

RemoteFinder: Eine graphische Oberfläche für SCP



Max Berger <max@berger.name>

20. Mai 2003

Übersicht

- Einführung
- Anforderungen
- Systementscheidungen
- Systementwurf
- Implementierung

Übersicht (cont.)

- Diskussion
- Fehlerbehandlung
- Schlussbemerkungen

Einführung

- Was ist SCP?
- Command-Line SCP
- SFTP
- Fugu

Was ist SCP?

- SSH: SecureShell
- SCP ist ein Subsystem von SSH
- Gedacht als sicherer Ersatz von rcp
- Kein festgeschriebener Standard

Command-Line SCP

- Setzt UNIX-Kenntnisse voraus
- Arbeitet auf Basis der Kommandozeile
- Im Normalfall 2 Terminals nötig: für Anzeige (SSH) und Kopieren (SCP)
- Kopieren von mehreren Dateien ziemlich kompliziert

SFTP

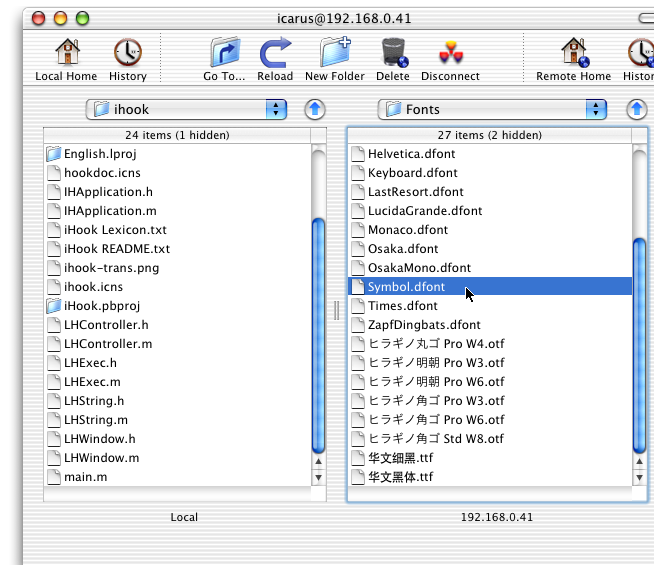
- SFTP ist ein Subsystem von SSH
- Soll zu FTP vergleichbare Funktionen bereitstellen

Aber:

- Namensverwechslung mit FTP/S
- Nicht so weit verbreitet wie SCP

Fugu

- Cocoa-Oberfläche für SFTP
- Projekt an der University of Michigan
- Oberfläche orientiert sich an bekannten FTP Clients
- SCP Unterstützung gegeben, aber sehr schlecht.



Anforderungen

- Funktionale
- Nicht-Funktionale

Funktionale

- Mehrere Verbindungen
- Kopieren sollte durchsuchen nicht unterbrechen
- Abgleich mit OpenSSH Konfiguration
- SSH-Protokoll
- Zukunft: Weitere Protokolle: FTP, SFTP, FTP/S, WebDAV

Nicht-Funktionale

- Benutzen einer bestehenden, erprobten SSH Implementation
- Mac OS X Design und Handling
- Cocoa Oberfläche
- Keine Abhängigkeit von externen Programmen
- Möglichst wenig Anforderungen an das Serversystem

Systementscheidungen

- Programmierumgebung
- Oberfläche
- SSH-Kern

Programmierungsumgebung

Wahl zwischen:

- Cocoa: Objektorientierter Ansatz, OpenStep Nachfolger
- Carbon: Klassischer C-Ansatz

und zwischen:

- Java: Plattform-unabhängig, aber nicht mehr mit Cocoa-Oberfläche, langsam
- ObjC: Native Sprache für Cocoa

Oberfläche

- Mac Benutzer!
- Keine Unix-Kenntnisse
- Do-what-I-mean

SSH-Kern

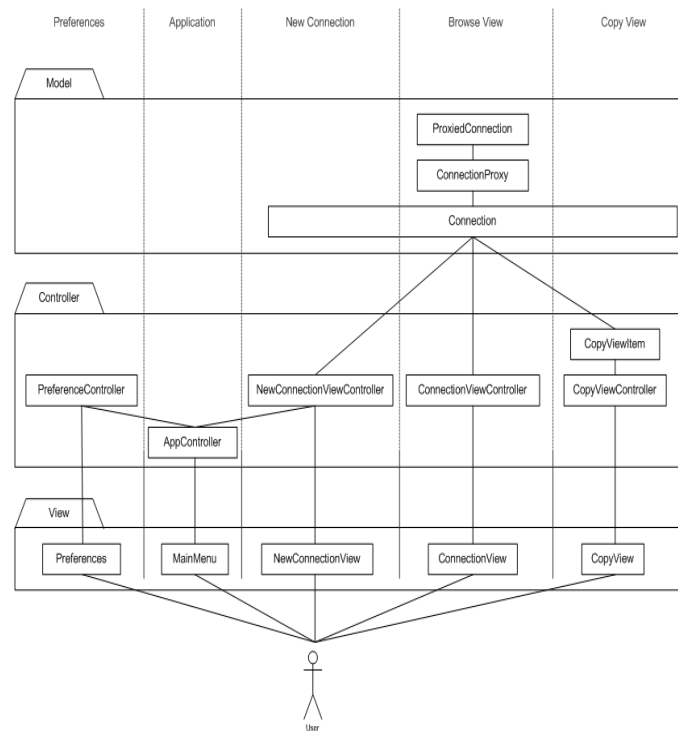
Wahl zwischen:

- OpenSSH: Quellcode auf Sicherheit, nicht Lesbarkeit optimiert
- FreeSSH: SSH-Bibliothek, seit 2001 nicht mehr weiterentwickelt
- LSH: Bisher nur SSH 2 Protokoll
- Putty: Läuft, bietet gutes Callback, wird in anderen Projekten verwendet

Systementwurf

- Klassentwurf
- Modell
- View
- Controller
- Ablaufbeispiel

Klassentwurf



Modell

- Connection: Hält Informationen über Verbindung
- ConnectionProxy: Sorgt für sauberes Multithreading
- ProxiedConnection: Tatsächliche Verbindung
- PLinkConnection: Implementation von ProxiedConnection
- PLinkConnectionManager: Sammelt PLinkConnections

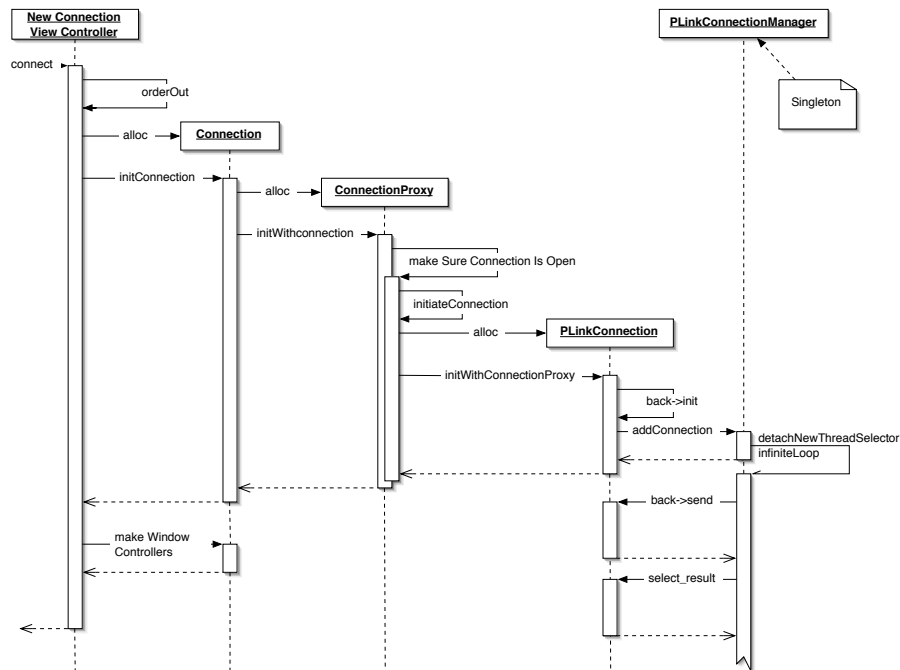
View

- Nahezu vollständig im ProjectBuilder erstellt
- Zu jedem Fenster gibt es mindestens eine Kontrollklasse

Controller

- PreferenceController: Allgemeine Einstellungen
- ApplicationController: Menu und Applikationssteuerung
- NewConnectionViewController: Neue Verbindung und Session Management
- ConnectionViewController: Browsing
- CopyViewController: Liste von CopyViewItems
- CopyViewItem: Informationen über Kopiervorgänge

Ablaufbeispiel



Implementierung

- Putty-Kern
- File-Promises

Putty-Kern

Bis auf wenige Punkte gute Wahl, aber:

- Variablen mussten umbenannt werden wegen Namenskonflikt
- Callback nicht immer konsistent
- Putty macht eigene Verbindungsverwaltung

File-Promises

- Neu in 10.2
- Für Dateien, die noch gar nicht existieren
- Benutzt undokumentierte Typen in Zwischenablage für Callback
- Kann nur aus `mousedown:` oder `mouseDragged:` aufgerufen werden.

Diskussion

- Geschwindigkeitsmessung
- Performance
- NetCat Modus

Geschwindigkeitsmessung

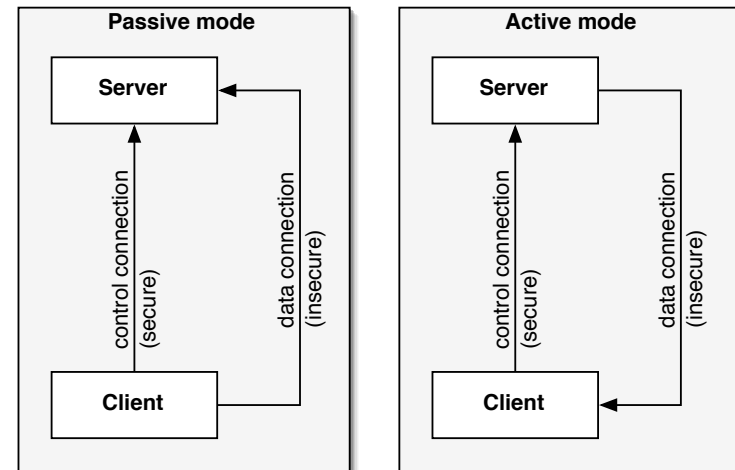
- Problem: Übertragungsgeschwindigkeit variiert sehr stark
- Traditionelle Methode (mit History) aufwendig
- Lösung: Adaptiver Tiefpassfilter $A_n = A_{n-1} + \varepsilon(R_n - A_{n-1})$

Performance

Programm	Upload	Download
FTP	8800	8800
Samba	7500	7500
SCP	5600	5600
Fugu	4600	4000
RemoteFinder	3600	2800

NetCat Modus

- geplant!
- angelehnt an FTP
- Wesentliche höhere Geschwindigkeit
- Daten sind unverschlüsselt, aber
- Passwörter und Verzeichnisse weiterhin verschlüsselt!



Fehlerbehandlung

- Sicherheitsfehler
- Verbindungsfehler
- Dateizugriffsfehler
- Programmfehler

Sicherheitsfehler

SSH-Serverschlüssel:

- Wird Angezeigt
- Dem Benutzer wird mitgeteilt, wie er diesen überprüfen kann
- Wird auf Wunsch gespeichert oder auch nicht.

Verbindungsfehler

- z.B. Netzkabel wird gezogen, Server rebootet
- Keine explizite Fehlerbehandlung
- Alle Fenster dieser Verbindung bleiben hängen, andere Verbindungen sind davon nicht betroffen

Dateizugriffsfehler

- z.B. keine Lese-rechte, keine Schreib-rechte, keine normale Datei
- Fehler werden in Fehlerprotokoll gesammelt
- Benutzer muss Fehlerprotokoll bestätigen, bevor er mit dieser Verbindung weiterarbeitet
- Andere Verbindungen nicht betroffen.

Programmfehler

- z.b. Segmentation-Fault oder allgemeine Schutzverletzung
- Häufigste Fehlerquelle: Null-Zeiger
- Seltener durch Verwaltung von ObjC, aber nicht ausgeschlossen
- Programmabsturz

Schlussbemerkungen

- Installation
- Neue Features
- Code Cleanup
- Download

Installation

- Am Server: Keine!
- Am Client: All-In-One Binary, optional in Applications Order und Dock ziehen

Neue Features

- Up-Download von Verzeichnissen
- Passwort-loses Einloggen (KeyChain, SSH-Key)
- Andere Protokolle
- Verschieben, Umbenennen

Code Cleanup

- Klassen wurden zu groß
- Teilweise: Verletzung des Schichtmodells
- Klassen haben nicht mehr die Bedeutung des Namens
- Klassennamen verwirrend / falsch

Download

RemoteFinder ist verfügbar unter <http://remotefinder.sf.net>