Wojskowa Akademia Techniczna

Wydział Elektroniki

**Programowanie w języku C**

Opis środowiska

Opracował:

mgr inż. Sebastian Szwaczyk

Warszawa 2020

**1. Wstęp**

Laboratoria w całości zostaną przeprowadzone wykorzystując komputery klasy PC z zainstalowanym systemem operacyjnym DEBIAN 9. Jako główną platformę do wykonania ćwiczenia wykorzystano technologię Docker.

**2. Instalacja środowiska**

Dzięki technologii Docker każdy ze studentów w prosty sposób może uruchomić środowisko testowe na swoim prywatnym komputerze.

Wymagania:

* Komputer klasy PC z systemem operacyjnym z rodziny Debian/Ubuntu (lub inny Linux z modułem openvswitch standardowo dołączonym do jądra) lub Mac OS
* Zainstalowany Docker CE (<https://www.docker.com/>)
* Zainstalowany klient GIT (<https://pl.wikipedia.org/wiki/Git_(oprogramowanie>))

Do przeprowadzenia zajęć laboratoryjnych został przygotowany obraz Dockera zawierający niezbędne oprogramowanie. Obraz Dockera dostępny jest w repozytorium ([https://hub.docker.com/r/sszwaczyk/prc-labs/](https://hub.docker.com/r/sszwaczyk/docker-sdn-labs/)), aby go pobrać należy wykonać komendę:

*docker pull sszwaczyk/prc-labs*

Po pobraniu obraz jest gotowy do użycia. W celu ułatwienia korzystania z tego obrazu przygotowano skrypty pozwalające na uruchomienia, dołączenia oraz zatrzymanie kontenera. Skrypty dostępne są poprzez repozytorium github ([https://github.com/sszwaczyk/](https://github.com/sszwaczyk/docker-sdn-labs)prc-labs), aby je pobrać należy wykonać komendę:

*git clone https://github.com/sszwaczyk/prc-labs.git*

Zostanie pobrany folder ze skryptami (run.sh, attach.sh, stop.sh).

**3. Praca z kontenerem**

**3.1 Uruchamianie**

Aby uruchomić kontener wystarczy uruchomić skrypt run.sh. Zawartość tego skryptu:

*#!/bin/bash*

*xhost +*

*docker run -tid --privileged --rm -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix --name=prc-labs sszwaczyk/prc-labs*

*attach.sh*

Skrypt wykonuje polecenie run w celu uruchomienia wcześniej pobranego obrazu sszwaczyk/prc-labs (jeśli obraz nie był pobrany to zostanie pobrany podczas działania skryptu). Następnie łączy się do uruchomionego kontenera. Opis pozostałych opcji można znaleźć w pomocy. Po uruchomieniu kontenera w konsoli powinien pojawić się jego ID oraz powinien być zalogowany użytkownik student:

*$ ./run.sh*

*access control disabled, clients can connect from any host*

*c74500f93de6b63e25ae556afbc9672500ac92d464be4a769bc4da4eafbaf31f*

*To run a command as administrator (user "root"), use "sudo <command>".*

*See "man sudo\_root" for details.*

*student@c74500f93de6:/$*

Uruchomione kontenery można sprawdzić przy pomocy polecenia (**w terminalu hosta a nie uruchomionego kontenera – inny terminal niż otwarty przez *run.sh*!**):

*docker ps*

Przykładowy wynik komendy docker ps:

*$ docker ps*

*CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES*

*9f22d9128c03 sszwaczyk/prc-labs "/bin/sh -c 'servi..." About a minute ago Up About a minute prc-labs*

Obraz Dockera został tak skonfigurowany, aby podczas startu był uruchamiany serwer SSH dzięki, któremu będzie możliwa praca z kontenerem.

**3.2 Łączenie z kontenerem**

W celu połączenia z kontenerem wystarczy użyć skryptu attach.sh. Zawartość skryptu:

*docker exec -t -i prc-labs bash -c 'su student'*

Po połączeniu dostajemy dostęp do bash jako użytkownik student.

**3.3 Łączenie z kontenerem po ssh (opcjonalnie – preferowane jest połączenie tak jak w 3.2)**

W celu połączenia z kontenerem wystarczy użyć skryptu attach\_ssh.sh. Zawartość skryptu:

*#!/bin/bash*

*ssh -X* [*student@172.17.0.2*](mailto:student@172.17.0.2)

Jak widać skrypt próbuje nawiązać połączenie ssh jako użytkownik student z adresem ip 172.17.0.2. Jest to domyślny adres ip w przypadku gdy uruchomiony jest tylko jeden kontener. Wykorzystywana jest tu również opcja -X pozwalająca na przesyłanie obrazu dla aplikacji graficznych (w przypadku wystąpienia problemów z przekazywaniem obrazu dla systemu Mac OS, można spróbować <https://fredrikaverpil.github.io/2016/07/31/docker-for-mac-and-gui-applications/>). Po nawiązaniu połączenia, należy podać hasło dla użytkownika student. Hasło to **student.**

Po udanym zalogowaniu można sprawdzić poprawność działania przesyłania obrazu wydając polecenie *eclipse*. Jeśli wszystko działa poprawnie powinno pojawić się GUI IDE eclipse.

Połączenie można zakończyć poleceniem *exit*. Rozłączenie połączenia ssh nie jest jednoznaczne z wyłączeniem kontenera!

**3.4 Zatrzymanie kontenera**

W celu zatrzymania działającego kontenera należy użyć skryptu stop.sh. Skrypt ten wykonuje jedną komendę:

*#!/bin/bash*

*docker stop prc-labs*

Zatrzymuje on kontener o nazwie prc-labs (ta sama nazwa została użyta przy poleceniu run oraz widać ją w wyniku wykonania komendy docker ps).

UWAGA!!! - Po zatrzymaniu kontenera wszystkie pliki na nim zapisane są tracone!!!

**3.4 Uruchamianie programów bez łączenia**

Przygotowano również skrypty pozwalające na uruchomienie eclipse i wireshark bezpośrednio:

wireshark\_root.sh

wireshark\_student.sh

eclipse\_root.sh

eclipse\_student.sh

Wystarczy wywołać odpowiedni ze skryptów i program zostanie uruchomiony z odpowiednimi uprawnieniami.

**4. Struktura obrazu**

W ramach zajęć będzie wykorzystywany następujący stos technologiczny:

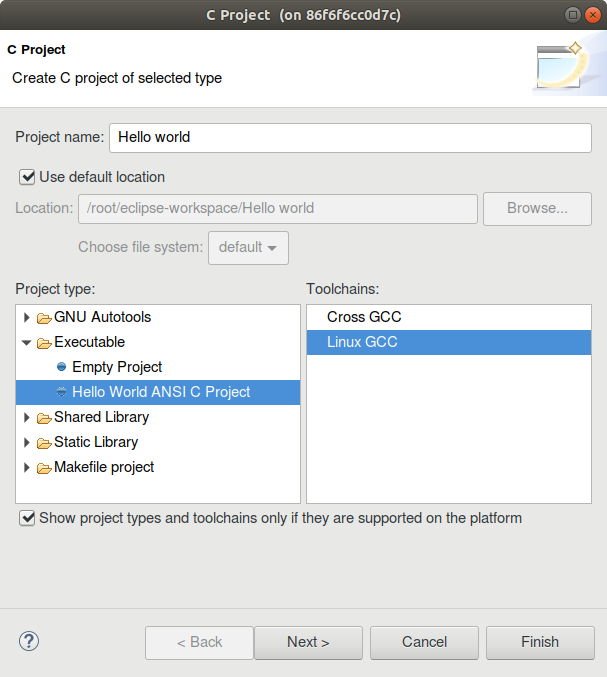
* gcc → kompilator języka C,
* Eclipse → środowisko programistyczne.

**4.1 Eclipse**

Jako środowisko programistyczne wybrano IDE Eclipse. Zostało ono pobrane do katalogu /home/student/eclipse. Dzięki dowiązaniu symbolicznemu można je uruchomić poleceniem *eclipse*. Dostępna wersja zawiera wszystkie niezbędne narzędzia ułatwiające pracę z kodem napisanym w językach C/C++.

**4.1.1 Hello World**

W celu stworzenia projektu w środowisku Eclipse wybieramy: *File→New→C Project* i następnie uzupełnić dane zgodnie z poniższym rysunkiem. Po uzupełniniu klikamy *Finish*.



Następnie należy skompilować projekt. Robimy to za pomocą ikony “młotka” w pasku górnym lub skrótem *CTRL+B*. Po zbudowaniu w konsoli powinien pojawić się komunikat oznaczający poprawną kompilację projektu:

*Building target: Hello world*

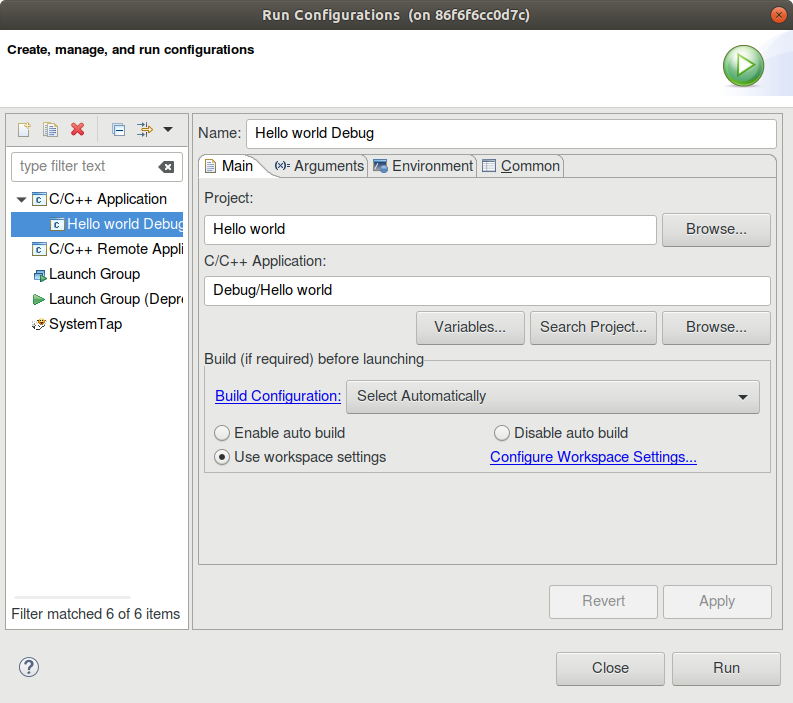
*Invoking: GCC C Linker*

*gcc -o "Hello world" ./src/Hello\ world.o*

*Finished building target: Hello world*

*06:42:52 Build Finished (took 573ms)*

Aby uruchomić wytworzony plik binarny należy wybrać *Run Configuration* rozwijane z menu obok przycisku do uruchamiania. Następnie należy wykonać dwukilk na *C/C++ Application* i powinna stworzyć się nowa konfiguracja uruchomieniowa dla projektu:



Po kliknięciu *Run* w konsoli powinien pojawić się komunikat:

!!!Hello World!!!

Kolejne uruchomienia można wykonać już przyciskiem uruchamiania w górnym pasku.

**5. Podsumowanie**

Obraz Dockera zawiera niezbędne narzędzia do wykonania zadań laboratoryjnych. Po przejściu niniejszej instrukcji Student powinien potrafić posługiwać się kontenerem Dockera oraz uruchomić program pod kontrolą IDE eclipse.