

**S T U • •**  
**• • • • •**  
**• F E I •**  
**• • • • •**

## **Zadanie č. 1**

Automaty a formálne jazyky

**Autor:** Bc. Martin Kiesel

# Obsah

<b>1</b>	<b>Zadanie</b>	<b>3</b>
	Zoznam inštrukcií Brainfuck . . . . .	3
	Modifikácie jazyka Brainfuck . . . . .	3
	Dodatočné inštrukcie jazyka AFJ . . . . .	3
<b>2</b>	<b>Riešenie</b>	<b>4</b>
	Vývojová špecifikácia . . . . .	4
	Používanie AFJ-interpretra . . . . .	4
	Príklady s vysvetlením . . . . .	4
	AFJ program (BONUS) . . . . .	5
	Obmedzenia AFJ-interpretra . . . . .	6
<b>3</b>	<b>Záver</b>	<b>6</b>

## 1 Zadanie

Naprogramujte program v C++, ktorý interpretuje zdrojový kód (modifikovaného) programovacieho jazyka Brainfuck (ďalej len, *AFJ*). Váš interpreter tento zdrojový kód vykoná (interpretuje) v súlade s platnými inštrukciami jazyka. Ako bonus naprogramujte v *AFJ* jazyku akýkoľvek program, s využitím niektorých inštrukcií (<, >, +, -, [, ], R, W) a opíšte čo robí.

### Zoznam inštrukcií Brainfuck

- < dekrementácia dátového pointera
- > inkrementácia dátového pointera
- , načítaj bajt zo vstupu a ulož ho na pozíciu, kde ukazuje dátový pointer
- . zapíš bajt do výstupu z pozície, kde ukazuje dátový pointer
- + inkrementuj hodnotu bajtu na pozícii, kde ukazuje dátový pointer
- dekrementuj hodnotu bajtu na pozícii, kde ukazuje dátový pointer
- [ ak je hodnota na pozícii, kde ukazuje dátový pointer nula (t.j. 0x00), skoč na inštrukciu za príslušným "]"
- ] ak nie je hodnota na pozícii, kde ukazuje dátový pointer nula (t.j. 0x00), skoč na inštrukciu za príslušným "["

### Modifikácie jazyka Brainfuck

Premapujte inštrukciu "." na znak "W", a "," na znak "R".

### Dodatočné inštrukcie jazyka AFJ

N nastav aktuálny bajt, na nulový tj. 0x00  
! zneguj aktuálny bajt, na ktorý ukazuje dátový pointer

## 2 Riešenie

Riešením zadania je konzolový program naprogramovaný v C++, ktorý po spustení interpretuje vyššie definovaný jazyk (tj. vykoná inštrukciu za inštrukciou) a skončí.

### Vývojová špecifikácia

AFJ-interpreter je vyvíjaný na operačnom systéme OS X, vo vývojovom prostredí Xcode (Version 7.2.1) v programovacom jazyku C++ s dialektom GNU++11. Systém riadenia revízií (SCM) je Git 2.7.2 a repozitár projektu je hostovaný na [GitHub-e](#). Dokumentácia je písaná v L<sup>A</sup>T<sub>E</sub>X s online kompilátorom [sharelatex](#). Všetky názvy metód, premenných a chybové hlásenia sú v anglickom jazyku.

### Používanie AFJ-interpretra

AFJ-interpreter (ďalej len program) používa getopt na parsovanie argumentov. Program akceptuje deväť argumentov a z toho päť s povinným parametrom. Medzi bezparametrové argumenty patria *-h*, *-v*, *-e*, *-x* a argumenty s povinným parametrom sú *-i*, *-s*, *-f*, *-p*, *-o*. Argumenty môžu byť napísané v akomkoľvek poradí, no argumenty *-h*, *-v*, *-e* ukončia vykonávanie programu po ich spracovaní.

Vysvetlenie argumentov:

- h vypíše nápovedu program
- v vypíše verziu programu
- e vypíše viac príkladov ako program spúšťať
- i [FILE] súbor so zdrojovým kódom v AFJ jazyku, ktorý sa má interpretovať (default ./source.afj)
- s "STRING" vstupné pole bajtov (uzavreté v úvodzovkách ak je použitá medzera)
- f [FILE] vstupné pole bajtov v súbore, program súbor číta binárne
- p [hex | str | hexstr | strhex] vypísanie výstupného poľa bajtov na obrazovku, argumentom je typ výstupu
- o [FILE] binárny súbor, kde sa uloží výstupné pole bajtov
- x nepýtať sa, či sa má výstupný súbor prepísať alebo nie (SÚBOR BUDE PREPÍSANÝ)

### Príklady s vysvetlením

Program je možné spustiť bez argumentov, v tom prípade je názov súboru zdrojového kódu predpokladaný ako "source.afj" a vstupné pole bajtov je

plné 0x00 (maximálna dĺžka tohto poľa je opísaná v subsekcii o obmedzeniach). Takto spustený program nemá žiaden výstup.

`./afj-interpreter -i source.afj -s "ABCDE" -o out.bin` spustí program so zdrojovým kódom v súbore "source.afj", vstupným poľom bajtov ABCDE a do súboru "out.bin" sa uloží obsah výstupného poľa. Ak súbor "out.bin" už existuje, program sa spýta (pred samotným interpretovaním) či sa má súbor prepísať, v prípade označenia "y" (yes) bude súbor prepísaný, naopak v prípade označenia "n" (no) sa súbor neprepíše, interpreter sa spustí v oboch prípadoch.

`./afj-interpreter -i source.afj -s "ABCDE" -o out.bin -x` sa program spustí rovnako ako v predchádzajúcom prípade, no neopýta sa na prepísanie súboru.

`./afj-interpreter -i source.afj -f stream.bin -p hexstr -o out.bin -x` sa program spustí rovnako ako v predchádzajúcom prípade, no vstupné pole bajtov sa prečíta binárne zo súboru "stream.bin" a na obrazovku (do konzoly) sa vypíše výstupné pole bajtov najprv hexadecimálne, a potom tlačiteľnými znakmi.

Je vhodné podotknúť, že prípony súborov (\*.afj, \*.bin) nemajú žiadnu úlohu a neovplyvňujú chod programu, teda slúžia čisto na demonštráciu.

## AFJ program (BONUS)

AFJ program načíta zo vstupného poľa bajtov slovo a uloží ho ako palindróm do výstupného poľa bajtov. Vstup: *abcdef* výstup *abcdef fedcbA*, vstup musí mať dĺžku šiestich znakov.

Zdrojový kód:

```
1  |
2      source.afj
3      AFJ source code
4
5      Created by Martin Kiesel on 01/03/16.
6      Copyright 2016 Martin Kiesel. All rights reserved.
7
8      run afj-interpreter such as:
9      ./afj-interpreter -i source.afj -s "tattar" -p strhex
10
11      Prints onomatopoeic palindrome tattarrattat
12      en.wikipedia.org/wiki/Palindrome#Long_palindromes
13
14  |
15
16  R W > R W > R W > R W > R W > R W [>+[-<[->>+<<]>>W<<]<]
```

Hore uvedený program obsahuje komentár, v ktorom sú aj znaky, ktoré by mal interpreter spracovať, no kým je komentár uzatvorený v zátvorkách "[", tak je možné v tele cyklu napísať akýkoľvek sled inštrukcií a nikdy sa nevykonajú, samotný program je na riadku 16.

Program najprv načíta (a hneď uloží do výstupného poľa tj. prvú polovičku palindrómu) šesť bajtov a poukladá ich vedľa seba na pásku. Nasledujú tri vnorené cykly, ktoré postupne kopírujú pomocou inštrukcií “+“ a “-“ znaky po jednom do bunky vždy o dve vpravo (a vypíšu znak na výstup). Vnútorňý cyklus `/-»+«/` slúži na samotné kopírovanie znaku, ďalší cyklus v poradí slúži na výpis, a vonkajší cyklus slúži na posun k ďalšiemu znaku (smerom “<“ vľavo). Najlepšie je tento program demonštrovaný v [Brainfuck Visualizeri](#).

### Obmedzenia AFJ-interpretra

Maximálny počet vstupných bajtov je 2 147 483 647 respektíve číslo `MAX_INT`, keďže to je maximálny možný počet indexov, ktorý vstupný tok dát môže mať.

Ak je AFJ program naprogramovaný tak aby sa nikdy neskončil, a teda je v nekonečnej slučke cyklu `while()`, nikdy sa na výstup nepošle nič na vypísanie (-p) pretože funkcia na vypísanie výstupného poľa sa nikdy nezavolá, to isté platí pre ukladanie do binárneho súboru (-o). Nekonečný program napísaný v AFJ jazyku je napríklad takýto: `+|W|`.

## 3 Záver

Výsledkom zadania je funkčný program, ktorý interpretuje AFJ jazyk (špecifikácia je v časti Zadanie), program vie pracovať s binárnymi súbormi (čítať a zapisovať), vie analyzovať zdrojový kód a vypísať chybovú hlášku v prípade, že počet inštrukcií reprezentovaných znakmi “[“ a “]“ nie je rovnaký (chybná syntax AFJ). Zdrojový kód interpretra je voľne dostupný na serveroch [GitHub-u](#).

Cieľ zadania sa mi podarilo splniť.

5. Marec 2016  
8787 – Martin Kiesel