

Tentamen

Nätverksprogrammering

Del 2

2011-06-03, 14.00-19.00

Tillåtna hjälpmedel för denna del av tentamen:

- Java snabbreferens.
- Kursboken: Java Network Programming av Eliotte Rusty Harold.
- Valfri lärobok i Java.
- Utskrift av OH-bilder från föreläsningarna.

Denna tentamen i kursen Nätverksprogrammering består av två delar – en del som innehåller frågor av teoretisk/principiell/utredande karaktär och en del som innehåller praktiska programmeringsuppgifter. Detta är del 2. Den ska du ha erhållit tillsammans med ett färgat tentamensomslag när du lämnade in din lösning på del 1 av tentamen.

För godkänt betyg på tentamen krävs sammanlagt minst 20 poäng på tentamen, varav minst 8 poäng på vardera deltentamen. För högre betyg krävs naturligtvis mer, så gör så många uppgifter du kan.

DownloadHelper

Uppgift

Edvard (Ned) Laddare är grundaren till ett företag i Lund. Hans idé är att skapa ett program för att ladda ner alla filer av en viss typ som refereras till från en given URL. Ned vill konkurrera med en välkänd add-on till webbläsaren Firefox: DownloadHelper (andraplats på listan över de mest populära add-ons). Ned vill i sin första version av programmet fokusera på att ladda ner alla PDF-filer som en webbsida refererar till och er uppgift är att hjälpa honom.

Du ska alltså hjälpa till med att skriva ett program som tar en URL som argument på kommandoraden, analyserar sidan och identifierar alla PDF-filer som sidan refererar till och därefter laddar ner dessa PDF-filer till hårddisken.

Bakgrund

DownloadHelper är en add-on till Firefox som tar en URL som in-data, analyserar HTML-koden för sidan och extraherar alla länkar. Den visar sedan ett fönster som innehåller en rad för varje refererad fil. Man kan sedan i fönstret ange vilka av dessa filer man vill ladda ner, t.ex. alla PDF-filer, och sedan starta nedladdningen. I vår tentamensuppgift begränsar vi oss dock till PDF-filer (referenser vars filnamn slutar på ".pdf") samt utesluter det grafiska gränssnittet – vi laddar helt enkelt ner samtliga PDF-filer som omnämns på sidan. Man kan även ange att DownloadHelper ska starta ett antal trådar (konfigurerbart antal) som laddar ner parallellt.

Figur 1 visar utseendet av webbsidan: <http://cs.lth.se/kurs/eda095/foerelaesningar/> och figur 2 visar resultat av den länkanalys som DownloadHelper gjort.

Föreläsningar

Här hittar du en kort beskrivning av vad som behandlades/planeras behandlas under de olika föreläsningarna tillsammans med referenser till relevanta avsnitt i kurslitteraturen. Overheadbilderna från föreläsningarna kommer även att finnas tillgängliga här och de läggs normalt upp senast samma dag som föreläsningen genomförs. Planeringen är preliminär.

Föreläsning 1 (16/3)
Innehåll: Kursintroduktion. Nätverksbegrepp. Protokoll. Meddelandesändning med UDP. Multicast.

Java Network Programming: 1-45, 150-176, 423-457, 470-492

- Overheadbilder - en bild per sida
- Overheadbilder - fyra bilder per sida
- Exempelprogram 1: Lookup.java
- Exempelprogram 2: Sender.java Receiver.java
- Exempelprogram 3: MCSender.java MCReader.java

Föreläsning 2 (17/3)
Innehåll: Java I/O - strömmar. Meddelandesändning med TCP. Icke-blockerande I/O.

Java Network Programming: 67-104, 275-341, 384-422

- Overheadbilder - en bild per sida
- Overheadbilder - fyra bilder per sida

Figur 1: Exempelwebbsida

http://cs.lth.se/kurs/eda095/tidigare_ar/	Tidigare år
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/f1-1x1.pdf	Overheadbild...
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/f1-2x2.pdf	Overheadbild...
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/Lookup.java	Lookup.java
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/Sender.java	Sender.java
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/Receiver.java	Receiver.java
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/MCSender.java	MCSender.java
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/MCReader.java	MCReader.java
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/f2-1x1.pdf	Overheadbild...
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/f2-2x2.pdf	Overheadbild...
http://fileadmin.cs.lth.se/cs/Education/EDA095/2011/lectures/ExerciseTailset.java	ExerciseTailset.java

Figur 2: Resultatet av DownloadHelpers länkanalys

Uppgifter

1. En enkeltrådad version av Neds program.

Du ska hjälpa Ned med att skriva en enkeltrådad version av hans program. Indata till programmet ges som argument på kommandoraden och ska bestå av en URL och en katalog på er lokal dator i vilken ni vill lagra filerna.

- a) Beskriv hur man får tillgång till en HTML parser i Java. Du behöver inte skriva koden – det räcker med en kort beskrivning.

(1p)

- b) Skriv en parser-callback som extraherar alla länkar från en vis URL och lagrar dem i en lista. Kalla klassen LinkExtractor.

(4p)

- c) Skriv ett huvudprogram som skapar ett LinkExtractor-objekt, extraherar alla länkar, filtrerar ut PDF-filerna ur URLerna som hämtas från LinkExtractor-objektet och laddar ner filerna i den angivna katalogen¹.

(5p)

2. En multitrådad version av Neds program.

Du ska hjälpa Ned med att skriva en multitrådad version av programmet. Indata till programmet i form av kommandoradsargument: URL, mapp på lokal dator och antal trådar.

- a) Varför kan det vara intressant att skriva en multitrådad version av nedladdningsprogrammet, dvs göra så att flera nedladdningar görs parallellt?

(1p)

- b) Skriv en klass Runner som ärver från Javas Thread-klass och som kapslar in koden för att ladda ner en PDF och sedan skriva den till den lokala katalogen. Tänk på att två Runner-objekt inte ska ladda ner samma PDF-fil (vi förutsätter dock att en fil bara refereras en gång per sida) och att samtliga trådar ska fortsätta ladda ner nya PDF-er så länge arbetet inte är slutfört (jobbalansering).

(5p)

- c) Ändra huvudprogrammet, och om nödvändigt andra delar av ditt program, så att det skapar det givna antalet Runner-objekt och laddar ner PDF-filerna parallellt.

(4p)

Glad sommar!

¹ Du kan skapa en ny fil på skivminnet genom att t.ex. använda en FileOutputStream.
Exempel: `file = new FileOutputStream("filename.pdf");`