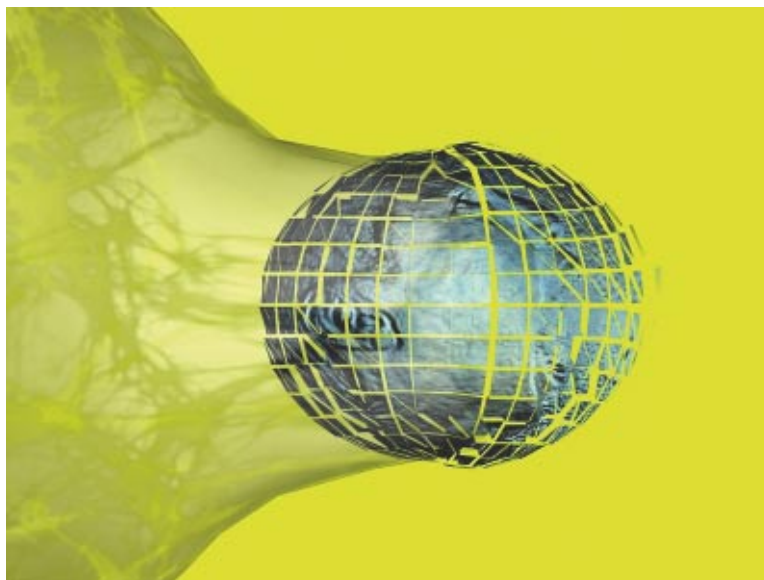


rTorrent – torrenty w konsoli

Łukasz Olender

Ostatnimi czasy dość dużo słyzy się o programach do ściągnięcia plików w sieci BitTorrent. Jedni preferują bogatego w funkcje Azureusa, drudzy lekkiego Bittornado, a jeszcze inny zachwalają prostotę i łatwość obsługi klienta wbudowanego w najnowsze wersje Opery. Cóż zatem wybrać mając do wyboru ogromną ilość programów do ściągnięcia popularnych torrentów?



linux@software.com.pl

Odpowiedź nie jest prosta, jednak ceniąc sobie funkcjonalność, szybkość działania, niskie użycie procesora oraz konsolowy interfejs możemy z powodzeniem dać szansę najnowszemu dziełu autorstwa Jari Sundell'a – tekstowemu klientowi rTorrent.

Wstęp

Te wymienione wyżej zalety, to niewątpliwe atuty programu, jednak na tym lista jego plusów się nie kończy. Do rysowania oraz obsługi interfejsu wykorzystano bibliotekę ncurses, co pozwala na uruchamianie rTorrenta w programach takich jak *screen*, czy *dtach* oraz daje nam możliwość łatwego, zdalnego sterowania programem poprzez SSH. Jak na porządnego klienta przystało, rTorrent pozwala na ściągnięcie pojedynczych plików z torrenta, oraz wspiera szyfrowanie. Warto wspomnieć także, że wykorzystuje on własną bibliotekę *libTorrent*, dzięki której – jak podaje Wikipedia – na łączach o dużej przepustowości jest w stanie wysłać dane kilkukrotnie szybciej niż oficjalny klient. Nie każdy może to dokładnie zmierzyć, jednak prędkość działania samego programu oraz użycie procesora czy pamięci z pewnością dadzą się odczuć oraz zadowolą wszystkich, którym zależy na dodatkowej komfortowej pracy z innymi programami na

słabszym komputerze. Przy pobieraniu czterech torrentów z łączną prędkością około 40KB/sek, top wskazywał użycie procesora i ramu na poziomie nie większym niż 1,5% dla procesora takowanego z prędkością 1GHZ oraz 4,5% dla pamięci o pojemności 384MB ram.

Instalacja

Instalacja programu zwykle jest bardzo prosta – rTorrent znajduje się w większości repozytoriów popularnych dystrybucji, skąd można go za pomocą odpowiedniego polecenia (*apt-get*, *pacman* itd.) łatwo pobrać i zainstalować. Jeśli jednak byłoby inaczej, zawsze możemy posłużyć się działem download [2] ze strony głównej programu [1]. Znajduje się tam kod źródłowy najnowszej jak i starszych wersji programu oraz link do repozytorium SVN, skąd zawsze możemy pobrać najświeższą wersję. Do prawidłowej kompilacji programu będą wymagane m.in. biblioteki *libsigt++*, *libcurl*, *libncurses*, oraz autorska *libTorrent*, którą możemy pobrać ze strony programu.

Pierwsze uruchomienie

Uruchomienie programu sprowadza się do wpisania w konsoli polecenia `rtorrent`. Warto jednak uruchomić go korzysta-

jąc ze screena, dzięki czemu nie będzie niepotrzebnie blokował terminala, a w razie potrzeby będziemy mogli szybko przywrócić jego widok poleceniem `screen -r` (więcej na temat screena znajdziemy w systemowym manualu bądź na stronie [3]). Po uruchomieniu, naszym oczom ukaże się widok główny interfejsu programu. Zanim jednak zaczniemy dodawać pierwsze torrenty do listy, niezbędne będzie prawidłowe skonfigurowanie programu za pomocą pliku konfiguracyjnego `.rtorrent.rc`, który standardowo znajduje się w katalogu domowym użytkownika. Jak to zrobić? Zawartość wraz z komentarzami pliku konfiguracyjnego znajduje się w Listingu 1.

Interfejs programu

Składa on się z dwóch najważniejszych ekranów między którymi poruszamy się wciskając klawisze lewo/prawo :

- widok główny;
- widok pobierania;
- oraz pasek statusu który jest obecny cały czas w dolnej części interfejsu.

Widok główny

Znajduje się w nim lista wszystkich torrentów oraz podstawowe informacje o nich takie jak:

- ilość pobranych/wysłanych danych;
- szybkość wysyłania/pobierania;
- ilość wysłanych danych;
- procent pobierania torrenta;
- przewidywany czas zakończenia ściągania;
- T – jeśli istnieje oznacza, że plik `.torrent` jest na dysku lokalnym;
- I – jeśli istnieje oznacza, że torrent nie odpowiada na działania rTorrenta;
- Share ratio.

Może przedstawiać wszystkie torrenty, posortowane wg. nazwy bądź podzielone ze względu na:

- stan (uruchomione bądź nie);
- stan pobierania (pobierane bądź pobrane);
- aktualnie seedowane torrenty;
- aktualnie haszowane torrenty.

Widok pobierania

Przedstawia on wszelkie wiadomości o wybranych torrentach; Znajdują się w nim następujące zakładki:

- *Peer list* – zakładka peerów. Składa się z 9 kolumn oznaczających po kolei:
- *IP* – numer IP peera;
- *UP* – prędkość wysyłania plików do danego peera;

- *DOWN* – prędkość pobierania plików od danego peera;
- *PEER* – łączna prędkość ściągania dane go peera;
- *C/RE/LO* – typ połączenia/inf. o zdalnym kliencie/inf. o lokalnym kliencie. (więcej na stronie [4]);
- *QS* – Kolejka do przychodzących/wychodzących plików. Pierwsza liczba oznacza numer kawałka torrenta (patrz Chunks seen), który będzie pobierał od nas dany peer, druga natomiast numer kawałka, który my będziemy pobierać od danego peera;
- *DONE* – Część pliku podana w procentach, którą posiada dany peer;
- *REQ* – Numer kawałka, który jest najwyżej w kolejce pobierania danego peera;
- *SNUB* – Pole to zaznaczone jest gwiazdką, jeżeli dany peer nie wysyła do nas plików;
- *Info* – pełne informacje o samym torrentie oraz o użyciu dysku i pamięci;
- *File list* – informacje o plikach składających się na dany torrent;
- *Tracker list* – informacje o trackerach;

Chunks seen – szczegółowe informacje o postępie pobierania w formie przedstawionych kawałków. Oznaczone są następująco:

- X – pobrane X – brak części X – w kolejce X – aktualnie pobierane, gdzie X jest jedną z wartości w systemie szesnastkowym od 0 do F. Liczby te symbolizują popularność

plików wśród osób, które nie są *seedami*. I tak 0 oznacza, że tylko osoby seedujące mają tę część, 1 oznacza, że prócz seedów jedna osoba ma daną część, 2 – dwie itd., aż do F czyli 15 i więcej.

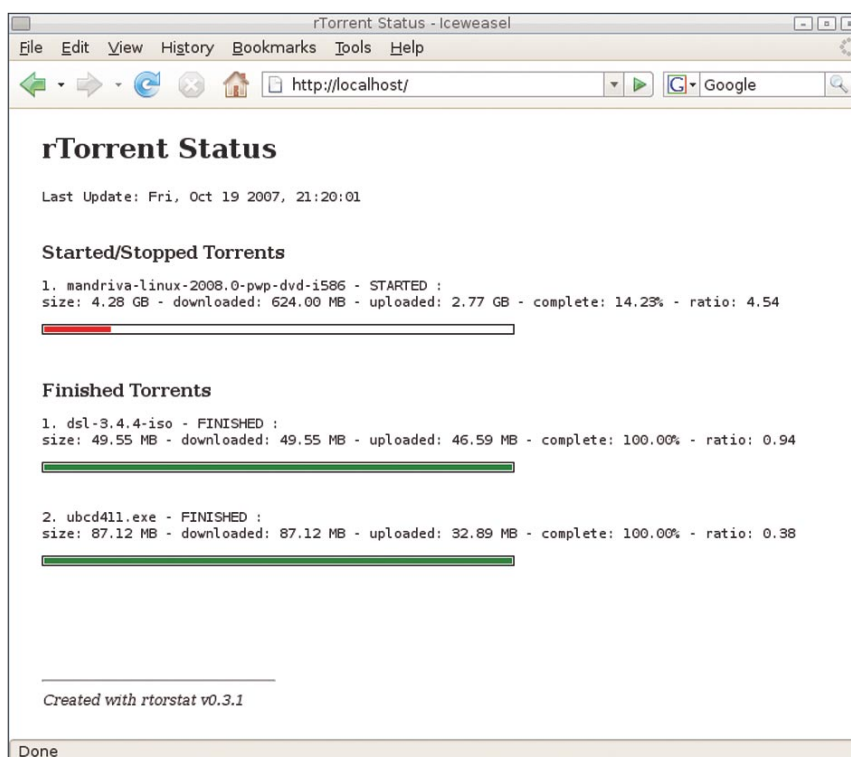
Status programu

Wyświetla on następujące informacje o stanie programu:

- *Throttle* – ograniczenia uploadu/downloadu programu podane w KB.
- *Rate* – prędkość wysyłania/pobierania w programie.
- *Port* – port używany przez rTorrenta.
- *U* – ilość peerów połączonymi z nami/max. ilość peerów, którzy mogą się z nami połączyć (w celu wysyłania do nich plików).
- *D* – ilość peerów/seedów połączonymi z nami/max. ilość peerów/seedów, którzy mogą się z nami połączyć (w celu pobrania od nich plików).
- *H* – ilość jednoczesnych zapytań HTTP/max. ilość jednoczesnych zapytań HTTP.
- *S* – ilość handshake'ów/ilość otwartych połączeń/max. ilość otwartych połączeń.
- *F* – ilość otwartych plików/max. ilość otwartych plików.

Uruchamianie programu wraz ze startem systemu

Korzystając przez dłuższy czas z programu pewnie dojdziemy do momentu, w którym



Rysunek 1. Rtorstat, czyli dowód, że małe jest piękne

chcielibyśmy, by program uruchamiał nam się przy starcie systemu. Twórca rTorrenta wyszedł na przeciw użytkownikom Debiana (i pochodnych) tworząc odpowiedni plik, który możemy łatwo dodać do skryptów startowych. Znajdziemy go na stronie [6]. Do działania będzie potrzebne ustawienie kilku opcji, co możemy zrobić edytując ten plik:

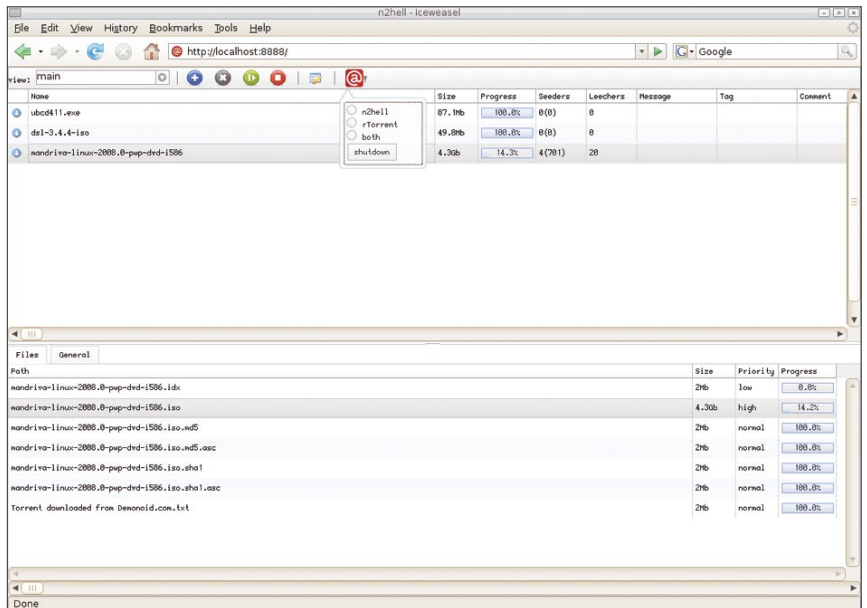
- `user="marcin"` – ustawia użytkownika marcin na tego, z kontekstu którego będzie uruchamiany rTorrent.
- `config="/home/user/.rtorrent.rc"` – wskazuje położenie i nazwę pliku konfiguracyjnego rTorrent.
- `options=""` – opcje, które domyślnie wstawilibyśmy po wpisaniu polecenia `rtorrent`. W tym przypadku rTorrent będzie uruchomiony z domyślnymi opcjami.
- `base="/home/user"` – katalog bazowy z którego będzie uruchomiony screen z rTorrentem.
- `srnname="rtorrent"` – nazwa sesji screena.

Jeżeli już odpowiednio skonfigurowaliśmy skrypt, zmieniamy jego nazwę na `rtorrent`, ustawiamy `chmod +x` oraz przenosimy do katalogu `/etc/init.d`. Ostatnim krokiem, jaki należy wykonać, jest wykonanie komendy `update-rc.d rtorrent defaults`. Przy następnym uruchomieniu komputera bądź wydaniu polecenia `/etc/init.d/rtorrent start` program będzie dostępny pod poleceniem `screen -r srnname`, gdzie `srnname`, to nazwa sesji screena. Uwaga! Skrypt ten może nie działać prawidłowo z najnowszymi, niestabilnymi wersjami rTorrenta.



BitTorrent – podstawowe pojęcia

- *Peer* – użytkownik, który w danym momencie pobiera plik (torrent) i udostępnia posiadane jego fragmenty;
- *Seed* – użytkownik posiadający kompletny plik (torrent) i udostępniający go innym osobom;
- *Tracker* – serwer przekazujący informacje (adresy IP) o innych użytkownikach pobierających dany plik;
- *Leecher* – użytkownik, który pobiera o wiele więcej danych, niż samemu udostępnia, ograniczając przepustowość łącza od siebie, korzystając ze zmodyfikowanego klienta, który umożliwia mu oszukiwanie, lub nie udostępniając swojej kopii pliku po jego pobraniu;
- *.torrent* – metaplik zawierający niezbędne informacje do rozpoczęcia pobierania pliku (plików);
- *Info hash* – 160-bitowa wartość zakodowana algorytmem SHA1 i zapisana w metapliku `.torrent`. Służy do identyfikacji przez trackera pliku `.torrent` oraz użytkowników ściągających te same dane.
- *Share ratio* – stosunek ilości danych wysłanych do pobranych;



Rysunek 2. N2hell – wygląd, funkcjonalność, łatwość instalacji – przepis na sukces?

Usprawnienie działania

Dochodząc do tego momentu możemy śmiało stwierdzić, że nasz rTorrent jest gotowy do używania. Jednak autor przygotował dla nas szereg dodatków, które z pewnością przydadzą się każdemu, kto chciałby z programu wycisnąć maksimum możliwości. Dodając do pliku konfiguracyjnego specjalne wpisy, możemy skorzystać z wbudowanych w program funkcji, które wywołują się w określonym czasie. Przykładowe, najczęściej używane podane są w Listingu 2. Dostosowanie ich do swoich potrzeb jest dość łatwe – po przeanalizowaniu składni myślę, że nikt nie będzie miał z tym problemów.

W tym miejscu mógłby nastąpić koniec artykułu – wszystko to, co niezbędne do sprawnego ściągania danych za pomocą

rTorrenta zostało opisane. Jednak program cieszy się dużą popularnością, inspirując jego użytkowników do tworzenia różnego rodzaju dodatków. Oto pokrótce opisane dwa z nich, moim zdaniem najciekawsze.

Rtorstat

Jest to niewielki skrypt napisany w pythonie, dzięki któremu możemy na bieżąco monitorować poprzez przeglądarkę internetową status pobierania plików. Zasada jego działania jest prosta – program pobiera dane o torrentach z katalogu sesji (ustawiamy to w pliku konfiguracyjnym za pomocą zmiennej `session`), a następnie generuje prostą stronę internetową zawierającą te informacje. *Rtorstat* znajdziemy na stronie [5]. Do działania niezbędne jest ustawienie następujących zmiennych w pliku `.rtorrent.rc`: `session_on_completion`, `session`, oraz `schedule = session_save` (więcej w Listingu 1. oraz 2.). Po ustawieniu tych zmiennych, pobraniu, rozpakowaniu i prze-

Listing 1. Konfiguracja serwera

```
scgi.server = (
    "/RPC2" => # RT_DIR
    ( "127.0.0.1" =>
        (
            "host" => "127.0.0.1", # Ip,
            którego rTorrent nasłuchuje
            "port" => 5000, # Port wpisany
        )
    )
    w .rtorrent.rc
    "check-local" => "disable"
)
)
```

**Listing 2.** Przykładowy plik konfiguracyjny .rtorrent.rc

```
# Katalog plików sesji
session = /mnt/hda5/torrent/sesja_pliki
# Domyślne położenie dla ściąganych plików
directory = /mnt/hda5/torrent/gotowe
# Limit uploadu dla programu w kilobajtach
upload_rate = 0
# Limit downloadu dla programu w kilobajtach
download_rate = 0
# Próbować połączeń UDP? Może to zwiększyć prędkość pobierania
use_udp_trackers = yes
# Maksymalna ilość połączeń wychodzących dla jednego torrenta;
max_uploads = 20
# Zakres portów używanych przez rTorrent
port_range = 6892-6892
# Rozpoczyna pobieranie wykorzystując losowy port z listy ustawionej przez
# zmienną port_range (powyżej)
port_random = no
# Minimalna i maksymalna ilość połączeń dla każdego torrenta
min_peers=10
max_peers=400
# Tak jak powyżej, jednak w odniesieniu do ściąganych plików.
min_peers_seed = 10
max_peers_seed = 50
# Określa, czy dane mają być sprawdzane po pobraniu torrenta
check_hash = no
# Po zakończeniu pobierania nie usuwa plików sesji (niezbędne dla
rtorstat).
session_on_completion = yes
```

Listing 3. .rtorrent.rc – zaawansowane funkcje

```
# Torrenty dodane do katalogu ~/torrenty będą automatycznie, co 10 sek
# dodawane do listy aktywnego pobierania
schedule = watch_directory,10,10,load_start=~/.torrent/*.*.torrent
# Wraz z usuwaniem torrenta z listy, usuwamy plik .torrent;
schedule = untied_directory,10,10,stop_untied=
# pobrane torrenty będą przenoszone do katalogu ~/torrenty/pobrane
on_finished = move_complete,"execute=mv,-u,$d.get_base_path=~/.torrenty/
pobrane ;d.set_directory=~/.torrenty/pobrane"
# Zatrzymaj upload, gdy ratio wyniesie ponad 2.00, przy przynajmniej
# 50MB wysłanych danych, bądź gdy ratio będzie większe od 3.00
schedule = ratio,10,10,"stop_on_ratio=200,50M,300"
# Wstrzymaj ściąganie, gdy miejsce na dysku będzie wynosiło 20MB lub mniej
schedule = low_diskpace,5,60,close_low_diskpace=20M
# Pierwszy wpis ustawia, że od godziny 1:00, maksymalna prędkość
pobierania będzie
# nieograniczona, drugi, że od 7:30 prędkość ta zmaleje do 25KB. Trzeci
analogicznie do.
# dwóch pierwszych. Można tworzyć nieograniczoną ilość takich wpisów.
schedule = throttle_1,01:00:00,24:00:00,download_rate=0
schedule = throttle_2,07:30:00,24:00:00,download_rate=25
schedule = throttle_3,21:00:00,24:00:00,download_rate=60
# Zapisuj status sesji co 5 minut (niezbędne dla rtorstat)
schedule = session_save,240,300,session_save=
```

ściu do katalogu ze skryptem wystarczy wpisać polecenie:

```
python rtorstat.py /ścieżka/do/sesji
> /ścieżka/do/index.html
```

by zastąpić plik *index.html* zawartością wygenerowaną przez *rtorstat*. Jeżeli chcemy, aby odświeżanie strony odbywało się automatycznie możemy stworzyć pętlę w bashu:

```
#!/bin/bash
while : ; do
sleep 2m
python rtorstat.py /mnt/hda5/
torrent/sesja_pliki/ > /var/www/
index.html
done
```

bądź jeśli mamy możliwość dodać zadanie do crona (za pomocą polecenia `crontab -e:\`)

```
*/2 * * * * python /ścieżka/do/
rtorstat/rtorstat.py /ścieżka/do/
sesji > /ścieżka/do/index.html
```

W obu przypadkach strona będzie odświeżana co 2 minuty. Oczywiście, jeżeli ktoś czuje się na siłach, może dowolnie zmodyfikować skrypt np. tak, aby integrował się z jego stroną domową, czy używał innych kolorów. Jednak to wszystko daje nam jedynie namiastkę komunikacji z rTorrentem. Chcąc zdalnie sterować programem poprzez www, powinniśmy zainteresować się nakładkami *rTWi* (które w tym artykule nie będą opisywane), wTorrent bądź n2hell, który to z wszystkich trzech jest najmłodszy. Zaletą wyróżniającą wTorrent jest możliwość łatwego sterowania rTorrentem przez wielu użytkowników, z których każdy może mieć swoje prywatne torrenty będące jednocześnie niewidoczne dla reszty. Ta opcja sprawia, że rTorrent wraz z tą nakładką może stać się programem do pobierania plików dla wszystkich użytkowników jednej maszyny, z którą każdy z nich będzie mógł się połączyć przez przeglądarkę w celu sprawdzenia stanu pobierania, czy dodania nowych torrentów. Niestety żadna z tych trzech nakładek nie współpracuje w pełni z Operą (testowana wersja to 9.23), jednak miejmy nadzieję, że w przyszłych wersjach powyższe wsparcie zostanie wprowadzone.

wTorrent

Choć sam wTorrent jest dość prosty w obsłudze, to do jego uruchomienia będzie niezbędne wykonanie kilku dodatkowych kroków. Przede wszystkim będziemy potrzebowali serwera

www Apache2 bądź lighttpd, oba ze skonfigurowaną obsługą php oraz scgi (jako, że lighttpd ma scgi wbudowane, dalszy opis będzie zakładał jego wybór – jeśli chcemy jednak Apache, zajrzyjmy na stronę główną wTorrenta [8]). Ponadto potrzebny będzie `xmllrpc-c`, SQLite oraz sam rTorrent w wersji ≥ 0.78 skompilowany z obsługą `xmllrpc-c` (`./configure --with-xmllrpc-c ; make ; make install`). Pierwszy krok konfiguracji to dodanie do `.rtorrent.rc` wpisu `scgi_port = 127.0.0.1:5000`. Następnie edytujemy `/etc/lighttpd/lighttpd.conf` – w sekcji `server.modules`

dodając linię `"mod_scgi"`, oraz na końcu pliku dopisując fragment: Listing 1.

Gdy już to zrobiliśmy, możemy przejść do pobrania wTorrenta ze strony [4], rozpakowania – koniecznie do katalogu, do którego lighttpd ma dostęp (domyślnie np. `/var/www/wtorrent`) oraz edycji jego pliku konfiguracyjnego `home.conf.php`, który znajduje się w podkatalogu `conf/`. Po upewnieniu się, że lighttpd ma prawa do zapisu i odczytu do tych plików możemy przeglądarką uruchomić plik `install.php`. Ostatnim krokiem będzie usunięcie tego pliku, uru-

chomienie rTorrenta oraz zrestartowanie serwera. Jeżeli podczas powyższych wskazówek nie wystąpiły żadne błędy, od tej chwili możemy cieszyć się schludnym, zielonym interfejsem wTorrenta, który uruchamiamy wpisując w przeglądarce `http://localhost/sciezka/do/wtorrent/index.php`.

N2hell

Jego przewagą nad wTorrentem jest przede wszystkim:

- łatwość konfiguracji;
- konieczność posiadania jedynie rTorrenta (z obsługą `xmllrpc-c`), javy (jre ≥ 1.50) oraz przeglądarki;
- możliwość uruchomienia także pod Windows czy Mac OS X.

Wady? Z pewnością zużycie zasobów – na komputerze mającym 384MB ram i procesor Athlon taktowany z częstotliwością 1GHZ użycie procesora w chwilach korzystania z interfejsu wynosiło nawet 7%, co przy teoretycznym (wskazany przez top) nie sięgającym 1% zużyciu zasobów podczas korzystania z wTorrent jest wręcz porażką szczególnie, jeżeli chcemy n2hell wykorzystać na starych komputerach będących zwykle routerami dla małej sieci. Głównie wpływa na to użycie javy, która będąc językiem interpretowanym z natury jest wolna. Dużym minusem może być też fakt, że n2shell korzysta ze swojego serwera www, który może nie być tak bezpieczny jak Apache czy lighttpd. Jeżeli jednak zdecydujemy się na korzystanie z niego, poniżej możemy przeczytać, jak to zrobić. Mając zainstalowany rTorrent z obsługą `xmllrpc-c` (patrz wTorrent) oraz javę, możemy ściągnąć [8], rozpakować wszystko oraz uruchomić z kontekstu roota plik `n2hell.sh` (`sh n2hell1.sh`). Wcześniej do pliku `~/rtorrent.rc` należy dodać wpis identyczny, jak w wTorrentcie: `scgi_port = 127.0.0.1:5000`. Ostatnim krokiem będzie wejście przeglądarką na adres: `http://localhost:8888`. Tak, to wszystko – pozostała nam jedynie graficzna konfiguracja nakładki, po której n2hell jest gotowy do działania. Prawda, że proste? 🧐



W Sieci

- [1] <http://libtorrent.rakshasa.no/>
- [2] <http://libtorrent.rakshasa.no/wiki/Download>
- [3] <http://jakilinux.org/aplikacje/konsola/screen-sztuczki/>
- [4] <http://libtorrent.rakshasa.no/wiki/RTorrentUserGuide>
- [5] <http://www.g-loaded.eu/2007/06/23/rtorstat-a-simple-rtorrent-status-web-page-generator/>
- [6] <http://libtorrent.rakshasa.no/attachment/wiki/RTorrentCommonTasks/rtorrentlnit.sh?format=raw>
- [7] <http://canbruixa.homelinux.net/wt/>
- [8] <http://code.google.com/p/n2hell/>



Klawiszologia

Dodawanie i usuwanie torrentów:

- `[backspace]` – Pozwala dodać torrenta do ściągnięcia z lokalnego bądź zdalnego komputera;
- `[ctrl]+[s]` – Rozpoczyna pobieranie;
- `[ctrl]+[d]` – Zatrzymuje bądź usuwa z listy zatrzymane wcześniej pobieranie;
- `[+],[–]` – Zmienia priorytet ściągnięcia torrenta;
- Kontrola prędkości:
- `[a],[s],[d]` – Zwiększa limit wysyłania o 1,5,50 KB;
- `[z],[x],[c]` – Zmniejsza limit wysyłania o 1,5,50 KB;
- `[A],[S],[D]` – Zwiększa limit pobierania o 1,5,50 KB;
- `[Z],[X],[C]` – Zmniejsza limit pobierania 1,5,50 KB;
- Interfejs:
- `[ctrl]+[q]` – Wyłącza program; wciśnięty podwójnie wymusza szybkie wyłączenie;
- `[góra],[dół]` – Wybierz element z listy;
- `[lewo]` – Powrót do poprzedniego ekranu;
- `[prawo]` – Przejście do widoku pobierania, bądź do wybranej zakładki z tego widoku;
- `[l]` – Pokaż log; spacja żeby wyjść;
- `[space]` – Zmień priorytet pobierania pliku (zakładka File list z widoku pobierania);
- `[1]` – Wyświetla wszystkie pobierania;
- `[2]` – Wyświetla wszystkie pobierania, posortowane wg nazwy;
- `[3]` – Wyświetla uruchomione torrenty;
- `[4]` – Wyświetla zatrzymane torrenty;
- `[5]` – Wyświetla pobrane torrenty;
- `[6]` – Wyświetla pobierane torrenty;
- `[7]` – Wyświetla haszowane torrenty;
- `[8]` – Wyświetla torrenty, które udostępniamy innym;



O autorze

Autor zajmuje się Linuxem od 4 lat rozpowszechniając go w lokalnych szkołach. Hobby to wszystkie aspekty związane z Debianem, Linuxem w ogólności oraz gra na klarnecie.

Kontakt z autorem: lukiasz@gmail.com