

Rechnernetze u. Netzwerkprogrammierung

Übungsblatt 7, 14.05.2012

Das Projekt dieses Semesters wird in Gruppen von **3 Personen** (+/-1 nach Absprache) bearbeitet. Geben Sie mir die **Gruppenzusammensetzung** bitte **bis zum 18.05.2011 per E-Mail** bekannt. Für jede Gruppe wird ein SVN-Repository auf einem FH-Server eingerichtet, das während der Bearbeitung des Projekts zu verwenden ist. Der Projektfortschritt muß mindestens wöchentlich anhand der SVN-Historie nachvollziehbar sein. Achten Sie daher bitte auch auf aussagekräftige Commit-Kommentare. Die Abgabe besteht aus:

1. dem vollständigen, gut nachvollziehbaren (Struktur / Kommentare) Python-Quelltext (alles im SVN) sowie
2. einer schriftlichen Dokumentation der Software (Aufbau, Funktionsweise und "Highlights" Ihres Protokolls und dessen Implementierung), elektronisch als PDF im SVN und unterschrieben auf Papier (Doku incl. Source-Listing, im Sekretariat abzugeben) und
3. einer Vorführung des Programms mit Erläuterung der "Highlights" im Quelltext, bei der jedes Gruppenmitglied Fragen zum gemeinsamen Werk beantworten können muss. Bitte vereinbaren Sie rechtzeitig einen Vorführungstermin bis Vorlesungsende, Termine für Vorführungen können bis zum Ende der ersten Klausurwoche dieses Semesters liegen.

Bewertungsgrundlage ist der Stand des bereitgestellten SVN-Repositories am **Montag, dem 25.06.2011, 18:00 Uhr** (Papier-Abgabe bis 18:29 Uhr dieses Tages im Sekretariat - ggf. Sekretariats-Briefkasten, unterschrieben von allen Gruppenmitgliedern). Eine verspätete Abgabe kostet für jeden angebrochenen Kalendertag zwei Notenstufen (so ähnlich wie im Parkhaus), Tage enden bzw. beginnen jeweils um 18:30 Uhr.

Jeder Abgabe ist eine von allen Bearbeitern unterschriebene Erklärung beizufügen, dass das Projektergebnis vollständig in Eigenleistung erstellt wurde bzw. verwendete Teile aus fremden Arbeiten deutlich als solche gekennzeichnet wurden. Erscheinen übereinstimmende Codeteile in mehreren Abgaben, so wird (Sie ahnen es schon) keine davon gewertet. Das Projekt muß auf der Linux-Installation in den Poolräumen der FH lauffähig sein.

Die Mission:

Jöndhard benötigt eine Datei-Synchronisationslösung, welche ein vorgegebenes Verzeichnis in seinem Home-Directory regelmäßig mit dem gleichen Verzeichnis bei seinen Freunden innerhalb des gleichen LAN abgleicht. Da er Software, die von Nichtmedieninformatikern entwickelt wurde, aus Erfahrung misstraut, bittet er Sie um Hilfe.

Entwickeln Sie bitte ein System wie folgt (das ist ein Rahmen - Details bitte selbst ausgestalten):

- Beim Start der zu entwickelnden Sync-Software „FloppBox“ wird ein String (Gruppenname) als Kommandozeilenparameter mitgegeben. Das ist gleichzeitig der Name des zu synchronisierenden Verzeichnisses (Unterverzeichnis des Startverzeichnisses). Existiert ein solches Sync-Verzeichnis noch nicht, bricht die FloppBox ab. Im LAN werden ab dann Veränderungen bei Dateien und Ordnern unter dem Sync-Verzeichnis zwischen FloppBoxen, die mit dem gleichen Gruppennamen gestartet wurden, synchron gehalten.
- Eine Veränderung im überwachten Dateibestand eines Rechners wird auf den anderen Rechnern nachgezogen (mindestens das Anlegen Dateien und Unterverzeichnishierarchien sowie Modifizieren von Dateien; schön (aber trickreicher) wäre auch noch „löschen“).

- Sollten in einem Überwachungsintervall auf mehreren FloppBoxen Modifikationen entdeckt werden, wird die jeweils neueste an alle anderen teilnehmenden Rechner verteilt. Eine (laut Zeitstempel) neuere Datei wird also nicht von einer älteren überschrieben. Gibt es unterschiedliche Dateien mit identischem Zeitstempel, wählt das System eine beliebige - Hauptsache, die Teilnehmer haben zeitnah die gleiche Datei vorliegen.
- Stets ist zu beachten, effizient zu arbeiten und unnötigen Rechen- und Kommunikationsaufwand zu vermeiden.
- Überlegen Sie sich bitte vorab mögliche Konfliktfälle und wie Sie damit umgehen wollen. Beispiel: Wenn ein neuer Teilnehmer mit noch leerem Sync-Verzeichnis hinzukommt, dürfen die anderen *nicht* aus Solidarität ihre Sync-Verzeichnisse ebenfalls löschen, denn das wäre eine potentielle Quelle großen Verdresses für dessen SyncGruppen-Mitglieder.
- Achten Sie auf einfache Nutzbarkeit und Robustheit (z.B.: Findet eine neue FloppBox "automatisch" einen Nachbarn und syncnt mit ihm ohne manuelle Konfiguration? Gibt es einen zentralen Agenten, dessen Ausfall das Funktionieren des Gesamtsystems in Frage stellt oder organisieren die Agenten sich "intelligent" untereinander? Was passiert bei zeitweisen Verbindungsunterbrechungen / Abbruch einer FloppBox?)
- Jeder SyncAgent loggt seine Entscheidungen Aktivitäten in nachvollziehbarer Weise in eine Datei „<gruppenname>.log“, die im Startverzeichnis der FloppBox liegt.

Sie können sowohl TCP als auch UDP zur Bearbeitung der Aufgabe einsetzen. Kommunikation zwischen den FloppBoxen erfolgt ausschließlich über selbstgeschriebenen Code auf Basis des socket-Bibliotheksmoduls (keine HighLevel-Python-Module für HTTP/FTP/...) zwischen von Ihnen selbstgeschriebenen Komponenten (kein externer HTTP/FTP/...-Server o.ä.). Ansonsten können die Module der Python-Standardbibliothek verwendet werden (keine anderen).

Hinweise:

- Die Python-Bibliotheksfunktion `os.path.getmtime()` erlauben das Abfragen der Modifikationszeit einer Datei oder eines Verzeichnisses als Zahl (Zeitstempel, Sekunden seit Beginn der Epoche), `os.path.getatime()` analog für die letzte (auch Lese-) Zugriffszeit.
- Das Anlegen/Löschen von Unterverzeichnissen oder Dateien in einem Verzeichnis ändert das Modifikationsdatum dieses übergeordneten Verzeichnisses (nutzbar zur Erkennung von Löschungen). Das Verändern einer Datei ändert nicht den Zeitstempel deren Ordner.
- Die Python-Bibliotheksfunktion `os.utime()` erlaubt das Setzen von Zugriffs- und Modifikationszeit einer Datei oder eines Ordners.
- Die Python-`hashlib` stellt eine Reihe von (starken) Hashfunktionen bereit. Wenn der (begrenzt lange) Hashwert zweier (im Prinzip beliebig langen) Dateiinhalte übereinstimmt, ist es für unsere Zwecke ausreichend wahrscheinlich, dass die Bytefolgen ebenfalls übereinstimmen.