Γλώσσες Προγραμματισμού ΙΙ

Αν δεν αναφέρεται διαφορετικά, οι ασκήσεις πρέπει να παραδίδονται στους διδάσκοντες σε ηλεκτρονική μορφή μέσω του συνεργατικού συστήματος ηλεκτρονικής μάθησης moodle.softlab.ntua.gr. Η προθεσμία παράδοσης θα τηρείται αυστηρά. Έχετε δικαίωμα να καθυστερήσετε το πολύ μία άσκηση.

Άσκηση 5 Εικονικές μηχανές

Προθεσμία παράδοσης: 6/12/2015

Υλοποιήστε την εικονική μηχανή που αποτελούσε μέρος του προγραμματιστικού διαγωνισμού του ICFP 2006 (www.boundvariable.org). Σας συνιστούμε να χρησιμοποιήσετε τη C ως γλώσσα υλοποίησης του διερμηνέα και να εκμεταλλευτείτε τις επεκτάσεις του GNU C Compiler για την αποδοτική υλοποίηση VM interpreters που αναφέρονται στις διαφάνειες της διάλεξης της 5/11/2015. Υποβάλετε τη λύση σας στο σύστημα αυτόματης υποβολής και ελέγχου προγραμμάτων grader.softlab.ntua.gr.

Προδιαγραφές της εικονικής μηχανής. Αντιγράφονται στα αγγλικά, όπως ακριβώς δόθηκαν για το διαγωνισμό του ICFP 2006.

```
Order for Construction          Standard Sand of Pennsylvania Co.

Client: Cult of the Bound Variable
Object: UM-32 ”Universal Machine”
-----------------------------------------------------------------
                                                    21 July 19106

Physical Specifications.
------------------------

The machine shall consist of the following components:

  * An infinite supply of sandstone platters, with room on each
    for thirty-two small marks, which we call ”bits.”

                         least meaningful bit
                                            |
                                            v
            .--------------------------------.
            |VUTSRQPONMLKJIHGFEDCBA9876543210|
            ‘--------------------------------’
             ^
             |
             most meaningful bit

            Figure 0. Platters

    Each bit may be the 0 bit or the 1 bit. Using the system of
    ”unsigned 32-bit numbers” (see patent #4,294,967,295) the
    markings on these platters may also denote numbers.
```

* Eight distinct general-purpose registers, capable of holding one platter each.

* A collection of arrays of platters, each referenced by a distinct 32-bit identifier. One distinguished array is referenced by 0 and stores the ”program.” This array will be referred to as the ’0’ array.

* A 1x1 character resolution console capable of displaying glyphs from the ”ASCII character set” (see patent #127) and performing input and output of ”unsigned 8-bit characters” (see patent #255).

Behavior.

The machine shall be initialized with a ’0’ array whose contents shall be read from a ”program” scroll. All registers shall be initialized with platters of value ’0’. The execution finger shall point to the first platter of the ’0’ array, which has offset zero.

When reading programs from legacy ”unsigned 8-bit character” scrolls, a series of four bytes A,B,C,D should be interpreted with ’A’ as the most magnificent byte, and ’D’ as the most shoddy, with ’B’ and ’C’ considered lovely and mediocre respectively.

Once initialized, the machine begins its Spin Cycle. In each cycle of the Universal Machine, an Operator shall be retrieved from the platter that is indicated by the execution finger. The sections below describe the operators that may obtain. Before this operator is discharged, the execution finger shall be advanced to the next platter, if any.

Operators.

The Universal Machine may produce 14 Operators. The number of the operator is described by the most meaningful four bits of the instruction platter.

```
            .--------------------------------.
            |VUTSRQPONMLKJIHGFEDCBA9876543210|
            `--------------------------------'
             ^^^^
             |
             operator number
```

Figure 1. Operator Description

Standard Operators.

Each Standard Operator performs an errand using three registers, called A, B, and C. Each register is described by a three bit segment of the instruction platter. The register C is described by the three least meaningful bits, the register B by the three next more meaningful than those, and the register A by the three next more meaningful than those.

```
                                    A     C
                                    |     |
                                    vvv   vvv
            .--------------------------------.
            |VUTSRQPONMLKJIHGFEDCBA9876543210|
            `--------------------------------'
             ^^^^                       ^^^
             |                          |
             operator number            B
```

Figure 2. Standard Operators

A description of each basic Operator follows.

Operator #0. Conditional Move.

The register A receives the value in register B, unless the register C contains 0.

#1. Array Index.

The register A receives the value stored at offset in register C in the array identified by B.

#2. Array Amendment.

The array identified by A is amended at the offset in register B to store the value in register C.

#3. Addition.

The register A receives the value in register B plus the value in register C, modulo 2^32.

#4. Multiplication.

The register A receives the value in register B times the value in register C, modulo 2^32.

#5. Division.

The register A receives the value in register B divided by the value in register C, if any, where each quantity is treated treated as an unsigned 32 bit number.

#6. Not-And.

Each bit in the register A receives the 1 bit if either register B or register C has a 0 bit in that position. Otherwise the bit in register A receives the 0 bit.

Other Operators.

The following instructions ignore some or all of the A, B and C registers.

#7. Halt.

The universal machine stops computation.

#8. Allocation.

A new array is created with a capacity of platters commensurate to the value in the register C. This new array is initialized entirely with platters holding the value 0. A bit pattern not consisting of exclusively the 0 bit, and that identifies no other active allocated array, is placed in the B register.

#9. Abandonment.

The array identified by the register C is abandoned. Future allocations may then reuse that identifier.

#10. Output.

The value in the register C is displayed on the console immediately. Only values between and including 0 and 255 are allowed.

#11. Input.

The universal machine waits for input on the console. When input arrives, the register C is loaded with the input, which must be between and including 0 and 255. If the end of input has been signaled, then the register C is endowed with a uniform value pattern where every place is pregnant with the 1 bit.

#12. Load Program.

The array identified by the B register is duplicated and the duplicate shall replace the ’0’ array, regardless of size. The execution finger is placed to indicate the platter of this array that is described by the offset given in C, where the value

0 denotes the first platter, 1 the second, et cetera.

The ’0’ array shall be the most sublime choice for loading, and shall be handled with the utmost velocity.

Special Operators.

One special operator does not describe registers in the same way. Instead the three bits immediately less significant than the four instruction indicator bits describe a single register A. The remainder twenty five bits indicate a value, which is loaded forthwith into the register A.

```
                 A
                 |
                 vvv
            .--------------------------------.
            |VUTSRQPONMLKJIHGFEDCBA9876543210|
            ‘--------------------------------’
             ^^^^   ^^^^^^^^^^^^^^^^^^^^^^^^^
             |      |
             |      value
             |
             operator number
```

Figure 3. Special Operators

#13. Orthography.

The value indicated is loaded into the register A forthwith.

Cost-Cutting Measures.

As per our meeting on 13 Febtober 19106, certain ”impossible behaviors” may be unimplemented in the furnished device. An exhaustive list of these Exceptions is given below. Our contractual agreement dictates that the machine may Fail under no other circumstances.

If at the beginning of a cycle, the execution finger does not indicate a platter that describes a valid instruction, then the machine may Fail.

If the program decides to index or amend an array that is not active, because it has not been allocated or it has been abandoned, or if the offset supplied for the access lies outside the array’s capacity, then the machine may Fail.

If the program decides to abandon the ’0’ array, or to abandon an array

```
that is not active, then the machine may Fail.

If the program sets out to divide by a value of 0, then the machine
may Fail.

If the program decides to load a program from an array that is not
active, then the machine may Fail.

If the program decides to Output a value that is larger than 255, the
machine may Fail.

If at the beginning of a machine cycle the execution finger aims
outside the capacity of the 0 array, the machine may Fail.
```

Είσοδος και έξοδος. Το πρόγραμμά σας θα δέχεται από τη γραμμή εντολών ακριβώς ένα όρισμα (`argv[1]`): το όνομα του αρχείου που περιέχει το πρόγραμμα που θα εκτελέσει η εικονική μηχανή.

Κατά τη διάρκεια της εκτέλεσης αυτού του προγράμματος, η εικονική μηχανή πρέπει να διαβάζει από την τυπική είσοδο και να γράφει στην τυπική έξοδο.

Περιορισμοί. Μπορείτε να θεωρήσετε δεδομένο ότι η εικονική σας μηχανή θα εκτελείται σε υπολογιστή αρχιτεκτονικής 32bit (το σύστημα αυτόματης υποβολής και ελέγχου είναι x86).

Πώς να ελέγξετε την εικονική μηχανή σας. Οι λύσεις σας θα βαθμολογηθούν με κριτήριο αφενός την ποιότητα του κώδικα, αφετέρου το χρόνο στον οποίο εκτελεί η εικονική μηχανή σας με επιτυχία το μετροπρόγραμμα (benchmark) που δίνεται στην ιστοσελίδα του διαγωνισμού (`sandmark.umz`).

Μπορείτε επίσης, αν θέλετε, να δοκιμάσετε να τρέξετε με την εικονική σας μηχανή το αρχείο που υλοποιούσε το κύριο μέρος του διαγωνισμού (`codex.umz`).