

Manuál k projektu z predmetu IPK  
**Implementácia SFTP serveru a klienta**

# Table of Contents

Zadanie.....	3
RFC 913.....	4
SFTP - Simple File Transfer Protocol.....	4
Formát príkazov.....	4
Formát odpovedí.....	4
SFTP server.....	5
Argumenty.....	5
Proces programu.....	5
Klient.....	6
Argumenty.....	6
Proces programu.....	6
Implementované príkazy.....	7
USER umeno.....	7
PASS heslo.....	8
LIST {F   V} cesta.....	9
CDIR nový-adresár.....	10
KILL súbor.....	11
NAME starý_názov.....	12
DONE.....	14
Spustenie príklady.....	15
Makefile.....	15
SFTP Server.....	15
Klient.....	15

## Zadanie

Vytvoriť SFTP (simple file transfer protocol) server a klienta, ktorý bude so serverom komunikovať pomocou SFTP protokolu. Naštudovať RFC 913, ktorý definuje použitie SFTP.

Pre môj osobný nedostatok času som niektoré časti z RFC 913 poupravil a nekontrolujem všetky možné chybové stavy. Všetky zmeny oproti RFC913 budú popísané ďalej.

# RFC 913

Popisuje možnú implementáciu protokolu Simple File Transfer Protocol.

Niektoré bližšie špecifikácie sú popísané v ďalších sekciách.

## SFTP - Simple File Transfer Protocol

SFTP je protokol, ktorý podporuje užívateľské vstupy, prenos súborov, výpis obsahu adresára, zmenu adresára, premenovanie a vymazanie súborov. Môže byť implementovaný pomocou rôzneho protokolu, ktorý je 8-bitovo zameraný. Využíva iba jedno TCP pripojenie.

Klient sa pripojí na vzdialeného host'a, na ktorý sa posielajú SFTP príkazy a získavajú sa odpovede.

## Formát príkazov

<príkaz> := <prkz> [<MEDZERA> <argumenty>] <NULL>

<prkz> := USER ! ACCT ! PASS ! TYPE ! LIST ! CDIR ! KILL ! NAME ! DONE ! RETR !  
STOR

## Formát odpovedí

<odpoved'> := <kód> [<správa>] <NULL>

<kód> := + | - | | !

<správa> môže obsahovať <CRLF>

## SFTP server

TCP server, ktorý naslúcha pre spojenie s klientom a následne komunikuje pomocou SFTP protokolu. Implementovaný v 2 súboroch a to *ipk-simpleftp-server.cpp* a hlavičkový súbor *ipk-simpleftp-server.hpp*. Server zvláda komunikáciu iba s jedným klientom v jednu dobu. Ukončí sa pri zle zadaných argumentoch, interných chýbách alebo pri jeho vypnutí (Ctrl + C).

## Argumenty

`./ipk-simpleftp-server {-i rozhranie} {-p port} -u cesta_k_súboru`

-i : rozhranie na ktorom server počúva, pokiaľ nieje zadaný, tak sa počúva na všetkých možných rozhraniach

-p : port na ktorý sa server pripojí pre počúvanie klientov, pokiaľ nieje zadaný, tak sa počíta s 115

-u : povinný argument, cesta k súboru, ktorý obsahuje zoznam užívateľských mien a k nim prislúchajúce heslá

## Proces programu

Ako prvé program spracuje argumenty. Následne sa začína vytvárať server. Na začiatku sa vytvorí file descriptor pre TCP soket. Ten sa použije pre nastavenie určitých vlastností, napríklad znova použitie adresy. V štruktúre *struct sockaddr\_in* sa nastaví port a rozhranie na ktorom sa bude odpočúvať. Následne sa nabinduje už vytvorený soket na prvky vytvorenej *sockaddr\_in* štruktúry.

Pokiaľ všetko doteraz prebehlo správne, tak server začína proces naslúchania. Pri pripojení klienta posiela uvítaciu správu ( na rozdiel od RFC 913 sa pošle iba kladná správa) a čaká na príkazy. Príjma príkazy od klienta na základe ktorých sa rozhoduje čo vykoná a akú dá odpoveď. Všetko potrebné vypisuje na štandardný výstup.

## Klient

Klient, ktorý slúži pre pripojenie na SFTP server. Umožňuje zadávať užívateľský vstup, ktorý slúži ako príkazy, ktoré má server vykonať. Ukončí sa pri zle zadaných argumentoch, interných chybách, nútenom vypnutí (Ctrl + C) alebo až keď server ukončí spojenie.

## Argumenty

`./ipk-simpleftp-client -h IP {-p port}`

-h : povinný argument, IPv4 alebo IPv6 adresa serveru

-p : port na ktorom server naslúcha

## Proces programu

Spracujú sa argumenty programu. Vytvorí sa TCP soket. Nastaví sa port a požadovaná IPv4 alebo IPv6 adresa servera. Pokiaľ všetko prebehne správne, tak sa začne proces pripájania. Po úspešnom pripojení na SFTP server získa uvítaciu správu po ktorej môže zadávať príkazy. Vypisuje všetky potrebné informácie na štandardný výstup.

# Implementované príkazy

Táto sekcia hovorí o implementovaných príkazoch a úpravách oproti RFC913. Na príkazy, ktoré nie sú implementované server odpovedá ich názvom.

## USER umeno

**umeno** – užívateľské meno, ktoré sa nachádza v databáze

Príkaz, ktorý je povinný zadať hneď po vytvorení spojenia so serverom.

### Možné odpovede zo servera:

*+User-id valid, send password*

V prípade nájdania mena v databáze mien a hesiel.

*-Invalid user-id, try again*

V prípade, že meno nie je nájdené.

### Zmena oproti RFC913

Využívam iba príkaz USER bez ACCT. Ktorý skontroluje výskyt užívateľského mena a pokiaľ je nájdené je nutné zadať príkaz PASS. Odpoveď „!*umeno*> logged in“ nie je implementovaná.

## Príklady

Klient

```
user name1
--- Server response ---
user name1
+User-id valid, send password
```

Server

```
--- NEW CLIENT IS CONNECTED ---
user name1
+User-id valid, send password
```

## PASS heslo

**heslo** – heslo na vzdialenom systéme, ktoré sa nachádza v databáze mien a hesiel

Tento príkaz musí nasledovať za príkazom *USER*.

### Možné odpovede zo servera:

*! Logged in*

Heslo bolo správne, prístup na vzdialeného hostiteľa je povolený.

*-Wrong password, ty again*

Heslo bolo nesprávne, prístup odmietnutý.

### Zmena oproti RFC913

Tento príkaz je nutné zadať bezprostredne po príkaze *USER*. Odpoveď „+Send account“ nieje implementovaná.

### Príklady

Klient

```
pass pass1
--- Server response ---
pass pass1
! Logged in
```

Server

```
pass pass1
! Logged in
```



## LIST {F | V} cesta

{F | V} – aspoň jeden z týchto argumentov je povinný

F – štandardný výpis

V – vypíše údaje o súbore/priečinku a to veľkosť a práva

cesta – cesta k adresáru v ktorom sa má akcia vykonať

Vypíše všetky súbory a priečinky v danom adresári.

### Možné odpovede zo servera:

-<error>

Error je dôvod prečo daná operácia zlyhala.

+<výpis>

Výpis je názov adresára, za ktorým pokračuje výpis všetkých súborov/priečinkov. V prípade argumentu „V“ sa pridajú aj dodatočné informácie.

### Zmena oproti RFC913

V tomto príkaze nedošlo ku zmene.

### Príklady

Klient

```
list V path
--- Server response ---
list V path
+path
file.txt 0 bytes, permission: -1
Dokumentacia.odt 180291 bytes, permission: 0
```

Server

```
list V path
+path
file.txt 0 bytes, permission: -1
Dokumentacia.odt 180291 bytes, permission: 0
```

## CDIR nový-adresár

**nový-adresár** – cesta k novému adresáru

Zmení aktuálny adresár na vzdialenom host'ovy na adresár predaný argumentom.

### **Možné odpovede zo servera:**

*!Changed working dir to <nový-adresár>*

Zmena prebehla úspešne.

*-Can't connect to directory because: <dôvod>*

Nedá sa zmeniť pre určitý dôvod.

### **Zmena oproti RFC913**

**Odpoved'** „+directory ok, send account/password“ nieje implementovaná. Pracovný adresár sa buď zmení na zadaný, alebo nezmení v prípade chyby. Ďalšie príkazy nie sú nutné.

### **Príklady**

Klient

```
cdir path
--- Server response ---
cdir path
!Changed working dir to path
```

Server

```
cdir path
!Changed working dir to path
```

## KILL súbor

Súbor – súbor ktorý sa ma odstrániť

Odstráni zadaný súbor.

### Možné odpovede zo servera:

+<súbor> *deleted*

Súbor bol vymazaný

-*Not deleted because* <dôvod>

Súbor nebol vymazaný pre nejaký dôvod.

### Zmena oproti RFC913

Žiadne zmeny neboli vykonané.

### Príklady

Klient

```
kill path/test  
  
--- Server response ---  
kill path/test  
+path/test deleted
```

Server

```
kill path/test  
+path/test deleted
```

## **NAME starý\_názov**

**starý\_názov** – názov súboru, ktorý sa ma premenovať

**TOBE nový\_názov**

**nový\_názov** – nový názov súboru

Premenuje súbor. Zadá sa príkaz *NAME*, po ktorom musí nasledovať príkaz *TOBE*.

### **Možné odpovede zo servera:**

**NAME**

*+File exists*

Súbor existuje. Očakáva sa *TOBE* príkaz.

*-Can't find <súbor>*

Súbor sa nedá nájsť.

**TOBE**

*+<starý\_názov> renamed to <nový\_názov>*

Starý názov bol úspešne premenovaný na nový názov.

*-File wasn't renamed because (dôvod)*

Súbor nebol premenovaný pre nejaký dôvod.

### **Zmena oproti RFC913**

Žiadne zmeny neboli vykonané.

## Příklady

### Klient

```
name path/test  
  
--- Server response ---  
name path/test  
File exists
```

```
tobe test_changed  
  
--- Server response ---  
tobe test_changed  
path/test renamed to test_changed
```

### Server

```
name path/test  
File exists
```

```
tobe test_changed  
path/test renamed to test_changed
```

# DONE

Správa klienta na server, že už nebude využívať viacero jeho služieb.

## Možné odpovede zo servera:

+We are done

Ukončenie spojenia servera s klientom.

## Zmena oproti RFC913

Žiadne zmeny neboli vykonané.

## Príklady

Klient

```
done
--- Server response ---
done
+We are done
+MIT-XX closing connection
```

Server

```
done
+We are done
```

# Spustenie príklady

## Makefile

```
daniel@daniel-ThinkPad-T450:~/Desktop/Main/VUTFIT/3.Ročník/IPK/Projekt2$ make
g++ -c -o ipk-simpleftp-server.o ipk-simpleftp-server.cpp
g++ -std=c++17 -Wall -Wextra -Werror -pedantic -g ipk-simpleftp-server.o -o ipk-simpleftp-server -pthread -lrt
```

## SFTP Server

```
daniel@daniel-ThinkPad-T450:~/Desktop/Main/VUTFIT/3.Ročník/IPK/Projekt2$ ./ipk-simpleftp-server -p 8080 -u path/file.txt
-----
SFTP server starts listening
-----

--- NEW CLIENT IS CONNECTED ---
█
```

## Klient

```
daniel@daniel-ThinkPad-T450:~/Desktop/Main/VUTFIT/3.Ročník/IPK/Projekt2$ ./ipk-simpleftp-client -p 8080 -h 127.0.0.1
-----
SFTP client is connected
-----

+MIT-XX SFTP Service

*** Enter commands ***
█
```

## Zdroje

RFC913 - <https://tools.ietf.org/html/rfc913>

Učivo preberané v predmete Počítačová komunikace a síte.