

## Instalacja oprogramowania na serwerze

System Linux z dystrybucji Debian jest bardzo elastycznym i prostym w obsłudze systemem operacyjnym. Ze względu na prostotę instalacji wybierany jest jako podstawa dla innych dystrybucji tegoż systemu – korzysta z niego m.in. Ubuntu oraz system Linux Mint (który korzysta także z Ubuntu), Edubuntu, routerowa Vyatta i wiele innych (lista wszystkich dystrybucji systemu Linux – [https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux\\_Distribution\\_Timeline.svg](https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)).

Zanim opisane zostaną sposoby instalacji oprogramowania w systemie Linux należy najpierw wspomnieć jak ta instalacja wygląda. Poszczególne programy są po prostu kopiowane do odpowiednich katalogów (jeżeli mają być dostępne dla wszystkich użytkowników systemu) lub dowiązywane dla danego użytkownika (jeżeli tylko on ma korzystać z danego programu). Ponadto Linux nie posiada żadnego rejestru, dodatkowych procedur deinstalacji oprogramowania itp. Równie dobrze użytkownik sam może skopiować odpowiednie pliki do katalogów – będzie to równoważne z instalacją (pliki konfiguracyjne znajdują się bezpośrednio w katalogu aplikacji).

Podstawowym źródłem nowego oprogramowania są bogate repozytoria systemowe. Repozytorium to po prostu serwer z oprogramowaniem. Administratorzy repozytoriów dbają o dodawanie do niego oprogramowania. Systemowe (domyślne) repozytoria posiadają kilka tysięcy aplikacji – zaczynając od systemowych, poprzez diagnostyczne czy użytkowe, na serwerowych kończąc. Prócz tego istnieje możliwość dodawania nowych repozytorium zarówno firm trzecich jak i prywatnych (opisane w dalszej części). Niepodważalną zaletą instalacji oprogramowania z repozytorium jest jego automatyczna aktualizacja – jeżeli system wykryje nowszą wersję danej aplikacji względem zainstalowanej lokalnie to pobierze i sam ją zaktualizuje. Odpada więc problem śledzenia zmian, pobierania i instalowania od nowa aplikacji/programu (i jeszcze ewentualnej deinstalacji poprzedniej wersji).

Innym sposobem instalacji jest pobranie plików samo-instalujących się w systemie. Tego typem plików są tzw. paczki (package) z rozszerzeniem deb (np. wine.deb). Instalacja tego typu plików działa tak jak instalacja w systemie Windows przy pomocy plików exe/msi.

Kolejnym sposobem instalacji jest tzw. archiwum samorozpakowujące (najczęściej posiadające rozszerzenie run, rzadziej sh). Ten typ instalacji odpowiada instalacji z archiwum SFX (WinRAR) w systemie Windows. Wybierany jest przez niektórych producentów oprogramowania ze względu na uniwersalność – działają na większości dystrybucji.

Programy/aplikacje dla systemu Linux mogą być także dostarczone po prostu w wersji spakowanej. W tym przypadku wystarczy rozpakować archiwum (najczęściej posiadające katalog aplikacji, ewentualnie trzeba wypakować zawartość do utworzonego przez siebie katalogu).

Ostatnim, „najtrudniejszym” sposobem instalacji oprogramowania jest „własnoręczna” kompilacja oprogramowania. Obecnie coraz rzadziej praktykowana, wykorzystywana głównie w przypadku bardzo specyficznych wymagań aplikacji (specyficzne biblioteki i/lub konfiguracje innego oprogramowania). Cała filozofia takiej instalacji sprowadza się do:

- a) rozpakowania pliku archiwum
- b) przejścia do tego katalogu
- c) wydania polecenia ./configure (czasem ./setup, ./config.sh lub podobnego)
- d) po pozytywnym zakończeniu konfiguracji wydania polecenia make install clean (powoduje zbudowanie aplikacji, zainstalowanie jej oraz wyczyszczenie katalogu budowania – by nie dublować plików)

### 1. Instalacja z repozytorium

Służą do tego następujące polecenia:

```
apt-get install <nazwa_pakietu>
```

lub

aptitude install <nazwa\_pakietu>

Pierwsze z zaprezentowanych poleceń jest poleceniem tzw. podstawowym. Istnieje w każdej dystrybucji Debiana oraz systemach Debianopochodnych. Jako <nazwa\_pakietu> podajemy nazwę aplikacji, jaką chcemy zainstalować w systemie. Oczywiście możemy wiedzieć co chcemy zainstalować, jednak nie znać nazwy, pod jaką aplikacja widnieje w repozytoriach. W takim wypadku wykorzystujemy inne polecenie:

apt-cache search <szukana\_fraza>

gdzie <szukana\_fraza> to potencjalna nazwa pakietu (bądź opisu pakietu).

Z kolei polecenie aptitude to skrypt napisany przy użyciu języka Python (wymaga posiadanie parsera tegoż języka). Pozwala na instalację, wyszukiwanie czy odinstalowanie pakietu/pakietów. Ponadto, w przeciwieństwie do apt, w przypadku konfliktów zainstalowanych pakietów bądź braku wzmaganych pakietów, proponuje najlepsze rozwiązania (jeżeli dane rozwiązanie nie przypadnie nam do gustu zaproponuje inne). Jako takie polecenie jest charakterystyczne jedynie dla dystrybucji Debian oraz Ubuntu (pozostałe pozwalają na instalację tego narzędzia lecz nie posiadają go domyślnie).

## **POZOSTAŁE MOŻLIWOŚCI APT**

a) usuwanie pakietów

apt-get remove <nazwa\_pakietu>

b) usuwanie pakietów wraz z plikami konfiguracyjnymi (tak zwane całkowite wykorzenienie pakietu)

apt-get purge <nazwa\_pakietu>

c) aktualizacja listy dostępnych pakietów w repozytoriach (aktualizacja bazy wiedzy naszego systemu)

apt-get update

d) aktualizacja pakietów do wersji dostępnej w repozytoriach

apt-get upgrade

e) aktualizacja dystrybucji naszego systemu (pełna aktualizacja wszystkich pakietów, wraz z powiązaniem)

apt-get dist-upgrade

f) naprawa uszkodzonych pakietów (problem z instalacją nowego pakietu/niespełnione zależności pakietu, których nie mógł naprawić instalator)

apt-get install -f

g) usunięcie nieużywanych/przestarzałych pakietów

apt-get autoremove

## DODAWANIE NOWEGO ADRESU REPOZYTORIUM

Przeważnie dodawanie nowych adresów repozytoriów do serwerowej wersji systemu nie jest najlepszym pomysłem z wielu względów. Najważniejszym jest oczywiście bezpieczeństwo – repozytoria nieoficjalne mogą w zasadzie zawierać, prócz potrzebnych aplikacji, także te potencjalnie niebezpieczne, szpiegujące bądź pozwalające na przejęcie kontroli nad naszym serwerem. Ponadto serwer posiadający aplikacje spoza oficjalnej dystrybucji może po prostu zacząć działać niestabilnie (w zależności od wersji aplikacji).

Czasem jednak znajdzie potrzeba używania nowszych wersji programów niż w oficjalnej wersji stabilnej. Wtedy trzeba edytować plik `/etc/apt/sources.list`, w którym znajdują się adresy naszych repozytoriów.

Adresy można edytować ręcznie np. w notatniku konsoli nano:

```
nano /etc/apt/sources.list
```

Po wydaniu tego polecenia otworzy nam się plik, który będzie wyglądał mniej więcej tak (w zależności od posiadanej wersji systemu):

```
#
# deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 NETINST Binary-1 2$
#deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 NETINST Binary-1 20$
deb http://ftp.pl.debian.org/debian/ wheezy main
deb-src http://ftp.pl.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
# wheezy-updates, previously known as 'volatile'
deb http://ftp.pl.debian.org/debian/ wheezy-updates main
deb-src http://ftp.pl.debian.org/debian/ wheezy-updates main
```

[ Wczytano 15 linii ]

^G Pomoc    ^O Zapisz    ^R Wczyt.plik   ^Y Poprz.str.   ^K Wytnij    ^C Bież.poz.  
^X Wyjdź    ^J Wyjustuj   ^W Wyszukaj   ^V Nast.str.   ^U Wklej    ^T Pisownia

Każdy wpis zaczyna się od słowa `deb` mówiącego narzędziu `apt`, że mamy do czynienia z adresem repozytorium; `deb` oznacza, że adres zawiera binarne paczki instalacyjne, natomiast `deb-src` iż serwer posiada kody źródłowe dostępnych aplikacji. Najczęściej są to dwa identyczne adresy aczkolwiek mogą się one od siebie nieznacznie różnić.

Kolejna część (po spacji) to po prostu adres URL serwera z paczkami; zamiast adresu domenowego może być adres IP (np. jeżeli tworzymy swoje lokalne repozytorium na potrzeby domowe/firmowe).

Trzecia część to nazwa aktualnej wersji wydania systemu. Mogą być podana nazwy kodowe (na rzucie mamy nazwę kodową `wheezy`; ponadto może być np. `sid` czy `jessie`) lub poprzez określenie typu wydania (`stable`, `oldstable`, `testing`, `experimental`, `unstable`).

Ostatnią częścią są tzw. komponenty. Może występować do 3 niezależnych komponentów (oddzielonych spacjami):

- `main` – oznacza, że pakiety w tym repozytorium składają się jedynie z takich, które zostały ujęte w głównym wydaniu systemu; są zgodne z polityką DFSG (Wytyczne Debiana dotyczące Wolnego Oprogramowania)

- contrib – paczki są kompatybilne z DFSG lecz mogą także być zależne od oprogramowania firm trzecich (tzw. oprogramowania niewolnego, firm trzecich itp.)
- non-free – paczki z aplikacjami nie spełniającymi zasad DFSG; mogą to być np. sterowniki do kart graficznych firm nVidia/AMD, kodeki DVD, oprogramowanie licencjonowane itp.

Przykładowa modyfikacja repozytoriów dla naszej wersji systemu pozwalająca także na korzystanie z aplikacji niewolnych:

```
deb http://ftp.pl.debian.org/debian/ wheezy main non-free contrib
deb-src http://ftp.pl.debian.org/debian/ wheezy main non-free contrib
```

```
deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-free
```

```
deb http://ftp.pl.debian.org/debian/ wheezy-updates main non-free contrib
deb-src http://ftp.pl.debian.org/debian/ wheezy-updates main non-free contrib
```

**INFORMACJA:** Rozpoczynając linię od znaku '#' (na zrzucie niebieski kolor) powodujemy wstawienie komentarza – cała linia tekstu jest pomijane przez narzędzie apt, a my możemy sobie opisać po co wstawiliśmy dane repozytorium bądź możemy na chwilę wyeliminować dane repozytorium (np. celem instalacji innego pakietu w systemie)

**PROSZĘ PAMIĘTAĆ, ŻE KAŻDRAZOWA ZMIANIA PLIKU SOURCES.LIST WYMAGA UŻYCIA POLECENIA:**

```
apt-get update
```

### **Przykłady adresów repozytoriów:**

Nowsze oprogramowanie (głównie praca biurowa):

```
deb http://ftp.pl.debian.org/debian/ wheezy-backports main
```

Kodeki multimedialne:

```
deb http://www.deb-multimedia.org/ stable main non-free
deb-src http://www.deb-multimedia.org/ stable main non-free
```

Dla przeglądarki Opera (inne aplikacje także mają swoje repozytoria):

```
deb http://deb.opera.com/opera/ stable non-free
```

**INFORMACJA:** System Ubuntu (i Ubuntopodobne) posiadają bardzo łatwy system dodawania repozytoriów, np. poprzez podanie polecenia:

```
add-apt-repository ppa:yannubuntu/boot-repair
```

Tak naprawdę jest to narzędzie systemu Debian. Aby z niego korzystać trzeba je po prostu zainstalować poprzez odpowiednią paczkę programów:

```
sudo apt-get install python-software-properties
```

lub poprzez (bardziej zalecane)

```
sudo apt-get install software-properties-common
```

Proszę także pamiętać, że repozytoria Ubuntu powinny bez problemu działać z systemem Debian (choć nie ma na to gwarancji!)

## NARZĘDZIE APTITUDE

Jeżeli chodzi o instalację poprzez narzędzie aptitude to polecenia są niemal identyczne jak w przypadku apt-get/apt-cache z tym, że wszystko odbywa się poprzez polecenie aptitude, np.

aptitude install <nazwa pakietu> - instaluje nowy pakiet

aptitude search <nazwa pakietu> - przeszukuje repozytorium celem znalezienia nazwy pakietu (opisu) pasującego do podanego wzorca <nazwa pakietu>

aptitude purge <nazwa pakietu> - usuwa pakiet instalacyjny wraz z plikami konfiguracyjnymi.

**INFORMACJA:** WSZYSTKIE polecenia systemowe posiadają swoje pliki pomocy, z których możemy skorzystać. Większość informacji, w przypadku wybrania języka polskiego, jest przetłumaczonych. Aby sprawdzić pomoc dla polecenia apt-get należy wydać polecenie:

apt-get - -help LUB apt-get -h

Jeżeli chcemy uzyskać pełne informacje na temat wybranego polecenia w konsoli wpisujemy:

man apt-get

Taki podręcznik przeglądamy za pomocą strzałek, klawiszy Page Up/Page Down. Jeżeli chcemy wyjść z podręcznika wciskamy klawisz [Q].

**INFORMACJA:** Wszystkie polecenia ingerujące w system operacyjny wykonujemy jako użytkownik główny (root). Zwykły użytkownik może co prawda wyświetlać sobie pliki pomocy oraz instalować oprogramowanie w swoim katalogu, jednak jeżeli aplikacja ma być dostępna dla wszystkich użytkowników systemu/wymaga dodania składników w systemie to musi być ona instalowana przez administratora systemu. Przełączania pomiędzy użytkownikiem a administratorem można dokonać poleceniem:

su

Po podaniu prawidłowego hasła zalogowani zostaniemy jako użytkownik root.

## 2. Instalacja z samodzielnych paczek instalacyjnych (pliki deb)

Jeżeli w repozytorium nie ma jakiejś aplikacji nie znaczy to, że jej nie ma. Prawdopodobnie będziemy musieli dokonać instalacji w sposób samodzielny, tj. pobrać plik (z rozszerzeniem deb) i zainstalować poprzez odpowiednie polecenie w systemie (dpkg).

### a) pobranie pliku instalacji

pierwszym etapem jest pozyskanie pliku instalacyjnego. O ile w systemach z nakładkami graficznymi wystarczy wejść na odpowiednią stronę i rozpocząć pobieranie, o tyle w konsoli ciężko będzie otworzyć jakąkolwiek stronę. Owszem są przeglądarki stron dostępne z poziomu linii poleceń, jednak może to być mało wygodne rozwiązanie. O wiele lepszym pomysłem jest pobranie paczki poprzez adres WWW za pomocą polecenia wget; wystarczy podać adres URL pliku do pobrania, a narzędzie wykona za nas całą resztę pobierając plik do katalogu, w którym aktualnie

przebywamy. Przykład polecenia:

wget <http://ftp.ps.pl/pub/apache//directory/apacheds/dist/2.0.0-M19/apacheds-2.0.0-M19-amd64.deb>

## **b) instalacja właściwa**

Jeżeli posiadamy już plik deb to instalujemy go poleceniem:

```
dpkg -i <nazwa pliku>
```

Od tego momentu posiadamy zainstalowany dodatkowy pakiet pod odpowiednią nazwą (apt-cache powinno go widzieć w spisie zainstalowanych paczek oprogramowania).

Usunięcie wskazanego pakietu z systemu

```
dpkg -r <nazwa pakietu>
```

albo

```
dpkg -P <nazwa pakietu>
```

Może się też zdarzyć, że pewne zgodności nie będą zachowane dla instalowanego oprogramowania. Wtedy możemy użyć polecenia - -force-\* . Powoduje ono wymuszenie pewnych uwarunkowań, np.

- -force-all – ustawia wymuszenie instalacji pomimo jakichkolwiek problemów (niezgodności) systemowych

- -force-downgrade – instalacja pakietu ze starszą wersją aplikacji niż jest zainstalowana w systemie

- -force-architecture – instalacja pakietu w przypadku niezgodności architektury sprzętowej (np. paczka 32 bitowa na 64 bitowy system)

## **3. Instalacja z archiwum samorozpakowującego/rozpakowanie do katalogu**

Dość popularnym sposobem dostarczania niektórych narzędzi jest umieszczanie wszystkiego w archiwum samorozpakowującym (np. sterowniki do kart graficznych, maszyny wirtualne) lub w prostu spakowane np. tarem, które wystarczy wypakować do katalogu i używać. Instalacje samorozpakowujące instaluje się poprzez:

a) pobranie zawartości (patrz punkt 2)

b) nadanie plikowi atrybutu wykonywalności (domyślnie ŻADEN POBRANY PLIK nie posiada praw wykonywania – względy bezpieczeństwa):

```
chmod +x <nazwa pliku>
```

c) rozpoczęcie instalacji:

```
./<nazwa pliku>
```

Bardzo ważną częścią polecenia jest './'. Bez tego przedrostka system będzie próbował wyszukać i wykonać polecenie poprzez dostępne aplikacje w systemie (które mogą mieć podobną/taką samą

nazwę). Przedrostek informuje, że system mam wykonać polecenie w katalogu bieżącym.

Jeżeli chodzi o rozpakowanie to sprawa jest jeszcze prostsza. Najczęściej „instalacja” sprowadza się do wydania polecenia:

```
tar -xf <plik archiwum>
```

Program tar wypakuje nam całe archiwum do bieżącej lokalizacji (jeżeli przebywamy np. w katalogu /home/test/Program to właśnie do tego katalogu zostanie wypakowane archiwum).

**WAŻNE:** Archiwa samorozpakowujące posiadają często także funkcję - -extract. Dzięki niej możemy otrzymać dostęp do plików znajdujących się w nim i np. podmienić ich zawartość przed instalacją. Przykładem wykorzystania takiej operacji jest np. dokładanie łątek do sterowników firmy AMD by te działały z najnowszymi jądrami systemu (np. 3.18.x)

#### 4. Kompilacja ze źródeł.

Niekiedy, w bardzo szczególnych przypadkach, zmuszeni będziemy do kompilowania oprogramowania ze źródeł. Przeważnie tego typu sytuacja ma miejsce w chwili gdy wybieramy mniej popularne narzędzie, dla którego nikt nie zajął się stworzeniem binarnej wersji. W tym wypadku aplikacja pobierana jest albo z repozytorium SVN (Subversion) albo git, ewentualnie dostarcza w skompresowanej postaci (najczęściej tar). Instalacja przebiega następująco:

a) pobranie aplikacji

- SVN:

```
wget --user=yourusername --ask-password -m -np <adres URL repozytorium SVN>
```

jeżeli svn jest dostępny dla anonimowego użytkownika można pominąć flagę user oraz ask-password

- git

```
git clone <URL do repozytorium git, najczęściej po HTTPS>
```

- jako spakowany plik (patrz poprzedni rozdział)

b) rozpakowanie (jeżeli było pobrane jako archiwum; pobranie poprzez SVN/git nie wymaga rozpakowywania – pobiera się pliki projektu a nie archiwum)

c) konfiguracja przed budowaniem

Tutaj najczęściej polega to na wydaniu polecenia ./configure, jednak nazwa skryptu może się różnić w zależności od projektu (trzeba zapoznać się z jego dokumentacją). Skrypty te mają sprawdzić ustawienia systemowe (ścieżki do bibliotek, zainstalowane komponenty itp.) by jak najlepiej skonfigurować narzędzie do naszych potrzeb

d) proces budowy rozpoczynamy poleceniem

```
make
```

e) po zbudowaniu projektu wydajemy polecenie

```
make install
```

f) jeżeli instalacja przebiegła poprawnie można po sobie posprzątać

```
make clean
```

Oczywiście istnieje możliwość użycia także polecenia 'make install clean' jednak w tym wypadku gdy kompilator napotka błąd możemy posiadać źle zainstalowaną aplikację, która to z kolei może prowadzić do destabilizacji całego systemu.

Opisywane wyżej instalacje (z kodów źródłowych) są, pomimo pewnych niedogodności,

najlepszym rozwiązaniem jeżeli chodzi o oprogramowanie. Dzięki temu każda aplikacja jest kompilowana ściśle pod nasz sprzęt/nasz system operacyjny co powoduje, że działa znacznie stabilniej niż wersja „dla wszystkich”. Proszę pamiętać, że repozytoria systemowe również posiadają kody źródłowe aplikacji – można pobrać ich kod, by następnie je skompilować i dopiero na końcu zainstalować.

#### ZADANIA:

1. Co to są backports? Jak dodać je do repozytoriów systemu. Czy po ich dodaniu coś się podziało (aktualizacje systemowe, jakieś pakiety zostały wyrzucone, itp.). Jaki wpływ może mieć dodanie repozytorium backports do systemu serwerowego?
2. Czy w ramach jednego polecenia można zainstalować kilka pakietów? Jeżeli tak to w jaki sposób?
3. Proszę do każdej opisanej metody automatycznej instalacji (apt, dpkg) wybrać minimum 5 najbardziej przydatnych opcji, które niemal zawsze będą przydatne przy obsłudze systemu serwerowego.
4. Proszę znaleźć jakie aplikacje mogą być przydatne przy codziennej administracji serwerem (np. do zarządzania plikami) i zainstalować je w swoim serwerze.

#### Źródła:

<http://www.debian.pl/threads/1408-Repozytoria-Debiana-wpisy-w-pliku-etc-apt-sources-list>

<https://wiki.debian.org/SourcesList>

<http://www.ccl.net/ccl/software/UNIX/updates/redhat/apt-howto/how-to-use-apt-cache.html>