

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Dokumentace k prvnímu projektu předmětu IPK
Varianta 2: Klient-server pro jednoduchý
přenos souborů

Petr Knetl
(xknetl00)

7. března 2018

Obsah

1	Zadání	2
2	Implementace	2
2.1	aplikace serveru	2
2.1.1	Konečný automat serveru	2
2.2	aplikace klienta	3
2.2.1	Konečný automat klienta	3
3	Příklad použití	3
4	Zdroje informací	4

1 Zadání

Zadáním bylo navrhnout a implementovat jednoduchý aplikační protokol architektury klient-server pro přenos souboru. Soubor je možné přenášet jak z klienta na server, tak i ze serveru ke klientovi.

2 Implementace

Implementace je rozdělena do dvou aplikací, jedna pro klienta a druhá pro server. Soubory jsou v obou směrech přesouvány po packetech o velikosti 256 bajtů. K přenosu souboru je využit *TCP protokol s blokujícími sockety*.

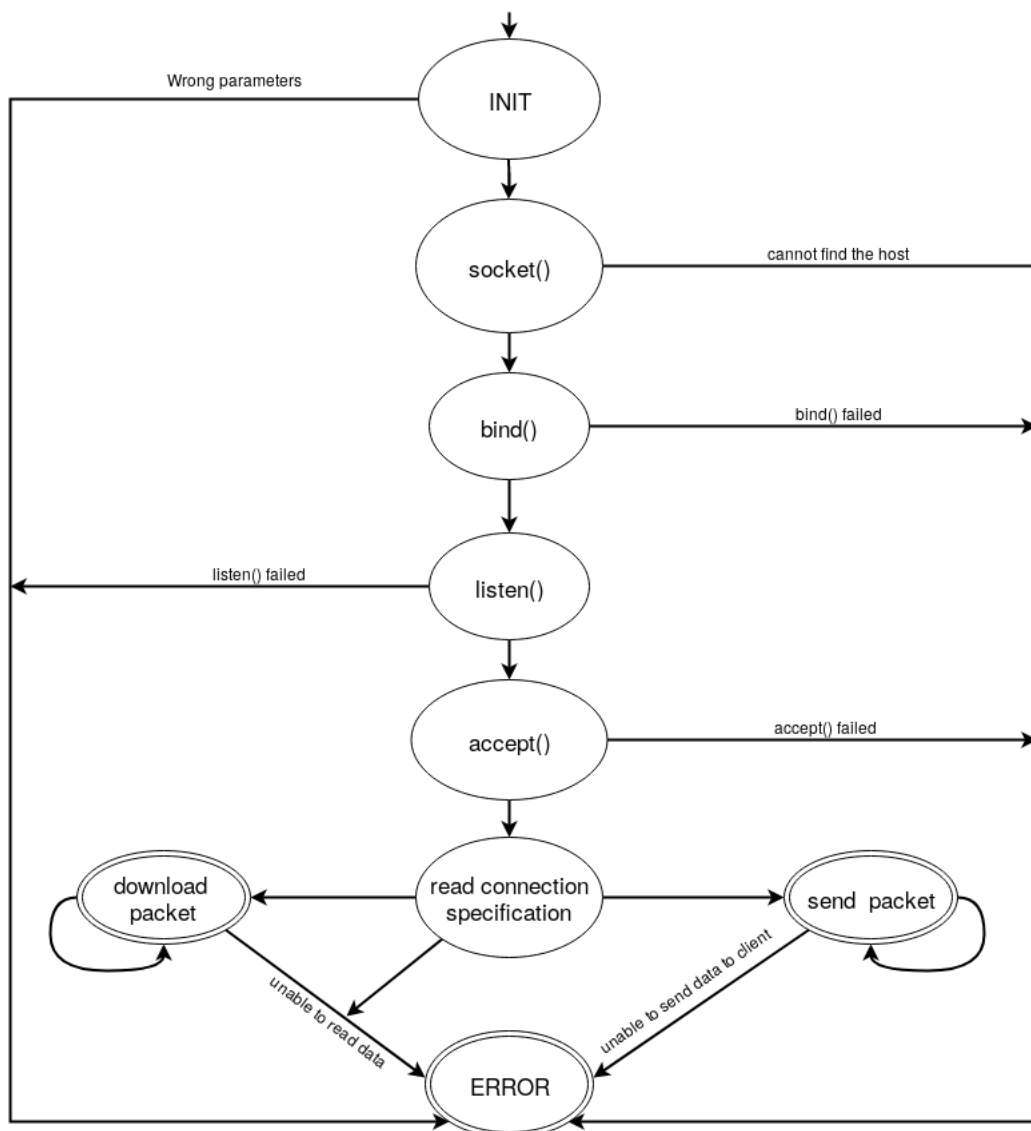
2.1 aplikace serveru

Server je spuštěn s jedním povinným parametrem, s číslem portu. Při spuštění inicializuje zdroje a vytváří socket pro navazování komunikace se serverem. Poté v nekonečné smyčce čeká dokud se klient nepokusí navázat komunikaci. K ukončení serveru dochází pouze zasláním SIGINT signálu.

Při úspěšném navázání komunikace s klientem je vytvořeno nový proces pro jeho obsluhu. Pokud již dochází ke komunikaci mezi serverem a jiným klientem, nově vzniklý proces čeká na konec jejich komunikace.

Jakmile je proces na řadě, vytváří druhý socket pro přenos dat. Poté čte ze strany klienta textovou zprávu, která blíže specifikuje navázané spojení. Ze zprávy získává název přenášeného souboru a informaci o tom zda se bude ze serveru číst, nebo na něj ukládat. Dále v případě zápisu na server je vytvořen nový soubor. Pokud soubor s tímto názvem již existuje, je přepsán. Pokud se klient pokouší ze severu kopírovat neexistující soubor, tak proces odesílá klientovi informaci o neexistujícím souboru a končí chybovou hláškou. Soubor se vždy ukládá do pracovního adresáře serveru.

2.1.1 Konečný automat serveru

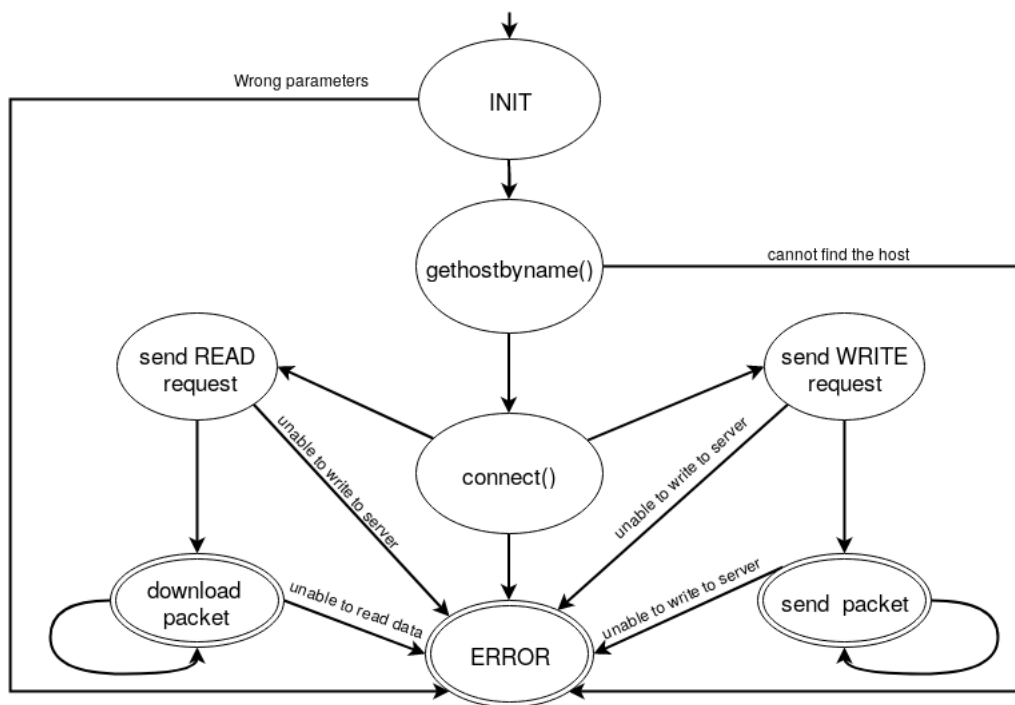


2.2 aplikace klienta

Klient je spuštěn s třemi povinnými parametry (název hosta serveru, číslo portu a názvem souboru ke čtení/odeslání). Klient inicializuje spojení se serverem skrze uvedený port.

Pokud klient čte soubor ze serveru, tak vytváří nový soubor v cestě zadané parametrem do kterého data ukládá. Pokud soubor již existuje, je přepsán. Pokud soubor na straně serveru neexistuje, klient přijímá tuto informaci a končí s chybovou hláškou. V případě zápisu na server klient ověří zda soubor existuje. Pokud ano, posílá data serveru. Po dokončení posílání/příjmu dat klient končí.

2.2.1 Konečný automat klienta



3 Příklad použití

strana klienta

```
./ipk-client -h merlin.fit.vutbr.cz -p 20010 -r ./dir/filename.txt
```

```
INFO: connection establised
```

```
INFO: directory "./dir/" was created
```

```
INFO: file "filename.txt" was downloaded from server on address  
merlin.fit.vutbr.cz
```

strana serveru

```
./ipk-server -p 20010
```

```
INFO: server inicialized and ready for connection(s) by client(s)
```

```
INFO: new client connected from address 95.101.244.102
```

```
INFO: file "filename.txt" sucessfully sent to client (79.141.244.102)
```

4 Zdroje informací

[1] Computer Networking , A Top - Down Approach Featuring the Internet. 6. Polytechnic University, Brooklyn: Addison - Wesley, 2013. ISBN 9780132856201.

[2] Berkley sockets. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2018-03-11].

Dostupné z: https://en.wikipedia.org/wiki/Berkeley_socket#saccept

[3] C++ Tutorial : Sockets - Server & Client - 2018 [online]. [cit. 2018-03-11]. Dostupné z: HONG, K. C++ Tutorial : Sockets - Server & Client - 2018 [online]. 2018 [cit. 2018-03-11].

Dostupné z: http://www.bogotobogo.com/cplusplus/sockets_server_client.php