



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ ΑΝΟΙΚΤΑ ΑΚΑΔΗΜΑΪΚΑ ΜΑΘΗΜΑΤΑ



Μεταφραστές

Παραγωγή τελικού κώδικα

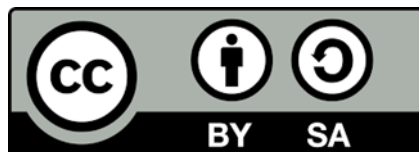
Διδάσκων: Επικ. Καθ. Γεώργιος Μανής



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- # Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- # Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Παραγωγή Τελικού Κώδικα

Γιώργος Μανής



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΛΛΗΝΙΚΗ ΥΠΟΥΡΓΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΣΠΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρκετικής
Επαγγελματικής Κατάρτισης

Τελικός Κώδικας



Η Γλώσσα Μηχανής

Καταχωρητές

$R[0], R[1], R[2], \dots, R[255]$

Ο καταχωρητής $R[0]$ χρησιμοποιείται σαν δείκτης στοίβας

Ο program counter συμβολίζεται με '\$ '

Πρόσβαση στη μνήμη

$M[\text{address}]$

$M[R[\dots]]$

$M[R[0]+\text{offset}]$

Η Γλώσσα Μηχανής

- # ini tr tr register
 - # outi so1 so1 register

 - # addi tr,so1,so2 tr=so1 + so2
 - # subi tr,so1,so2 tr=so1 - so2
 - # muli tr,so1,so2 tr=so1 * so2
 - # divi tr,so1,so2 tr=so1 / so2
-

Η Γλώσσα Μηχανής

movi tr,sol

tr := sol

tr	sol
μνήμη	register
register	μνήμη
register	αριθμός
register	R[0] (ο SP)
register	PC

Η Γλώσσα Μηχανής

- # jmp addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # cmpi so1,so2 so1,so2 registers
 - # jb addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # jbe addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # ja addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # jae addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # je addr addr label, M[..], R[..], δ/ση(αριθμός)
 - # jne addr addr label, M[..], R[..], δ/ση(αριθμός)
-

Η Γλώσσα Μηχανής

- # Ο καταχωρητής SR είναι ένας πίνακας 8 θέσεων και τον επηρεάζουν η *cmpr* και ο έλεγχος για overflow.
 - # Η *cmpr* επηρεάζει το SR[0] και το SR[1]. Αν οι δυο αριθμοί είναι ίσοι το SR[0] γίνεται 1 και αν ο πρώτος αριθμός είναι μεγαλύτερος από τον δεύτερο τότε το SR[1] γίνεται 1. Ο έλεγχος υπερχείλισης επηρεάζει το SR[7] και αν υπάρχει υπερχείλιση τότε γίνεται 1.
 - # Όταν η εντολή είναι *jb*, *jbe*, *ja*, *jae*, *je*, *jne*, *jo* ελέγχει τα SR[0] και SR[1] και αν ισχύουν οι κατάλληλες συνθήκες πηγαίνει τον PC στο σωστό σημείο.
-

Η Συνάρτηση glnvcode

- # Μεταφέρει στον R[255] τη διεύθυνση μίας μη τοπικής μεταβλήτης

movi R[255], M[4+R[0]] R[255]=M[4+R[0]]

όσες φορές χρειαστεί:

movi R[255], M[4+R[255]] R[255]=M[4+R[255]]

movi R[254], offset R[254]=offset

addi R[255], R[254], R[255] R[255]=R[254]+R[255]

Η Συνάρτηση loadnr(v,r)

- # αν v είναι αριθμός:

movi R[r], v R[r]= v

- # αν v είναι global μεταβλητή - δηλαδή του κυρίως προγράμματος

movi R[r], M[600+offset] R[r]= M[600+offset]

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

movi R[r], M[offset+R[0]] R[r]= M[offset+R[0]]

Η Συνάρτηση loadnr(v,r)

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

movi R[255], M[offset+R[0]] R[255]=M[offset+R[0]]

movi R[r], M[R[255]] R[r]=M[R[255]]

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

glnvcode() ώστε ο R[255] να έχει την σωστή δ/ση

movi R[r], M[R[255]] R[r]=M[R[255]]

Η Συνάρτηση loadnr(v,r)

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

glncode() ώστε ο $R[255]$ να έχει την σωστή δ/υση

movi $R[255], M[R[255]]$ $R[255]=M[R[255]]$

movi $R[r], M[R[255]]$ $R[r]=M[R[255]]$

Η Συνάρτηση `storev(r,v)`

- ✦ αν v είναι global μεταβλητή - δηλαδή του κυρίως προγράμματος

`movi M[600+offset], R[r] M[600+offset] = R[r]`

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

`movi M[offset+R[0]], R[r] M[offset+R[0]] = R[r]`

Η Συνάρτηση `storev(r,v)`

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

`movi R[255], M[offset+R[0]]` $R[255]=M[\text{offset}+R[0]]$

`movi M[R[255]], R[r]` $M[R[255]] = R[r]$

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

`glnvcode()` ώστε ο $R[255]$ να έχει την σωστή δ/ση

`movi M[R[255]], R[r]` $M[R[255]] = R[r]$

Η Συνάρτηση $storev(r,v)$

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

`glncode()` ώστε ο $R[255]$ να έχει την σωστή δ/νση

`movi R[255], M[R[255]]` $R[255]=M[R[255]]$

`movi M[R[255]],R[r]` $M[R[255]]=R[r]$

Εντολές Αλμάτων

`jump,_,_,w`

`jmp Lw`

`relop,x,y,w` `relop={=,<>, >, >=, <, <=}`

`loadvr(x,1)` `R[1]=x`

`loadvr(y,2)` `R[2]=y`

`cmpi R[1], R[2]`

`condjmp Lw` `comdjmp={je,jne,jb,jbe,ja,jae}`

Η `cmp` επηρεάζει τον καταχωρητή SR και οι

`je,jne,jb,jbe,ja,jae`

τον ελέγχουν και κάνουν ή όχι το άλμα υπό συνθήκη

Εκχώρηση

:=, x, _, z

loadvr(x,1) R[1]=x

storerv(1,z) z=R[1]

Εντολές Αριθμητικών Πράξεων

op, x, y, z $op = \{+, -, *, /\}$

$loadvr(x, 1)$ $R[1] = x$

$loadvr(y, 2)$ $R[2] = y$

$op\ R[3], R[1], R[2]$ $op = \{addi, subi, muli, divi\}$

$storerv(3, z)$ $z = R[3]$

μπορεί για το αποτέλεσμα της op να χρησιμοποιηθεί και ένας από τους $R[1]$ και $R[2]$

Εντολές Εισόδου-Εξόδου

out,x,_,_

loadvr(x,1)

R[1]=x

outi R[1]

in,x,_,_

ini R[1]

storerv(1,x)

x =R[1]

Επιστροφή Τιμής Συνάρτησης

:=,x,_,return_value

loadvr(x,1) R[1]=x

movi R[255],M[8+R[0]]

movi M[R[255]],R[1]

αποθηκεύεται ο x στην διεύθυνση που είναι αποθηκευμένη στην 3η θέση του εγγραφήματος δραστηριοποίησης.

Παράμετροι Συνάρτησης

par,x,CV,_ (παράμετρος με τιμή)

loadvr(x,255) R[255]=x

movi M[d+R[0]], R[255]

$d = \text{framelength} + 12 + 4 * i$

$i = \text{αύξοντας αριθμός παραμέτρου}$

Παράμετροι Συνάρτησης

- # `par,x,REF,_` (παράμετρος με αναφορά)
 - Αν η καλούσα και η μεταβλητή `x` έχουν το ίδιο βάθος φωλιάσματος:
 - Αν η παράμετρος `x` είναι στην καλούσα συνάρτηση τοπική μεταβλητή ή παράμετρος που έχει περαστεί με τιμή:

```
movi R[255],R[0]
```

```
movi R[254],offset
```

```
addi R[255], R[254],R[255]
```

```
movi M[d+R[0]], R[255]
```

Παράμετροι Συνάρτησης

- # `par,x,REF,_` (παράμετρος με αναφορά)
 - Αν η καλούσα και η μεταβλητή `x` έχουν το ίδιο βάθος φωλιάσματος:
 - Αν η παράμετρος `x` έχει περαστεί στην καλούσα συνάρτηση με αναφορά:

```
movi R[255],R[0]
```

```
movi R[254],offset
```

```
addi R[255], R[254],R[255]
```

```
movi R[1], M[R[255]]
```

```
movi M[d+R[0]], R[1]
```


Παράμετροι Συνάρτησης

- # `par,x,REF,___` (παράμετρος με αναφορά)
 - Αν η καλούσα και η μεταβλητή `x` έχουν διαφορετικό βάθος φωλιάσματος:
 - Αν η παράμετρος `x` είναι στην καλούσα συνάρτηση τοπική μεταβλητή ή παράμετρος που έχει περαστεί με τιμή:
`glnvcode()`
`movi M[d+R[0]], R[255]`
 - Αν η παράμετρος `x` έχει περαστεί στην καλούσα συνάρτηση με αναφορά:
`glnvcode()`
`movi R[1], M[R[255]]`
`movi M[d+R[0]], R[1]`
-

Παράμετροι Συνάρτησης

par,x,RET,_

γεμίζουμε το 3ο πεδίο της κληθείσας συνάρτησης με την δ/νση της προσωρινής μεταβλητής όπου θα επιστραφεί η τιμή

```
movi R[255], R[0]
```

```
movi R[254], offset της προσωρινής μεταβλητής
```

```
addi R[255],R[254], R[255]
```

```
movi M[frameLength+8+R[0]], R[255]
```

Κλήση Συνάρτησης

- # Αρχικά γεμίζουμε το 2ο πεδίο της στοίβας της καλούμενης συνάρτησης με την δ/νση της αρχής της πατρικής συνάρτησης της κληθείσας, ώστε η καλούμενη να ξέρει που να κοιτάξει για τις μεταβλητές στις οποίες έχει πρόσβαση.

- Αν η καλούσα και η κληθείσα έχουν το ίδιο βάθος φωλιάσματος (καλούσα και κληθείσα έχουν τον ίδιο πατέρα):

movi R[255], M[4+R[0]]

movi M[frameLength+4+R[0]], R[255]

- Αν καλείται συνάρτηση με μεγαλύτερο βάθος φωλιάσματος από την καλούσα (η καλούσα είναι ο πατέρας της κληθείσας):

movi M[frameLength+4+R[0]], R[0]

Κλήση Συνάρτησης

επιστροφή στην καλούσα:

movi R[255],framelength

subi R[0], R[0], R[255] (δείκτης στοίβας)

end_block,x,_,_

jmp M[R[0]]

Αρχή και Τερματισμός Προγράμματος

- # Αρχή προγράμματος
 - Άλμα στην πρώτη εκτελέσιμη εντολή

- # Τερματισμός προγράμματος
 - halt,_,_,_
 - halt

Τέλος Ενότητας



Χρηματοδότηση

- # Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- # Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Ιωαννίνων**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- # Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

Έκδοση 1.0 διαθέσιμη εδώ.

<http://ecourse.uoi.gr/course/view.php?id=1125>.

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Ιωαννίνων, Διδάσκων:
Επικ. Καθ. Γεώργιος Μανής. «Μεταφραστές.
Παραγωγή τελικού κώδικα». Έκδοση: 1.0. Ιωάννινα
2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://ecourse.uoi.gr/course/view.php?id=1125>.

Σημείωμα Αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή, Διεθνής Έκδοση 4.0 [1] ή μεταγενέστερη.



- [1] <https://creativecommons.org/licenses/by-sa/4.0/>
-