

Πανεπιστήμιο Μακεδονίας
Τμήμα Εφαρμοσμένης Πληροφορικής

Mic-1 MMV

Προσομοιωτής Αρχιτεκτονικής Η/Υ

2010

Περιεχόμενα

| | |
|--------------------------------------------------------------------|----|
| Εισαγωγή..... | 3 |
| 1 Ξεκινώντας..... | 3 |
| 1.1 Περιεχόμενα της διανομής του προγράμματος..... | 3 |
| 1.2 Εγκατάσταση..... | 4 |
| 1.3 Παράδειγμα 1 - Φόρτωση και εκτέλεση ενός προγράμματος JVM..... | 4 |
| 1.4 Παράδειγμα 2 - Συμβολομετάφραση ενός προγράμματος JAS..... | 5 |
| 1.5 Παράδειγμα 3 - Νέα συμβολομετάφραση ενός προγράμματος JAS..... | 6 |
| 1.6 Παράδειγμα 4 – Δοκιμάζοντας διαφορετικές ταχύτητες..... | 6 |
| 2 Λειτουργίες του προγράμματος..... | 7 |
| 2.1 Επισκόπηση..... | 7 |
| 2.1.1 Προτιμήσεις εκκίνησης..... | 7 |
| 2.1.2 Γρήγορη επισκόπηση..... | 8 |
| 2.1.3 Εκτέλεση του προσομοιωτή..... | 10 |
| 2.1.4 Καθορισμός παραμέτρων..... | 11 |
| 2.2 Έλεγχοι και Πεδία..... | 12 |
| 2.2.1 Command Console – Κονσόλα εντολών..... | 12 |
| 2.2.2 Μενού επιλογών..... | 13 |
| 2.2.3 Πεδίο Εντολών..... | 13 |
| 2.2.4 Πεδία εισόδου / εξόδου (Input / Output Consoles)..... | 14 |
| 2.2.5 Περιοχές μνήμης..... | 14 |
| 2.2.6 Παράθυρο Μικρο-Κώδικα..... | 15 |
| 2.2.7 Περιοχή Αρχιτεκτονικής..... | 15 |
| 2.2.8 Παράθυρα Assembler | 16 |

Εισαγωγή

Το έγγραφο αυτό περιέχει οδηγίες για την εγκατάσταση και τη χρήση του προσομοιωτή **Mic-1 MMV**. Ο Mic-1 MMV χρησιμοποιείται ως εκπαιδευτικό λογισμικό σε συνδυασμό με το τέταρτο κεφάλαιο (“Επίπεδο Μικρο-αρχιτεκτονικής”) του βιβλίου “Η Αρχιτεκτονική των Υπολογιστών – Μια δομημένη προσέγγιση” του Andrew S. Tanenbaum. Το λογισμικό έχει υλοποιηθεί εξ ολοκλήρου σε Java (JDK v1.4), και θα πρέπει να τρέχει χωρίς τροποποιήσεις χρησιμοποιώντας οποιοδήποτε Java Virtual Machine (JVM), v1.4 ή νεότερη έκδοση.

1 Ξεκινώντας

Η ενότητα αυτή περιέχει όλα όσα χρειάζεστε για να αποσυμπιέσετε και να εγκαταστήσετε το πρόγραμμα. Περιέχει επίσης αρκετά παραδείγματα για να καταλάβετε πώς το Mic-1 MMV λειτουργεί.

1.1 Περιεχόμενα της διανομής του προγράμματος

Η διανομή του προγράμματος έρχεται σε ένα συμπιεσμένο αρχείο **Mic1MMV.zip**, και αν το αποσυμπιέσετε θα δείτε του ακόλουθους φακέλους και αρχεία.

bin

- **Mic1MMV_hr.jar**
- **Mic1MMV_lr.jar**
Εκτελέσιμα Java jar αρχεία που εκτελούν τον προσομοιωτή (hi-res και low-res εκδόσεις αντίστοιχα – υψηλής και χαμηλής ανάλυσης).
- **runMic1.bat**
Ένα batch αρχείο για την εκτέλεση του προσομοιωτή (όχι απαραίτητο πλέον).

lib

- **ijvm.conf**
Ένα αρχείο ρυθμίσεων για τον ijvm assembler. Περιέχει την περιγραφή της γλώσσας assembly δηλαδή τον κωδικό εντολής (opcode – operation code), το μνημονικό (mnemonic), και τους τύπους τελεστών (operand types) για κάθε εντολή.
- **mic1.properties**
Ένα αρχείο ιδιοτήτων.
- **GNU.TXT**
Αντίγραφο της άδεια χρήσης GNU - General Public License.

Examples

- **MAL**
Ένας φάκελος που περιέχει το mic1ijvm.mal, το αρχείο πηγαίου κώδικα της micro assembly (MAL file) για τον προκαθορισμένο IJVM διερμηνέα, και το mic1ijvm.mic1, δηλαδή το αντίστοιχο αρχείο μικροκώδικα.
- **JAS-IJVM**
Ένας φάκελος που περιέχει αρκετά παραδείγματα προγραμμάτων Integer Java Virtual Machine (IJVM), αρχεία πηγαίου κώδικα (jas αρχεία) και αρχεία κώδικα γλώσσας μηχανής (ijvm αρχεία).

doc

- **UserGuide.jar**
Μια standalone έκδοση αυτού του οδηγού. Για εκτέλεση, διπλό κλικ ή μέσω της γραμμής εντολών :
java -jar UserGuide.jar
- **UserGuide_hs.jar**

Ένα σύνολο βοήθειας Java για αυτό τον βοηθό.

src

- **mic1mmv.zip**

Ένα αρχείο zip που περιέχει τον πηγαίο κώδικα για το [Mic-1 MMV](#).

1.2 Εγκατάσταση

Το λογισμικό [Mic-1 MMV](#) απαιτεί το Java Runtime Environment (JRE) 1.4 ή νεότερη έκδοση. Επισκεφτείτε το σύνδεσμο <http://java.sun.com/j2se/> να βρείτε το JDK ή το JRE για την πλατφόρμα σας.

- **Windows NT/XP αποσυμπίεση**

Για την αποσυμπίεση τρέξτε το αυτο-εξαγόμενο zip αρχείο [Mic1MMV.exe](#). Ο φάκελος [Mic1MMV](#) θα δημιουργηθεί περιέχοντας, όλα τα αρχεία του προγράμματος, όλα τα αρχεία πηγαίου κώδικα και τη τεκμηρίωση.

- **Unix / Linux αποσυμπίεση**

Για την αποσυμπίεση του [Mic1MMV.tar.gz](#) σε λειτουργικό σύστημα τύπου Unix, θα χρειαστεί το πρόγραμμα gunzip GNU ή άλλο συμβατό λογισμικό αποσυμπίεσης, και το πρόγραμμα tar. Εάν χρησιμοποιείτε gunzip, σε ένα κέλυφος εντολών, πληκτρολογήστε:

```
$ gunzip mic1.tar.gz  
$ tar xvf Mic1MMV.tar
```

Μπορείτε να το κάνετε αυτό με μία εντολή εάν χρησιμοποιείτε το GNU tar:

```
$ tar xzf Mic1MMV.tar.gz
```

Αυτό θα δημιουργήσει ένα φάκελο με το όνομα [Mic1MMV](#), όπως ακριβώς και στα Windows.

Υπάρχουν δύο jar αρχεία στο υποφάκελο bin του φακέλου [Mic1MMV](#). Το αρχείο [Mic1MMV_lr.jar](#) είναι σχεδιασμένο να λειτουργεί με χαμηλή ανάλυση οθόνης, ενώ το [Mic1MMV_hr.jar](#) είναι για υψηλή ανάλυση οθόνης. Εάν χρησιμοποιείτε ανάλυση 1280 x 960 και υψηλότερη, θα πρέπει να χρησιμοποιήσετε το [Mic1MMV_hr.jar](#), διαφορετικά το [Mic1MMV_lr.jar](#).

Σημείωση: Όταν εκτελέσουμε το [Mic1MMV](#), αυτόματα φορτώνεται ένα πρόγραμμα μικροκώδικα. Κατά την πρώτη εκτέλεση αυτού του προγράμματος είναι το [mic1ijvm.mic1](#), ο διερμηνέας μικροκώδικα IJVM, που περιγράφεται στο κεφάλαιο 4 του βιβλίου “Η αρχιτεκτονική των Υπολογιστών – Μια δομημένη προσέγγιση” του Andrew S. Tanenbaum.. Μπορείτε να φορτώσετε ένα διαφορετικό πρόγραμμα μικροκώδικα στο [Mic1MMV](#) καθώς εκτελείται, ή ακόμη και να καθορίσετε ένα διαφορετικό μικροπρόγραμμα να φορτώνεται κατά την εκκίνηση. Ανατρέξτε στον Οδηγό χρήσης για λεπτομέρειες.

Δοκιμάστε να εκτελέσετε και τα δύο αρχεία, το [Mic1MMV_hr.jar](#) και το [Mic1MMV_lr.jar](#). Μπορείτε έπειτα να μετονομάσετε αυτό που θέλετε να χρησιμοποιείτε σε [Mic1MMV.jar](#). Για την εκτέλεση του μπορείτε να κάνετε δεξί κλικ πάνω στο αρχείο και να επιλέξετε “Ανοιγμα με Sun Java Runtime”, είτε να δώσετε την εντολή από την κονσόλα:

```
java-jar mic1mmv_lr.jar ή java-jar mic1mmv_hr.jar
```

1.3 Παράδειγμα 1 - Φόρτωση και εκτέλεση ενός προγράμματος IJVM

Το παράδειγμα αυτό δείχνει πώς μπορείτε να επεξεργαστείτε και να συμβολομεταφράσετε ένα πρόγραμμα JAS, και να φορτώσετε το παραγόμενο πρόγραμμα IJVM.

1. Τρέξτε το πρόγραμμα όπως περιγράψαμε πιο πάνω στην ενότητα 1.2 Εγκατάσταση.
2. Επιλέξτε από το μενού **File | Assemble / Load JAS file**
3. Στον επιλογέα αρχείου, μεταβείτε στο φάκελο "examples/JAS-IJVM Examples".

4. Επιλέξτε το πρόγραμμα `ijvmtest.jas`. Το IJVM πρόγραμμα θα πρέπει να εμφανιστεί στην περιοχή Method Area.
5. Θα πρέπει να σας εμφανιστεί παράθυρο με το μήνυμα *Assembling ijevmtest.jas ..., assembly complete* και να κάνετε κλικ στο κουμπί **Load**.
6. Επιλέξτε **Prog Speed** στην περιοχή Command Console (ένα radio button ονομαζόμενο "Prog").
7. Κάνετε κλικ στο κουμπί στην περιοχή Command Console για να ξεκινήσει η ερμηνεία (interpretation) του IJVM προγράμματος από το προκαθορισμένο μικροπρόγραμμα. Μετά από μια σύντομη περίοδο, ενώ ο εξομοιωτής λειτουργεί, θα πρέπει να δείτε τα παρακάτω στην περιοχή "Output Console": OK

1.4 Παράδειγμα 2 - Συμβολομετάφραση ενός προγράμματος JAS

Παρουσιάζουμε ένα παράδειγμα το οποίο μας δείχνει πώς μπορείτε να επεξεργαστείτε και να συμβολομεταφράσετε ένα πρόγραμμα JAS, και να φορτώσετε το παραηόμενο πρόγραμμα IJVM.

1. Βρείτε το αρχείο [examples/JAS-IJVM Examples/ijvmtest.jas](#)
2. Ανοίξτε το με κάποιον editor.
3. Πηγαίνετε στην γραμμή 451.
4. Αντικαταστήσετε τις γραμμές:

```
OK:      BIPUSH 79
         OUT
         BIPUSH 75
         OUT
         HALT
```

με

```
OK:      BIPUSH 65
         OUT
         BIPUSH 79
         OUT
         BIPUSH 75
         OUT
         HALT
```

και σώστε τις αλλαγές.

5. Στον προσομοιωτή επιλέξτε από το μενού **File | Assemble/Load IJVM program** Περιηγηθείτε στο φάκελο με τα παραδείγματα όπως και προηγούμενος "examples/JAS-IJVM", και επιλέξτε το [ijvmtest.jas](#).
6. Θα πρέπει να σας εμφανιστεί παράθυρο με το μήνυμα *Assembling ijevmtest.jas ..., assembly complete* και να κάνετε κλικ στο κουμπί **Load**.
7. Επιλέξτε **Prog Speed** και κάντε κλικ στο κουμπί , όπως και πριν. Τώρα θα πρέπει να δείτε στο Output Console:

AOK

1.5 Παράδειγμα 3 - Νέα συμβολομετάφραση ενός προγράμματος JAS

Μόλις ένα αρχείο JAS έχει φορτωθεί και συμβολομεταφραστεί, μπορεί να συμβολομεταφραστεί εκ νέου και να φορτωθεί χρησιμοποιώντας την επιλογή του μενού **Assemble/Load | Current JAS Assemble/Load** χωρίς να απαιτείται περαιτέρω η επιλογή αρχείου. Το παράδειγμα αυτό δείχνει επίσης τι συμβαίνει όταν υπάρχει και κάποιο σφάλμα σε ένα πρόγραμμα JAS.

1. Μετά την ολοκλήρωση του παραδείγματος 1, τις αλλαγές που κάναμε στο αρχείο **ijvmtest.jas**, αφαιρέστε την άνω και κάτω τελεία μετά το OK και αποθηκεύστε το αρχείο.

```
OK
BIPUSH 65
OUT
BIPUSH 79
OUT
BIPUSH 75
OUT
HALT
```

2. Στον προσομοιωτή επιλέξτε από το μενού **Assemble/Load | Current JAS Assemble/Load**. Το παράθυρο *Assembling ijevmtest.jas* θα πρέπει να εμφανίζει το ακόλουθο μήνυμα λάθους:


```
IJVM Assembler...
1433: Invalid instruction: ok
1424: Invalid goto label: ok
```

3. Κλείστε το παράθυρο και ξανά επεξεργαστείτε το αρχείο, τοποθετώντας πίσω την άνω και κάτω τελεία και αφαιρώντας τις γραμμές

```
BIPUSH 65
OUT
```

4. Τώρα το πρόγραμμα μοιάζει όπως ήταν αρχικά. Δοκιμάστε το Βήμα 2 και πάλι, αυτή τη φορά χωρίς λάθη. Φορτώστε το πρόγραμμα, κάνετε κλικ στο Reset κουμπί και τρέξτε το ξανά. Το αποτέλεσμα στην περιοχή Output Console είναι “OK”.


1.6 Παράδειγμα 4 – Δοκιμάζοντας διαφορετικές ταχύτητες






Σε αυτό το παράδειγμα θα δείτε πως ο εξομοιωτής λειτουργεί με διαφορετικές ταχύτητες. Λέγοντας ταχύτητα εννοούμε το σύνολο υπολογισμού το οποίο λαμβάνει χώρα αμέσως μόλις πατηθεί το κουμπί  Θα δείτε επίσης πώς να χρησιμοποιήσετε το Delay Mode για να κάνετε επίδειξη των ενεργειών του προσομοιωτή.

1. Τρέξτε τον προσομοιωτή και από το μενού επιλέξτε **File | Assemble/Load IJVM program Επιλέξτε** από τα παραδείγματα το πρόγραμμα **examples/JAS-IJVM/add.jas**.

2. Δοκιμάστε το πρώτα σε **Prog** ταχύτητα. Αυτό το πρόγραμμα προσθέτει 2 δεκαεξαδικούς αριθμούς. Τρέξτε το πρόγραμμα και πληκτρολογήστε το ακόλουθο στο Input Console:

```
1234
5678
Όταν τελειώσετε, το Output Console θα πρέπει να περιέχει
1234
+5678
=====
000068AC
```

Μπορείτε να βάλετε άλλο ζευγάρι αριθμών για πρόσθεση στο Input Console. (Βεβαιωθείτε να χρησιμοποιήσετε κεφαλαία A ... F για τους δεκαεξαδικούς αριθμούς.) Κλικ στο  όταν τελειώσετε.

3. Τώρα πατήστε Reset, και αλλάξτε την ταχύτητα σε IJVM Speed.
4. Κάντε κλικ στο κουμπί  αρκετές φορές, και προσέξτε την τονισμένη περιοχή στην Method area. Σε κάθε κλικ που κάνουμε εκτελείτε και μια απλή εντολή IJVM. Μπορείτε επίσης να κάνετε κλικ στο κουμπί  για ένα βήμα πίσω.
5. Τώρα επιλέξτε **Delay On** (άλλο radio button) και κάντε κλικ στο κουμπί Reset και έπειτα στο κουμπί Run, όπως στο προηγούμενο βήμα. Παρατηρήστε ότι η ακολουθία των εντολών μικροκώδικα για την ερμηνεία της τρέχουσας εντολής IJVM εμφανίζεται, και η διαδρομή των δεδομένων απεικονίζεται στην Architecture View (το αριστερό παράθυρο που δείχνει τους καταχωρητές και τους διαύλους).
6. Επιλέξτε **Delay Off** και ταχύτητα ρολογιού (**Clock Speed**), έπειτα πατήστε το  και τρέξτε μερικά βήματα. Κάθε βήμα δείχνει την εκτέλεση μιας μικροεντολής. Το κουμπί  μας πηγαίνει ένα βήμα πίσω. Τώρα από το μενού επιλέξτε **Microstore | View Microstore**. Το Microstore παράθυρο θα εμφανιστεί, και μπορείτε να δείτε την εκτέλεση του μικροπρογράμματος ενώ κάνετε κλικ στο Run. Δοκιμάστε το με **Delay On** για να δείτε τους υποκύκλους όπως και πριν.
7. Τέλος, επιλέξτε **Delay Off** και **SubClock** ταχύτητα, στη συνέχεια το κουμπί reset και το κουμπί εκτέλεσης πολλές φορές. Κάθε βήμα τώρα αντιστοιχεί σε ένα τέταρτο του κύκλου. Το κουμπί  θα λειτουργεί όπως θα περιμένατε, αλλά το **Delay On** δεν έχει καμία επίδραση σε αυτή τη ταχύτητα.

2 Λειτουργίες του προγράμματος

Το πρόγραμμα *Mic-1 MMV* συνδυάζει τρεις κύριες λειτουργίες σε μια εφαρμογή:

1. Περιλαμβάνει ένα συμβολομεταφραστή για μετάφραση γλώσσας assembly JAS σε γλώσσα μηχανής Integer Java Virtual Machine (IJVM).
2. Περιλαμβάνει ένα μικρο-συμβολομεταφραστή για τη μετάφραση της γλώσσας μικροπρογραμματισμού (Register Transfer Language) MAL, σε μικροκώδικα Mic1.
3. Παρέχει έναν προσομοιωτή για την εκτέλεση IJVM προγραμμάτων στην αρχιτεκτονική Mic1.

2.1 Επισκόπηση

Η ενότητα αυτή ξεκινά με προτιμήσεις εκκίνησης και έπειτα γίνεται μια γρήγορη επισκόπηση του προγράμματος με εικόνες για τις διάφορες περιοχές του. Τέλος γίνεται μία αναφορά στους διάφορους μηχανισμούς του προγράμματος.

2.1.1 Προτιμήσεις εκκίνησης

Όταν το *Mic-1 MMV* εκτελεστεί, αρχικά αναζητά το αρχείο **mic1.properties** για να φορτώσει τις ιδιότητες και τις ρυθμίσεις. Σε περίπτωση που δεν βρεθεί φορτώνεται ένα εσωτερικό αρχείο αρχείο ιδιοτήτων. Το εσωτερικό αρχείο ιδιοτήτων, καθώς επίσης και τα υπόλοιπα εσωτερικά προεπιλεγμένα αρχεία του συστήματος, βρίσκεται στο κατάλογο *mic/resources* των jar αρχείων.

Στη συνέχεια, το *Mic-1 MMV* αναζητά ένα προεπιλεγμένο πρόγραμμα μικροκώδικα (mic1) να φορτώσει. Αυτό το προεπιλεγμένο πρόγραμμα καθορίζεται στις Προτιμήσεις (Preferences). Εάν δεν έχει καθοριστεί προεπιλεγμένο πρόγραμμα μικροκώδικα, ή αν εάν δεν μπορεί να φορτωθεί (π.χ. εσφαλμένη μορφή του αρχείου), τότε φορτώνεται ένα εσωτερικό πρόγραμμα. Το πρόγραμμα αυτό αντιστοιχεί σε εκείνο που περιγράφεται στο τέταρτο κεφάλαιο του βιβλίου του Tanenbaum.

Έπειτα το *Mic-1 MMV* αναζητά ένα αρχείο ρύθμισης παραμέτρων IJVM, το οποίο καθορίζει τους κωδικούς πράξης (opcode), τα μνημονικά και του τύπους παραμέτρων που χρησιμοποιούνται από τον προσομοιωτή, όταν μεταγλωττίζει JAS σε IJVM κώδικα μηχανής. Το όνομα του αρχείου ρυθμίσεων καθορίζεται στις προτιμήσεις (Preferences), και ονομάζεται *ijvm.conf*. Το *Mic-1 MMV*

αρχικά ψάχνει για το αρχείο στον κατάλογο *mic/ontko/resources* των jar αρχείων. Αν δεν βρεθεί, τότε φορτώνεται η εσωτερική έκδοση, που αντιστοιχεί στη γλώσσα όπως ορίζεται στο βιβλίο του Tanenbaum, “Η αρχιτεκτονική των Υπολογιστών – Μια δομημένη προσέγγιση”.

2.1.2 Γρήγορη επισκόπηση

Αυτή η ενότητα περιέχει μια εικονική περιήγηση του προσομοιωτή, που δείχνει τα κύρια μέρη και τα χαρακτηριστικά του.

Μέρη του προσομοιωτή

The screenshot shows the Microstore simulator interface with several key components labeled:

- Architecture View:** Shows the internal state of the processor, including registers (MAR, MDR, PC, MBR, SP, LV, CPP, TOS, OPC, H) and buses (C Bus, B Bus, A Bus). It also displays the ALU and Shifter.
- Method Area:** Displays the current instruction and its microcode sequence.
- Constant Pool:** Shows the memory locations for constants used in the instruction.
- Stack Area:** Shows the current stack contents.
- Command Console:** A text input area for entering commands.
- Instruction View:** Shows the current instruction being executed, including the Microcode (MPC), Microcode Register (MIR), and the ALU/Shift Register (AR).
- Microstore View:** Shows the current state of the microstore, including the current instruction and its microcode sequence.
- JAS Assembler View:** A window for assembling JAS code, showing the assembly process and completion.
- MAL Assembler View:** A window for assembling MAL code, showing the assembly process and completion.

Χαρακτηριστικά

Preference and Breakpoint capabilities.

Method area shows IJVM program. Current instruction is highlighted in red.

“Live” diagram (based on Tanen. Figure 4.1) animates data path when running at low speed.

Speed levels can be adjusted to illustrate micro and macro level processing. Single-stepping runs both forward and backward.

MIR shows microinstruction breakdown and current IJVM instruction.

Separate Input and Output Consoles.

The simulator interface includes a central diagram of the processor with registers (MAR, MDR, PC, MBR, SP, LV, CPP, TOS) and buses (C Bus, B Bus, A Bus). A control panel at the bottom features buttons for 'Reset', 'Step Back', 'Step Forward', 'Clock', 'IJVM', and 'Prog'. Below the control panel, the Microprogram Register (MIR) is displayed with a table of microinstructions:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 7 |
| A | J | J | J | S | F | E | E | I | I | C | W | R | F | B |
| D | M | A | A | L | R | N | N | N | N | | R | D | E | |
| R | P | M | M | L | A | A | B | V | C | | R | D | E | |
| | C | N | Z | 8 | 1 | | | A | | | | | | |

Below the MIR table, the instruction breakdown is shown: (9) (2) (9) (4).

Separate Micro store window displays and tracks current microprogram

Integrated JAS and MAL assemblers

```

Microstore
0000: goto 0x2
0001: PC=PC+1;goto 0xd1
0002: PC=PC+1;fetch;goto (MDR)
0003: H=IDS;goto 0x4
0004: TOS=MDR-H+MDR;wr;goto 0x2
0005: H=IDS;goto 0x6
0006: TOS=MDR-MDR-H;wr;goto 0x2
0007: H=IDS;goto 0x8
0008: TOS=MDR-H AND MDR;wr;goto 0x2
0009: H=IDS;goto 0xa
000a: TOS=MDR-H OR MDR;wr;goto 0x2
000b: MDR=TOS;wr;goto 0x2
000c: goto 0xd
000d: TOS=MDR;goto 0x2
000e: MAR=SP;goto 0xf
000f: H=MDR;wr;goto 0xd1
    
```

```

Assembling ...
Assembling src.jas:
IJVM Assembler...
assembly complete
    
```

```

Assembling ...
Assembling mic.ijvm.mat
Microcode Assembler...
assembly complete
    
```



2.1.3 Εκτέλεση του προσομοιωτή

Ο *Mic-1 MMV* σας επιτρέπει να δείτε την διερμηνεία του IJVM προγράμματος και την εκτέλεση μέσα από ένα μικροπρόγραμμα. Σε αυτό το παράδειγμα, χρησιμοποιούμε έναν εσωτερικό συμβολομεταφραστή JAS για να παράγουμε ένα πρόγραμμα IJVM από τον πηγαίο κώδικα και να εκτελέσουμε το πρόγραμμα στον προσομοιωτή.

Συμβολομετάφραση και φόρτωση του echo.jas

- Εκκινήστε τον προσομοιωτή
- Επιλέξτε **File | Assemble / Load JAS file ...**. Και φορτώστε το αρχείο **echo.jas** από τα παραδείγματα.
- Φορτώστε το στον πρόγραμμα.

Εκτέλεση του echo.ijvm

Ο προσομοιωτής μπορεί να εκτελεστεί σε διαφορετικές ταχύτητες, προκειμένου να παρατηρήσουμε την εκτέλεση του **echo.ijvm** σε διαφορετικά επίπεδα διακριτότητας. Η έννοια «βήμα» εξαρτάται από την επιλεγμένη ταχύτητα, όπως περιγράφεται παρακάτω. Ένα βήμα προς τα εμπρός συμβαίνει για κάθε κλικ του κουμπιού  Ένα βήμα αντίστροφο (αναίρεση της τελευταίας ενέργειας), εμφανίζεται για κάθε κλικ του κουμπιού .

Subclock speed

Κάθε βήμα αντιστοιχεί στο 1/4 του κύκλου μικροεντολής. Αυτή η ταχύτητα είναι κατάλληλη για την παρατήρηση της κίνησης των δεδομένων στην περιοχή Architecture View.


Clock speed

Κάθε βήμα αντιστοιχεί σε έναν κύκλο μικροεντολής. Αν η καθυστέρηση είναι επιλεγμένη (Delay), η κίνηση των δεδομένων μέσω της Architecture View θα παρουσιαστεί.


IJVM speed

Κάθε βήμα αντιστοιχεί στην εκτέλεση μιας εντολής IJVM. Αν η καθυστέρηση είναι επιλεγμένη (Delay), η κίνηση των δεδομένων μέσω της Architecture View θα παρουσιαστεί.

Prog speed

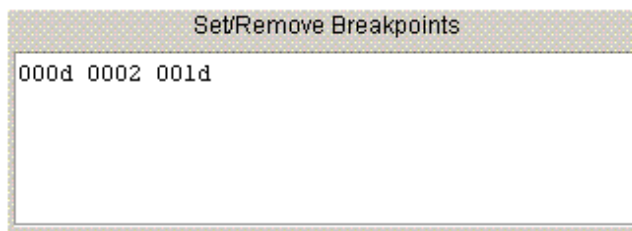
Κάνοντας κλικ στο κουμπί βήματος, το πρόγραμμα IJVM που θα εκτελεστεί έως την ολοκλήρωση του. Το κουμπί Stop () μπορεί να χρησιμοποιηθεί για αναστολή της εκτέλεσης. Αν η καθυστέρηση είναι επιλεγμένη (Delay), η εκτέλεση των επιμέρους μικροεντολών θα εμφανιστεί.

Όλες οι εισοδοί και οι έξοδοι αντίστοιχα, εμφανίζονται στις Input και Output κονσόλες.

Σημείωση: το κουμπί Reset () μπορεί να χρησιμοποιηθεί για επανεκκίνηση του προσομοιωτή, ανά πάσα στιγμή. Γίνετε επαναφορά των καταχωρητών και η μνήμη επιστρέφει στην αρχική της τιμή. Είναι καλή πρακτικής να κάνετε Reset μετά από αλλαγές ταχυτήτων.

Καθορισμός σημείων παύσεων (Breakpoints)

- Τα Breakpoints στην Method Area ορίζονται στις παραμέτρους (Preference panel). Επιλέξτε στο μενού **Preferences | Edit Preferences ...** για να ανοίξετε το παράθυρο με τις παραμέτρους.
- Εισάγετε στο κατάλληλο παράθυρο διευθύνσεις breakpoint (σε δεκαεξαδικό) χωρισμένες με το κενό, όπως και την παρακάτω εικόνα.



- Κλείστε το παράθυρο παραμέτρων. Οι καταχωρήσεις στην Method Area που αντιστοιχούν σε σημεία παύσης πρέπει να είναι με μπλε χρώμα.
- Τρέξτε το σε **Prog** speed. Ο προσομοιωτής θα σταματήσει κάθε φορά που βρίσκει ένα breakpoint.
- Μπορείτε να ξανά τοποθετήσετε breakpoint οποιοδήποτε στιγμή.

2.1.4 Καθορισμός παραμέτρων

Το μενού προτιμήσεων (Preferences) έχει παραμέτρους για την επεξεργασία, τη φόρτωση και την αποθήκευση. Οι προτιμήσεις συνήθως αποθηκεύονται σε ένα αρχείο που ονομάζεται **mic1.properties**. Μετά την εκκίνηση, το αρχείο αυτό φορτώνεται από τον τρέχοντα κατάλογο, αν υπάρχει, αλλιώς ένα εσωτερικό αρχείο ιδιοτήτων φορτώνεται.

Παρακάτω περιγράφονται οι διάφορες προτιμήσεις που μπορούμε να καθορίσουμε στο παράθυρο προτιμήσεων (Preferences).

Demo Delay

Μέγεθος σε χιλιοστά του δευτερολέπτου για την καθυστέρηση εκτέλεσης, όταν η καθυστέρηση (Delay) είναι ενεργοποιημένη στο control panel.

Method Area Base

Βασική διεύθυνση της μνήμης, στην περιοχή Method area.

Constant Pool Base

Βασική διεύθυνση της μνήμης, στην περιοχή Constant Pool.

Constant Pool Size

Αριθμός bytes του Constant Pool. Αυτός είναι γενικά πολύ μικρότερος από το χώρο εικονικών διευθύνσεων.

Stack Area Base

Βασική διεύθυνση της μνήμης, στην περιοχή Stack area.

Stack Size

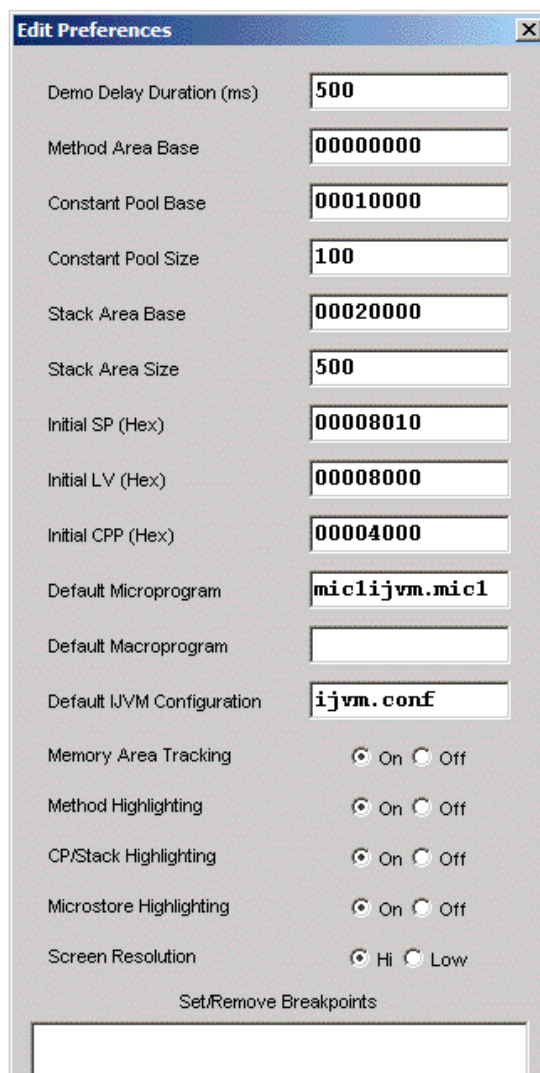
Αριθμός bytes του Stack. Αυτός είναι γενικά πολύ μικρότερος από το χώρο εικονικών διευθύνσεων.

Initial SP, LV, CPP

Αρχικές τιμές των καταχωρητών SP, LV, CPP. Οι τιμές είναι στο δεκαεξαδικό σύστημα.

Screen Resolution

Η επιλογή **hi** είναι για αναλύσεις οθονών 1280 x 960 και υψηλότερες. Το **low** είναι για οθόνες με μικρότερη ανάλυση.



Default Microcode Prog

Προκαθορισμένο πρόγραμμα Μικροκώδικα που φορτώνεται κατά την εκκίνηση.

Default Macrocode Prog

Προκαθορισμένο πρόγραμμα IJVM που φορτώνεται κατά την εκκίνηση.

Memory Area Tracking

Όταν είναι ενεργοποιημένη προκαλεί αυτόματη ολίσθηση μνήμης. (εκτός της Prog)

Method Highlighting

Όταν είναι ενεργοποιημένη Υπογραμμίζονται οι τρέχουσες εντολές IJVM. (εκτός της Prog)

CP/Stack Highlighting

Όταν είναι ενεργοποιημένη Υπογραμμίζονται οι συνεχείς δραστηριότητες στο Constant Pool (εκτός της Prog).

Microstore Highlighting

Όταν είναι ενεργοποιημένη, υπογραμμίζονται οι συνεχείς δραστηριότητες στο παράθυρο Microstore (μόνο για **Subclock** ή **Clock** ταχύτητες).

Set/Remove Breakpoints

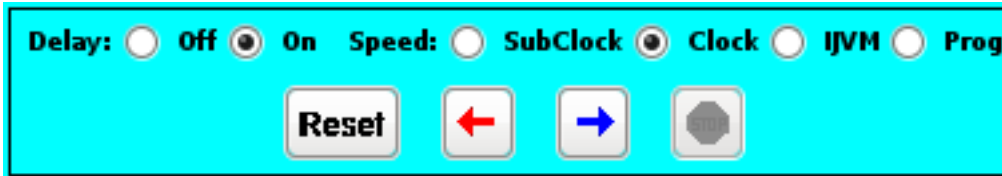
Ορίζουμε τα σημεία παύσης που χρησιμοποιούνται κατά τη διάρκεια εκτέλεσης.

2.2 Έλεγχοι και Πεδία

2.2.1 Command Console – Κονσόλα εντολών

Αυτή η ενότητα περιγράφει τις εντολές λειτουργιών που βρίσκονται στην περιοχή Command Console.



Command Console



Καθυστέρηση (Delay)

Όταν ενεργοποιηθεί, τότε οι οπτικές διαδρομές δεδομένων εμφανίζονται στην περιοχή architecture view.

Ταχύτητα (Speed)

Προσδιορίζει την έννοια του βήματος ( ). Αυτές οι επιλογές ταχύτητας επιτρέπουν την εξέταση των λεπτομερειών λειτουργίας του συστήματος σε διαφορετικά επίπεδα διακριτότητας.

SubClock

Ένας υποκύκλος από τη λειτουργία της μηχανής εκτελείται με κάθε βήμα. Οι διαδρομές δεδομένων απεικονίζονται στην περιοχή architecture view.

Clock

Ένας πλήρης κύκλος λειτουργίας της μηχανής εκτελείται με κάθε βήμα. Με καθυστέρηση (**Delay On**), οι υποκύκλοι απεικονίζονται στην περιοχή architecture view.





IJVM

Μία ολοκληρωμένη εντολή IJVM εκτελείται με κάθε βήμα. Με καθυστέρηση (**Delay On**), οι υποκύκλοι απεικονίζονται στην περιοχή architecture view.

Prog

Το πρόγραμμα εκτελείτε αδιάκοπα μέχρι να διακοπεί. Με καθυστέρηση (**Delay On**), οι κύκλοι απεικονίζονται στην περιοχή architecture view.

Κουμπιά

| | |
|-------------------------------------------------------------------------------------|---------------------------------------------------|
|  | Θέτει στις αρχικές τιμές τα μέρη του προσομοιωτή. |
|  | Εκτελεί ένα βήμα <i>πίσω</i> . |
|  | Εκτελεί ένα βήμα <i>μπροστά</i> . |
|  | Σταματάει τον προσομοιωτή. |

2.2.2 Μενού επιλογών

File

Load IJVM program

Φορτώνει ένα IJVM πρόγραμμα (IJVM αρχείο) στην μνήμη του προσομοιωτή. Ένα προκαθορισμένο IJVM πρόγραμμα μπορεί να οριστεί στα Preferences.

Load Micro program

Φορτώνει ένα πρόγραμμα μικροκώδικα (MIC1 αρχείο) στο παράθυρο microstore. Αυτό το πρόγραμμα θα πρέπει να είναι σε μορφή mic1. Κατά την εκκίνηση, το *Mic-1 MMV* ψάχνει να βρει για το προκαθορισμένο πρόγραμμα μικροκώδικα στα Preferences.

Assemble / Load JAS file

Συμβολομετάφραση και φόρτωση ενός JAS αρχείου που επιλέγεται από τον χρήστη.

Assemble / Load MAL file

Συμβολομετάφραση και φόρτωση ενός MAL αρχείου που επιλέγεται από τον χρήστη.

Preference

Edit Preferences

Επιτρέπει την προβολή και την επεξεργασία των προτιμήσεων (preferences).

Load Preferences

Φορτώνει τις ιδιότητες από ένα αρχείο.

Save Preferences

Αποθηκεύει τις προτιμήσεις (preferences) σε ένα αρχείο προτιμήσεων. Εάν το mic1.properties αρχείο βρίσκεται στο κατάλογο εκκίνησης, θα φορτωθεί όταν το πρόγραμμα ξεκινήσει.

Microcode Store

View Microstore

Άνοιγμα του Microcode Store για προβολή.

Assemble/Load

Current JAS Assemble / Load

Εκτελεί τον IJVM *Assembler* σε αρχείο πηγής JAS, του τρέχοντος IJVM προγράμματος.

Current MAL Assemble / Load

Εκτελεί τον MIC1 *Assembler* σε αρχείο πηγής MAL, του τρέχοντος μικροπρογράμματος.

Help

Mic-1 MMV Help

Ανοίγει το εγχειρίδιο βοήθειας. Επίσης ανοίγει με το κουμπί του πληκτρολογίου F1.

Mic-1 MMV Help On

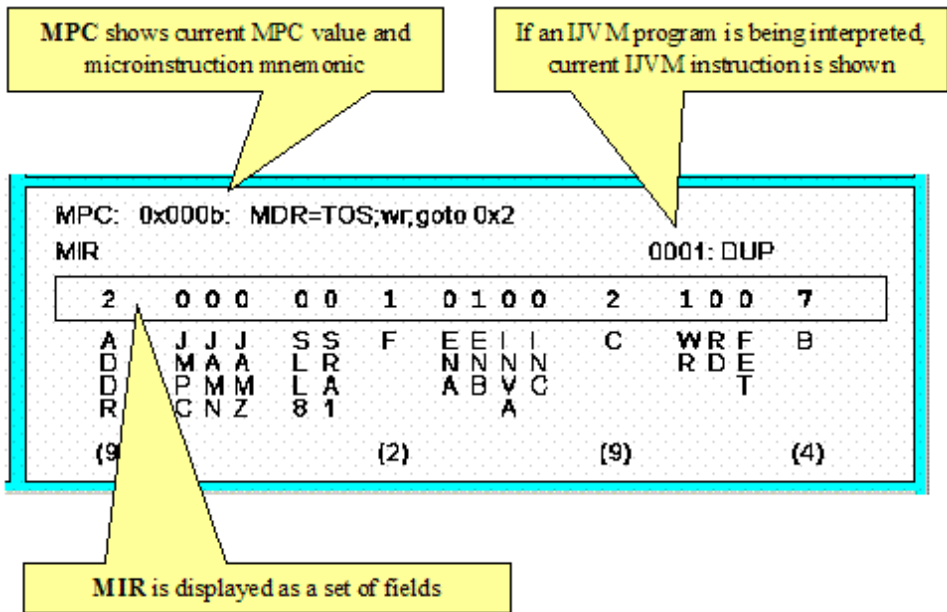
Οι χρήστες μπορούν να χρησιμοποιήσουν αυτή την επιλογή για να αποκτήσουν βοήθεια για οποιοδήποτε γραφικό (GUI) αντικείμενο. Όταν επιλέγετε ο δέκτης του ποντικιού αλλάζει μορφή.

About *Mic-1 MMV*

Πληροφορίες σχετικά με την άδεια και την έκδοση του προγράμματος.

2.2.3 Πεδίο Εντολών

Η περιοχή αυτή παρέχει μια προβολή της μονάδας μικροεντολών του *Mic-1 MMV*. Το MPC δείχνει την τρέχουσα διεύθυνση της μικροεντολής και τη συμβολική αναπαράσταση. Το MIR δείχνει την τρέχουσα μικροεντολή που αντιστοιχίζεστε με τα υποπεδία. Εάν μία εντολή IJVM διερμηνεύεται, εμφανίζεται στο Instruction view.



2.2.4 Πεδία εισόδου / εξόδου (Input / Output Consoles)

Τα πεδία εισόδου / εξόδου παρέχουν ξεχωριστούς χώρους για την εισαγωγή και την πληροφόρησή μας, για τα δεδομένα του προγράμματος.

| Input Console | Output Console |
|---------------|----------------|
| 1234 | 1234 |
| 2.1.1.1 | -2.1.1.1 |
| | ----- |
| | 0000.1.177 |

2.2.5 Περιοχές μνήμης

Η παρακάτω εικόνα δείχνει στο Mic-1 MMV τις περιοχές μνήμης σε τρία διαφορετικά τμήματα. Η διεύθυνση βάσης και το μέγεθος της κάθε περιοχής καθορίζονται από τις προτιμήσεις του προγράμματος. Ο χρήστης μπορεί επίσης να ορίσει προτιμήσεις, για την παρακολούθηση και προβολή της μνήμης με την βοήθεια χρωματικών επισημάνσεων. Σημειώστε ότι η Method Area εμφανίζει επιμέρους bytes, ενώ η Constant Pool και Stack Area εμφανίζονται ως 4-byte λέξεις.

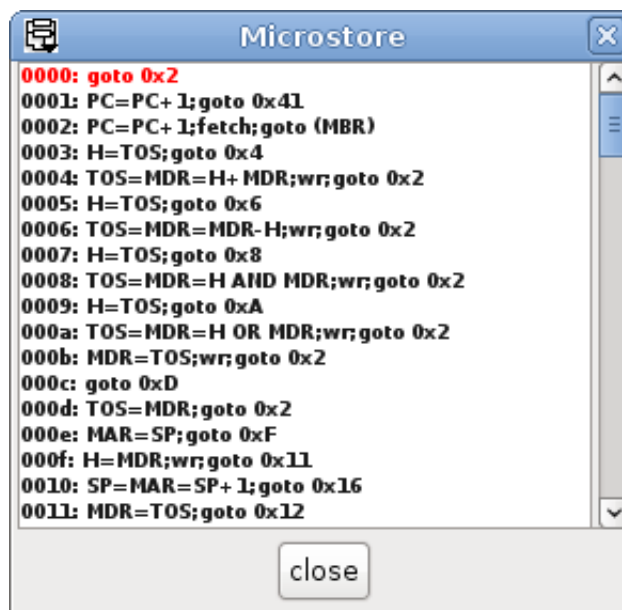
| Method Area | | |
|-------------------------|----------------------|--------------------|
| 0048: 00 | 0050: 00 | 0058: 36 |
| 0049: 36 ISTORE 01 | 0051: 41 | 0059: 59 DUP |
| 004a: 01 | 0052: 59 DUP | 005a: 10 BIPUSH 3a |
| 004b: fc IN | 0053: 10 BIPUSH 30 | 005b: 3a |
| 004c: 59 DUP | 0054: 30 | 005c: 64 ISUB |
| 004d: 10 BIPUSH 0a | 0055: 64 ISUB | 005d: 9b IFLT 001a |
| 004e: 0a | 0056: 9b IFLT 0036 | 005e: 00 |
| 004f: 9f IF_ICMPEQ 0041 | 0057: 00 | 005f: 1a |

| Constant Pool | | |
|--------------------|--------------------|--------------------|
| 00010000: 00000040 | 00010003: 00000094 | 00010010: 00000000 |
| 00010004: 00000043 | 0001000c: 00000000 | 00010014: 00000000 |

| Stack Area | | |
|--------------------|--------------------|--------------------|
| 00020030: 00000000 | 00020040: 00000000 | 00020050: 00003000 |
| 00020034: 00000000 | 00020044: 00003013 | 00020054: 00000000 |
| 00020038: 00000000 | 00020048: 00000000 | 00020058: 00000000 |
| 0002003c: 00000000 | 0002004c: 00000010 | 0002005c: 00000030 |

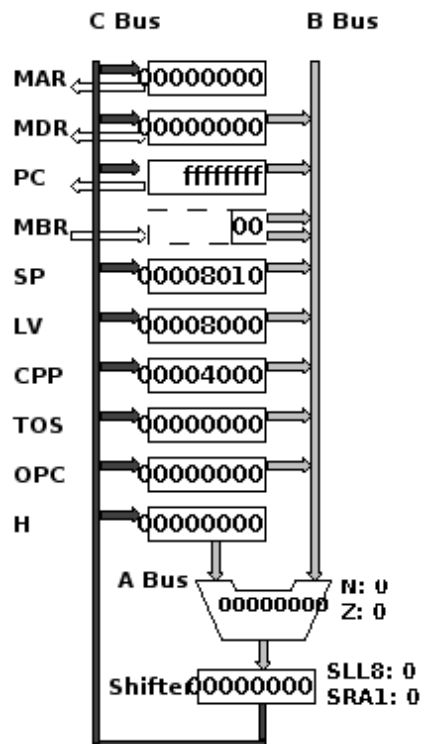
2.2.6 Παράθυρο Μικρο-Κώδικα

Το Microstore παράθυρο δείχνει τον έλεγχο του μικροπρογράμματος του *Mic-1 MMV*. Μπορείτε να το ανοίξετε επιλέγοντας **Microcode Store | View Microstore** από το μενού.



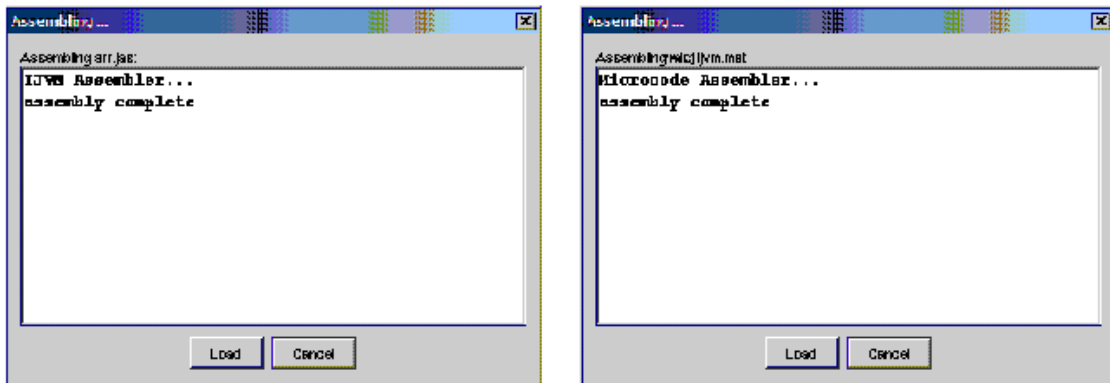
2.2.7 Περιοχή Αρχιτεκτονικής

Η περιοχή Architecture δείχνει όλους τους καταχωρητές και όλες τις διαδρομές δεδομένων του *Mic-1 MMV*. Όταν επιλέξουμε ταχύτητες **SubClock** και **Clock**, οι διαδρομές δεδομένων σχηματίζονται στην περιοχή με διάφορες χρωματικές αποχρώσεις.



2.2.8 Παράθυρα Assembler

Τα παράθυρα Assembler JAS και MAL εμφανίζουν την assembly ενός IJVM και ενός μικροπρογράμματος αντίστοιχα. Εμφανίζονται κάθε φορά που ο χρήστης φορτώνει ένα νέο αρχείο JAS ή MAL, και κάνει την επιλογή από το μενού **File | Assemble/Load JAS file ...** ή **File | Assemble/Load MAL file ...**. Εάν δεν παραχθεί κάποιο λάθος κατά την διαδικασία, το αντίστοιχο αντικείμενο μπορεί να φορτωθεί.



3 Τροποποίηση MAL και JAS

Για να ορίσουμε μια νέα εντολή στη γλώσσα JAS εκτελούμε τα παρακάτω βήματα.

- (1) Δημιουργούμε ένα αντίγραφο του αρχείου `ijvm.conf`. Με ένα απλό διορθωτή προσθέτουμε στο τέλος του αρχείου το `opcode`, το `mnemonic` και μια εξήγηση για την χρήση της εντολής.
- (2) Δημιουργούμε ένα αντίγραφο του αρχείου `mic1ijvm.mal`, έστω με όνομα `mic1ijvm_1.mal`. Με ένα απλό διορθωτή προσθέτουμε ένα `label` στο τέλος των `labels` των εντολών (βρίσκονται στην

αρχή του αρχείου). Στη συνέχεια, σε κάποιο σημείο του αρχείου εισάγουμε σε γλώσσα MAL τη περιγραφή της νέας εντολής JAS (τελειώνουμε με enter).

(3) Ανοίγουμε το αρχείο jar που χρησιμοποιούμε με ένα πρόγραμμα διαχείρισης συμπιεσμένων αρχείων. Δεν το αποσυμπιέζουμε, απλά το ανοίγουμε για επεξεργασία.

(4) Τοποθετούμε το νέο ijvm.conf στο κατάλογο mic/ontko/resources (το ήδη υπάρχον αρχείο το μετονομάζουμε προσωρινά, έστω σε ijvm.conf.orig.)

(5) Τοποθετούμε το νέο ijvm.conf στο κατάλογο mic/resources (το ήδη υπάρχον αρχείο το μετονομάζουμε προσωρινά, έστω σε ijvm.conf.orig.). Εναλλακτικά τοποθετούμε το ίδιο αρχείο ijvm.conf στο home directory μας.

(6) Εκτελούμε το αρχείο jar που μόλις τροποποιήσαμε. Συμβολομεταφράζουμε το αρχείο mic1ijvm_1.mal και φορτώνουμε το νέο μικρο-κώδικα mic1ijvm_1.mic1. Αυτό απαιτείται να γίνει μια φορά, κατόπιν απλά μπορούμε να φορτώνουμε το mic1 αρχείο.

(7) Συμβολομεταφράζουμε και φορτώνουμε το πρόγραμμα JAS που περιέχει τη νέα εντολή. Προχωρούμε στην εκτέλεσή του.