

# IPK: Přenos souborů

## Popis aplikačního protokolu

### 1 Úvod

Cílem projektu bylo vytvoření aplikačního protokolu pro jednoduchý přenos souborů mezi klientem a serverem, pomocí protokolu TCP.

### 2 Implementace

Samotná implementace se skládá z velmi jednoduchého protokolu. První požadavek zasílá klient, kde formát vypadá následovně:

```
IPK VER CMD FILENAME
```

Jednotlivé elementy mají následující význam:

**IPK** Označení protokolu, v našem případě je to všude právě IPK

**VER** Verze protokolu, v aktuální implementaci se nijak nekontroluje

**CMD** Příkaz, který serveru říká, zda-li chceme soubor nahrávat či stahovat (PUT/GET)

**FILENAME** Název souboru, se kterým se bude provádět předchozí operace

S pomocí výše uvedených informací jsme nyní schopni sestavit skutečné příklady:

```
IPK 0.1 PUT image.jpg      # Nahrání souboru image.jpg na server
IPK 0.1 GET invoice.pdf    # Stáhnutí souboru invoice.pdf ze serveru
```

Na tyto žádosti server odpovídá následující zprávou:

```
XX STATUS_MSG
```

kde **XX** je číslo stavového kódu a **STATUS\_MSG** zpráva, která se k danému kódu vztahuje. V případě, že je kód roven nule, je žádost validní a je možno přistoupit k vlastnímu přenosu dat. Z důvodu synchronizace musí klient před začátkem odesílání/přijímání dat odeslat serveru zprávu **READY**, čímž říká, že je připraven k přenosu. V opačném případě je spojení ukončeno a klient může z výše uvedené zprávy zjistit, kde došlo k chybě.

Po této krátké výměně zpráv již dochází k vlastnímu přenosu, který je ukončen jednou ze stran - v případě žádosti **GET** se jedná o server (přečtení konce souboru), v případě **PUT** jde o klienta (tentýž důvod).

Server také obsahuje kontrolu jmen souborů, aby klient nemohl pracovat se soubory mimo aktuální složku serveru.

### 3 Příklady komunikace

Samotný server přijímá pouze jediný parametr **-p** společně s označením portu, na kterém bude poslouchat:

```
./server -p 12345
```

Nyní je server připraven pro požadavky klienta:

```
./client -h localhost -p 12345 -u myfile
Uploading file 'myfile'
Waiting for server
File uploaded
```

Na serverové části můžeme vidět následující výstup:

```
[worker] spawned
[worker] PUT request for file 'myfile'
[worker] waiting for client
[worker] file saved
[worker] quitting
```

#### Výstup s žádostí **GET**:

```
KLIENT:
./client -h localhost -p 12345 -d myfile
Downloading file 'myfile'
Waiting for server
File downloaded
```

```
SERVER:
[worker] spawned
[worker] GET request for file 'myfile'
[worker] waiting for client
[worker] file sent
[worker] quitting
```

#### Příklad zpracování neplatného souboru:

```
KLIENT:
./client -h localhost -p 12345 -d ../notmyfile
Can't GET file ../notmyfile
Error: 4: INVALID_FILE
An error has occurred during command processing
```

```
SERVER:
[worker] spawned
[worker] received invalid file name
[worker] quitting
```