

Praca z kontenerami Dockera dla Lego Mindstorms EV3[4]

Jakub Arnold Postępski

22 marca 2018

1 Wstęp

Kostki Lego Mindstorms EV3[4] stosowane w laboratorium mają procesor z rdzeniem ARM926EJ-S (architektura ARM, taktowanie 300 MHz, 64 MB RAM). Stosowany system operacyjny ev3dev (oparty na Debian Wheezy). Komunikacja z kostkami przez SSH i Wi-Fi. System operacyjny zestawu instalowany jest na karcie SD.

Konfiguracja sprzętowa spowalnia proces użytkowania całego zestawu. Niewydajny proces kompilowania programów w języku C++. Wolne działanie interpretera języka Python. Wolne działanie instalatora oprogramowania apt. Przy pomocy środowiska Docker[2] opisano:

- metodę przygotowania gotowego OS z już zainstalowanymi pakietami
- proces kompilacji skróśnej

Kopiowanie gotowego kontenera na kartę SD przy pomocy programu Brickstrap[1].

2 Docker

Docker[2] to środowisko do obsługi kontenerów podobnych w działaniu do wirtualnych maszyn. Możliwe uruchamianie kontenerów z innymi architekturami niż system maszyna gospodarza. Demon dla systemu operacyjnego. Możliwość tworzenia kontenera, zachowania zmodyfikowanego kontenera oraz pobrania gotowego kontenera z dedykowanych repozytoriów. Uruchomiony kontener posiada własną etykietę. Obraz może być uruchomiony wiele razy jako różne kontenery. Do jednego kontenera można uruchomić wiele terminali.

3 Operacje na kontenerze Dockera

Dla systemu z rodziny Ubuntu. Polecenia dla użytkownika z odpowiednimi uprawnieniami.

3.1 Instalacja

Instalacja:

```
apt install docker.io
```

Uruchomienie demonów:

```
service docker.io
```

3.2 Uruchomienie kontenera

Ściągnięcie obrazu kontenera[3]:

```
docker pull ev3dev/ev3dev-jessie-ev3-generic
```

Nadanie tagu obrazowi:

```
docker tag ev3dev/ev3dev-jessie-ev3-generic ev3
```

Wyświetlenie dostępnych obrazów:

```
docker images
```

Uruchomienie i logowanie do kontenera:

```
docker run -it --name <nazwa_kontenera> ev3
```

Dodatkowo, w celu ułatwienia pracy można stosować poniższe:

- W nowej konsoli hosta, wyświetlenie uruchomionych kontenerów:

```
docker ps
```

- Logowanie do już utworzonego kontenera:

```
docker exec -it <nazwa_kontenera> bash
```

- Zabicie kontenera, z systemu hosta:

```
docker kill <nazwa_kontenera>
```

3.3 Praca z kontenerem

Po zalogowaniu do kontenera można wykonywać operacje jak dla maszyny wirtualnej. Aby nie stracić efektów pracy należy zapisać migawkę kontenera z systemu hosta.

```
docker commit <tag_obrazu> <nazwa_kontenera>
```

Powoduje to utworzenie nowego obrazu. Dzięki temu możemy wytworzyć dowolne modyfikacje systemu (takie jak instalacja oprogramowania) i potem przenieść na kartę SD.

Jeśli kontener zostanie uruchomiony poleceniem *run* z parametrem *-v*

```
docker run -it -v <katalog_hosta>:<katalog_kontenera>  
--name <nazwa_kontenera> <nazwa_obrazu>
```

to katalog o danej nazwie zostanie odtworzony w kontenerze. Dzięki temu mamy łatwą możliwość kopiowania plików pomiędzy kontenerem a maszyną hosta. Po zainstalowaniu w kontenerze:

```
apt-get install unzip bzip2 build-essential
```

otrzymujemy możliwość kompilacji skrośnej. Dla przykładowego programu, w kontenerze, we współdzielonym katalogu wykonujemy:

```
g++ -o hello hello.c
```

```
./hello
```

Następnie jesteśmy w stanie przesłać gotowy program na rzeczywistą kostkę, z systemu hosta. Po przejściu do katalogu współdzielonego:

```
scp hello robot@192.168.18.74:~/hello
```

3.4 Zapis kontenera na karcie

Należy wykonywać w konsoli hosta oraz pamiętać aby wytworzyć nową migawkę obrazu przed zapisem na kartę
Ściągniecie repozytorium:

```
git clone git://github.com/ev3dev/brickstrap
```

Skrypt programu w podkatalogu

```
brickstrap/src
```

Dla kontenera o podanym tagu tworzy obraz karty SD[1]:

```
./brickstrap.sh create-tar <nazwa_obrazu_dockera> ev3.tar  
./brickstrap.sh create-image ev3.tar ev3ros.img
```

Sprawdzenie którym urządzeniem jest karta SD:

```
fdisk -l
```

Zapis obrazu na kartę SD:

```
dd if=ev3ros.img of=/dev/sdb bs=4M
```

Jeśli hasło nie zostało zmienione, po uruchomieniu kostki należy się zalogować użytkownik: *robot*, hasło *master*

Literatura

- [1] Brickstrap github. <https://github.com/ev3dev/brickstrap>. Online; accessed 22-March-2018.
- [2] Docker website. <https://www.docker.com/>. Online; accessed 22-March-2018.
- [3] Ev3dev website. <http://www.ev3dev.org/>. Online; accessed 22-March-2018.
- [4] Lego mindstorms ev3. <https://www.lego.com/pl-pl/mindstorms>. Online; accessed 22-March-2018.