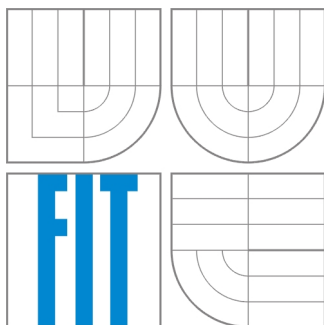




Zálohování news složek přes NNTP protokol
(projekt do předmětu ISA)

17.10.2006

Kamil Dudka, xdudka00@stud.fit.vutbr.cz

Fakulta informačních technologií

Vysoké Učení Technické v Brně

1. Úvod

Aplikace *nntpbackup* je velmi jednoduchý klient pro zálohování zpráv z NNTP severu. Je řešen jako konzolová aplikace. Neočekává žádný vstup od uživatele. všechny potřebné informace se klientovi předávají jako parametry příkazové řádky. Pokud během zálohování nedojde k žádné chybě, program skončí s chybovým kódem 0. Chybové hlášky jsou vypisovány na stderr. Pokud nastane nekritická chyba během stahování jedné skupiny, klient se pokusí zálohovat ostatní skupiny. Zprávy jsou ukládány (v jejich původní podobě) do textových souborů. Jednotlivé zprávy jsou od sebe odděleny prázdnými řádky. Jméno souboru je automaticky zvoleno stejné jako je název diskusní skupiny. Pokud soubor již existuje, bude bez varování přepsán.

2. NNTP protokol

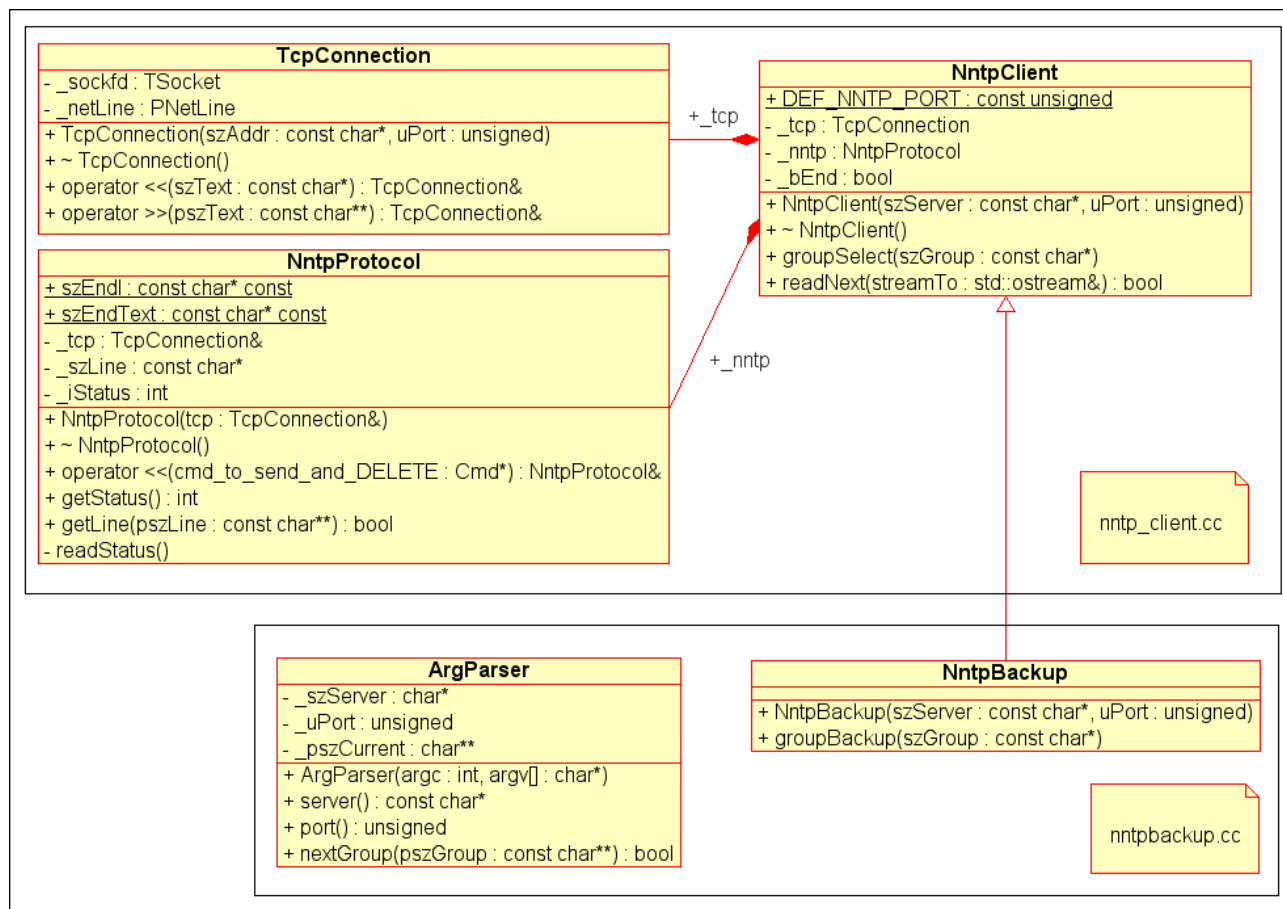
Pro komunikaci se serverem byl použit NNTP protokol. Zkratka NNTP znamená Network News Transfer Protocol. Jedná se o standardizovaný protokol (RFC 977), který se nyní pokusím stručně charakterizovat.

Protokol používá TCP spojení pro přenos dat, výchozí port, na kterém server naslouchá je 119. Komunikuje se prostým textem kódovaným ASCII. Jednotlivé řádky jsou ukončeny CR-LF sekvencí. Klient zasílá serveru příkazy, na které server odpovídá. Příkaz je ukončen CR-LF sekvencí a jeho maximální délka je 512 znaků včetně CR-LF. V programu jsou použity příkazy GROUP, ARTICLE, NEXT a QUIT.

Na příkazy server reaguje zasláním *stavové odpovědi* (obsahující stavový kód a stavovou frázi). Za stavovou odpověď může následovat *odpověď textová*. Stavová odpověď je vždy zaslána jako první, na samostatném řádku ukončeném CR-LF. Za stavovou odpověď může následovat víceřádková odpověď textová. Jednotlivé řádky jsou od sebe opět odděleny CR-LF. Konec textové odpovědi je indikován zasláním tečky na samostatném řádku. Pokud řádek přenášeného textu začíná tečkou, je potřeba nahradit tečku dvěma tečkami.

3. Objektový model

Objektový model aplikace je nastíněn na následujícím obrázku:



Třída `TcpConnection` zapouzdřuje komunikaci se serverem přes sockety. Komunikační protokol aplikační vrstvy je zpřístupněn rozhraním třídy `NntpProtocol`, ta se stará o posílání NNTP příkazu serveru a reagování na jeho stavovou odpověď. Vždy je předpokládána kladná odpověď serveru. Pokud server odpoví chybovým kódem, je zahozena odpovídající výjimka. Metoda `getLine()` této třídy zajišťuje zpřístupnění jednotlivých řádků textové odpovědi serveru – to v praxi znamená jednotlivých řádků zprávy. Automaticky je provedena redukce zdvojených teček na začátku řádku.

Propojení aplikační vrstvy na rozhraní transportní vrstvy zajišťuje třída `NntpClient` s velmi jednoduchým rozhraním. Samotná aplikační logika (zálohování do souboru) je skryta v třídě `NntpBackup`. Na začátku aplikace je vytvořena jedna instance této třídy, voláním metody `groupBackup()` jsou pak zálohovány jednotlivé diskusní skupiny. Zpracování argumentů příkazové řádky zajišťuje třída `ArgParser`.

4. Implementace

Nejnižší vrstva aplikace, která pracuje s rozhraním *BSD sockets* operačního systému je implementována v jazyce C. Samotný objektový model aplikace je pak implementován v C++. Aplikace je tvořena třemi moduly:

- `tcp_client` – nejnižší vrstva aplikace (transportní vrstva)
- `nntp_client` – implementace aplikační vrstvy (NNTP protokol)
- `nntpbackup` – aplikační logika (zálohování do souboru)

5. Návod k použití

Automatickou kompilaci a sestavení aplikace ze zdrojových kódů provedete zadáním příkazu `make` v kořenovém adresáři archivu. Pro výpis parametrů aplikace použijte parametr `--help`. Jednoduchý příklad použití:

```
./nntpbackup news.fit.vutbr.cz vutbr.news.test vutbr.courses.isa
```