

# Μεταγλωττιστής από minibasic σε Jackal 3.0

Μάθημα: Προηγμένα Θέματα Θεωρητικής Πληροφορικής (CST319 / W10)

Διδάσκων: Νικόλαος Καββαδίας

[nkavnn@uop.gr](mailto:nkavnn@uop.gr)

20/03/2010

## Αντικείμενο της εργασίας

Αντικείμενο αυτής της εργασίας είναι η ανάπτυξη και σχεδιασμός μεταγλωττιστή από τη γλώσσα minibasic σε κώδικα για την αρχιτεκτονική επεξεργαστή Jackal 3.0. Ο μεταγλωττιστής θα πρέπει να μπορεί να χρησιμοποιηθεί σε εκτελέσιμη μορφή από τη γραμμή εντολών του κατάλληλου τερματικού σε Cygwin/Windows ή Linux. Ο μεταγλωττιστής θα πρέπει να σχεδιαστεί σύμφωνα με τις αρχές του δομημένου προγραμματισμού για την ευκολότερη συντήρηση και επέκταση του κώδικα.

Η γραμματική της minibasic δίνεται στο Σχήμα 1 σε μορφή BNF (Backus-Naur Form), ενώ ένα παράδειγμα εφαρμογής γραμμένο σε minibasic δίνεται στο Σχήμα 2.

```
program : stmt_list
        ;

stmt_list : /* nothing */
          | stmt stmt_list
          ;

stmt      : "print" expr
          | "let" T_var '=' expr
          | "for" expr "do" stmt
          | "begin" stmt_list "end"
          | "if" expr "then" stmt
          ;

expr      : T_const
          | T_var
          | '(' expr ')'
          | expr '+' expr
          | expr '-' expr
          | expr '*' expr
          ;
```

Σχήμα 1: Γραμματική για τη minibasic.

```
let b = 1
print 1
for 10 do
begin
  let c = a + b
  print c
  let a = b
  let b = c
end
```

Σχήμα 2: Υπολογισμός της ακολουθίας Fibonacci στη minibasic.

Η αρχιτεκτονική Jackal 3.0 αποτελεί μία πραγματική μηχανή (επεξεργαστή) η οποία διαθέτει πλήρες περιβάλλον μεταγλώττισης, συμβολομετάφρασης και προσομοίωσης, αναρτημένο στο διαδίκτυο [jack]. Διαθέτει τα εξής χαρακτηριστικά:

- 12 βασικές λειτουργίες (εντολές)
- ο χρήστης μπορεί να προσθέσει μέχρι τέσσερις επιπλέον εντολές
- 16 καταχωρητές
- κάθε αριθμητική εντολή δέχεται μέχρι τρία ορίσματα καταχωρητών.

Στη συνέχεια στον Πίνακα 1 δίνεται μία σύνοψη των εντολών του Jackal, ενώ στο Σχήμα 3 δίνονται οι διαμορφώσεις των εντολών για τον ίδιο επεξεργαστή.

**Πίνακας 1:** Ρεπερτόριο εντολών του Jackal 3.0.

Μνημονικό	Κωδικός Λειτουργίας (opcode)	Λειτουργία
ADD	0000	Arithmetic Integer Addition
SUB	0001	Arithmetic Integer Subtraction
AND	0010	Logical Bitwise And
OR	0011	Logical Bitwise Inclusive Or
NAND	0100	Logical Bitwise Nand
SLA	0101	Arithmetic Shift Left
SRA	0110	Arithmetic Shift Right
LD	0111	Load from Memory
ST	0111	Store to Memory
LIL	1000	Immediate Low Byte Load
LIH	1001	Immediate High Byte Load
CMP	1010	Logical Register Data Comparison
BRN	1011	Branch if Negative
BRZ	1011	Branch if Zero
BRP	1011	Branch if Positive
JMP	1011	Jump

INSTRUCTION	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD	0	0	0	0	DESTINATION				SOURCE ONE				SOURCE TWO			
SUB	0	0	0	1	DESTINATION				SOURCE ONE				SOURCE TWO			
AND	0	0	1	0	DESTINATION				SOURCE ONE				SOURCE TWO			
OR	0	0	1	1	DESTINATION				SOURCE ONE				SOURCE TWO			
NAND	0	1	0	0	DESTINATION				SOURCE ONE				SOURCE TWO			
SLA	0	1	0	1	DESTINATION				SOURCE				OFFSET			
SRA	0	1	1	0	DESTINATION				SOURCE				OFFSET			
LD	0	1	1	1	DESTINATION				SOURCE*				MODE			
ST	0	1	1	1	SOURCE				DESTINATION*				MODE			
LIL	1	0	0	0	DESTINATION				IMMEDIATE							
LIH	1	0	0	1	DESTINATION				IMMEDIATE							
CMP	1	0	1	0	UDO				SOURCE ONE				SOURCE TWO			
BRN/BRZ/BRP/JMP	1	0	1	1	MODE				OFFSET							
UDI	1	1	0	0	UDO											
UDI	1	1	0	1	UDO											
UDI	1	1	1	0	UDO											
UDI	1	1	1	1	UDO											

\*For the LD/ST instruction, the source/destination operand is a register holding a memory address.

**Σχήμα 3:** Κωδικοποίηση των εντολών της αρχιτεκτονικής Jackal 3.0.

Στην ιστοσελίδα του μαθήματος έχει αναρτηθεί αναλυτικός οδηγός για την αρχιτεκτονική Jackal 3.0.

Η ενδιάμεση αναπαράσταση του μεταγλωττιστή θα πρέπει να είναι αρχικά κάποιας μορφής αφηρημένο συντακτικό δένδρο (AST). Ως ενδεικτική υλοποίηση ενός AST και των βοηθητικών ρουτινών για τη διαχείρισή του αναφέρεται ο σημασιολογικός αναλυτής της minibasic στην ιστοσελίδα των Μεταγλωττιστών του NTUA [minibasic]. Στη συνέχεια θα πρέπει να παράγεται γραμμικοποιημένος κώδικας ο οποίος θα χρησιμοποιεί εντολές της αρχιτεκτονικής Jackal 3.0.

Στα πλαίσια της εργασίας ζητούνται τα εξής:

- να επεκταθεί η γραμματική της minibasic ώστε να υποστηρίζει:
  - ακέραια διαίρεση “/” και υπολογισμό ακέραιου υπολοίπου “%”
  - τους τελεστές σύγκρισης “==” και “/=”
- η αρχιτεκτονική Jackal 3.0 πρέπει να επεκταθεί από την εντολή PLOT η οποία έχει τη σύνταξη:

PLOT rA, rB, rC

όπου: οι τιμές των καταχωρητών rA και rB καθορίζουν ένα σημείο (pixel) στο επίπεδο της μορφής (x,y) με την αρίθμηση να αρχίζει από το (0,0) το οποίο βρίσκεται στο άνω αριστερό άκρο του επιπέδου. Το pixel αυτό θα “χρωματίζεται” από τον κωδικό χρώματος που υπάρχει στον καταχωρητή rC και έχει την τιμή 0 ως 255 (π.χ. 0 είναι το μαύρο – NOIR και 255 το λευκό – WHITE).
- θα υποστηρίζει τη μνήμη απεικόνισης γραφικών CANVAS διαστάσεων [512 x 512] στην οποία και αποθηκεύονται οι τιμές χρώματος για τα εικονοστοιχεία. Η διεύθυνση ενός εικονοστοιχείου δίνεται από τη σχέση  $rA * 512 + rB$ .

Για την επιλογή κώδικα να χρησιμοποιηθεί αλγοριθμική τεχνική η οποία να καλύπτει με τα δένδρα της ενδιάμεσης αναπαράστασης, πραγματοποιώντας κατάλληλη διαπέραση τύπου DFS (Depth-First Search).

Για τον καταμερισμό καταχωρητών μπορεί να υλοποιηθεί είτε κάποια απλή τεχνική είτε ο αλγόριθμος γραμμικής σάρωσης (linear-scan register allocation) των Poletto και Sarkar, ο οποίος αποτελεί καλό συμβιβασμό ανάμεσα στην ευκολία ανάπτυξης, την ταχύτητα εκτέλεσής του και τις επιδόσεις του τελικού προγράμματος συμβολομεταφραστή. Για παράδειγμα, μία απλή τεχνική θα χρησιμοποιεί πάντα εντολές LD για την φόρτωση των μεταβλητών σε καταχωρητές, και στη συνέχεια μετά την εκτέλεση των όποιων υπολογιστικών εντολών πάνω στους καταχωρητές, εντολές ST για την αποθήκευσή τους πίσω στη μνήμη δεδομένων.

## **Βιβλιογραφικές αναφορές**

[bison] Bison homepage. <http://www.gnu.org/software/bison/bison.html>

[flex] Flex (The Fast Lexical Analyzer) homepage. <http://flex.sourceforge.net>

[GCC] The GNU Compiler Collection homepage. <http://gcc.gnu.org>

[jack] Jackal 3.0 development tools. <http://sourceforge.net/projects/jackcc/>

[Ker90] Brian Kernighan and Dennis Ritchie, The C Programming Language, 2nd edition, Prentice Hall, 1990.

[Lev09] John Levine, Flex & Bison, O'Reilly Publishing, 2009.

[minibasic] minibasic homepage. <http://courses.softlab.ntua.gr/compilers/2009a/examples/minibasic/>

[Nie10] Tom Niemann, "A Compact Guide to Lex & Yacc."

<http://www.epaperpress.com/lexandyacc/>

[Pat04] D.A. Patterson and J.L. Hennessy, Computer Architecture: the Hardware-Software Interface Approach, 3rd edition, Morgan Kaufmann Publishers, San Fransisco, CA, 2004.

## **Παράδοση και βαθμολόγηση της εργασίας**

Στην εργασία του μαθήματος, ο φοιτητής καλείται να παραδώσει

- τον πηγαίο κώδικα του μεταγλωττιστή
- παραδείγματα δοκιμής της λειτουργίας του μεταγλωττιστή
- τεχνική αναφορά η οποία θα τεκμηριώνει τόσο τη διαδικασία της ανάπτυξης όσο και του σχεδιασμού του μεταγλωττιστή
- συνιστώμενες προδιαγραφές συστήματος του τελικού χρήστη για τη δημιουργία του μεταγλωττιστή με κτίσιμο (make) από τον πηγαίο κώδικα

Η εργασία παραδίδεται σε ηλεκτρονική μορφή (PDF της εργασίας + αρχεία κώδικα) στο email του διδάσκοντα. Οι φοιτητές μπορούν να παραδώσουν τις εργασίες τους το αργότερο μέχρι και την ημερομηνία διεξαγωγής των εξετάσεων περιόδου Ιουνίου-Ιουλίου 2010 για το μάθημα. Εργασία η οποία θα παραδοθεί μετά το πέρας αυτής της ημερομηνίας, θα βαθμολογηθεί ώστε να ληφθεί υπόψη για τις εξετάσεις της επόμενης περιόδου.

Μια εργασία βαθμολογείται με άριστα το δέκα (10), ο οποίος βαθμός και προστίθεται στο βαθμό της γραπτής εξέτασης πολλαπλασιασμένος με τον παράγοντα 0.3.

Η συγκεκριμένη προγραμματιστική εργασία είναι ατομική.