



**IPK – počítačové komunikace a sítě
2017/2018**

Projekt 1: Klient-server pro jednoduchý přenos souborů

Tomáš Pazdiora
login: xpazdi02

12. března 2018

Obsah

1 Zadání.....	3
1.1 Konvence jména klientské aplikace a jejích povinných vstupních parametrů.....	3
1.2 Konvence jména serverové aplikace a jejích povinných vstupních parametrů.....	3
2 BSD Sockety.....	4
3 Protokol IPKFTP.....	5
4 Formát IPKPacket.....	6
5 Serverový a klientský stavový automat.....	7
6 Struktura programu.....	8
7 Reference.....	9

1 Zadání

Navrhnete vlastní aplikační protokol, kterými poté spolu budou komunikovat klient a server.

Umístění souborů na straně serveru je v aktuálním adresáři, ve kterém se nachází server. Spuštění klienta předpokládá provedení pouze jedné operace a to uložení nebo přečtení souboru se zadaným jménem. Komunikace mezi serverem a klientem bude schopna se vypořádat se ztrátami paketů (a to buď na aplikační, nebo transportní vrstvě).

1.1 Konvence jména klientské aplikace a jejích povinných vstupních parametrů

`./ipk-client -h host -p port [-r|-w] file`

- host (IP adresa nebo fully-qualified DNS name) identifikace serveru jakožto koncového bodu komunikace klienta
- port (číslo) cílové číslo portu
- -r značí, že klient bude ze serveru soubor číst
- -w značí, že klient bude na server soubor zapisovat
- file (cesta) určuje cestu k souboru, se kterým se bude manipulovat (buď se soubor odkazovaný cestou uploadne do pracovního adresáře serveru, nebo se obsah souboru stejného jména v pracovním adresáři serveru downloadne na místo specifikované cestou)

např.: `./ipk-client -h eva.fit.vutbr.cz -p 55555 -r myfile.xml`

1.2 Konvence jména serverové aplikace a jejích povinných vstupních parametrů

`./ipk-server -p port`

- port (číslo) číslo portu, na kterém server naslouchá na připojení od klientů.

např.: `./ipk-server -p 55555`

2 BSD Sockety

3 Protokol IPKFTP

IPKFTP je v protokol pro přenos souborů mezi počítači s využitím protokolu TCP. Veškerá komunikace probíhá prostřednictvím packetů formátu IPKPacket (jenž je popsán v následující kapitole). Podrobný popis protokolu pomocí stavových automatů je k nalezení v kapitole [5].

4 Formát IPKPacket

IPKPacket je binární formát pro komunikaci pomocí protokolu IPKFTP. Tento formát není zabezpečený/šifrovaný, ale obsahuje cyklický redundantní součet pro kontrolu chyb dat. Tento formát nemá fixní velikost, ani maximální velikost. Proto je vhodné využít nižší protokol, který je schopný rozdělit tento formát na menší části. V případě protokolu IPKFTP je použit protokol TCP.

offset	velikost	jméno
0h	6 bytů	signatura „IPKFTP“
6h	1 byte	verze protokolu
7h	1 byte	status *
8h	8 bytů	celková velikost zprávy
10h	volitelné	jméno souboru (null terminated)
...	volitelné	data souboru
konec - 4h	4 byty	CRC32 zprávy (polynom 0x04C11DB7)

Tabulka 1: formát IPKPacket

* **status:**

hodnota	název	význam
0	REQUEST	žádost o soubor (vyžaduje jméno souboru)
1	OFFER	nabídka souboru (vyžaduje jméno a data souboru)
2	PING	ping
3	OK	ok
4	ERROR	chyba
5	INACCESSIBLE	nedostupný soubor

Tabulka 2: položka „status“

5 Serverový a klientský stavový automat

6 Struktura programu

7 Reference

- [1] Stephan Brumme, *Fast CRC32*, 4. únor 2015, <http://create.stephan-brumme.com/crc32/#sarwate>
- [2] Ed Rosten, *Compile time lookup tables*, 12. květen 2016, <https://deathandthepenguinblog.wordpress.com/2016/05/12/compile-time-lookup-tables/>
- [3] Evan Klitzke, *Blocking I/O, Nonblocking I/O, And Epoll*, 10. leden 2017, <https://eklitzke.org/blocking-io-nonblocking-io-and-epoll>