

AKADEMIA GÓRNICZO-HUTNICZA

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki



KATEDRA INFORMATYKI

SyncFile

Document wizyjny dla projektu SyncFile

Wersja 0.1-5 z dnia 2006-11-14

Kierunek, rok studiów:

Informatyka, IV rok

Przedmiot:

Grupa (projekt):

Rład!

Niniejsze opracowanie powstało w trakcie i jako rezultat zajęć dydaktycznych z przedmiotu wymienionego na stronie tytułowej, prowadzonych w Akademii Górniczo-Hutniczej w Krakowie (AGH) przez osobę (osoby) wymienioną (wymienione) po słowach „Prowadzący zajęcia” i nie może być wykorzystywane w jakikolwiek sposób i do jakichkolwiek celów, w całości lub części, w szczególności publikowane w jakikolwiek sposób i w jakiegokolwiek formie, bez uzyskania uprzedniej, pisemnej zgody tej osoby (tych osób) lub odpowiednich władz AGH.

Copyright © 2006 Akademia Górniczo-Hutnicza (AGH) w Krakowie

1. Spis treści

1.	Spis treści	2
2.	Opis problemu	3
3.	Cel produktu	3
4.	Funkcjonalność	4
4.1.	Klient	4
4.2.	Komunikacja klient-sewer	4
4.3.	Interfejs użytkownika	4
5.	Architektura systemu	5
5.1.	Komponenty	5
5.2.	Sekwencja działania	6

2. Opis problemu

Problemem w dobie rozwoju komunikacji jest zachowanie spójności i integralności przechowywanych danych. Wraz rozwojem internetu praktycznie zniknęły ograniczenia związane z dostępem do danych, jednak ze względu na rozproszenie powstał problem synchronizacji.

Najlepszym sposobem na zachowanie spójności jest przechowywanie danych na centralnym serwerze, do którego dostęp jest synchronizowany uniemożliwiając równoczesną edycję współdzielonych danych. Takie podejście ma zasadniczą wadę – ograniczona szybkość przesyłania pakietów przez sieć oraz związane z tym opóźnienia.

Innym rozwiązaniem jest przechowywanie kopii danych z serwera w lokalnym systemie plików. Znika wtedy problem opóźnień, jednak na użytkownika nakładany jest obowiązek ręcznej synchronizacji plików z użyciem różnego rodzaju narzędzi.

3. Cel produktu

Produkt ma za zadanie poprawić komfort pracy w środowisku rozproszonym, w którym występuje konieczność zachowania spójności danych znajdujących się na różnych urządzeniach. Jest on dedykowany dla zwykłego użytkownika, w związku z tym ma zapewnić możliwie intuicyjny interfejs połączony z funkcjonalnością wymagającą minimum czynności administracyjnych (tj. konieczność ręcznego wymuszania synchronizacji danych z serwerem).

Ze względu na istniejące rozwiązania wspierające prace programistów związane z utrzymaniem kodu, takie jak CVS, SourceSafe, naszym celem nie jest produkt zapewniający podobną funkcjonalność. Jest to związane z faktem, że w tego typu przedsięwzięciach nie zalecana jest automatyczna synchronizacja źródeł tworzonej aplikacji – takie działania powinny się odbywać na żądanie i pod kontrolą programisty.

4. Funkcjonalność

4.1. Klient

- Monitorowanie wywołań funkcji systemowych WIN API związanych z operacjami na plikach (otwieranie i zamykanie) – ładowanie biblioteki *hook.dll* do przestrzeni adresowej jądra systemu.
- Rozproszona kontrola otwieranych plików – przy wystąpieniu żądania otwarcia monitorowanego pliku (przechwytywanie wywołań WIN API) wysyłane jest prośba o pozwolenie na otwarcie pliku do centralnego serwera oraz blokowany jest dostęp do pliku dla pozostałych użytkowników – rozproszone zajmowanie zasobów.
- Plik otwarty przez innego użytkownika na czas edycji jest *lockowany* uniemożliwiając dostęp do niego z innych programów na pozostałych komputerach.
- Automatyczna synchronizacja plików, na których zakończona została edycja oraz zostały wprowadzone zmiany (przechwytywanie wywołania systemowego zamykania pliku). Wysyłanie zawartości pliku na serwer.
- Po stronie klienta w rejestrze systemowym przechowywana jest lista plików, których zawartość klient zamierza synchronizować

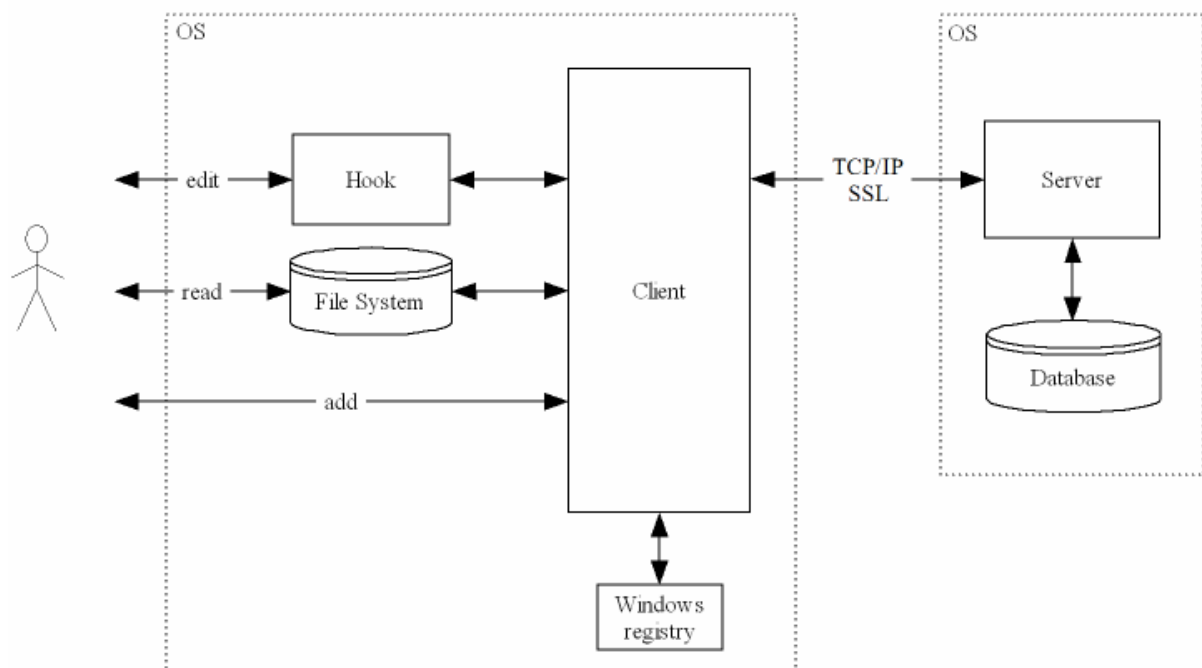
4.2. Komunikacja klient-serwer

- Komunikacja z wykorzystaniem TCP/IP
- Wykorzystanie szyfrowania połączeń na warstwie transportowej - SSL
- Serwer jako samodzielnie działająca aplikacja.
- Do klientów propagowane są tylko różnice między plikiem oryginalnym a zmodyfikowanym – ograniczenie obciążenia sieci.
- Ograniczony dostęp do repozytorium dla użytkowników – wykorzystanie login/hasło
- Przechowywanie danych o wersjach w bazie danych.

4.3. Interfejs użytkownika

- Możliwość subskrypcji na wybrane pliki, wypisanie się z subskrypcji.
- Przenoszenie monitorowanych plików w obrębie dysku.
- Dodawanie plików do repozytorium oraz rozpoczęcie synchronizacji na dysku lokalnym.
- Kasowanie plików ze zdalnego repotorium.
- Włączenie/wyłączenie monitorowania zmian.

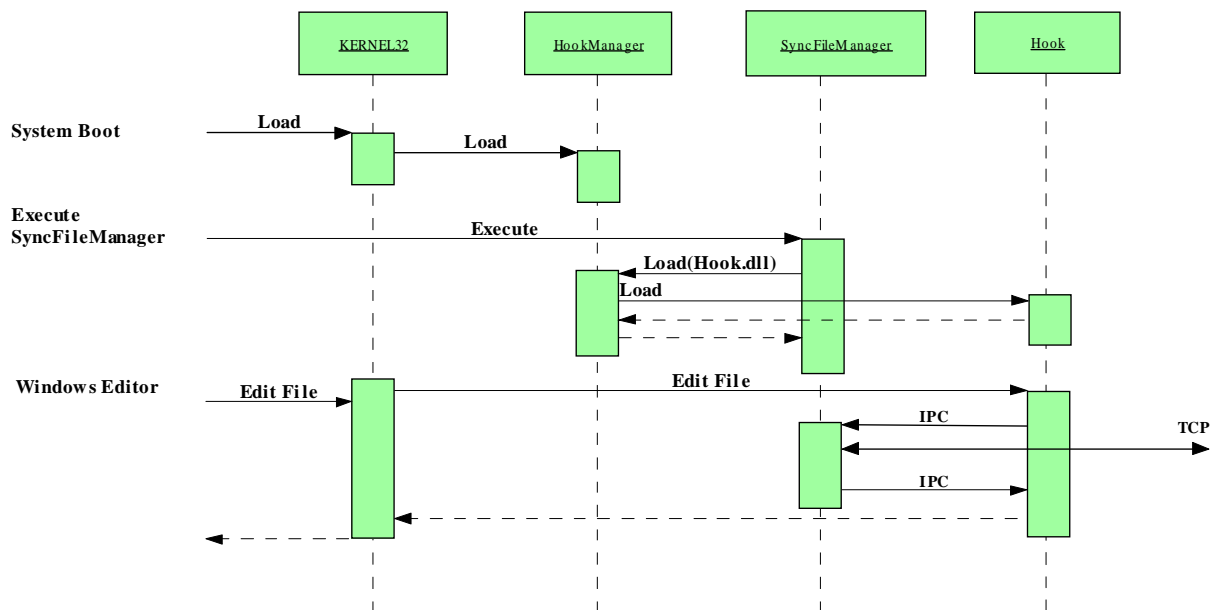
5. Architektura systemu



5.1. Komponenty

- HookManager – biblioteka dll ładowana w czasie startu systemu przez jądro systemu. Jest odpowiedzialna za zarządzanie ładowaniem bibliotek obsługujących żądania obsługi wywołań systemowych. Przez okres działania systemu oczekuje na konieczność załadowania *hook.dll* zleconą przez zewnętrzny program (*SyncFile*).
- Hook – biblioteka dll – biblioteka obsługi wywołań systemowych odpowiedzialna za przechwytywanie operacji otwierania i zamykania pliku oraz odpytanie serwera o pozwolenia
- SyncFile – graficzny interfejs do zarządzania synchronizowanymi plikami, dodanie sybskrypcji na plik, dodanie pliku do repozytorium, zarządzanie aktualizacjami.
- Serwer – centralny punkt systemu obsługujący przechowywanie aktualnych wersji plików oraz zarządzający rozproszonym blokowaniem zasobów.
- Baza danych – odpowiedzialna za niezawodne przechowywanie danych związanych z synchronizacją danych.

5.2. Sekwencja działania



1. Załadowanie biblioteki *HookManager* przez jądro systemu *Kernel32* w czasie startu.
2. Oczekiwanie na żądanie załadowanie biblioteki obsługi wywołań systemowych przez *HookManager*.
3. Start programu *SyncFile* i załadowanie biblioteki obsługi wywołań systemowych *Hook.dll* przez *HookManager*.
4. Oczekiwanie na wywołanie systemowe wykonane przez zewnętrzny program (*Windows Editor*).
5. Wywołanie funkcji WIN API związanej z obsługą pliku – otwieranie/zamykanie – przez *Windows Editor*.
6. Obsługa zadania przez *Hook.dll*. Jeśli plik jest monitorowany przez aplikacje to żądanie jest przekazywane do *SyncFile* z wykorzystaniem mechnizmów systemowej komunikacji międzyprocesowej – WIN IPC.
7. Wysłanie zapytania do serwera przez *SyncFile* z wykorzystaniem kanału TCP/IP.
8. Zezwolenie lub zabronienie dostępu do pliku wysłane do *Hook.dll* z wykorzystaniem IPC.
9. Zezwolenie lub zabronienie otwarcia pliku dla *WindowsManager*.