

Zadanie na 1.počítačové cvičenie

3. apríla 2018

1 Zadanie

Implementujte tento riadkovo orientovaný protokol.

1. Server poskytuje službu „čítanie súboru `zoznam.txt`”.
2. Klient vždy posiela riadok tvaru `"POZIADAVKA parameter"`. *parameter* nemusí byť uvedený ale ak je, je to vždy nejaké slovo obsahujúce nebiele znaky.
3. Server vždy posiela ako prvý riadok odpovede riadok tvaru `"n popis stavu"`, prípadne nasledovaný obsahom odpovede. *n* je vždy trojčiferné číslo.
4. Na začiatku klient pošle požiadavku `VERSION 1.0` s číslom verzie. Server odpovie `100 OK` nasledovaný na ďalšom riadku číslom verzie `1.0`.
5. Ak klient pošle ako prvú požiadavku čokoľvek iné ako `VERSION` server pošle klientovi status `203 Need version` a *uzavrie spojenie*¹
6. Zvyšok komunikácie prebieha takto:

C→S	S→C
<code>LENGTH</code>	stav <code>100 OK</code> , nasledovaný riadkom obsahujúcim počet riadkov v súbore
<code>READ n</code> , <i>n</i> je nezáporné číslo	stav <code>100 OK</code> , nasledovaný riadkom <i>n</i> zo súboru, prvý riadok má číslo <code>0</code>
	stav <code>200 Index too big</code> ak je <i>n</i> príliš veľké vzhľadom na počet riadkov.
	stav <code>201 Bad index</code> ak <i>n</i> nie je nezáporné číslo.
čokoľvek iné	stav <code>202 Unknown command</code>

7. Komunikácia končí, ak klient uzavrie spojenie.

¹Toto mi pripadá rozumnejšie ako čakať. Klient je zjavne chybné napísaný program, netreba s ním mať zľutovanie.

2 Ďalšie pokyny

- Pre analýzu vstupného riadku na strane servra použite funkciu `re.match`
- Snažte sa zdrojový kód písať rozumne tak, aby ste ho mohli ľahko rozšíriť, povedzme pridať ďalší príkaz alebo status.
- Protokol budem testovať pomocou programu, nedodržanie protokolu bude chyba. Uvedomte si najmä, že nepíšete interaktívny program, na druhej strane socketu je *program*, nie človek.

3 Príklad komunikácie

Súbor `zoznam.txt` má 7 riadkov, obsah je takýto:

```
prvy
druhy
treti
stvrty
piaty
siesty
```

Komunikácia je napr. takáto:

```
C->S VERSION 1.0
S->C 100 OK
S->C 1.0
C->S LENGTH
S->C 100 OK
S->C 6
C->S READ 1
S->C 100 OK
S->C druhy
C->S READ 0
S->C 100 OK
S->C prvy
C->S READ 100
S->C 200 Index too big
C->S READ 7
S->C 200 Index too big
C->S READ -1
```

```
S->C 201 Bad index  
C->S READ ahahha  
S->C 201 Bad index  
C->S BIMBAM  
S->C 202 Unknown command
```

a potom server zavrie socket komunikujúci s klientom.