

Spätný prekladač pre MIPS architektúru

Bc. Kryštof Kiss

Zadanie

- Analyzujte MIPS architektúru, inštrukčný cyklus a samotné inštrukcie. Navrhňte postup dekodovania inštrukcií. Implementujte spätný prekladač v jazyku python s možnosťou čítania zdrojových dát zo súboru. Porovnajte výstupy s inými dostupnými spätnými prekladačmi.

Formát inštrukcií

Analýza

- Register Instructions
- Immediate Instructions
- Jump Instructions

R-Type

6	5	5	5	5	6
opcode	rt	rs	rd	shift	func

I-Type

6	5	5	16
opcode	rt	rs	immediate/offset

J-Type

6	26
opcode	offset

Trieda Disassembler

Návrh riešenia

- obsahuje 4 metódy
 - inicializácia (načítanie JSON súboru a parsovanie dát, načítanie vstupných dát)
 - dekóder
 - disassemble (iteruje cez jednotlivé inštrukcie)
 - decode_instruction (identifikuje inštrukciu, dekóduje použité registre a funcie, vráti dekódovanú inštrukciu)

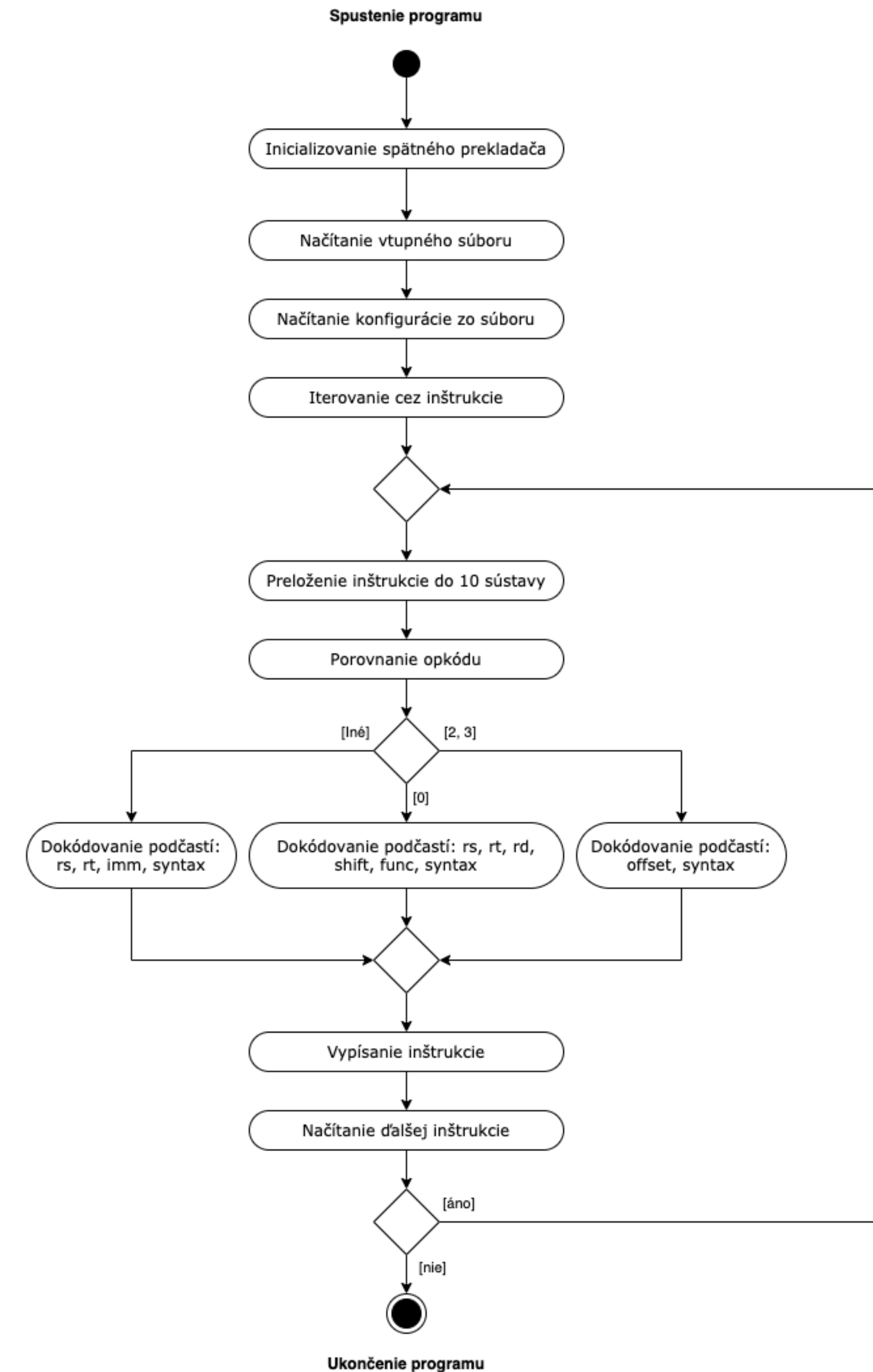
Vstup

Návrh riešenia

- JSON súbor s parsovanými inštrukciami v špecifickom formáte
 - slúži na inicializáciu MIPS Disassembléru, inicializácia zahŕňa:
 1. inicializáciu registrov
 2. inicializáciu podporovaných opkódov
 3. inicializáciu inštrukčnej sady (opkódy, funckódy, syntax inštrukcií)

Aktivity diagram

Návrh riešenia



Implementácia

mips.json

```
"opcode": "0b11111000000000000000000000000000",
"r_type_format": {
  "rs": "0b00000011111000000000000000000000",
  "rt": "0b00000000000011111000000000000000",
  "rd": "0b0000000000000000000011111000000000",
  "shift": "0b000000000000000000000000111110000000",
  "func": "0b000000000000000000000000000000111111"
},
"j_type_format": {
  "offset": "0b00000011111111111111111111111111"
},
"i_type_format": {
  "rs": "0b00000011111000000000000000000000",
  "rt": "0b00000000000011111000000000000000",
  "imm": "0b000000000000000000001111111111111111"
}
```

```
"26": {
  "type": "R",
  "opcode": 0,
  "func": 26,
  "syntax": "div $rs, $rt"
},
"27": {
  "type": "R",
  "opcode": 0,
  "func": 27,
  "syntax": "divu $rs, $rt"
},
"32": {
  "type": "R",
  "opcode": 0,
  "func": 32,
  "syntax": "add $rd, $rs, $rt"
},
```

Vstup/Výstup

Ukážka

```
less disassembler_python/example_input.txt

0x23bdfffc
0xafbf0000
0x20040002
0xaf848000
0x20050003
0xaf858004
0x0c10000b
0xaf828008
0x8fbf0000
0x23bd0004
0x03e00008
0x00851020
0x03e00008
disassembler_python/example_input.txt (END)
```

```
kisso@MacBook-Pro:~/Developer/school/disassembler-python

→ disassembler-python git:(main) ✗ python3 main.py -e
0x23bdfffc -> addi $sp, $sp, 0x0000fffc
0xafbf0000 -> sw $ra, 0x00000000($sp)
0x20040002 -> addi $a0, $zero, 0x00000002
0xaf848000 -> sw $a0, 0x00008000($gp)
0x20050003 -> addi $a1, $zero, 0x00000003
0xaf858004 -> sw $a1, 0x00008004($gp)
0x0c10000b -> jal 0x0010000b
0xaf828008 -> sw $v0, 0x00008008($gp)
0x8fbf0000 -> lw $ra, 0x00000000($sp)
0x23bd0004 -> addi $sp, $sp, 0x00000004
0x03e00008 -> jr $ra
0x00851020 -> add $v0, $a0, $a1
0x03e00008 -> jr $ra
→ disassembler-python git:(main) ✗
```


Ďakujem za pozornosť

Link na disassembler:

- <https://github.com/kisso/disassembler-python>

Zdroje

- <http://www.cs.uwm.edu/classes/cs315/Bacon/Lecture/HTML/ch05s03.html>
- [https://commons.wikimedia.org/wiki/File:MIPS_Architecture_\(Pipelined\).svg](https://commons.wikimedia.org/wiki/File:MIPS_Architecture_(Pipelined).svg)