

Zadání 'jádra' úloh

Logovací server

Zadání:

Implementujte server, který nabídne základní možnost logování. Server bude mít možnost obsluhovat vícero 'logů' – funkce `listLogs()`, `createLog(name)`, `attach2Log(name)`. Log bude identifikován textovým řetězcem (max délka 100). Log bude reprezentován souborem. Logy budou cyklické (maximálně 10 záznamů) a jeden jejich záznam bude mít následující položky:

- časové razítko záznamu – čas přijmutí záznamu na server (String – '2008-09-23 14:34:56')
- původce – od koho záznam pochází (String)
- důležitost – úroveň závažnosti (číslo z intervalu 1-5; 5 = debug, info, warn, error, fatal = 1)
- zpráva – vlastní zpráva (String)

Každý log musí podporovat operace `insert(logItem)`, `show()`, `showLast()`, kde `insert(logItem)` vloží nový záznam do logu (parametrem musí být buď objekt reprezentující záznam nebo jednotlivé složky záznamu, nesmí to být jediný String), `show()` zobrazí celý obsah logu (vrací pole `String[]`), `showLast()` zobrazí poslední záznam v logu (vrací jednotlivé položky záznamu – ne String).

V rámci zjednodušení není nutné, aby v době běhu byl synchronizován obsah 'programového' logu se souborovou reprezentací (v tomto případě je však nutné implementovat funkci `sync()`, která se postará o tuto synchronizaci). Soubor reprezentující log nemusí být 'cyklický' (tzn. nemusí v něm být jen posledních 10 záznamů). Po startu logovacího serveru bude stav serveru synchronizován s uloženými soubory (tzn. vrátí se do stavu, ve kterém byl ukončen).

Klient má možnost přihlásit se k serveru, vybrat/vytvořit log, poté odesílat posloupnost jednotlivých příkazů, přepnout se do jiného logu, ... Musí být umožněno, aby mohlo být připojeno více klientů pracujících s různými logy.

Vnitřní struktury

Jméno logovacího souboru bude odpovídat jménu logu (tzn. log se jménem `system03` bude mít svou reprezentaci v souboru `system03.log`).

Např.:

Log jménem `system03`:

obsah souboru `system03.log` po uložení 2 zpráv od serveru 'Kapral'

```
2008-09-09 14:23:34;Kapral;3;Tohleže jsou přišitý knoflíky? Tohleto? Seber je, no!  
2008-09-09 14:27:56;Kapral;5;Zas ti to spadlo. Asi vítr, ne? Postav to, no!
```

Dávkové zpracování

```
./server [IP_adresa] [cislo_portu]
```

, kde `IP_adresa` a `cislo_portu` specifikuje, kde má server naslouchat

```
./client IP_adresa cislo_portu konf_soubor
```

, kde `IP_adresa` a `cislo_portu` specifikují, kde je potřeba hledat server a v souboru je posloupnost příkazů pro klienta. Příkazy mohou mít například následujících tvar:

```
listlogs  
cratelog syatem03  
attach system03  
insert Kapral 5 "Zas ti to spadlo. Asi vítr, ne? Postav to, no!"  
showlast  
show  
sync
```

Ukázkový výstup:

Obsah souboru *konf_soubor*

```
listlogs
create system03
listlogs
attach system03
insert srv2 2 "Test logu"
showlast
insert srv3 5 "Test logu2"
show
sync
```

Výstup z klientské aplikace

```
>> listlogs
<< Logs:
>> create system03
<< Log 'system03' created
>> listlogs
<< Logs: system03
>> attach system03
<< Attached to 'system03'
>> insert srv2 2 "Test logu"
<< Log 'system03' - message "srv2;2;Test logu" inserted
>> showlast
<< Log 'system03' - last message: "2008-09-09 13:45:34;srv2;2;Test Logu"
>> insert srv3 5 "Test logu2"
<< Log 'system03' - message "srv3;5;Test logu2" inserted
>> show
<< Log 'system03' log01: "2008-09-09 13:48:34;srv3;5;Test logu2"
<< Log 'system03' log02: "2008-09-09 13:45:34;srv2;2;Test Logu"
>> sync
<< Log 'system03' - synchronized with file
```