디렉토리 관리 기능을 추가한 자바 웹서버의 설계 및 구현

김혜연*, 황효선, 장도임, 조동섭
이화여자 대학교 컴퓨터학과

Design and Implementation of Java Web Server added to Directory Management

Hye-Yeon Kim*, Hyo-Sun Hwang, Do-Im Chang, Dong-Sub Cho
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

기존의 웹서버에는 디렉토리 관리를 위한 기능이 없다. 즉 서버의 파일 시스템에 디렉토리를 만들고, 지우고 다른 곳으로 옮기는 등의 기능을 웹서버만으로는 수행할 수 없다. 이러한 기능을 수행하기 위해서는 ftp 서버와 같은 프로그램을 서버 시스템에 설치하여 ftp클라이언트 프로그램에 의해 이 일을 수행할 수 있는데 이는 서버 시스템의 부하를 증대시키는 요인이 될 수 있다. 그러므로 웹서버에 디렉토리 관리에 대한 기능을 추가하여 추가적인 프로그램 없이 간단하게 디렉토리 관리를 할 수 있도록 하여 유용성을 확장할 수 있다. 이러한 기능은 강의 및 숙제제출을 위해 파일 업로드 및 다운로드가 빈번하게 일어나는 특징이 있는 인터넷 강의 시스템에 적합하다. 또한 동시에 많은 사용자들이 접속할 수 있어야 하는 인터넷 강의를 위해 서버의 부담이 크지 않은 웹서버가 필요하다.

본 논문에서는 이러한 요구를 반영해서 디렉토리 관리 기능을 추가한 웹서버를 제시하였다. 웹서버에 파일처리를 위한 디렉토리서버를 추가하여 보다 빠르게 파일처리를 할 수 있도록 하고 아파치 같은 기존의 웹서버보다 시스템 부하를 줄여 동시에 많은 사용자들의 요청을 받아들일 수 있도록 하는 서버를 구축하였다.

1. 서 론

정보 통신의 발달로 시간과 장소의 한계가 극복 가능해짐에 따라 웹브라우저만 가지고 인터넷으로 강의를 들을 수 있는 인터넷 강의에 대한 관심이 급증하고 있다. 이러한 인터넷 강의는 강의 및 숙제제출을 위해 파일 업로드 및 다운로드가 빈번하게 일어난다는 특징이 있으며 동시에 많은 사용자들이 접속을 원활하게 처리할 수 있어야 한다.

그러므로 빈번하게 일어나는 파일 관리를 잘 하는 것은 서버의 성능을 개선시킬 수 있는 중요한 요소가 될 수 있다. 파일 관리 기능을 위해서는 ftp서버 및 클라이언트 같은 별도의 프로그램이 필요하다. 이것은 서버의 부담을 증대시킬 수 있는데 많은 사용자의 요구를 한꺼번에 처리해야 하는 인터넷 강의 시스템의 경우 서버 시스템의 부하를 감소시키는 것이 중요하다. 그러므로 기존의 웹서버에 디렉토리 관리를 위한 기능을 추가하여 파일 및 디렉토리 관리 시 서버의 부하를 줄일 수 있도록 디자인하는 것이 유용하다고 할 수 있다.

본 논문에서는 서버의 부담을 줄이기 위해 디렉토리 관리 기능을 추가한 웹 서버를 설계하고 이를 구현하여 웹서버의 성능을 향상시켰다.

본 논문의 구성은 다음과 같다. 2장 본론에서는 배경 지식 및 웹 서버에 대한 설계 및 구현에 대해 설명하고 3장에서는 결론 및 향후 연구를 보인다.

2. 본 론

2.1 쓰레드

쓰레드란 실행중인 프로그램 내에서 독립적으로 수행되는 하나의 순차적인 제어의 흐름이다. 클라이언트 서버 프로램과 같이 독립적인 일을 병행 처리해야 하는 경우에 C프로그램 등을 사용하여 여러 개의 다중 프로세스로 프로그램을 구현하는 것이 일반적이었다. 그런데 자바에서는 이를 다중 쓰레드를 이용하여 구현할 수가 있다.

프로세스가 서로 완전히 독립적으로 수행되는 것인데 비해 쓰레드들은 변수나 파일 같은 리소스들을 서로 공유하면서 프로그램의 작은 부분을 독립적으로 실행한다.

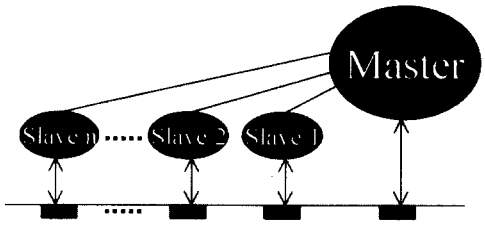
한 프로그램 내의 각각의 프로세스는 상태 값을 공유하며 상호 밀접한 관계를 가지고 동작하여야 할 필요가 있다. 특히 여러 프로세스를 만들고 이들을 총괄적으로 관리하는 것이 필요한 경우가 많은데, 이런 경우 쓰레드를 만들어 사용하면 프로그래밍 하기가 쉬울 뿐 아니라 실행 속도 개선 및 빠른 반응 시간을 사용자에게 제공할 수 있다.

2.2 웹 서버

일반적인 웹 서버는 여러 개의 클라이언트의 요구를 동시에 처리하기 위해서, 사용자의 연결 요청이 있으면 새로운 프로세스를 생성하고 그 프로세스가 클라이언트와의 연결을 유지하면서 사용자의 요구를 처리한다. 즉 웹서버는 여러 개의 독립적인 프로세스가 각각 하나의 클라이언트의 요청에 대한 수행을 전담한다. 이러한 클라이언트 서버 구조를 병렬 연결형 서버라고 한다. [3,4]

[그림 1] 은 이러한 서버와 클라이언트를 나타낸 것이다.

본 논문은 99년도 정보통신부 산·학·연 공동 기술 개발의 사업의 지원을 받았음



[그림 1] 병행 연결형 서버

본 논문에서 제안하는 웹서버는 사용자의 연결 요청을 받아들여 프로세스를 생성하는 대신 쓰레드를 생성하여 사용자와의 연결을 유지하여 서버의 부하를 줄였다. 프로세스는 프로그램 전체를 생성하는 반면에 쓰레드는 필요한 작은 부분만을 생성하기 때문에 프로세스보다 생성 속도가 빠르고 실행속도도 빠르다.

2.3 파일 및 디렉토리 관리 기능

빈번한 파일 업로드 및 다운로드가 일어나기 때문에 성능 향상을 위해 웹서버에 파일관리 기능을 추가하였다.
서버 시스템의 파일 및 디렉토리를 원격으로 관리한다. 즉 서버의 파일 관리 기능(delete, move, access, refresh 등) 을 웹브라우저만 있으면 어느 곳에서도 수행할 수 있다.

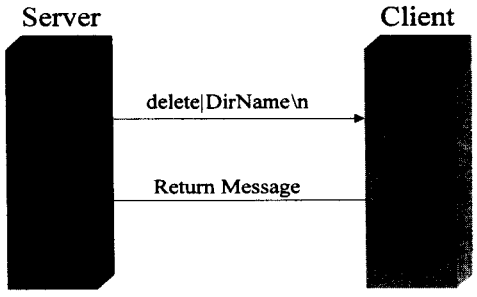
2.3.1 파일 관리의 기능

구현된 파일 관리 기능은 다음과 같다.

1) refresh - 현재 디렉토리 밑의 파일 리스트를 보여준다.
2) access - 특정 디렉토리 밑의 파일 리스트를 보여준다.
3) delete - 특정 디렉토리 및 파일을 지운다. 단 디렉토리가 비어있지 않으면 지울 수 없다.
4) make - 디렉토리를 만들어 준다.
5) move - 디렉토리를 다른 경로로 옮겨준다.

2.3.2 프로토콜

프로토콜은 서버와 클라이언트가 통신하는 규약이다. 디렉토리 관리서버와 클라이언트와의 프로토콜은 [그림 2]와 같다.



[그림 2] 디렉토리 관리 서버와 클라이언트와의 통신

서버에서 클라이언트로 리턴해 주는 Return Message는 [표 1], [표 2]와 같다.

delete,move,make 성공		delete,move,make 실패
현재디렉토리 refresh 성공	현재디렉토리 refresh 실패	"0\n에러메시지\n"
"1\n1\n디렉토리 리스트/end\n"	"1\n0\n에러메시지\n"	

[표 1] delete, add, move 일 때의 Return Message

add, refresh 성공	add, refresh 실패
"1\n디렉토리리스트/end"	"0\n에러메시지\n"

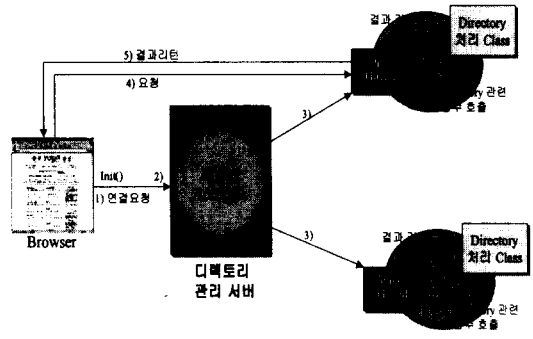
[표 2] add, refresh일 때의 Return Message

delete, move, make는 수행 후 현재 디렉토리 리스트를 서버로 전송한다.(refresh 기능 수행)

2.3.3 클라이언트와 서버의 수행 과정

[그림 3]은 클라이언트로부터 요청이 들어왔을 때 디렉토리 관리서버가 요청을 처리하는 과정을 보여준다.

1) Client는 Socket을 생성한 후 디렉토리 관리 서버에 연결을 시도한다.
2) 디렉토리 관리 서버(AppletAcceptor) 가 client의 연결을 받아들인다.
3) AppletAcceptor는 실제적으로 파일 관련된 일을 전담할 쓰레드를 생성한 후 그 쓰레드와 클라이언트와의 연결을 설정한다. (AppletThread라는 쓰레드가 이 일을 담당한다.)
4) AppletThread는 클라이언트의 요청이 있으면 해당하는 method를 호출하는 것에 의해 파일 관련된 일 (delete, move, access, refresh 등)을 수행함.
5) 성공인지 실패인지와 그에 따른 리턴 메시지를 브라우저에게 반환
6) 클라이언트 프로그램이 종료될 때까지 4) - 5)를 반복해서 수행함

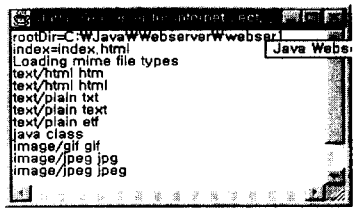


[그림 3] 클라이언트로부터 요청을 받았을 때 서버의 개념도

2.3.4 구현

클라이언트 프로그램은 웹 브라우저에 의해 수행될 수 있는 애플릿으로 구현했고, 디렉토리 관리서버는 Java의 쓰레드를 사용하여 구현하였다. 서버와 클라이언트는 소켓 통신으로 메시지를 주고받는다. 쓰레드는 프로세스보다 생성되는 크기가 작기 때문에 서버의 부하도 적다. 그렇기 때문에 쓰레드로 구현한 디렉토리 관리 서버는 클라이언트가 요청했을 때 응답시간 및 수행시간이 빠르다.[1,2,5]

[그림 4] 는 웹서버가 실행되었을 때의 화면이다.
웹서버가 시작되면 디렉토리 관리 서버도 같이 시작된다.



[그림 4] 웹서버의 실행화면

다음은 서버 쪽 프로그램의 주된 클래스이다.

1) class AppletAcceptor extends Thread
이 클래스는 디렉토리 관리 서버이다. 쓰레드를 사용해서 만들어졌으며 웹 서버가 살아있는 한 메모리에 상주하여 사용자의 요청을 기다리고 있다.
클라이언트로부터 연결 요청이 오면 받아들이고 그 일을 전담하는 쓰레드를 할당해 준다.

2) class AppletThread extends Thread
쓰레드를 상속받아 만들어진 클래스이다. 클라이언트와의 연결을 유지하고 있는 동안에 클라이언트로부터 요청이 들어오면 해당하는 일을 해주는 쓰레드이다. 클라이언트 프로그램이 종료되면 이 쓰레드는 자동 종료된다.

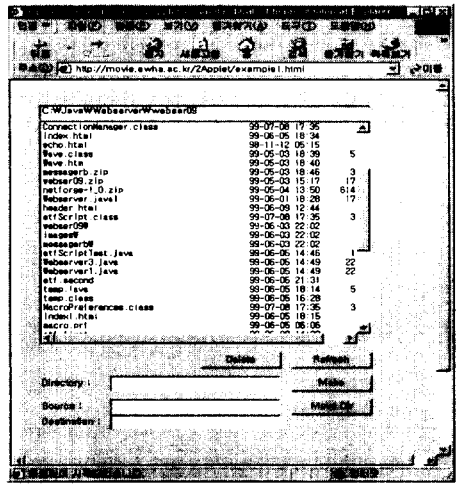
3) class DirAction
이 클래스는 디렉토리 관리에 대한 일을 전담하는 함수이다. 이 클래스의 주된 함수에는 access(), refresh(), delete(), move(), add()등이 있다. 이 함수들이 실제적으로 서버 시스템의 디렉토리 관리에 대한 일을 수행한다.

클라이언트가 디렉토리 처리에 대한 요청을 하면 AppletThread는 DirAction클래스를 생성하고 DirAction클래스의 함수들을 호출하는 것에 의해 실제적인 기능을 수행한다.

[그림 5]는 파일 처리를 위한 애플릿이 실행되고 있는 웹브라우저의 화면이다.
애플릿은 자바의 AWT 및 Swing API를 사용하여 구현하였다[6]. 애플릿이 시작될 때 서버와의 연결이 설정되며 프로그램이 끝날 때 연결이 끊어진다. 각 버튼을 누를 때마다 디렉토리 처리를 위한 해당 메시지가 서버로 전송된다.

다음은 클라이언트 프로그램이 시작될 때 초기화를 위해 가장 먼저 불려지는 함수인 init()의 수행 과정이다.

- 애플릿을 초기화한다.
- 소켓을 생성해서 디렉토리 관리 서버와 연결을 시도한다.
- 디렉토리 관리 서버가 연결요청을 받아들이면 클라이언트와 연결이 설정된다.
- 서버는 루트 디렉토리 리스트를 클라이언트한테 넘겨주면 클라이언트는 애플릿의 List에 그 내용을 보여준다.



[그림 5] 서버의 파일 관리를 위한 애플릿

3. 결론 및 향후 연구

인터넷 강의 시스템에 적용될 수 있는 디렉토리 관리 기능을 추가한 웹서버를 설계 및 구현하였다. 파일 업로드 및 다운로드가 많은 인터넷 강의의 특성을 고려하여, 웹서버에 파일 및 디렉토리 관리를 전담하는 서버(디렉토리관리서버)를 따로 유지함으로서 사용자에게 일관된 인터페이스를 제공하고 시스템의 부하를 줄이게 되었다. 웹서버 및 디렉토리관리서버는 자바의 쓰레드를 사용하여 사용자의 연결을 받아들이기 때문에 일반적인 웹서버보다 연결 요청 시의 반응 속도 및 실행 속도가 빠르다.

이러한 서버/클라이언트 구조는 애플릿의 제약사항이었던 서버의 디렉토리 엑세스를 가능하게 하였다. 앞으로 확장된 자바 웹 서버와 강력한 사용자 인터페이스를 제공하는 애플릿을 결합하여 보다 다양한 기능을 인터넷 환경에서 제공할 수 있게 확장할 것이다. 그리고 현재 구현된 시스템에는 디렉토리 관리를 할 수 있는 사용자가 따로 정해져 있지 않기 때문에 보안상의 문제가 있다. 그러므로 서버에 디렉토리마다 보안 등급을 설정해 등급에 따라 권한을 다르게 처리할 수 있는 보안관리 기능을 확장하려 한다.

[참 고 문 헌]

[1] Steve Potts, " Java 1.2 How-To" , Earthweb press, 1998
[2] Mike Cohn외 5인, " Developer's reference JAVA ", Sams.net, 1997
[3] Merlin and Conrad Hughes, Michael Shoffner, Maria Winslow, " Java Network Programming ", Manning , 1996
[4] Elliotte Rusty Harold, "Java Network Programming", O'REILLY, 1997
[5] Cay S. Horstmann, Gary Cornell, "Core Java 1.2 Volume I-Fundamentals ", Sun Microsystems, 1999
[6] David M. Geary, "graphic JAVA 1.2 Mastering the JFC", Sun Microsystems, 1999