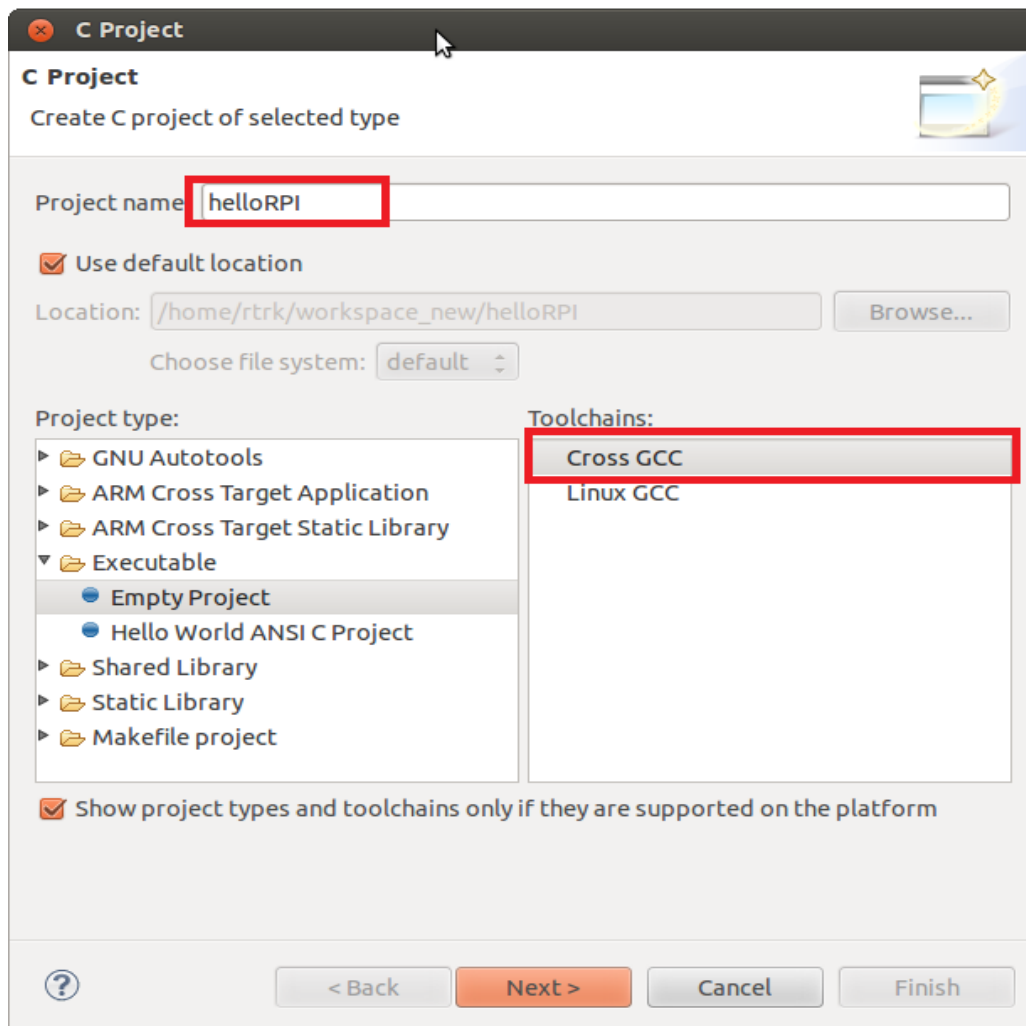




Вежба 1

У овој вежби упознаћемо се са употребом програмских алата намењених за превођење, асемблирање и повезивање програма намењених за ARM архитектуру.

1. Покрените развојно окружење *Eclipse*
2. Направите нови Це пројекат и назовите га „*helloRPI*“. Приликом прављења новог пројекта означити да ће бити коришћен скуп алата *Cross GCC*. Дијалог за прављење новог пројекта је приказан на слици испод.

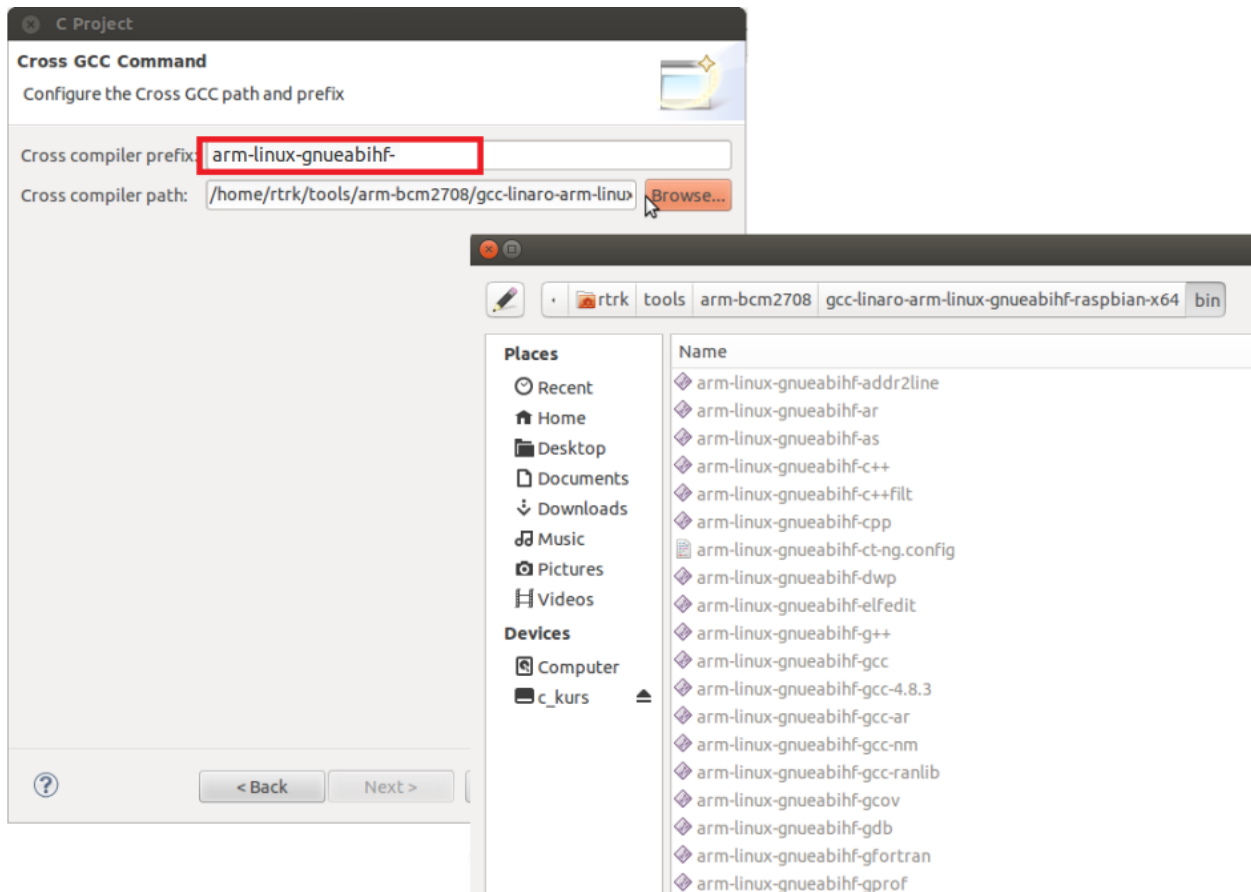


3. У наредном кораку потребно је задати који скуп алата ће бити коришћен:

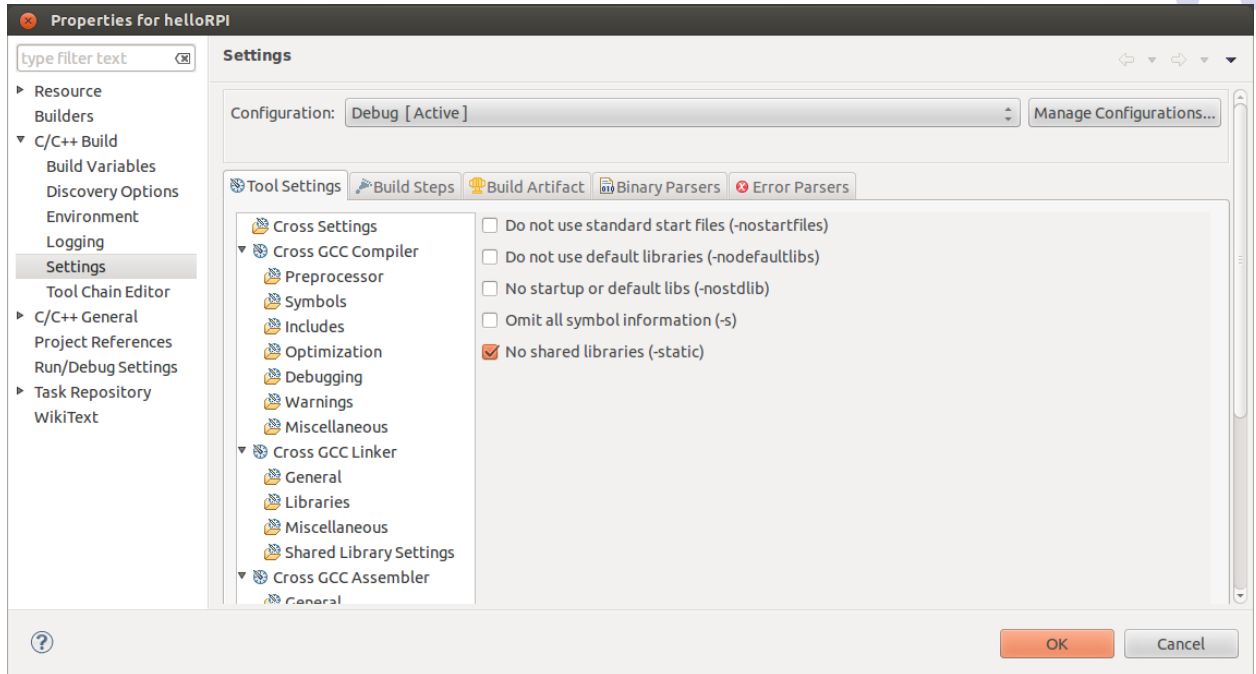


Летња пракса 2018

- Префикс: arm-linux-gnueabihf-
- Путања до алата: /home/rtrk/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin



4. Додајте нову датотеку у пројекат, са називом *main.c* и у оквору *main* функције додајте испис поруке „Hello RPi“.
5. Отворите подешавања пројекта (Project->Properties). У оквиру подешавања за повезивач (C/C++ Build -> Settings -> Cross GCC Linker), укључите опцију за повезивање коришћењем статичких библиотека (*No shared libraries*)



6. Преведите програм одабиром команде *Project -> Build*
7. Отворите конзолу и проверите да ли су коришћени програмски алати које сте навели приликом прављења пројекта и да ли је превођење програма извршено успешно.

Building file: ../main.c

Invoking: Cross GCC Compiler

```
arm-linux-gnueabi-gcc -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"main.d" -MT"main.o" -o "main.o" "../main.c"
```

Finished building: ../main.c

Building target: helloRPI

Invoking: Cross GCC Linker

```
arm-linux-gnueabi-gcc -static -o "helloRPI" ../main.o
```

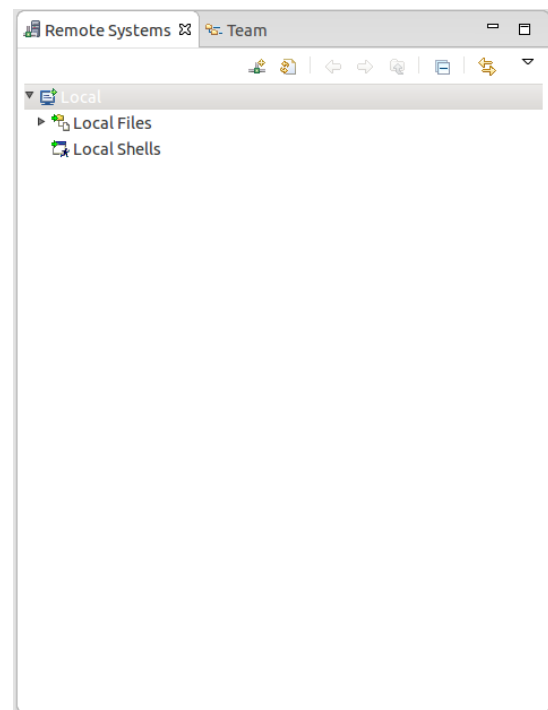
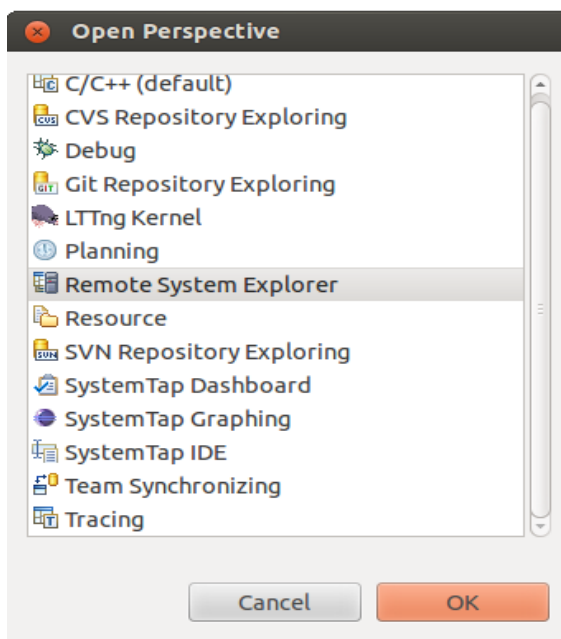
Finished building target: helloRPI



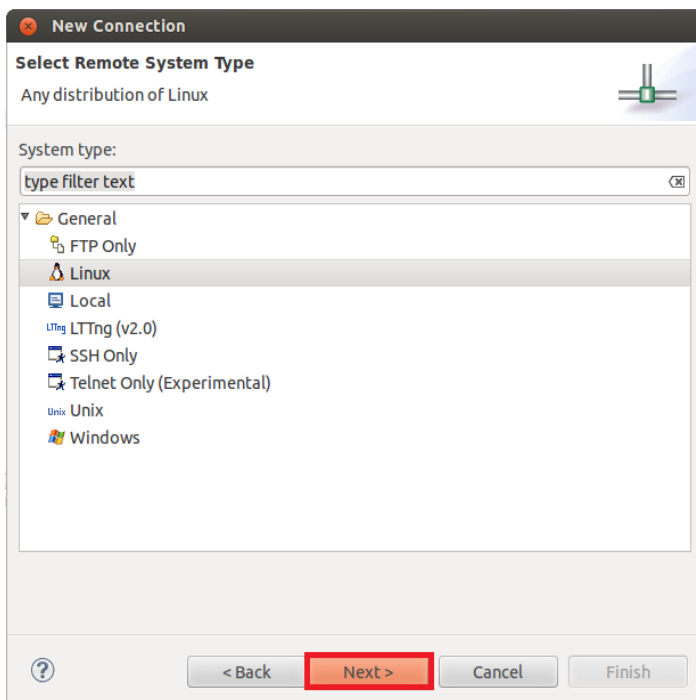
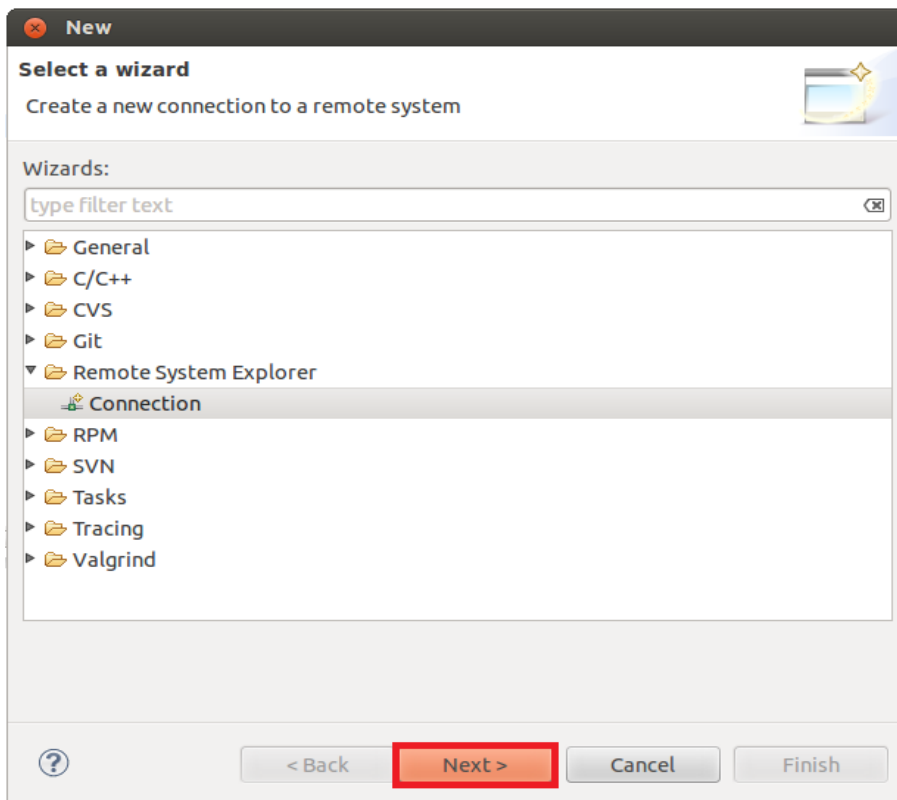
Вежба 2

У овој вежби ће бити показано на који начин се врши подешавање приступа RPI развојној плочи из развојног окружења.

1. У *Eclipse* развојном окружењу отворите „*Remote System Explorer*“ перспективу (*Window -> Open Perspective -> Other ...*).



2. У следећем кораку је потребно дефинисати начин повезивања за нови уређај. Додавање нове дефиниције се може урадити покретањем чаробњака из са *File -> New -> Other -> Connection*. Кораци су приказани на следећим сликама.





New Connection

Remote Linux System Connection

Define connection information

Parent profile:

vm-linux

Host name:

10.80.10.100

Connection name:

my-raspberry-pi

Description:

☒ Verify host name

[Configure proxy settings](#)

?

< Back

Next >

Cancel

Finish

New Connection

Files

Define subsystem information

Configuration

☐ dstore.files

☐ ftp.files

☒ ssh.files

Properties

Property	Value
----------	-------

Available Services

Ssh / Sftp File Service

SSH Connector Service

SSH Settings

Description

Work with files on remote systems using the Secure Shell (ssh) protocol.

?

< Back

Next >

Cancel

Finish



New Connection

Processes
Define subsystem information

Configuration	Properties		
<input type="checkbox"/> dstore.processes	<table border="1"><thead><tr><th>Property</th><th>Value</th></tr></thead><tbody></tbody></table>	Property	Value
Property	Value		
<input checked="" type="checkbox"/> processes.shell.linux			

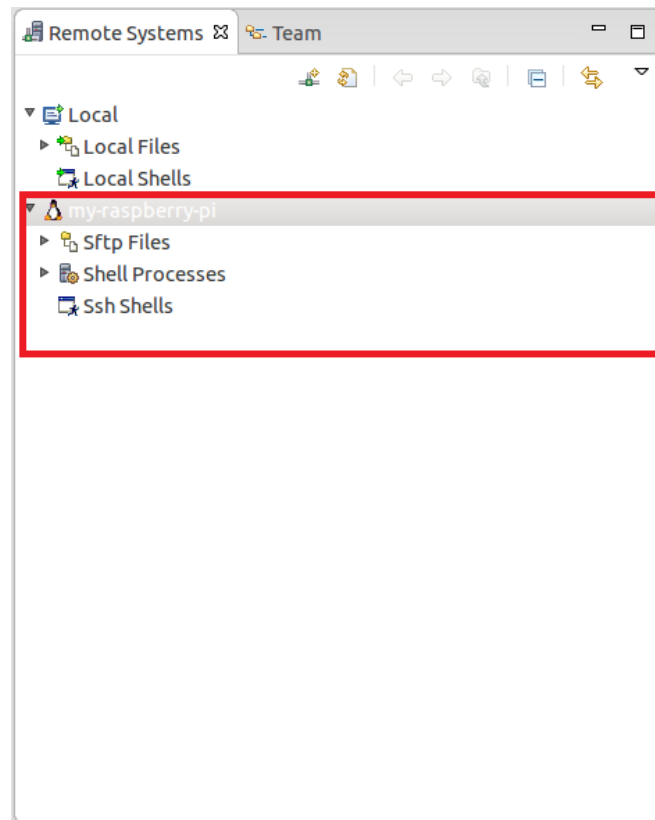
Available Services

- Shell Process Service

Description

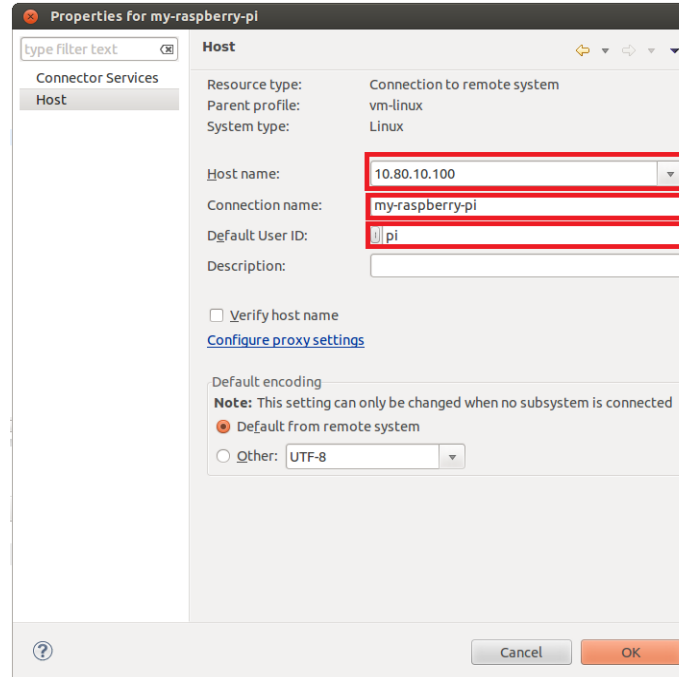
This configuration allows you to work with processes on remote linux systems using any contributed Shell subsystem.

? < Back **Next >** Cancel Finish

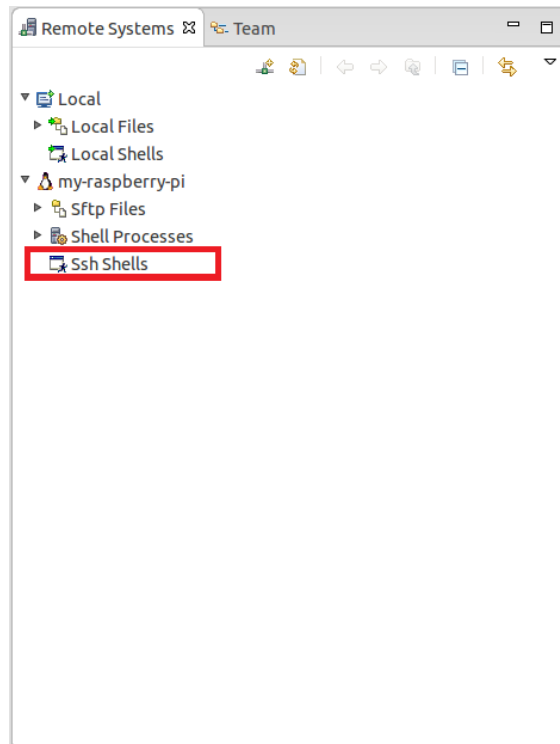


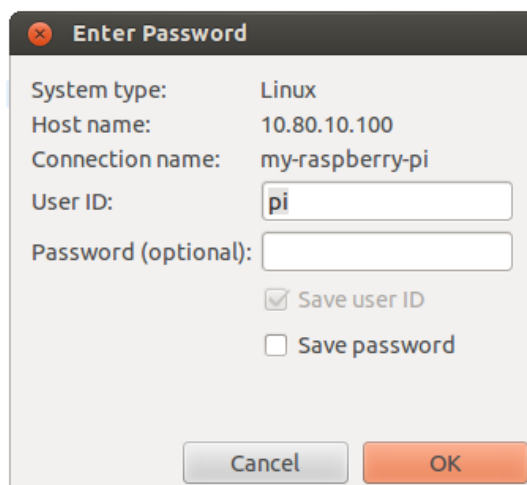


3. Подесите подразумевени налог за приступ RPI развојној плочи. Дијалог за подешавање је приказан на следећој слици.



4. Покрените SSH терминал у оквиру *Eclipse* развојног окружења.

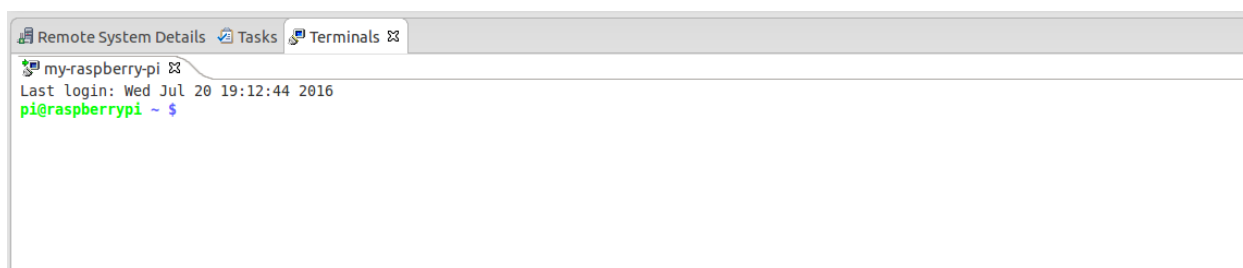


Enter Password

System type: Linux
 Host name: 10.80.10.100
 Connection name: my-raspberry-pi
 User ID:
 Password (optional):
☒ Save user ID
☐ Save password

Cancel OK



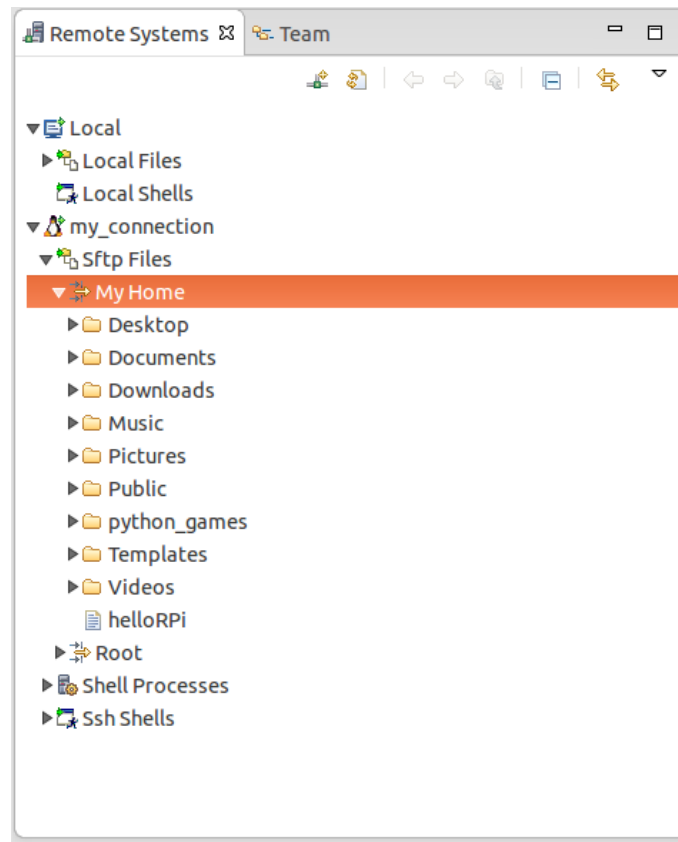
```
Remote System Details Tasks Terminals
my-raspberry-pi
Last login: Wed Jul 20 19:12:44 2016
pi@raspberrypi ~ $
```

5. Након успешног логовања, извршите **lscpu** команду. На конзоли која представља SSH терминал би требале бити неке исписане информације о CPU архитектури.



```
Remote System Details Tasks Terminals
my-raspberry-pi
Last login: Wed Jul 20 19:12:44 2016
pi@raspberrypi ~ $ lscpu
Architecture: armv7l
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
pi@raspberrypi ~ $
```

6. Пронађите и копирајте helloRPI извршну датотеку коју сте добили као резултат превођења у првој вежби.
7. У оквиру „Remote Systems“ погледа, раширите део стабла са називом „Sftp Files“. Налепите претходно копирану датотеку у оквиру тог стабла у директоријум *Home*. Након овог корака, извршна датотека креирана у првој вежби би требала бити прекопирана на RPI развојну плочу.



8. Отворите SSH терминал (конзолу) и проверите да ли је извршна датотека копирана (**ls** команда).
9. Претходно прекопирана извршна датотека нема права да може да се извршава. Додајте право за извршавање датотеке и покрените извршну датотеку.

```
chmod +x helloRPI
```

```
./helloRPI
```

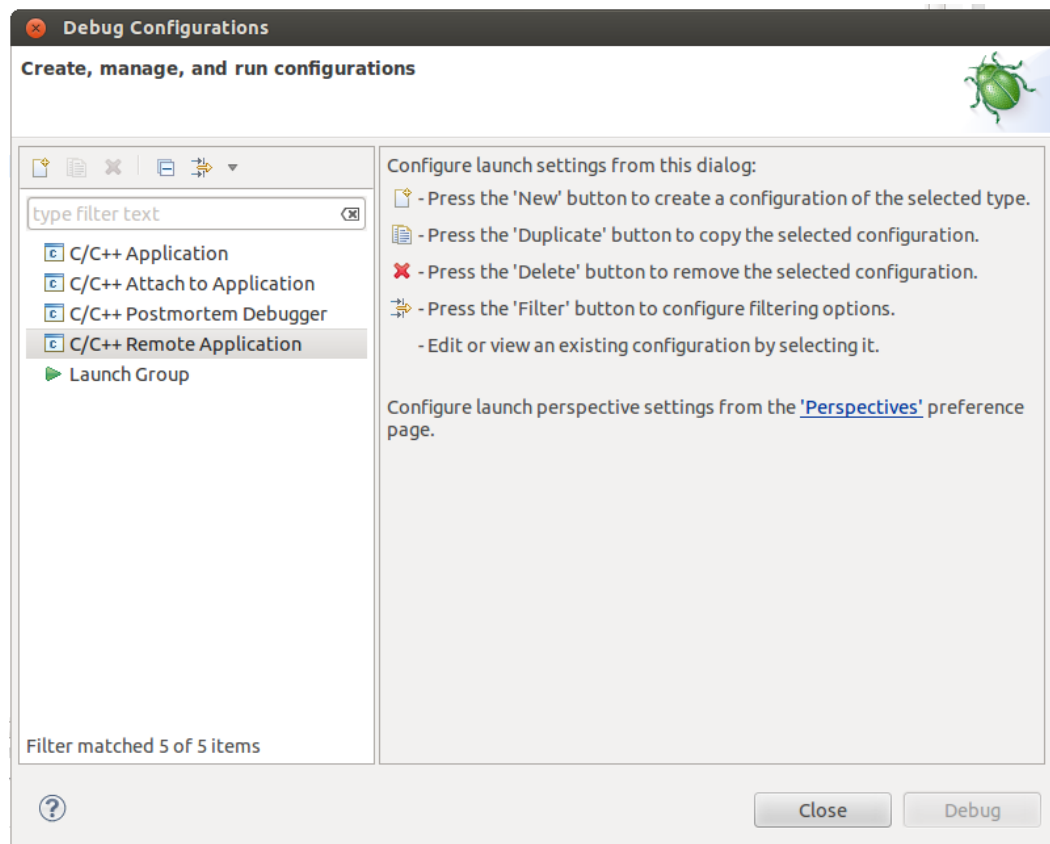
10. Након покретања програма, на конзоли би требало бити исписано „Hello RPI“.



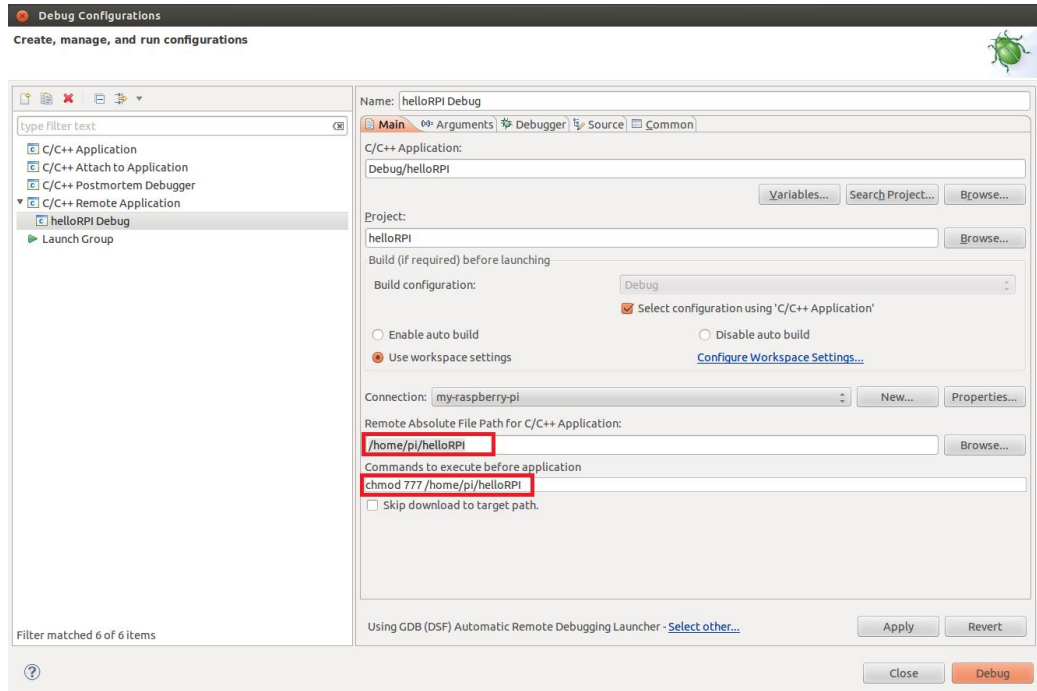
Вежба 3

У овој вежби ће бити показано на који начин се врши подешавања развојног окружења у циљу контролисаног извршавања програма.

1. У Eclipse развојном окружењу отворите дијалог за подешавање *Build/Run* конфигурација (*Run -> Debug Configurations ...*).
2. Додајте нову конфигурације у оквиру „C/C++ Remote Application“ групе.



3. Попуните апсолутну путању до апликације на циљној платформи (*Remote Absolute File Path for C/C++ Application*).
4. Додајте команду за додавање права извршавања, коју је неопходно извршити пре покретања апликације.

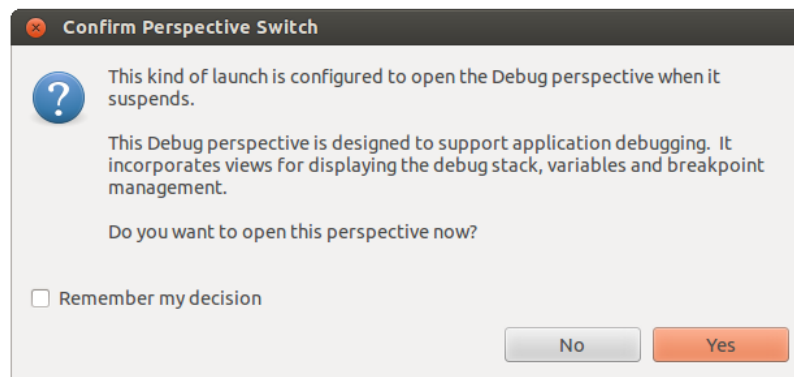


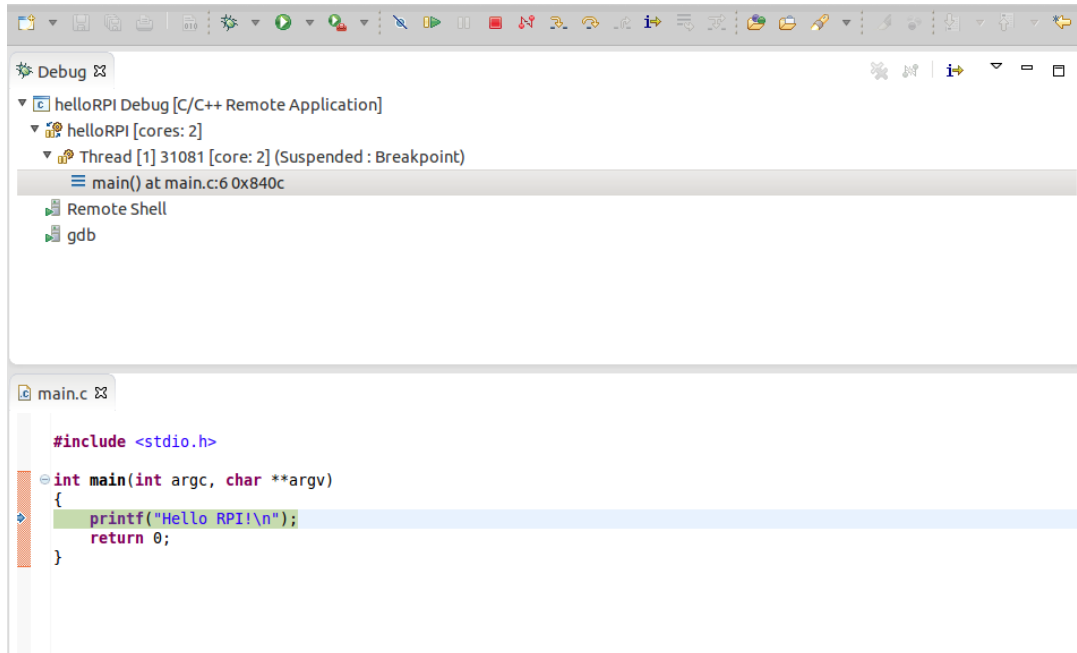
5. Пребаците се на таб *Debugger* и наведите путању до дебагера на рачунару:

- `/home/rtrk/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian-x64/bin/arm-linux-gnueabi-hf-gdb.`

6. Кликните на дугме *Apply* па затим на дугме *Debug*.

7. Након покретања контролисаног извршавања програма, *Eclipse* ће вам понудити да промените перспективу. Потврдите.





Вежба 4

Анализирати програм дат у датотеци example1.c. Програм илуструје једну врсту неухватљивог бага. Коју врсту? Разумети шта се у програму дешава. Покренути програм неколико пута и утврдити да ли се резултати разликују.

Вежба 5

Анализирати програм дат у датотеци example2.c. Програм илуструје читање из неиницијализоване меморије. Разумети шта се у програму дешава. Покренути га неколико пута и утврдити да ли се резултати разликују. Коментарисати резултате.

Вежба 6

У датотеци program.c дат је програм који задати низ најпре уређује у нерастућем редоследу, а онда исписује на екран елементе уређеног низа докле год су већи од вредности задате симболом TRESHOLD. Програм уз елементе низа исписује и индексе које су елементи имали у почетном низу.



Разумети програм. Проверити да ли ради исправно. Која функција не ради исправно? Ставити тачку прекида на почетак те функције. Корак по корак извршавати функцију и пратити вредности. Уочити где се грешка дешава и отклонити је. Поново проверити програм.