

POLITECHNIKA WARSZAWSKA

Maciej Spychalski

**Synchronizacja plików i katalogów przy pomocy
RSYNC**

PROJEKT INDYWIDUALNY

wykonany pod kierunkiem
dr inż. Krzysztofa Siwka

Warszawa, 2010r.

Spis treści

1	Słowem wstępu	3
1.1	Założenia projektu	3
2	Programy do synchronizacji	3
2.1	Pakiety serwer/klient	4
2.1.1	Allway Sync	4
2.1.2	GoodSync	5
2.2	Dysk wirtualny	5
2.3	Kompatybilne z rsyncd	6
2.4	Podsumowanie	7
3	Rsync	7
3.1	Parametry	8
3.2	Przykłady użycia	8
3.2.1	lokalnie	9
3.2.2	zdalnie	10
3.2.3	poprzez usługę rsyncd	10
3.2.4	za pomocą SSH	12
4	cwRsync	12
4.1	Pobieranie i instalacja	13
4.2	Instalacja Cygwin’a	14
4.3	Skrypty	15
4.3.1	Wersja pełna/minimalna Cygwina	15
4.3.2	Synchronizacja lokalna	15
4.3.3	Synchronizacja zdalna	17
4.3.4	Poprzez usługę rsyncd	17
4.3.5	Za pomocą SSH	18
4.3.6	Podsumowanie	18
4.4	Ustawianie kluczy w SSH	19
5	Testowanie	20
6	Wnioski	20
	Literatura	21

1 Słowem wstępu

Każdy z nas zapewne staje często przed problemem kopiowania dużej ilości danych, czy to z nośników wymiennych, czy też z drugiego komputera. Czasami jednak jest to kopiowanie zaledwie kilku nowych, bądź zmienionych, plików, tak jak to ma miejsce podczas tworzenia kopii bezpieczeństwa. Nie ma wtedy sensu powielanie całych katalogów, a jedynie tych plików, które są zmodyfikowane lub nie znajdują się na naszych stacjach. Z pomocą przychodzą nam liczne programy, stworzone specjalnie, by ułatwić i zautomatyzować tę żmudną pracę.

Programy do synchronizacji plików to narzędzia, które automatycznie kopiują, przenoszą, nadpisują i usuwają pliki pomiędzy wybranymi folderami. Jest to bardzo przydatna opcja, gdy korzystamy z kilku komputerów, a na każdym z nich chcemy mieć aktualne wersje danych. Synchronizacja polega więc na uaktualnieniu plików we wskazanych folderach do najnowszych dostępnych wersji oraz na selekcji jedynie tych, które takiej aktualizacji wymagają, co znacznie przyspiesza ten proces (więcej w przykładzie w rozdziale 3.2.1). Co więcej, możliwa jest ona w obie strony, także po szyfrowanym kanale.

1.1 Założenia projektu

Celem mojego projektu było zaprojektowanie i oprogramowanie systemu do synchronizacji plików pomiędzy wieloma komputerami. W pierwszej jego części należało wykonać przegląd istniejących rozwiązań zarówno komercyjnych jak i darmowych. Zadaniem praktycznym było oprogramowanie systemu synchronizacji przy użyciu programu rsync lub pokrewnego.

Zadanie polegało na napisaniu odpowiednich plików wsadowych „bat”, które „jednym kliknięciem” pozwolą na synchronizację dowolnych katalogów i plików. Ważny jest tu dokładny opis parametrów skryptu, aby ułatwić konfigurację danych do synchronizacji, lokalizacji i adresu serwera. Dokładna instrukcja tworzenia tych plików oraz działania, znajduje się w rozdziale 4.3.

2 Programy do synchronizacji

Wśród długiej i ciągle rozrastającej się listy oprogramowania do synchronizacji, znaleźć możemy pakiety, które składają się z serwera i klienta, korzystają z internetowego wirtualnego dysku, bądź posiadające program kliencki, do łączenia się z Linuxowym serwerem RSYNC.

2.1 Pakiety serwer/klient

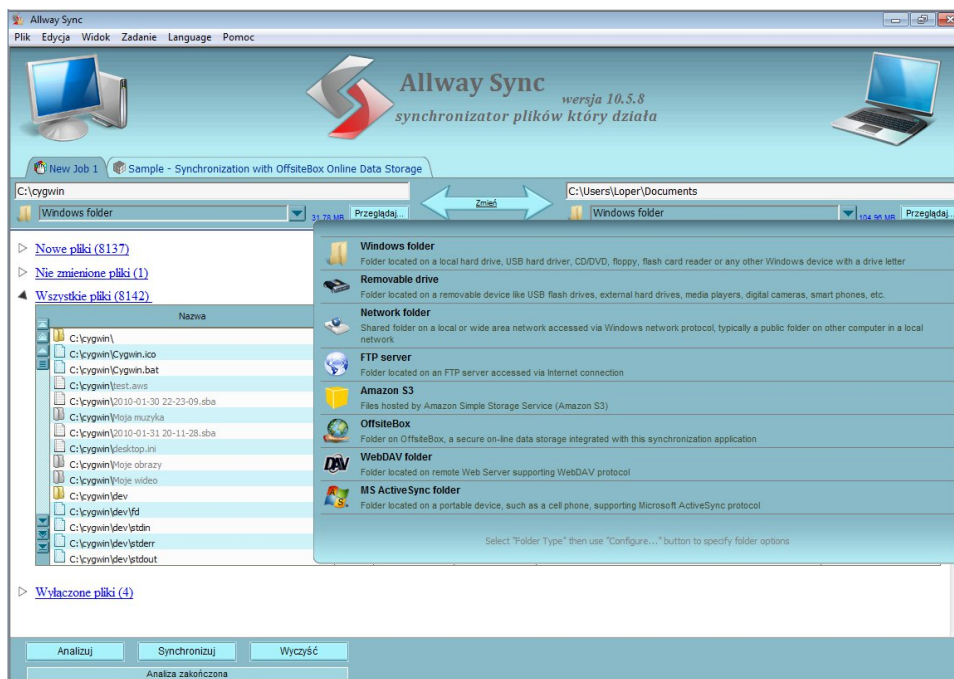
Lista ta jest bardzo obszerna, jednakże postanowiłem zainstalować i przetestować kilka z nich.

2.1.1 Allway Sync

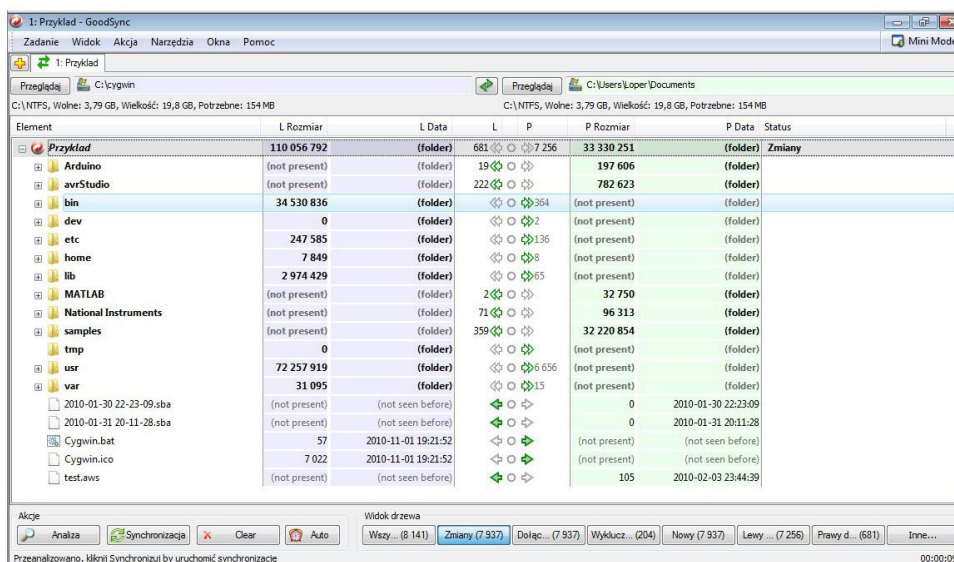
Uwagę moją przykuł darmowy **Allway Sync** [1], który posiada bardzo intuicyjną obsługę (choćby dzięki polskiej nakładce oraz licznym podpowiadziom), działanie na więcej niż jednym katalogu, synchronizację po sieci lokalnej i internetowej oraz z wymiennym dyskiem (patrz rys. 1) oraz, jako jeden z niewielu, umożliwiał instalację na pendrive'ie lub zewnętrznym dysku twardym (wersja **Allway Sync 'n' Go**) i posiadał dość rozbudowaną opcję automatycznej synchronizacji: przy starcie programu, po podłączeniu urządzenia, wykryciu zmian w folderach, co określony czas i oczywiście o podanej godzinie.

2.1.2 GoodSync

Godny uwagi jest także **GoodSync** firmy **Siber Systems** [2] - również darmowy (oczywiście istnieje jego płatna wersja, oferująca więcej niż 3 zaplanowane zadania po 100 plików w każdym z nich). Oprócz podstawowych opcji, obecnych w każdym z testowanych przeze mnie programów, wyróżniał się bardzo dobrą analizą zależności między zadaniami (zmiany w jednym z nich, powodują natychmiastową reakcję w innym zadaniu, posiadającym te same ustalone foldery, a na przykład inną lokalizację docelową), kopiowanie po SMB, FTP, WebDAV i SSH, szyfrowanie i kompresję kopii, przejrzysty interfejs (patrz rys. 2), dobrą automatyzację zadań (choć tu nie zadziałała mi opcja synchronizacji przez FTP po wykryciu połączenia z internetem). Program również dostępny w polskiej wersji językowej.



Rysunek 1: Allway Sync - okno główne programu z listą źródeł synchronizacji

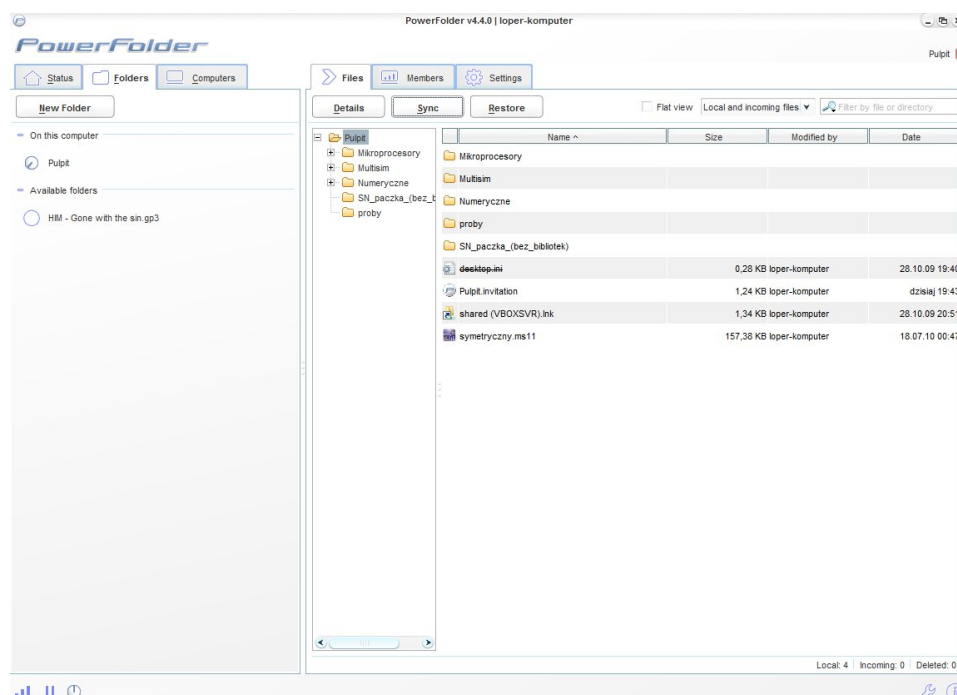


Rysunek 2: GoodSync - przykład porównywania zawartości folderów

2.2 Dysk wirtualny

W tej kategorii warto wspomnieć o **PowerFolder Basic** [3] (patrz rys. 3). Jest to wieloplatformowa aplikacja napisana w języku Java. Posiada bardzo wygodną możliwość zarządzania, dostępnego z poziomu przeglądarki oraz moduł, który działa jako systemowy proces i monitoruje wszelkie zmiany. Producent oferuje 1GB dysk wirtualny (obecnie na okres 371 dni od rejestracji), również dostępny przez przeglądarkę. Ciekawą opcją jest wykonywanie (plików „bat”) po zakończonej synchronizacji, dobrą kompresję danych „w locie”, bezpieczne przesyłanie danych (HTTPS, WebDAV przez SSL), ochronę przed przypadkowym usuwaniem plików, sumy kontrolne i wznowianie przerwanych transferów.

Minusem jest tu konieczność posiadania w miarę szybkiego łącza i plików o małych rozmiarach, w przeciwnym wypadku kopiowanie może trwać bardzo długo.



Rysunek 3: PowerFolder - przesyłanie na wirtualny dysk

2.3 Kompatybilne z rsyncd

Niestety, większość aplikacji nie posiada możliwości łączenia się z usługą **rsyncd** (nasłuchującą standardowo na porcie 873). Jedynym programem umożliwiającym synchronizację kompatybilną z **rsync**, jaki udało mi się znaleźć, jest **cwRsync** [4]. Jako, że założeniem mojego projektu indywidualnego (opisanym na wstępie) było stworzenie systemu synchronizacji katalogów i plików pomiędzy klientem na systemie Microsoft Windows a serwerem **rsyncd** na systemie Linux, będzie to idealne rozwiązanie do wykonania, postawionego przeze mnie, zadania. Z tej racji, programowi temu poświęciłem osobny, 4. rozdział.

2.4 Podsumowanie

Aplikacje te posiadają, na swoich stronach internetowych, wyczerpujące opisy możliwości, zrzuty ekranu, obszerną dokumentację, a nawet (w przypadku chociażby **PowerFolder**) kurs obsługi w formie animacji flashowej i własną stronę Wiki. Wiele z nich dostępnych jest w polskiej wersji językowej, co dodatkowo ułatwia pracę. W miarę rosnącej konkurencji, programy te prześcigają się w oferowanych możliwościach, które z dnia na dzień są coraz bogatsze. Większość tego typu aplikacji możemy znaleźć na takich stronach, jak chociażby:

- <http://www.sciagnij.pl>
- <http://www.idg.pl>
- <http://programy.pcworld.pl>

pod zakładką **Synchronizacja danych**.

Niestety wiele z nich, delikatnie mówiąc, nie jest godnych polecenia. Mają one zbyt wiele denerwujących reklam, bądź kłopotliwy interfejs i konfigurację, co skutecznie zniechęca do korzystania z nich, na co dzień, osoby, które cenią sobie zasadę „minimum wysiłku przy maksimum korzyści”.

Celowo zrezygnowałem tu z rozwiązań komercyjnych, ponieważ darmowe programy oferują zazwyczaj taką samą funkcjonalność, jak drogie oprogramowanie.

3 Rsync

Rsync (z angielskiego „**R**emote **s**ynchronization” - zdalna synchronizacja) to protokół synchronizacji plików przez sieć. Zwykle systemy przesyłania różnic (takie jak popularny diff/patch) wymagają istnienia obu wersji po jednej stronie, na podstawie których tworzona jest lista różnic, a następnie przesyłana przez sieć.

Rsync działa w odmienny sposób - przez sieć wysyłany jest spis plików z hashami bloków (zwykle ok. 1 kB), po czym na drugiej maszynie program sprawdza, które z fragmentów już posiada. Daje to bardzo dobre rezultaty i umożliwia radzenie sobie z sytuacjami, które dla patch/diff byłyby trudne do realizacji, jak przeniesienia plików. Rsync umożliwia też dostęp na bieżąco, w przeciwieństwie do „raz na dzień” w przypadku patch/diff. [5]

3.1 Parametry

składnia: `rsync [PARAMETRY] ŹRÓDŁO [CEL]`

Wśród wielu parametrów, jakie posiada rsync, warto wymienić najczęściej stosowane z nich (opracowano na podstawie [6]):

- a - tryb archiwalny (odpowiadający „-rlptgoD”)
- v - bardziej szczegółowe informacje
- r - rekursywnie wszystkie katalogi
- l - zachowanie dowiązań symbolicznych
- u - pomiń, jeśli nowsze u odbiorcy
- p - zachowaj prawa dostępu
- t - zachowaj czasy modyfikacji
- o - zachowaj właściciela
- g - zachowaj grupę
- e - deklaruje powłokę do użycia
- z - kompresuj dane podczas kopiowania
- n - test działania (bez wprowadzania zmian)
- h - liczby w czytelnej formie
- P - pokaż pasek postępu
- D - zachowaj potoki i nazwane gniazda
- L - kopiuje pliki, na które wskazują dowiązania, zamiast nich
- q - nie wypisuj błędów

Kompletny zbiór znaleźć można pod adresem [6] lub wykonując `man rsync` na systemach z rodziny Unix.

3.2 Przykłady użycia

Synchronizację można przeprowadzić między katalogami na lokalnym urządzeniu lub też na zdalny serwer: łącząc się z demonem `rsyncd` (zazwyczaj na porcie 873) lub poprzez SSH (standardowo - port 22). Zasadę działania najłatwiej tłumaczyć się na przykładach. Podam więc kilka z nich, opisując je pokrótce.

3.2.1 lokalnie

składnia: `rsync [PARAMETRY] ŹRÓDŁO [CEL]`

Często synchronizujemy dane z innego folderu do bieżącego katalogu. W takim wypadku można pominąć podawanie parametru `[CEL]` - traktowany jest on jako aktualny folder.

Przykład:

`rsync -av /home/user/katalog /home/tmp` - synchronizowanie katalogu `/home/user/katalog` w trybie archiwalnym (rekursywnie, z zachowaniem: praw dostępu, właściciela, grupy, dowiązań symbolicznych, czasów dostępu, urządzeń) z katalogiem `/home/tmp` z wypisywaniem szczegółowych informacji.

W rezultacie otrzymujemy:

```
$ rsync -av /home/user/katalog /home/tmp
sending incremental file list
./
Makefile
hello
hello.asm
sent 1413 bytes  received 72 bytes  2970.00 bytes/sec
total size is 1189  speedup is 0.80
```

Najpierw zostaje wysłana lista plików do synchronizacji. Następnie porównywana jest ona z zawartością katalogu docelowego `/home/tmp` - przesyłane są tylko te dane, których nie było w tym folderze lub ich odpowiedniki są

nowsze (porównywana jest data modyfikacji).

Dla porównania można zobaczyć, co się stanie, kiedy pojawi się nowy plik w katalogu źródłowym:

```
$ touch /home/user/katalog/nowy_plik
$ rsync -av /home/user/katalog /home/tmp
sending incremental file list
nowy_plik
sent 167 bytes  received 31 bytes  396.00 bytes/sec
total size is 1204  speedup is 6.08
```

Przesłany został jedynie `nowy_plik` - nie ma sensu kopiować całej reszty, skoro nie różni się ona niczym w obu folderach - jest to znacząca oszczędność czasu i łącza.

3.2.2 zdalnie

wysyłanie: `rsync [PARAMETRY] ŹRÓDŁO [LOGIN@]HOST:CEL`

pobieranie: `rsync [PARAMETRY] [LOGIN@]HOST:ŹRÓDŁO [CEL]`

przykłady:

`rsync -avz serwer:/var/log /home/tmp` - synchronizowanie, w trybie archiwalnym, z użyciem kompresji, logów z katalogu `/var/log` na komputerze zdalnym (`serwer`¹) do `/home/tmp` na komputerze lokalnym.

Działa to tak samo w drugą stronę (z komputera lokalnego na serwer):

`rsync -avz /home/tmp serwer:/var/log`.

Nie musimy synchronizować całych katalogów - możemy wybrać konkretne pliki (dla przykładu `plik1`, `plik3` i `plik4`:

`rsync -av /home/tmp/plik{1,3,4} serwer:/var/tmp` lub pliki z rozsze-

rzeniem `.jpg`: `rsync -av /home/tmp/*.jpg serwer:/home/user/zdjecia`.

3.2.3 poprzez usługę rsyncd

wysyłanie: `rsync [PARAMETRY] ŹRÓDŁO [LOGIN@]HOST::MODUŁ`

`rsync [PARAMETRY] ŹRÓDŁO rsync://[LOGIN@]HOST[:PORT]/CEL`

¹można tu podać adres albo nr IP hosta (nazwa „serwer” służy do zobrazowania kopowania pomiędzy różnymi urządzeniami w sieci)

pobieranie: `rsync [PARAMETRY] [LOGIN@]HOST:MODUŁ [CEL]`

`rsync [PARAMETRY] rsync://[LOGIN@]HOST[:PORT]/ŹRÓDŁO [CEL]`

W tym przypadku, w adresie, zamiast jednego dwukropka, używamy dwóch (::) lub podajemy adres `rsync://` (z opcjonalnym numerem portu). Dużym ułatwieniem jest tu listowanie dostępnych modułów (należy podać jedynie adres lub nr IP serwera): `rsync serwer::`. Należy zauważyć tu pewną różnicę - zamiast katalogu źródłowego/docelowego podaje się tu moduł. Lista modułów przechowywana jest na zdalnym komputerze w pliku `/etc/rsyncd.conf`².

Przykład wykonania takiego polecenia:

```
$ rsync 192.168.128.3::
shared          wspoldzielony katalog
pi              projekt indywidualny
backup          archiwum kopii
(...)
```

oraz zawartość `/etc/rsyncd.conf`:

```
$ cat /etc/rsyncd.conf
[shared]
    path=/home/shared
    comment = wspoldzielony katalog
[pi]
    path=/home/loper/Politechnika/PI
    comment = projekt indywidualny
    read only = no
    auth users = spychalm
    secrets file = /etc/rsync_users
[backup]
    path=/NAS/backup
    comment = archiwum kopii
(...)
```

Jak widać. katalogi mogą być zabezpieczone przed nieautoryzowanym dostępem. Cała konfiguracja sprowadza się do podania lokalizacji pliku (istnieje tu pewna dowolność w wyborze lokalizacji i nazwy pliku, jednak wiele

²zależnie od dystrybucji, może to być także `/etc/rsyncd/rsyncd.conf`, bądź inny

źródła podaje folder „/etc/” lub „/etc/rsync/” za typowy; należy pamiętać o nadaniu praw odczytu i zapisu jedynie dla użytkownika root przy pomocy np.: „chmod 600 /etc/rsync_users; chown root.root /etc/rsync_users”) secrets file = plik_z_hasłami (jak wyżej) oraz na umieszczeniu w nim linii z nazwą użytkownika oraz hasłem, oddzielonych dwukropkiem:

```
$ sudo cat /etc/rsync_user
test:test
spychalm:tajnehaslo
(...)
```

Dla zobrazowania:

```
$ rsync -auv 192.168.128.3::shared
receiving incremental file list
drwxrwx---          4096 2011/01/09 19:54:16 .
drwxr-x---          4096 2011/01/09 20:02:29 skrypty
-rw-----          300 2011/01/09 20:02:29 skrypty/ssh.bat
-rw-----          215 2011/01/09 20:02:24 skrypty/zdalny.sh

sent 29 bytes  received 138 bytes  334.00 bytes/sec
total size is 716  speedup is 4.29
```

3.2.4 za pomocą SSH

Czasem zachodzi potrzeba synchronizacji poufnych danych, bądź sytuacja, gdy na serwerze nie jest uruchomiony `rsyncd`. Posłużyć możemy się wtedy opcją kopiowania przez tunel SSH³. W takim przypadku używamy parametru „-e”, podając (w cudzysłowie) „ssh [-l <uzytkownik>] [-p <port>] ...”. Dla bieżącego użytkownika oraz portu 22 będzie to po prostu: `rsync -av -e „ssh” serwer:/var/tmp`. Zobaczmy jak wygląda takie wywołanie (do serwera sshd na porcie 12345):

```
$ rsync -avn -e "ssh -p 12345" serwer:~/sync
Warning: Permanently added '[serwer]' (RSA) to the list of known hosts.
receiving incremental file list
-rwxr--r--          146 2010/11/01 13:52:51 sync_skrypt

sent 11 bytes  received 47 bytes  116.00 bytes/sec
total size is 146  speedup is 2.52 (DRY RUN)
```

³o ile ta usługa jest obecna, a my posiadamy aktywne konto

Proszę zwrócić tu uwagę na zastosowanie przeze mnie parametru „-n”, oznaczającego „suche uruchomienie” (DRY RUN) - pozwala to na sprawdzenie, bez kopiowania, jaki wynik da wykonanie polecenia.

4 cwRsync

cwRsync to skrót od **c**(yg)**w**(in)**R**(emote)**sync**(hronization) - to pakiet łączący funkcjonalność Cygwin’a [7] i **rsync**’a. Został zoptymalizowany pod kątem szybkiej i łatwej instalacji na systemach z rodziny Windows. Jednak, w razie jakichkolwiek problemów, opisałem tu cały proces instalacji i konfiguracji.

Jako, że Cygwin posiada 2 wersje: pełną oraz „okrojona”, przedstawię obie z nich (osoby zainteresowane pełną wersją lub posiadające zainstalowanego Cygwina, mogą przejść od razu do rozdziału 4.2).

4.1 Pobieranie i instalacja

Pierwszym krokiem, jaki musimy podjąć, jest ściągnięcie programu. Pobieramy instalator o nazwie **cwRsync_xxx_Installer.zip** (gdzie **xxx** to numer wersji⁴), wchodząc na stronę <http://sourceforge.net/projects/sereds/files/cwRsync>. Następnie rozpakowujemy zawartość pobranego archiwum i uruchamiamy instalator (**cwRsync_xxx_Installer.exe**). Podczas instalacji zalecane jest zachowanie domyślnego katalogu aplikacji: **C:\Program Files\cwRsync**. Otwierając go, znajdziemy przykładowy⁵ skrypt synchronizacji oraz katalog główny **bin**. Zawartość przedstawia rysunek 4. Jak widać, znajdziemy tam znajome programy⁶: „rsync.exe”, „ssh.exe”, „ssh-keygen.exe”. Mając zainstalowany cwRsync, możemy przejść do tworzenia plików wsadowych (rozdział 4.3).

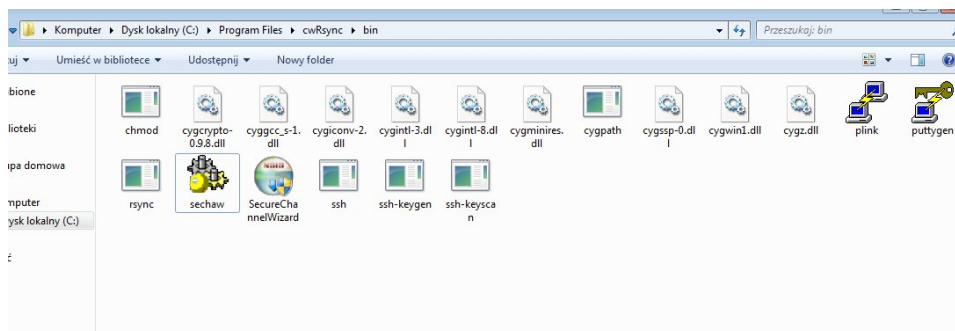
4.2 Instalacja Cygwin’a

Alternatywą dla pomniejszonej wersji Cygwina (a dokładniej - bibliotek „dll” obsługujących Linuxowe API), jest jego pełna wersja. Polecam ją osobom, które chciałyby mieć choć część funkcjonalności Linuxowej na Windowsie. Z dostępnych programów mogę wymienić chociażby: „vim”, „apache”,

⁴w momencie pisania tego sprawozdania, najnowszą wersją była 4.0.5; plik około 4MB

⁵radzę tu korzystać z przygotowanych przeze mnie skryptów (rozdział 4.3)

⁶wymienione programy będą niezbędne w dalszej części projektu



Rysunek 4: Zawartość folderu „C:\Program Files\cwRsync\bin”

„tetex”, „octave”, „cron” i wiele innych. Opis ten dedykowany także dla osób, które już posiadają ten program zainstalowany, a chcą uzupełnić go o opcję synchronizacji.

Proces instalacji przedstawię w kilku prostych krokach:

1. pobieramy najnowszą wersję Cygwin’a ze strony <http://www.cygwin.com/setup.exe>
2. podczas instalacji wybieramy kolejno (po każdym klikając dalej):
 - „Install from Internet”
 - podajemy katalog do instalacji, np. `C:\cygwin`
 - podajemy miejsce, gdzie mają trafiać ściągane pliki, np. `C:\cyg_install`; po instalacji, można je usunąć
 - jeśli łączymy się przez proxy, ustawiamy je w tym kroku, w przeciwnym wypadku - „Direct Connection”
 - jako stronę, z której będą pobierane dane, możemy wybrać dowolną (wracamy i wybieramy inną, gdyby któraś z nich nie działała)
 - w tym kroku rozwijamy listę „Net” i wybieramy z niej „rsync” - klikamy na **Skip**, aż pojawi się najnowszą wersja (patrz rys. 5); można tu skorzystać także z wyszukiwarki
 - jeżeli chcemy korzystać z szyfrowanego połączenia poprzez SSH, wybieramy pakiet **openssh**, także z listy „Net”
 - na zakończenie instalacji możemy ustawić sobie skróty na pulpicie i/lub w Menu Start


```
@echo off
cls
setlocal
```

```
SET PARAMETRY=avu
SET SCIEZKA=C:\cygwin
SET PATH=%SCIEZKA%\bin;%PATH%
```

```
rsync -%PARAMETRY% /cygdrive/c/katalog/ /cygdrive/d/inny_katalog
```

Jeżeli chcemy, dodajemy deklarację zmiennych:

```
SET CEL=/cygdrive/d/inny_katalog
SET ZRODLO=/cygdrive/c/katalog/
```

Mamy wtedy: `rsync -%PARAMETRY% %ZRODLO% %CEL%`, dzięki czemu wygląda to przejrzystiej, a my sami ustawiamy jedynie zmienne, nie martwiąc się o składnię polecenia.

Krótkie wyjaśnienie poszczególnych instrukcji:

`@echo off`, `cls`, `setlocal` - wyłącza ciągłe wypisywanie ścieżki, czyści ekran i włącza możliwość definiowania zmiennych lokalnych, z których to będziemy korzystać,

`SET PARAMETRY=...` - parametry synchronizacji,

`SET SCIEZKA=...` - ścieżka, gdzie zainstalowano Cygwina (patrz 4.3.1),

`SET PATH=%SCIEZKA%\bin;%PATH%` - dopisywanie lokalizacji Cygwina do zmiennej systemowej, w której wyszukiwane są programy, dzięki czemu możemy użyć samej nazwy `rsync`,

`SET CEL=...` - podajemy miejsce docelowe,

`SET ZRODLO=...` - podajemy lokalizację źródłową; tutaj wymagane jest dłuższe wyjaśnienie.

W Windowsie ścieżki składają się z litery dysku (np. C, D, E), dwukropka i odwróconego ukośnika (np. C:\katalog). Użycie znaku dwukropka, w składni polecenia `rsync`, zostałoby potraktowane jako ścieżka do katalogu zdalnego, dlatego stosujemy tu konwencję znaną z UNIX'a, to jest: zamieniamy odwrócone ukośniki na nieodwrócone, a w miejsce litery dysku wpisujemy `/cygdrive/<litera_dysku>`. W ten sposób nasza przykładowa lokalizacja wygląda następująco: `/cygdrive/c/katalog`.

4.3.3 Synchronizacja zdalna

Dla synchronizacji zdalnej nie trzeba zbytnio zmieniać powyższego skryptu - wystarczy modyfikacja jednej zmiennej. Przy wysyłaniu na zdalny serwer, jako cel podajemy [LOGIN@]HOST:CEL, na przykład:

```
SET cel=spychalm@192.168.100.5:/home/spychalm/testowy.
```

Pobieranie jest analogiczne - ustawiamy, jako źródło, [LOGIN@]HOST:ŹRÓDŁO.

Nie będę tutaj omawiać szczegółów, ponieważ wyjaśniłem wszystko w rozdziale 3.2.2.

Przykład pobierania danych:

```
@echo off
```

```
cls
```

```
setlocal
```

```
SET ZRODLO=~ /www/
```

```
SET CEL=/cygdrive/c/strona_www
```

```
SET PARAMETRY=av
```

```
SET SCIEZKA=C:\cygwin
```

```
SET PATH=%SCIEZKA%\bin;%PATH%
```

```
rsync -%PARAMETRY% spychalm@volt.iem.pw.edu.pl:%ZRODLO% %CEL%
```

4.3.4 Poprzez usługę rsyncd

Podobnie jak poprzednio, skorzystamy z wcześniejszego opisu składni polecenia **rsync**, wracając do rozdziału 3.2.3. Można ustawić tu zmienne zgodnie ze schematem, jak przy synchronizacji zdalnej, jednak pokażę inne rozwiązanie. Zmienimy składnię polecenia **rsync -%PARAMETRY% %ZRODLO% %CEL%**.

Dla wysyłania:

```
rsync -%parametry% %zrodlo% [LOGIN@]HOST::MODUL
```

lub

```
rsync -%parametry% %zrodlo% rsync://[LOGIN@]HOST[:PORT]/%cel%
```

podając oczywiście login, host i moduł lub port i cel, do którego się odwołujemy.

Dla odbierania postępujemy analogicznie. Jedną z modyfikacji może być tu:

```
rsync -%parametry% 192.168.100.5::wazne_dane /cygdrive/c/pobrane.
```

4.3.5 Za pomocą SSH

Zastosować można tu co najmniej dwa rozwiązania: do parametrów dopisujemy `e "ssh [-l <uzytkownik>] [-p <port>] ..."` lub to samo umieszczamy po `rsync -%parametry%`. Przykład: `set PARAMETRY=ave "ssh -l spychalm"`. Cały skrypt mógłby wyglądać następująco:

```
@echo off
cls
setlocal

SET LOGIN=spychalm
SET PORT=4321
SET HOST=edi.iem.pw.edu.pl
SET ZRODLO=/cygdrive/c/strona_www/
SET CEL=~ /www
SET PARAMETRY=auve "ssh -l %LOGIN% -p %PORT%"
SET SCIEZKA=%PROGRAMFILES%\cwRsync
SET PATH=%SCIEZKA%\bin;%PATH%

rsync -%PARAMETRY% %ZRODLO% %HOST%:%CEL%
```

4.3.6 Podsumowanie

Do każdego ze skryptów można dopisać na końcu linijkę `PAUSE` - umożliwi ona wyświetlenie wyniku operacji - nie zamknie okna, jeśli wywołaliśmy skrypt bezpośrednio (to znaczy klikając na niego - nie z poziomu konsoli). W przypadku używania tych plików wsadowych w połączeniu z „Harmonogramem Zadań”, autostartem - kiedy jest automatycznie startowany, lepiej pominąć tą opcję (a nawet parametr *verbose* (`-v`)), ponieważ w takiej sytuacji nie potrzebujemy okna z informacją.

Oczywiście każdy może napisać od nowa lub dowolnie edytować te skrypty, stąd zamysł podania przeze mnie składni w rozdziale 3. Celowo, za każdym razem, zmieniałem dane i parametry dla pokazania, że są one jedynie przykładem i każdy ustala je według własnych upodobań.

4.4 Ustawianie kluczy w SSH

Łącząc się ze zdalnym hostem poprzez SSH, jesteśmy za każdym razem pytani o hasło. Ciągłe jego wprowadzanie może okazać się niewygodne. Z pomocą przychodzi autoryzacja przy pomocy kluczy - dzięki czemu możemy ustawić zupełnie inne hasło lub nawet puste, czym zajmiemy się w tym rozdziale. Nie będę tu omawiał szczegółowo wszystkich parametrów komendy `ssh-keygen`, ponieważ jest ona dokładnie opisana w internecie[8].

Całość sprowadza się do wykonania kilku kroków (działa tylko z pełną wersją Cygwina):

1. **Uruchamiamy Cygwina**, z poziomu pulpitu (skrót), lokalizacji programu (np. `C:/cygwin/Cygwin.bat`) lub Menu Start (Cygwin Bash Shell).
2. **Generujemy klucze: publiczny i prywatny** - `$ssh-keygen -t rsa`. Na pytanie o hasło/passphrazę wciskamy enter. Polecenie wygeneruje dwa pliki, domyślnie w katalogu domowym w „ssh”. Pierwszy o nazwie „id_rsa” zawiera klucz prywatny, który nie może być nikomu udostępniany. Drugi o nazwie „id_rsa.pub” zawiera klucz publiczny, który wysyłamy na dowolny serwer, z którym zamierzamy ustanawiać połączenie.
3. **Wysyłamy klucz publiczny na serwer**, z którym chcemy łączyć się bez hasła. Użyjemy do tego polecenia⁸
`scp ~/ssh/id_rsa.pub [uzytkownik]@host:~/`
4. **Logujemy się na zdalny host**, jeszcze przy pomocy hasła. Tam poleceniem `cat ~/id_rsa.pub >> ~/ssh/authorized_keys`⁹ dodajemy nasz klucz do listy kluczy autoryzowanych do łączenia się z naszym hostem.

Od tej pory mamy możliwość logowania się bez hasła do zdalnego serwera, a tym samym nasza zautomatyzowana synchronizacja nie będzie nas za każdym razem pytała o hasło, oczywiście zachowując bezpieczny kanał transmisji plików.

⁸można użyć tu innego sposobu wysyłania, jednak ten jest najprostszy

⁹jeżeli katalog „ssh” nie istnieje, zakładamy go przy pomocy „mkdir ssh”

Dużym ułatwieniem jest skorzystanie z gotowego programu „ssh-copy-id”[9], który wykonuje wysyłanie i dopisanie klucza na zdalnym komputerze za nas. Generujemy klucze, a następnie uruchamiamy `ssh-copy-id [uzytkownik]@host`.

5 Testowanie

Cały proces instalacji i konfiguracji przeprowadziłem początkowo na maszynie wirtualnej z zainstalowanym systemem Windows 7. Następnie całość powtórzyłem na innej maszynie wirtualnej z Windowsem XP oraz na komputerach, z zainstalowanymi tymi wersjami Windowsa, jako ich macierzystymi systemami operacyjnymi. Przykładowe programy (z rozdziału 2.) testowane były tylko na systemach wirtualnych, z racji braku Windowsa na moim notebooku. Na żadnym z tych urządzeń nie zauważyłem problemu z pobieraniem, instalacją czy konfiguracją `cwRsync`'a. Jednakże, może się tak zdarzyć, że podane przeze mnie strony internetowe lub wersje programów, nie będą dostępne. Należy wówczas skorzystać z wyszukiwarki internetowej i za jej pomocą znaleźć serwery lustrzane lub strony, gdzie podane oprogramowanie będzie możliwe do pobrania.

6 Wnioski

Z synchronizacją danych spotykamy się na co dzień. Większość z nas kopiuje dane z jednego nośnika na inny - jedni robią to „ręcznie”, inni mają do tego skrypty. Osobiście uważam, że druga opcja jest znacznie korzystniejsza, zwłaszcza, gdy chcemy, by ten proces wykonywał się automatycznie.

Pomimo dostępności mnogiej liczby programów do tego celu oraz przetestowania przeze mnie części z nich, stwierdzam, że żaden nie zastąpił szybkiego i łatwego rozwiązania konsolowego, a mianowicie programu `rsync`. Po poświęceniu chwili czasu, mamy napisane różnorakie pliki wsadowe, do użycia, w zależności od potrzeb. Możemy je umieszczać na serwerach czy terminalach, nawet tych bez środowiska graficznego i uruchamiać je zdalnie. W systemach z rodziny Windows, dodając go do „Harmonogramu Zadań”, a w Linuxie - do zadań `cron`'a, nie musimy ciągle pamiętać o zrobieniu kopii zapasowej. Uważam to za ogromne ułatwienie zwłaszcza dla osób, które mają z tym problemy.

Literatura

- [1] AllwaySync, <http://allwaysync.com>, dostęp 31.10.2010
- [2] GoodSync, <http://www.goodsync.com>, dostęp 31.10.2010
- [3] PowerFolder, <http://www.powerfolder.com>, dostęp 31.10.2010
- [4] cwRsync, <http://www.itefix.no/i2/node/10650>, dostęp 31.10.2010
- [5] Wikipedia, *Rsync*, <http://pl.wikipedia.org/wiki/Rsync>, dostęp 31.10.2010
- [6] Rsync Documentation, *rsync man page*, <http://samba.anu.edu.au/ftp/rsync/rsync.html>, dostęp 1.11.2010
- [7] Cygwin, <http://www.cygwin.com>, dostęp 3.11.2010
- [8] ssh-keygen, *ssh-keygen man page*, <http://linuxmanpages.com/man1/ssh-keygen.1.php>, dostęp 6.12.2010
- [9] ssh-copy-id, *ssh-copy-id man page*, <http://linux.die.net/man/1/ssh-copy-id>, dostęp 9.01.2011