Projektspecifikation

Programmering i C++ (DVB021)

Martin Boldt
School of Engineering
Blekinge Institute of Technology

2004-01-12

Introduktion

Projektuppgiften i kursen DVB021 går ut på att skriva ett eget förenklat antivirusprogram i C++ (alltså *inte* C). Det finns en befintlig virusdatabas som ert program skall använda sig av. Denna databas innehåller beskrivningar av filer som är klassade som virus. Er uppgift är att skriva ett program som traverserar (går igenom) en katalog i filsystemet samt alla dess underkataloger, och dess underkataloger osv. För varje fil programmet hittar skall det kontrollera ifall den matchar någon av beskrivningarna i virusdatabasen. Varje gång programmet påträffar en fil som finns med i virusdatabasen så skall detta rapporteras till en logfil.
För att göra uppgiften mer överskådlig så bryter vi ner uppgiften i mindre och mer lätthanterliga delar (divide and conquer). Genom att lösa respektive del för sig så kommer arbetet förhoppningsvis att gå lättare. Projektet kan brytas ner i följande tre delar: filtraversering, virusdatabasen och filidentifiering.

Gruppindelning

Det är tillåtet att göra denna uppgift i grupp om två studenter. De som vill får även göra uppgiften individuellt. Oavsett om man utför uppgiften själv eller i grupp om två så skall samtliga delar i denna specifikation uppfyllas. Studenterna ansvarar själva för att samordna gruppindelningen. Observera att om ni löser uppgiften i grupp så måste *båda* i gruppen kunna förklara *all* kod som gruppen skrivit. Om någon i gruppen inte kan utföra detta så kommer denna person att självständigt få genomföra en omredovisning.

Systemspecifikation och examinering

Ni får själva välja ifall ni vill lösa uppgiften under Windows eller Unix. Examinationen kommer att utföras under Windows XP eller FreeBSD 4.10. Om uppgiften löses under Windows är det ett krav att ni skickar in er Visual Studio projektfil. Under FreeBSD skall en fungerande *Makefile* bifogas. Vid examineringstillfället så kommer ert antivirusprogram att testas genom att köra det på en testkatalog där ett förutbestämt antal filer skall identifieras som virus.

Del 1 - Filtraversering

Ert program skall ges ett startbibliotek och skall sedan gå igenom alla filer i det samt alla underkataloger, och dess underkataloger osv. Denna typ av

fillistningar brukar lösas med *rekursiva* funktionsanrop, alltså en funktion som anropar sig själv för varje underkatalog. Namnet på startbiblioteket skickar ni in till programmet som ett argument i kommandoprompten, t.ex:

```
./dvb021proj MyStartDir/
```

Observera även att ni inte får använda externa program för att traversera filsystemet eftersom ni skall skapa en egen lösning i C++. Om ni stöter på problem med att det inte går att öppna filer så kan det bero på att ni har för många bibliotek öppna samtidigt. Kom ihåg skillnaden mellan *DepthFirst*[1] och *BreadthFirst*[2] vid rekursiv traversering genom en katalogstruktur.

Unixtips, se `opendir()` och `readdir()`.
Windowstips, se `FindFirstFile()` och `FindNextFile()`.

Del 2 - Virusdatabasen

Det är i virusdatabasen som beskrivningarna av virus finns lagrade. I vårt fall är det egentligen ingen databas utan en helt vanlig textfil, med namn `signatures.db`. Varje rad i textfilen innehåller en virusdefinition. Ni kan alltså läsa tecken för tecken på varje rad tills ni stöter på ett radslut `'\n'` eller så använder ni er av en funktion för att läsa filer rad för rad. Varje virusdefinition innehåller två värde som är åtskilda av ett '='. Det första värdet är en textsträng som innehåller *namnet* på viruset. Denna textsträng får inte överstiga 32 tecken, inkl. NULL-terminering. Efter namnet följer ett '=' och därefter kommer en beskrivning av innehållet i virusfilen, se exempel nedan.

```
TestVirus.D=255044462d312e340a332030206f62
```

Det andra värdet, alltså virusbeskrivningen, har ingen maxlängdsbegränsning. Vidare är beskrivningen lagrad i hexadecimal notation. Detta betyder att varje byte i virusfilen (*en* byte) beskrivs med *två* tecken i virusdefinitionen. Se exempel för `TestVirus.D` nedan.

Byte nr:	0	1	2	3	4	5	5	7
Hexvärde:	0x25	0x50	0x44	0x46	0x2d	0x31	0x2e	0x34
Decimalt:	37	80	68	70	45	49	46	52
Ascii:	'%'	'P'	'D'	'F'	'-'	'1'	'.'	'4'

Virusbeskrivningen börjar alltid från byte 0 (noll) i den misstänkta virusfilen. I tabellen ovan kan vi se att de första byten i virusbeskrivningen är en textsträng som innehåller `"%PDF-1.4"` osv. Vi kan även sluta oss till att antalet byte som finns beskrivet i virusbeskrivningen är längden på beskrivningen delat med två. Så om vi har en virusbeskrivning som är 50 tecken lång så beskriver den de 50/2=25 första byten i virusfilen.

Det format på AV-databas som vi använder här är detsamma som opensource programmet ClamAV använder. ClamAV klarar dock ett antal extra features så som att en virusbeskrivning inte *alltid* behöver börjar på den första byten i virusfilen. Dessutom finns det möjlighet att använda *wildcards* som t.ex. `*` eller `?` i virusbeskrivningarna hos ClamAV. Som en extra uppgift kan ni lägga till support för ClamAV-signaturer. Om ni gör det så kan ni använda deras färdiga virusdatabaser för att identifiera riktiga virus. Mer information finns i dokumentet `signatures.pdf` på siten `www.clamav.net`.

[1] Se följande länk: http://en.wikipedia.org/wiki/Depth-first_search
[2] Se följande länk: http://en.wikipedia.org/wiki/Breadth-first_search

Del 3 - Filidentifiering

För varje fil ni stöter på vid filtraverseringen så skall ni jämföra den mot samtliga virusdefinitioner i virusdatabasen. Observera att ni alltid börjar med att jämföra första byten i filen mot första byten i virusbeskrivningen, såvida ni inte löst extrauppgiften ovan. Om filen stämmer överens med en av virusdefinitionerna så skall ni logga detta i en logfil med namn `dvb021.log`. I logfilen skall det finnas information om vilken fil som är infekterad, sökvägen till filen samt namnet på det virus som antas ha infekterat filen.

Inlämning

Deadline för uppgiften hittar ni på Idenet under kurssidan. Innan ni skickar in er inlämningsuppgift skall ni kontrollera att ni gjort följande:

- Uppfyller samtliga krav i denna specifikation.
- Angett namn samt mailadress **i samtliga filer** ni lämnar in, samt i mailet till examinatorn, alltså även era kodfiler.
- Skickat in er projektfil från Visual Studio om ni kodat under Windows.
- Skickat in er *Makefile* om ni kodat under FreeBSD.
- Packat alla era inlämningsfiler som ett Rar- eller Zip-arkiv.

Då ni gjort detta kan ni maila er packade fil till följande mailadress:

deadline_dvb021@ipd.bth.se

Lycka till!