

FIT VUT v Brně

Počítačové komunikace a sítě
2017/2018

IPK – Projekt číslo 1

Klient-server pro jednoduchý přenos souborů

Obsah

Obsah.....	2
Základní informace.....	3
FSM.....	4
Klient.....	4
Server	5
Zajímavé pasáže	6
Zápis/získávání hlavičky	6
Kontrola dostupnosti souboru	6
Generování cesty k souboru	6
Demonstrace činnosti aplikace.....	7
Bibliografie.....	8

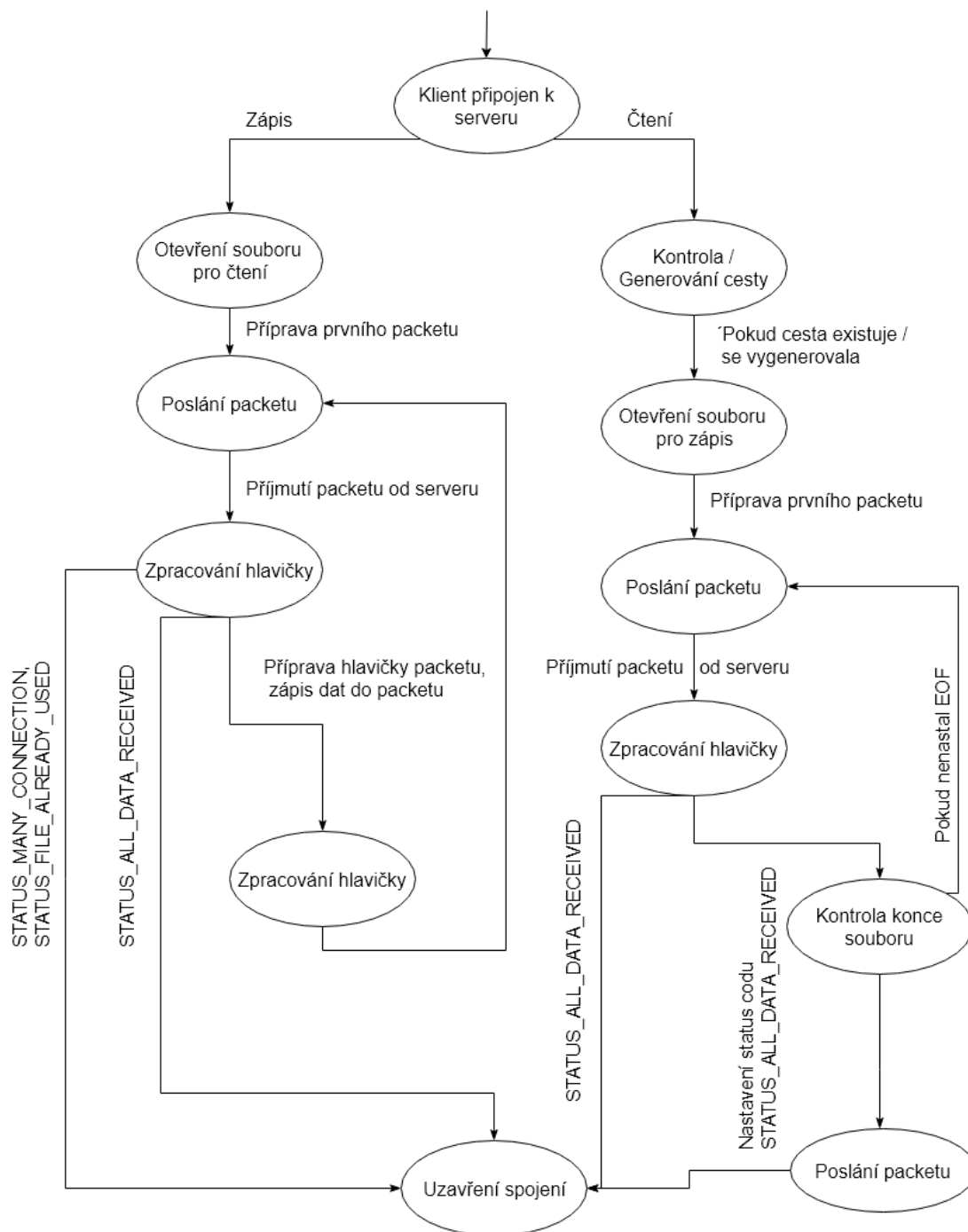
Základní informace

Server je implementován jako konkurentní neblokující server. To znamená, že dokáže obsloužit více klientů zároveň. Proto je potřeba kontrolovat aktuálně využívané soubory, aby nedošlo například, že jeden klient zapisuje do souboru a druhý, který je spuštěn ve stejný okamžik chce data číst. V tomhle případě by klient, který přijde jako druhý dostane od serveru zprávu, že se soubor už využívá a spojení se přeruší. Kvůli kontrole používaných souborů je server omezen na 200 klientů, kteří jsou připojeni v jeden okamžik. Aplikace je naprogramována v jazyce C. Využívá protokol TCP, který oproti UDP řeší ztrátu dat, nebo pořadí packetů na transportní vrstvě. V aplikaci jsou nastaveny dva timeouty. Jeden je nastavený pro socket, kde je timeout nastaven na 10 sekund. Druhý je na serveru pro select, kde je nastavena minuta, tzn. že server vyžaduje, aby klienta poslal packet na server v zadané době jinak dojde k uzavření spojení.

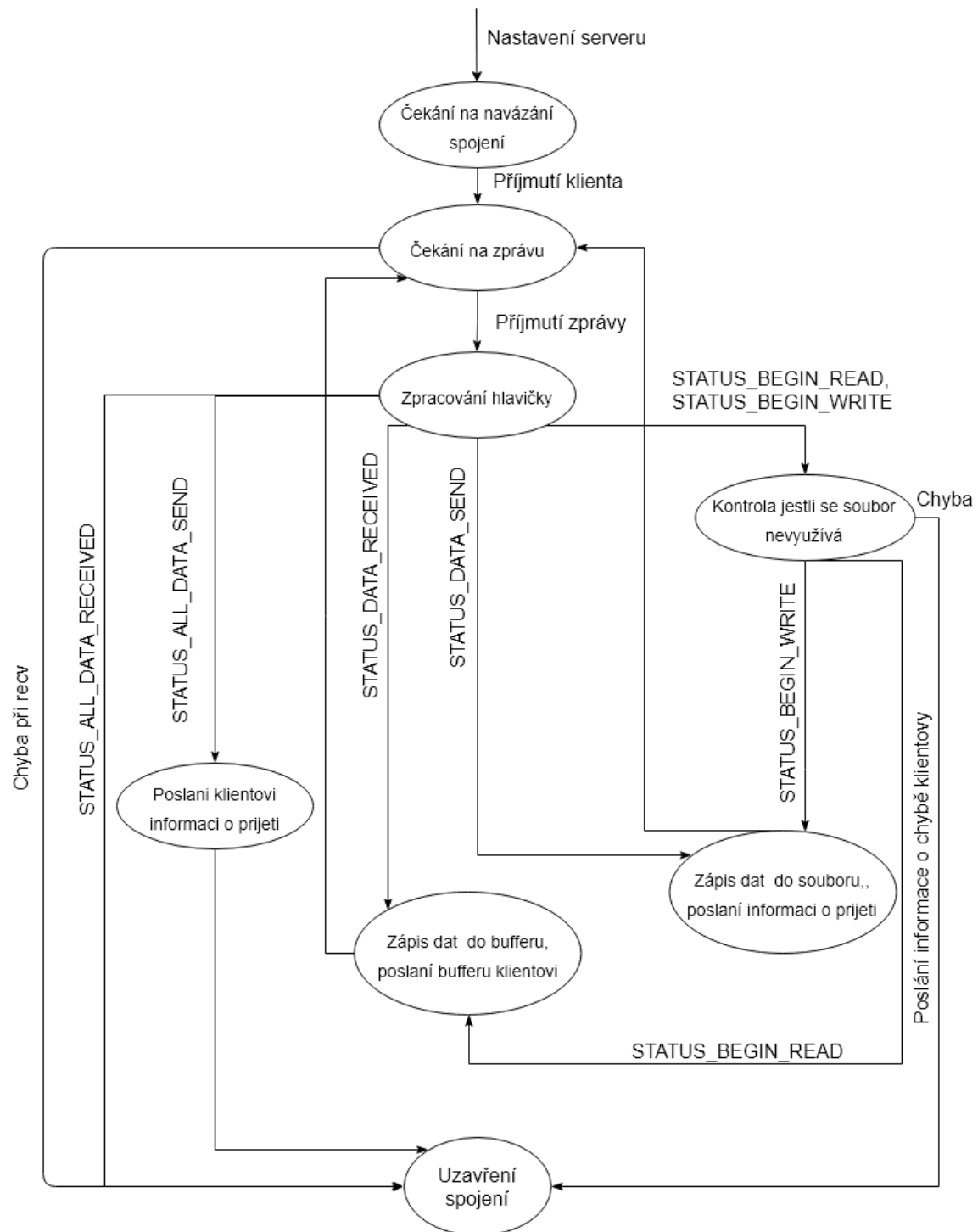
Velikost packetů je konstantní a to 4 kB. První packet od klienta obsahuje povinnou hlavičku obsahující status code, délku dat, délku názvu souboru, název souboru a popřípadě načtená data ze souboru pokud se jedná o zápis na server. Server musí zkontrolovat dostupnost souboru. Kontrola se řeší přes hashe názvů, aby se nemuseli uchovávat řetězce. Další zprávy od klienta už obsahují pouze status code, délku dat a popřípadě data ze souboru, když se jedná o zápis na server. Zprávy ze serveru obsahují status code, délku dat a data ze souboru, pokud se jedná o čtení ze serveru.

FSM

Klient



Server



Zajímavé pasáže

Zápis/získávání hlavičky

Hlavička má vždy 2 bajty v prvních 4 bitech je uložen status code, který tím pádem může nabývat hodnot 0-15. Dalších 12 bitů určuje délku dat, které jsou uloženy ve zprávě. Pokud se jedná o začátek přenosu souborů, tj. STATUS_BEGIN_WRITE a STATUS_BEGIN_READ, tak se navíc ukládá název souboru. Ten jsem se rozhodl uložit tak, že první bajt obsahuje délku názvu souboru. Za tím se nachází už samotné bajty, které obsahují název. Maximální délka názvu je 255 bajtů.

Kontrola dostupnosti souboru

Jelikož každého klienta obsluhuje jiný proces serveru, bylo potřeba ukládat informaci o souborech ve sdílené paměti. Rozhodl jsem se ukládat informaci o názvu souboru ve formě hashu. To z důvodu, aby se nemusel ukládat celý řetězec, ale pouze integer. Pro hashování jsem použil algoritmus djb2 od Dana Bernsteina. Abych nemusel řešit realokaci sdílené paměti zvolil jsem maximální hodnotu připojených klientů na 200. Přišlo mi to jako vhodné číslo jelikož není potřeba moc paměti a zároveň jsem si řekl, že je to dostatečný počet klientů, kteří jsou připojení ve stejný okamžik.

Generování cesty k souboru

Může se stát, že chceme zapisovat do souboru, který zatím neexistuje a neexistuje ani cesta k tomuhle souboru. Proto musíme zkontrolovat existenci zadané cesty, popřípadě ji vygenerovat. Je potřeba cestu generovat postupně. Pomocí funkce dirname se dá získat cesta k souboru bez samotného názvu souboru. A poté cestu postupně zpracovávat po jednotlivých částech, které jsou rozděleny znakem '/'.

Demonstrace činnosti aplikace

Rozhodl jsem se demonstrovat aplikaci tak, že na serveru merlin spustím server, ze svého pc pošlu soubor na server. Poté soubor budu číst ze serveru do jiné složky, a nakonec soubory porovnáím pomocí nástroje diff.

1. První je potřeba spustit server. Zvolil jsem číslo portu 50050.

```
xendry02@merlin: ~/ipk/server$ ./ipk-server -p 50050
```

2. Poté spustím klienta tímhle příkazem

```
$ ./ipk-client -h merlin.fit.vutbr.cz -p 50050 -w test-file.avi
```

3. Soubor se začne zapisovat na server. Abych demonstroval jednu z funkcionalit, tak stejný příkaz zapnu ještě jednou během toho co se soubor na server zapisuje. Klient dostane od serveru zprávu o využití souboru a vypíše uživateli hlášku Connection failed: File is already use.
4. Poté co je soubor celý zapsán na server, tak klient vypíše hlášku All data send a ukončí se. Přišel čas pro čtení souboru ze serveru, čtení spustím následujícím příkazem

```
./ipk-client -h merlin.fit.vutbr.cz -p 50050 -r test/test-file.avi
```

Jelikož složka test zatím neexistuje, dojde k její vygenerování.

5. Po získání všech dat ze serveru, klient vypíše hlášku All Data received a aplikace se ukončí.
6. Pro kontrolu souboru spustím nástroj diff.

```
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client$ diff test/test-file.avi test-file.avi
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client$
```

Diff nic nevypsal to značí, že je vše v pořádku. Pro jistotu ještě zkontroluji velikosti souborů.

```
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client$ ls -l
total 182600
-rwxrwxrwx 1 root root      92 Feb 26 21:46 data.txt
-rwxrwxrwx 1 root root  18856 Mar 12 10:24 ipk-client
-rwxrwxrwx 1 root root   1672 Mar  8 21:24 poznamky.1.txt
-rwxrwxrwx 1 root root   3346 Mar 10 11:42 poznamky.2.txt
-rwxrwxrwx 1 root root 208247 Feb 26 21:34 Screenshot2017-10-2001.18.12.png
drwxrwxrwx 0 root root    512 Mar 12 10:58 test
-rwxrwxrwx 1 root root 186744832 Mar 10 11:56 test-file.avi
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client$ cd test/
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client/test$ ls -l
total 182368
-rwxrwxrwx 1 root root 186744832 Mar 12 10:59 test-file.avi
xendry02@DESKTOP-S9SIMAN:/mnt/c/Users/rykk0/Desktop/IPK-Projekt/projekt1/client/test$
```

7. Velikosti sedí a to znamená, že přenos souboru na server a následné čtení ze serveru proběhlo v pořádku.

Bibliografie

[0] Hash Functions [online]. [cit. 2018-03-11]. Dostupné z:

<http://www.cse.yorku.ca/~oz/hash.html>

[1] Socket Programming in C/C++ [online]. [cit. 2018-03-11]. Dostupné z:

<https://www.geeksforgeeks.org/socket-programming-cc/>

[2] Protokol TCP – I. [online]. [cit. 2018-03-11]. Dostupné z:

<http://www.earchiv.cz/a93/a305c110.php3>