ίσο με το τρέχον

```
lw $t0,-offset($sp)
sw $tr,($t0)
```

Βοηθητικές Συναρτήσεις – storerv

- αν v είναι τοπική μεταβλητή, ή τυπική παράμετρος που περνάει με τιμή και βάθος φωλιάσματος μικρότερο από το τρέχον

  gnlvcode(v)

  sw $tr,($t0)

- αν v είναι τυπική παράμετρος που περνάει με αναφορά και βάθος φωλιάσματος μικρότερο από το τρέχον

  gnlvcode(v)

  lw $t0,($t0)

  sw $tr,($t0)

Εντολές Αλμάτων

- jump, “_”, “_”, label

  j label

- relop(?),x,y,z

  loadvr(x,1)

  loadvr(y,2)

  branch(?),$t1,$t2,z branch(?) : beq,bne,bgt,blt,bge,ble

Εκχώρηση

- :=, x, “_”, z

  loadvr(x,1)

  storerv(1,z)

Εντολές Αριθμητικών Πράξεων

- op x,y,z

  loadvr(x,1)

  loadvr(y,2)

  op $t1,$t1,$t2 op: add,sub,mul,div

  storerv(1,z)

Εντολές Εισόδου-Εξόδου

- out “_”, “_”, x

  li $v0,1

  li $a0, x

  syscall

- in “_”, “_”, x

  li $v0,5

  syscall

  το αποτέλεσμα γράφεται στον $v0

Επιστροφή Τιμής Συνάρτησης

- retv “_”, “_”, x

  loadvr(x,1)

  lw $t0,-8($sp)

  sw $t1,($t0)

  αποθηκεύεται ο x στη διεύθυνση που είναι αποθηκευμένη στην 3^{η} θέση του εγγραφήματος δραστηριοποίησης

- εναλλακτικά μπορούμε να γράψουμε το αποτέλεσμα στον $v0, και μετά πρέπει να φροντίσουμε να το πάρουμε από εκεί

  loadvr(x,1)

  move $v0,$t1

Παράμετροι Συνάρτησης

- πριν από την πρώτη παράμετρο, τοποθετούμε τον $fp να δείχνει στην στοίβα της συνάρτησης που θα δημιουργηθεί

```
add $fp,$sp,framelength
```

Παράμετροι Συνάρτησης

- par,x,CV, _

  loadvr(x,0)

  sw $t0, -(12+4i)($fp)

  όπου i ο αύξων αριθμός της παραμέτρου

Παράμετροι Συνάρτησης

- par,x,REF, _
  - αν η καλούσα συνάρτηση και η μεταβλητή x έχουν το ίδιο βάθος φωλιάσματος, η παράμετρος x είναι στην καλούσα συνάρτηση τοπική μεταβλητή ή παράμετρος που έχει περαστεί με τιμή

    add $t0,$sp,-offset

    sw $t0,-(12+4i)($fp)

Παράμετροι Συνάρτησης

- par,x,REF, _
  - αν η καλούσα συνάρτηση και η μεταβλητή x έχουν το ίδιο βάθος φωλιάσματος, η παράμετρος x είναι στην καλούσα συνάρτηση παράμετρος που έχει περαστεί με αναφορά

    lw $t0,-offset($sp)

    sw $t0,-(12+4i)($fp)

Παράμετροι Συνάρτησης

- par,x,REF, _
  - αν η καλούσα συνάρτηση και η μεταβλητή x έχουν διαφορετικό βάθος φωλιάσματος, η παράμετρος x είναι στην καλούσα συνάρτηση τοπική μεταβλητή ή παράμετρος που έχει περαστεί με τιμή

    gnlvcode(x)

    sw $t0,-(12+4i)($fp)

Παράμετροι Συνάρτησης

- par,x,REF, _
  - αν η καλούσα συνάρτηση και η μεταβλητή x έχουν διαφορετικό βάθος φωλιάσματος, η παράμετρος x είναι στην καλούσα συνάρτηση παράμετρος που έχει περαστεί με αναφορά

    gnlvcode(x)

    lw $t0,($t0)

    sw $t0,-(12+4i)($fp)

Παράμετροι Συνάρτησης

- par,x,RET, _

  γεμίζουμε το 3ο πεδίο του εγγραφήματος δραστηριοποίησης της κληθείσας συνάρτησης με τη διεύθυνση της προσωρινής μεταβλητής στην οποία θα επιστραφεί η τιμή

  ```
  add $t0,$sp,-offset

  sw $t0,-8($fp)
  ```

Κλήση Συνάρτησης

- call, _, _, f

  αρχικά γεμίζουμε το 2ο πεδίο του εγγραφήματος δραστηριοποίησης της κληθείσας συνάρτησης με την διεύθυνση του εγγραφήματος δραστηριοποίησης του γονέα της, ώστε η κληθείσα να γνωρίζει που να κοιτάξει αν χρειαστεί να προσπελάσει μία μεταβλητή την οποία έχει δικαίωμα να προσπελάσει, αλλά δεν της ανήκει

  - αν καλούσα και κληθείσα έχουν το ίδιο βάθος φωλιάσματος, τότε έχουν τον ίδιο γονέα

    lw $t0,-4($sp)

    sw $t0,-4($fp)

  - αν καλούσα και κληθείσα έχουν διαφορετικό βάθος φωλιάσματος, τότε η καλούσα είναι ο γονέας της κληθείσας

    sw $sp,-4($fp)

Κλήση Συνάρτησης

- στη συνέχεια μεταφέρουμε τον δείκτη στοίβας στην κληθείσα

  add $sp,$sp,framelength

- καλούμε τη συνάρτηση

  jal f

- και όταν επιστρέψουμε παίρνουμε πίσω τον δείκτη στοίβας στην καλούσα

  add $sp,$sp,-framelength

Κλήση Συνάρτησης

- μέσα στην κληθείσα
  - στην αρχή κάθε συνάρτησης αποθηκεύουμε στην πρώτη θέση του εγγραφήματος δραστηριοποίησης την διεύθυνση επιστροφής της την οποία έχει τοποθετήσει στον $ra η jal

    sw $ra,($sp)

  - στην τέλος κάθε συνάρτησης κάνουμε το αντίστροφο, παίρνουμε από την πρώτη θέση του εγγραφήματος δραστηριοποίησης την διεύθυνση επιστροφής της συνάρτησης και την βάζουμε πάλι στον $ra. Μέσω του $ra επιστρέφουμε στην καλούσα

    lw $ra,($sp)

    jr $ra

Αρχή Προγράμματος και Κυρίως Πρόγραμμα

- το κυρίως πρόγραμμα δεν είναι το πρώτο πράγμα που μεταφράζεται, οπότε στην αρχή του προγράμματος χρειάζεται ένα άλμα που να οδηγεί στην πρώτη ετικέτα του κυρίως προγράμματος

  ```
  j Lmain
  ```

- στη συνέχεια πρέπει να κατεβάσουμε τον $sp κατά framelength της main

  ```
  add $sp,$sp,framelength
  ```

- και να σημειώσουμε στον $s0 το εγγράφημα δραστηριοποίησης της main ώστε να έχουμε εύκολη πρόσβαση στις global μεταβλητές

  ```
  move $s0,$sp
  ```

ευχαριστώ !!!