

Pracownia Technik Obliczeniowych

Instalowanie oprogramowania

Paweł Daniluk

Wydział Fizyki

Wiosna 2016



Standardowy układ katalogów

Systemy UNIXowe mają z reguły ten sam układ katalogów.

| | |
|--------|---|
| /bin | Podstawowe pliki wykonywalne dostępne dla wszystkich (np. cat, ls, cp). |
| /boot | Pliki rozruchowe (np. kernel, initrd). |
| /dev | Pliki urządzeń (ang. device; np. /dev/null). |
| /etc | Pliki konfiguracyjne. |
| /home | Katalogi domowe użytkowników . |
| /lib | Biblioteki dla programów z katalogów /bin i /sbin. |
| /mnt | Punkt montowania innych niż natywny systemów plików. |
| /media | Punkty montowań dla nośników wymiennych. |
| /opt | Opcjonalne aplikacje. |
| /proc | Wirtualny system plików proc informujący o stanie systemu. |
| /root | Katalog domowy użytkownika root. |
| /sbin | Pliki wykonywalne do zarządzania systemem (np. init, route, ifup). |
| /tmp | Pliki tymczasowe (mogą być kasowane przy restarcie). |
| /usr | Drugorzędowa hierarchia dla danych, dane tylko do odczytu. |

Standardowy układ katalogów

| | |
|--------------|---|
| /usr/bin | J. w., ale nie wymagane do uruchomienia systemu. |
| /usr/include | Standardowe pliki nagłówkowe. |
| /usr/lib | J. w. |
| /usr/sbin | J. w., ale nie wymagane do uruchomienia systemu. |
| /usr/share | Dane niezależne od architektury. |
| /usr/src | Kody źródłowe. |
| /usr/local | Trzeciorzędowa hierarchia danych lokalnych. |
| /var | Pliki często ulegające zmianom (n.p. logi, bazy danych). |
| /var/lock | Pliki blokady zasobów będących w użyciu. |
| /var/log | Logi różnych aplikacji. |
| /var/mail | Skrzynki pocztowe użytkowników. |
| /var/run | Informacje o działaniu systemu od ostatniego jego uruchomienia. |
| /var/spool | Miejsce przechowania oczekujących zadań. |

Systemy pakietów

Każdy nowoczesny UNIX/Linux ma jakiś system zarządzania instalacją oprogramowania, który zapewnia:

- 1 możliwość deinstalacji
- 2 zachowanie zależności (wymagań i wykluczeń)
- 3 aktualizację wersji

Przykładowe systemy pakietów

- 1 RPM (RedHat i pokrewne)
- 2 APT (Debian i pokrewne)
- 3 MacPorts, Homebrew, Fink (dla OS X)

Niektóre narzędzia mają własne systemy pakietów. Np.:

- 1 Python
- 2 LaTeX
- 3 R

Kiedy system pakietów zawodzi

Oprogramowanie naukowe (i nie tylko) nie spełnia opisanych standardów i wymaga specjalnego podejścia do instalacji.

Czasem trzeba własnoręcznie skompilować narzędzie.

Gdzie instalować

Można instalować w:

- 1 `/usr/local` – wymaga uprawnień superużytkownika, instalacja dostępna dla wszystkich
- 2 `$HOME/local` – u siebie każdy może wszystko
- 3 jako moduł

Etapy instalacji

- 1 Pobranie i rozpakowanie źródeł
- 2 Przygotowanie kompilacji – weryfikacja wymagań, konfiguracja
- 3 Kompilacja
- 4 Instalacja (kopiowanie w docelowe miejsce)

Standardowa procedura

```
./configure
```

```
make
```

```
make install
```


configure

Sprawdza, czy kompilacja jest możliwa. Lokalizuje potrzebne biblioteki. Ustawia parametry kompilacji.

- 1 miejsce instalacji
- 2 moduły do skompilowania
- 3 inne opcje

make

Nadzoruje kompilację.

Pożyteczna opcja

make -j

Pakiety Pythonowe

Python szuka pakietów w bieżącym katalogu oraz domyślnych lokalizacjach.

```
>>> sys.path
['', '/home/pawel', '/usr/lib64/python26.zip',
'/usr/lib64/python2.6', '/usr/lib64/python2.6/plat-linux2',
'/usr/lib64/python2.6/lib-tk', '/usr/lib64/python2.6/lib-old',
'/usr/lib64/python2.6/lib-dynload',
'/usr/lib64/python2.6/site-packages']
```

Pakiety, które mają być dostępne dla wszystkich, można instalować w `site-packages`.

Repozytorium pakietów Pythonowych.

<http://pypi.python.org>

Setuptools

Zestaw narzędzi do instalowania pakietów. Zawiera `easy_install`.

Instalacja

http://bootstrap.pypa.io/ez_setup.py

Zadanie 1

Pobierz kod źródłowy najnowszej wersji Pythona 2, skompiluj i zainstaluj w swoim katalogu domowym.

Zadanie 2

Zainstaluj setuptools.

Zadanie 3

Zainstaluj ipython.