

Programowanie Aplikacji Sieciowych

Synchronizacja plików

Piotr Korniak

Michał Ochmiński

Oskar Wal

Program automatycznie wykrywa zmiany w folderze, następnie wysyła go na serwer który rozsyła go do pozostałych użytkowników zalogowanych tym samym loginem.

Serwer napisany jest w Pythonie. Loginy i zahashowane hasła przechowywane są w pliku sample.json w formie słownika. Przy uruchamianiu programu dane kopiowane są do słownika, dzięki czemu logowanie polega na porównaniu wartości słownika dla loginu z zahashowanym hasłem wprowadzonym przez użytkownika. Jeżeli w słowniku nie ma podanego loginu, tworzone jest nowe konto i folder o nazwie loginu. Haszowanie odbywa się za pośrednictwem ręcznie zaimplementowanego algorytmu RSA.

Klient łączy się z serwerem, po czym następuje zastąpienie folderu lokalnego użytkownika na folder użytkownika umieszczony na serwerze. W przypadku jakiegokolwiek zmiany w folderze lokalnym użytkownika wysyłana jest informacja o zmianie na serwer, a następnie klient z serwerem łączą się na nowym porcie w celu przesłania swoich plików. Klient wysyła swój folder lokalny, który to jest zamieniany z istniejącym na serwerze. Następnym krokiem jest wysłanie przez serwer do wszystkich podłączonych klientów informacji o tym, że nastąpiła zmiana i wymagana jest aktualizacja. Serwer z klientem łączą się na nowym porcie w celu aktualizacji plików lokalnych. Następnie zamieniany jest folder lokalny użytkownika na folder przesłany z serwera.

Klient po połączeniu wysyła komunikat "Hl\r\n\r\n", na co serwer wysyła informację zwrotną "HELLO\r\n\r\n\r\n"

Następnie serwer na podstawie wysłanego przez klienta loginu i hasła (LOGIN\r\n\r\nUSERNAME\r\n\r\nPASSWORD\r\n\r\n\r\n) rozpoczyna proces weryfikacji. Po pozytywnej weryfikacji, serwer wysyła do klienta port oraz długość pliku (260 __PORT__ __ROZMIAR_PLIKU__\r\n\r\n\r\n). Klient łączy się do socketu serwera na podanym porcie w celu pobrania archiwum w którym znajdują się pliki klienta.

Jeśli klient wykryje zmiany w folderze, po 5 sekundach od ostatniej zmiany wyśle do serwera komunikat o lokalnych zmianach wraz z numerem portu oraz rozmiarze pliku do przesłania (CHANGE\r\n\r\n__PORT__\r\n\r\n__ROZMIAR__\r\n\r\n\r\n). Serwer łączy się do socketu klienta na podanym porcie w celu pobrania archiwum w którym znajdują wszystkie pliki klienta wraz ze zmianami.

Po otrzymaniu nowych plików od klienta serwer rozpoczyna rozsyłanie zmian do wszystkich klientów o tym samym loginie przesyłając im numer portu na którym zostanie stworzony socket (260 __PORT__ __ROZMIAR_PLIKU__\r\n\r\n)

