



ŠOLSKI CENTER CELJE
POKLICNA IN TEHNIŠKA ELEKTRO IN KEMIJSKA ŠOLA

Oddaljeno posodabljanje software-a (Programske opreme)

(raziskovalna naloga)

Mentor:

Damjan Luc, univ.dipl.inž.elektroteh.

Avtor:

Alen Stanko E-4.e

Celje, 5.3.2008

1 KAZALO

1	KAZALO	2
2	POVZETEK IN KLJUČNE BESEDE	3
2.1	Povzetek / Abstract.....	3
2.2	Ključne besede / Keywords.....	3
2.3	Kratice in okrajšave.....	4
3	UVOD	5
3.1	Opis / predstavitev raziskovalnega problema.....	5
3.2	Cilji.....	5
3.3	Opis raziskovalnih metod.....	5
4	Visual Studio	6
4.1	Programski Jezik C#.....	6
4.2	Uporabljene Knjižnice.....	6
4.3	Uporabljene Windows Forme	7
5	Problemi in rešitve delov programa	9
5.1	Shranjevanje podatkov	9
5.2	Prenos datotek	10
5.3	Zagon Aplikacije	10
5.4	Funkcija Dodaj	11
6	Navodila za uporabo.....	11
6.1	Uporaba Povezanega seznama	11
6.2	Uporaba Menu-ja.....	11
6.3	Uporaba Forme Dodaj.....	12
6.4	Gumb Posodobi	13
6.5	Navodila za dodajanje obstoječe ini datoteke	13
6.6	Navodila za dodajanje nove ini datoteke.....	14
7	ZAKLJUČEK.....	16
9	PRILOGE	18

2 POVZETEK IN KLJUČNE BESEDE

2.1 Povzetek / Abstract

V tej nalogi sem naredil aplikacijo za posodabljanje programov z uporabo HTTP protokola s pomočjo Windows Visual Studio 2005 Professional v jeziku C Sharp. Program posodablja programe s pomočjo ini datotek , ki so shranjene na disku.

In this work I made an application for updating programs with the use of http protocol and the help of Windows Visual Studio 2005 Professional in the programming language C sharp. The application updates programs with the help of ini files, which are saved on the hard disk.

2.2 Ključne besede / Keywords

Program za posodabljanje

Programski jezik C sharp

Microsoft Visual Studio 2005

2.3 Kratice in okrajšave

- MVS → Microsoft Visual Studio 2005
- C# → C sharp programski jezik
- HTTP → Hyper Text Transfer Protocol
- FTP → File Transfer Protocol

3 UVOD

3.1 Opis / predstavitev raziskovalnega problema

Odločil sem se narediti aplikacijo, ki posodablja programsko opremo prek HTTP strežnika, ker me je zanimalo, kako težko je v programskem jeziku C# dostopati do podatkov na strežniku. Za to aplikacijo potrebujemo Server, kjer imamo dostop do dodajanja in odvzemanja podatkov in seveda internetno povezavo.

3.2 Cilji

Cilj raziskovalne naloge je bil narediti, aplikacijo ki posodablja aplikacije preko HTTP strežnika in tudi naučiti uporabljati C# v okolju Microsoft Visual Studio 2005, ter naučiti tudi dostopati do strežnikov s pomočjo C# programskega jezika.

3.3 Opis raziskovalnih metod

Pri nalogi sem uporabil Microsoft Visual Studio 2005 in tudi zastonski program za postavitvev FTP strežnika (Filezilla) preko katerega sem potem prenašal datoteke na HTTP strežnik, ki sem jih nastavil da so bile gostovane na mojem računalniku. Z ini knjižnico, ki sem jo naredil sam, sem potem naredil neko vrsto podatkovne baze na podlagi katere se dostopa do podatkov, ki so na strežniku, kot določa trenutno prebrana ini datoteka.

4 Visual Studio

4.1 Programski Jezik C#

V programu MVS, sem uporabljal programski jezik C#, ki je bolj enostaven za uporabo, saj je veliko bolj prilagojen za .Net framework podjetja Microsoft. C# je objektni programski jezik in je že samo zaradi tega delo z njim zelo olajšano. Pri C# je dostop do razredov in objektov čisto drugačen kot pri C++.

4.2 Uporabljene Knjižnice

Seveda so v aplikaciji vključene standardne knjižnice, ki se uporabljajo za prikaz »Windows application-a« :

- knjižnica **System**, v kateri so vključene vse knjižnice in se napiše zato da je dostop do knjižnic pri deklaraciji krajši.
- knjižnica **Collections.Generic**, za generične zbirke, ki so bolj varne in imajo boljše delovanje
- knjižnica **ComponentModel**, za to da ustvari kako bo program deloval in kako bo zgrajen. Vsebuje glavne razrede in možnosti spreminjanja za implementiranje atributov in konverterjev tipov, označevanje datotečnih virov, in licenčnih virov.
- knjižnica **Data** omogoča dostop do ADO.NET arhitekture in dovoli da se na enkrat obdeluje več podatkovnih virov
- knjižnica **Drawing** omogoča dostop do GDI+ grafične funkcionalnosti. Razredi kot so **Rectangle** in **Point** so najbolj primitivne GDI+ funkcije. Lahko se še riše z razredom **Pen**, abstrakten razred, kot je **Brush** je pa za polnitev notranjosti in oblik.
- knjižnica **Text** omogoča dostop do bolj zapletenih GDI+ grafičnih funkcij. Razredi te knjižnice so namenjeni zbirki **Fontov** oz. pisav.
- knjižnica **Windows.Forms**, je glavna knjižnica v kateri si lahko zbiramo razrede za izdelovanje **Windows Formov**, ki so vključeni v Microsoft Windowsovem okolju.

To so bile glavne knjižnice, potem je še bila knjižnica, ki sem jo sam naredil in to je **Ini.cs**, v njej je en razred in to je **IniFile**. Ima en konstruktor, kjer definiramo pot do tega IniFile-a. In dva operatorja z katerim beremo in pišemo, sta **IniWriteValue** in **IniReadValue**.

```
Public string IniReadValue (string Section, string Key)
{
    StringBuilder temp = new StringBuilder (255);
    Int i = GetPrivateProfileString (Section, Key, "", temp, 255, this.path);
    return temp.ToString ();
}
```

Primer: IniReadValue operator

Uporabljeni sta bili še dve knjižnici, ki se kot standardne ne vključijo, vendar so ključnega pomena pri izdelavi moje aplikacije:

- knjižnica **IO**, vsebuje tipe in razrede, ki dovoljujejo branje in pisanje v datoteke in datotečne tokove in tipe, za osnovno podporo datotek in direktorijev.

Potrebujem jo za prestavljanje datotek, iz root direktorija pri programu, do določene poti, ki je shranjena v ini datoteki.

- knjižnica **Net**, je za osnovne najbolj uporabljene internetne protokole npr. HTTP in FTP potrebujem ga zaradi razreda webclient , ki vsebuje operator **DownloadFile** in z njim lahko uporabim podatek v ini knjižnici.

4.3 Uporabljene Windows Forme

Windows forme so objekti, ki so avtomatično generirani v okolju MVS, vsi imajo svoje metode in operatorje in tudi podrobnosti. Bili bi samo objekti, če ne bi imeli še grafičnih lastnosti, ki se uporabljajo na celotni aplikaciji, ko jo odpreš, zato da je končen produkt uporabniku veliko bolj prijazen, kot pa da bi bil zapisan za delovanje v konzoli.

Na aplikaciji so uporabljene naslednje forme:

- **Main Form**, kar je glavni obrazec na katerega vstavljamo vse ostale obrazce. Se obnaša, kot nosilec le teh. Na njem se lahko uporabljajo event, oblikovanje, dodajanje lastnosti itd.
- **Menu Strip**, ki je trigger oz. sprožilec. Je uporabljen namesto gumba, saj je veliko bolj pregledno, kot če bi na formo vstavil, 10-12 gumbov, ki so za izvajanje osnovnih operacij, kot na primer odpiranje nove forme, ali pa copy, paste, exit itd.
- **Buttons**, so osnovni nosilec eventov oz. dogodkov, preko njih napišemo del programa, ki se bo izvedel ob kliku gumba, je precej enostavno za uporabiti, lahko mu spreminjamo lastnosti, npr. enable/disable
- **Text Box**, so uporabljeni za izpis osnovnih Stringov, ki se prikažejo v text boxu. Lahko ga tudi uporabljamo kot razred v programu za shrambo Stringov, ki jih trenutno uporabljamo, kar je seveda potratno, vendar se lahko uporabi pri enačenju trenutno uporabljenih dinamičnih text boxov in potem vstavi v že obstoječega.
- **Label**, uporabljen za napis, lahko spreminjamo lastnosti, kot so naprimer napis, ki je trenutno v uporabi, potem če je viden ali ne itd.
- **Checkbox**, enostavna forma, ki se uporabi, da preverimo če uporabnik želi da je določen podatek, ali pač tisto na, kar se checkbox navezuje, res.
- **Dodatna forma**, je forma ki se uporabi ob glavni formi, vendar se obnaša isto, kot glavna forma samo, da je razlika v tem, da ni glavna in se ne prikaže ob zagonu programa.

5 Problemi in rešitve delov programa

5.1 Shranjevanje podatkov

Prvi problem pri izvedbi moje aplikacije je bilo dejstvo, da moram nekako shraniti podatke, ki bodo bistvo pri izvajanju. Ti podatki so, trenutna verzija programa, ki se posodablja, podatek o lokaciji programa, ki se posodablja ter lokacija podatkov za naslednje posodabljanje. V podatkih za posodabljanje potrebujemo še lokacijo nahajanja posodobljenega programa in verzijo programa, ki ga bomo prenesli z interneta.

Ta problem sem rešil z uporabo že omenjene knjižnice ini.cs. Z uporabo te knjižnice lahko iz ini datoteke preberem vse potrebne podatke. Vendar zaradi tega potrebuje vsak program, ki ga posodabljam takšno ini datoteko, kot je predpisana v tem poročilu.

Zgled predpisane ini datoteke, ki je uporabljena, kot naslavljanje .ini datotek iz interneta:

```
[Prop]
  Filename = nekaj.exe
  Version = 1001
  File Path=C: /nekjenadisku/
[Naslov]
  Net = http://določen_naslov.com/
  IniName=ini.ini
```

S takšno ini datoteko lahko potem prenašamo podatke z interneta. Na strežniku se nahaja še ena ini datoteka s pomočjo katere preverimo verzijo exe datoteke na strežniku in potrebo po prenosu. Če je na sistemu nameščena ista verzija kot se nahaja na strežniku, datoteke ne prenašamo saj ni potrebe.

Na tak način potem shranjujem podatke in vedno vem od kod, kje in kako moram delati z datotekami in programom ki se posodablja.

5.2 Prenos datotek

Bistven del moje aplikacije je prenos datotek s strežnika, vendar je pa tudi eden izmed lažjih. Z uporabo knjižnice **System.Net** in določenih razredov v tej knjižnici, ki uporabljajo standardne internetne protokole je prenašanje datotek zelo enostavno.

Po kliku na gumb posodobi, se določene ini datoteke odprejo in z njihovo pomočjo lahko potem uporabljam podatke, ki jih preberem iz ini datotek.

Celoten potek posodabljanja bom tudi opisal:

Najprej se odpre ini datoteka in se prebere HTTP naslov .ini datoteke na strežniku.

Potem se prenese ini datoteka in se primerja z ini datoteko na sistemu, kar pomeni da preverim, če je na strežniku novejša verzija programa, ki se posodablja.

S pomočjo nove ini datoteke se preveri naslednji HTTP naslov, ki se uporabi za prenos novejše verzije programa. Potem se še s pomočjo standardnih Windows ukazov prenese .exe datoteka in se shrani tako, da prepíše datoteko, ki je nameščena na sistemu.

To je celoten proces, ki se izvede ob posodabljanju.

5.3 Zagon Aplikacije

Ob zagonu aplikacije se za večjo funkcionalnost in tudi preglednost iz datoteke *root.ini* izpišejo vse ini datoteke, ki se uporabljajo za posodabljanje. To se zgodi tako, da se na povezanem seznamu, v tako imenovani, **DataGridView** izpišejo naslovi ini datotek. Vse ini datoteke se potem lahko v povezanem seznamu označijo za uporabo v procesu posodabljanja z uporabo checkbox ikon (odkljukamo).

5.4 Funkcija Dodaj

S to formo lahko dodajamo nove ini datoteke v povezan seznam, s tem dosežemo, da je ustvarjanje ini datotek za posodobitev veliko lažja.

V formi so obrazci za dodajanje že obstoječih ini datotek in seveda tudi za vnos novih ini datotek za posodabljanje na novo dodanih programov. Ti obrazci so zelo podobni predlogi in morajo biti pravilno izpolnjeni drugače program javi napako.

6 Navodila za uporabo

Za pravilno uporabo moje aplikacije je potrebno upoštevati predpostavke in pravilne tipe vnosov v zahtevana tekstovna polja. Zahteva se tudi znanje akcij določenih gumbov, ki so porazdeljeni po formi. Zaradi tega sem se odločil narediti navodila za uporabo, ki bodo opisala funkcije določenih gumbov, ter kako uporabiti gumbe in dodatno formo.

6.1 Uporaba Povezanega seznama

Na povezanem seznamu se najdejo ini datoteke, ki jih uporabljamo za posodabljanje programov, ki so nameščeni na računalniku. V seznamu se najdejo podatki o tem, kje se ini datoteka nahaja in kako se imenuje program, ki se posodablja. Ob teh podatkih je še checkbox, ki se uporabi za omogočitev posodabljanja programa, saj se le ta posodobi le v primeru če je checkbox odkljukan. Z uporabo gumba Refresh se pa po dodajanju novih ini datotek ali že obstoječih posodobi povezan seznam in prikaže te podatke. (glej sliko spodaj)

6.2 Uporaba Menu-ja

V menu-ju se najdejo za to aplikacijo pogosti ukazi, kot so Exit in Dodaj. Ob kliku na izbiro Exit se program zapre, ob kliku na jezik Dodaj pa se odpre forma ki se imenuje Dodaj. S to formo lahko potem dodajamo novo ali pa že obstoječo ini datoteko.

6.3 Uporaba Forme Dodaj

Z uporabo forme Dodaj lahko, kot že omenjeno dodajamo nove ali že obstoječe ini datoteke. Na tej formi se najdejo tekstovna polja, v katera vpisujemo zahtevane podatke. Opis polj od začetka do konca (glej sliko spodaj):

- Prvo je polje, ki od uporabnika zahteva pot, do že obstoječega ini datoteke, ki mora biti v obliki: »C:\Pot\Pot\naslov.ini«
- Naslednje polje je ločeno s črto skozi formo in se uporablja za dodajanje nove ini datoteke, ki se tudi na novo ustvari. Najprej je potrebno vnesti mesto, kjer bo nova ini datoteka v obliki: »C:\pot\pot\naslov.ini«. Če navedemo le ime datoteke v obliki »ime.ini«, se bo le ta ustvarila v root mapi.
- Potem je polje za ime datoteke, ki se posodablja. Vanj vpišemo datoteko v obliki: »Naslov.exe«. To pomeni, da se bo ta datoteka prepisala, ko bo na strežniku nova verzija. Pot do datoteke navedemo kasneje.
- V naslednje polje vnesemo verzijo programa, ki kot privzeto predlaga: »1001. Ta podatek se potem uporablja za ugotavljanje ali je na strežniku nova verzija programa, saj le če je nova verzija višja od trenutne se program posodobi.
- V naslednje polje vnesemo pot, kjer se nahaja program. Uporablja se zato, da lahko prenesene datoteke, shranimo na pravo mesto na disku, kar pomeni da po prenosu datoteke program prekopira preneseno datoteko prek že obstoječe in na tak način posodobi program, ki je naveden za posodabljanje. Predpis je: »C:\pot\pot«, brez imena datoteke, saj se le ta vnese polje »Ime datoteke, ki se posodablja«.

- Potem je polje, kjer je spletni naslov nove ini datoteke za posodabljanje, ki se ob kliku na gumb Posodobi prenese. Predpostavka za HTTP naslov je: »http:\\naslov\«

Na formi so potem še checkbox-i, za izbiro kateri .ini file bomo dodali. Npr. če dodajam novo ini datoteko izberem checkbox dodajanje obstoječe ini datoteke, ki je spodaj. Seveda v obratnem primeru izberem checkbox za obstoječo ini datoteko, če želim spremeniti podatke v že obstoječi ini datoteki.

Uporaba gumba dodaj je potem v primeru, ko želiš shraniti vpisane podatke, v primeru da sta oba checkbox-a odključana se dodajanje ne izvede. Enako se zgodi, če noben checkbox ni odkljukan. Potem če so polja prazna se tudi nič ne izvede, saj program javi napako.

V to formo je tudi potrebno dosledno vnašati po predpisanih vrednostih, saj drugače v ini datoteko ne zapišejo pravilni podatki in tako zelo otežimo proces ali pa ga celo spremenimo v nemogočega.

Ima tudi gumb Exit za izhod iz te forme in ne celotnega programa, tako da ob pritisku na ta gumb zapremo formo Dodaj. Vendar pred tem moramo pritisniti na gumb dodaj, če želimo ustvariti novo ini datoteko ali pa dodati v staro.

6.4 Gumb Posodobi

Je vitalni del aplikacije in ob kliku le tega se začne izvajati proces, ki posodobi odključane programe na povezanem seznamu. Ob kliku tega gumba se izvede proces, katerega delovanje je opisano pri Opisu funkcij na prejšnjih straneh.

6.5 Navodila za dodajanje obstoječe ini datoteke

Zdaj bom na kratko predstavil stvari, ki jih je potrebno narediti pri dodajanju obstoječe ini datoteke:

- Najprej moramo zagnati program, ko se le ta zažene preveri koliko ini datotek je shranjenih v *root.ini* in jih izpiše na povezan seznam,

- Potem na menu-ju kliknemo na jeziček Dodaj ali pa uporabimo nastavljeno bližnjico *Ctrl+D*,
- Na formi Dodaj odključamo dodajanje obstoječe ini datoteke in vpišemo v tekstovno polje pot obstoječe ini datoteke,
- Pritisnemo na gumb dodaj in se ini datoteka doda v root.ini,
- Kliknemo gumb Exit in pridemo nazaj na glavno formo,
- Na glavni formi pritisnemo gumb Refresh in tako se nam na povezanem seznamu pojavi na novo dodana ini datoteka,
- Odključamo dodano ini datoteko in pritisnemo na gumb Posodobi,
- Po prenosu in kopiranju exe datoteke lahko zapremo program s klikom na gumb Exit, ali pa z bližnjico Alt+F4.



Slika 1: Primer pravilno vnesene nove ini datoteke

6.6 Navodila za dodajanje nove ini datoteke

Pri dodajanju nove ini datoteke začnemo, kot pri dodajanju obstoječe vendar so določene spremembe, tako da je najbolje da celoten proces spet opišem:

- Najprej moramo zagnati program, ko se le ta zažene preveri koliko ini datotek je shranjenih v *root.ini* in jih izpiše na povezan seznam,
- Potem na menu-ju kliknemo na jeziček Dodaj ali pa uporabimo nastavljeno bližnjico *Ctrl+D*,

- Na formi Dodaj odkljukamo dodajanje nove ini datoteke in v tekstovno polje vpišemo pot nove ini datoteke, ime ini datoteke, ime programa, pot kjer se program, internetni naslov ini datoteke na strežniku, ter trenutno verzijo programa.
Pri vnosu podatkov moramo upoštevati že prej omenjene omejitve za imena in poti datotek.
- Pritisnemo na gumb Dodaj in ini datoteka se doda v *root.ini*,
- Kliknemo gumb Exit in pridemo nazaj na glavno formo,
- Na glavni formi pritisnemo gumb Refresh in tako se nam na povezanem seznamu pojavi na novo dodana ustvarjena ini datoteka,
- Odkljukamo dodano ini datoteko in pritisnemo na gumb Posodobi,
- Po prenosu in kopiranju exe datoteke lahko zapremo program s klikom na gumb Exit, ali pa z bližnjico Alt+F4.

7 ZAKLJUČEK

V nalogi mi je uspelo ustvariti aplikacijo, ki prek HTTP protokola prenese in posodobi na računalniku že obstoječ program. Uporaba programa je enostavna in mogoče na nekaterih mestih bolj komplicirana. Še posebej zapletena je pri dodajanju novih ini datotek, saj oseba brez predznanja o uporabi računalnika ne bi znala dodati ini datoteke.

Naučil sem se bolje uporabljati programski jezik C#, ki je precej enostaven za uporabo in če si dovolj interesiran zelo močno orodje za pisanje programov. Sedaj tudi znam s tem jezikom prenašati datoteke preko HTTP in FTP protokola.

Težave so bile pri strežniku, potem pri avtentikaciji z uporabo **WebClient** razreda, saj najprej **NetworkCredential** razred ni dovoljeval dostopa in pri iskanju dobre in funkcionalne shrambe podatkov o programih ter verzijah za posodabljanje.

Boljše delovanje bi zagotovil z uporabo podatkovne baze, ki bi veliko reči zelo olajšala.

Lahko pa bi bilo vse v eni ini datoteki in bi se vsi podatki brali iz nje.

VIRI IN LITERATURA

- <http://www.geekpedia.com>
- Zapiski pri predmetu APJ
- J. Price, M. Gunderloy, Visual C# .NET, 2002, Sybex
- A. S. Tanenbaum, Computer Networks, Fourth Edition, 2003, Pearson Education, Inc.

9 PRILOGE

CD s programom