

Programátorská dokumentácia započtového programu

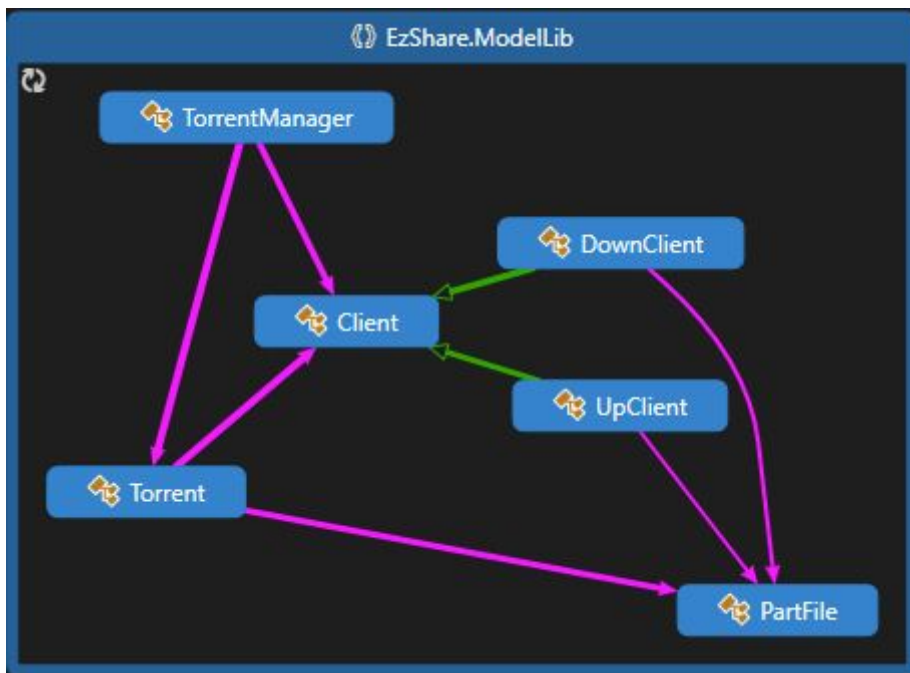
Pomocou aplikácie je možné zdieľať súbory s viacerými počítačmi na základe peer to peer architektúry - komunikácia prebieha medzi každými dvoma klientmi.

Súbory sa dajú zdieľať nasledovným spôsobom: zdieľajúci vytvorí tzv. torrent z existujúceho súboru v jeho počítači. Následne vytvorí malý súbor, ktorý obsahuje informácie o torrente (meno, unikátne ID, veľkosť zdieľaného súboru), a informácie o zdieľajúcej aplikácii (IP počítača a port). Tento súbor následne odošle všetkým, s ktorými chce súbor zdieľať. Vždy, keď sa nový klient pripája, pripojí sa najskôr na IP a port z .share súboru, odtiaľ zistí informácie o všetkých klientoch, ktorí majú daný torrent a pripojí sa aj k nim.

Program je rozdelený na 2 veľké časti: GUI a ModelLib. ModelLib je koncipovaná ako (dynamicky linkovaná) knižnica objektov a funkcií. Je v nej obsiahnutý dátový model, a implementácia sieťového protokolu používaného aplikáciou. GUI je okenné užívateľské rozhranie postavené nad touto knižnicou.

ModelLib

Tu sú znázornené hlavné komponenty knižnice:



TorrentManager obsahuje zoznam Torrentov, ktoré spravuje. Vytvára a pripája nových TCP klientov, ktorých priaduje k príslušným torrentom pre sťahovanie (DownClient). Obsahuje funkciu, ktorá počúva na porte (TcpListener) a pripája nových klientov pre odosielanie (UpClient). Na jednom počítači by mala bežať práve jedna inštancia tejto triedy.

Torrent reprezentuje jeden zdieľaný súbor na sieti. Je v ňom odkaz na práve jeden PartFile. Poskytuje metódu pre stiahnutie súboru. Obsahuje zoznam klientov (trieda Client), ktorí pracujú so súborom.

PartFile reprezentuje súbor uložený na disku. Súbor je rozdelený na časti. O každej časti si pamätá jej stav (či je prítomná na disku). Poskytuje funkcie na prácu so súborom (zápis, čítanie).

Client obaľuje TcpClienta a reprezentuje jedno TCP spojenie. Poskytuje funkcie na odosielanie a prijímanie informácií po sieti.

DownClient dedí od Clienta a slúži na sťahovanie častí súboru. Prijme časť zo siete a zapíše ju do súboru

UpClient dedí od Clienta a slúži na odosielanie častí súboru. Načíta časť zo súboru a odošle ju po sieti.

GUI

GUI využíva WinForms a tvorí užívateľské rozhranie aplikácie. Hlavné okno aplikácie má jednu inštanciu triedy TorrentManager. V tabuľke sa zobrazujú informácie o torrentoch z managera a tlačidlami je pokrytá funkcionálna poskytovaná knižnicou ModelLib.

Dá sa z nej vyvolať form TorrentProperties, ktorý poskytuje detailnejšie informácie o jednom Torrente. Využíva sa tu ProgressViewer, ktorý graficky reprezentuje status jednotlivých častí súboru PartFile.

Protokol a sieťová komunikácia

Sťahovanie prebieha nasledovne: Náš klient sa najskôr pokúsi nadviazať spojenie s klientom, ktorého IP adresa a port sú uvedené v .share súbore. Ak je nedostupný, torrent síce zostane v tabuľke, no je nepoužiteľný - nemá skadiaľ stiahnuť súbor. Ak sa pripojenie podarí, pošle sa unikátne ID, na základe ktorého vzdialený klient vie, o ktorý torrent máme záujem. Následne pošle informácie (IP a port) všetkých, o ktorých vie, že majú k dispozícii daný torrent. Náš klient sa rovnakým spôsobom pokúsi pripojiť aj k týmto klientom.

Následne začína sťahovanie. To prebieha po častiach, ktoré majú 1 MiB. Klient si udržiava informácie o tom, ktoré už má stiahnuté a ktoré ešte nie. Na tie, ktoré ešte nemá postupne posiela náhodne požiadavky všetkým pripojeným klientom. Ak je u nich časť dostupná, prebehne samotné sťahovanie.

Kvôli návrhu programu aplikáciu nie je možné out of the box používať na zdieľanie súborov na Internete bežnými používateľmi. Sťahujúci sa musí totiž pripojiť priamo na počúvajúceho hostiteľa (hostiteľov) súboru. Dnes však má málokto verejnú IP, a väčšina počítačov je schovaná za NATom pod správou poskytovateľa internetu, priame pripojenie teda nie je možné. Dá sa však bez problémov používať v prostredí kde na seba počítače priamo vidia - firemnom, domácom, atď.

Zmeny oproti špecifikácii

Funkcia zastaviť torrent. Táto funkcia splynula s funkciou pauza. Nenašiel som medzi tými to dvoma stavmi torrentu rozdiel, preto som implementoval iba jednu.

Funkcia protokolu "Požiadavok na zistenie stavu súboru". Tu som pristúpil k zjednodušeniu protokolu. Torrent na strane klienta, ktorý by mal odpovedať na tento požiadavok je buď v stave *Seeding*, v tom prípade má dostupné všetky časti, a teda nemá zmysel sa dotazovať na to, ktoré sú dostupné. Dostupná bude každá časť, o ktorú bude požiadané.

V druhom prípade, že stav je *Downloading* síce nie sú dostupné všetky časti, no to, ktoré časti sú dostupné sa stále mení. Tým pádom by poslaná informácia bola okamžite zastaraná. Sťahujúci klient skúša posilať požiadavky náhodne, kým sa netrafí na časť, ktorú zdieľajúci má dostupnú. Myslím, že overhead je tu nie je veľký.