



Carleton
UNIVERSITY

Canada's Capital University

Advance topics in software engineering
Server Client Implementation Using TCP Protocol
Group-A

Project User Manual

SUBMITTED TO:

Prof. Cristina Ruiz Martin

SUBMITTED BY:

Hitin Sarin – 101134217

Jaspreet Kaur – 101148265

Prerit Sikerwal – 101128282

Saksham Mal – 101132131

Installing GCC Compiler in Ubuntu:

Below are the commands for Installing gcc 7.2 and g++ 7.2 and making them default compiler in Ubuntu machine. Before proceeding with the installation please verify if the machine has the gcc compiler already installed in it. To check that you have g++-7 and gcc-7, Open Ubuntu terminal using "CONTROL+ALT+t" and type on the terminal the following command and press ENTER:

```
$ g++ -v
```

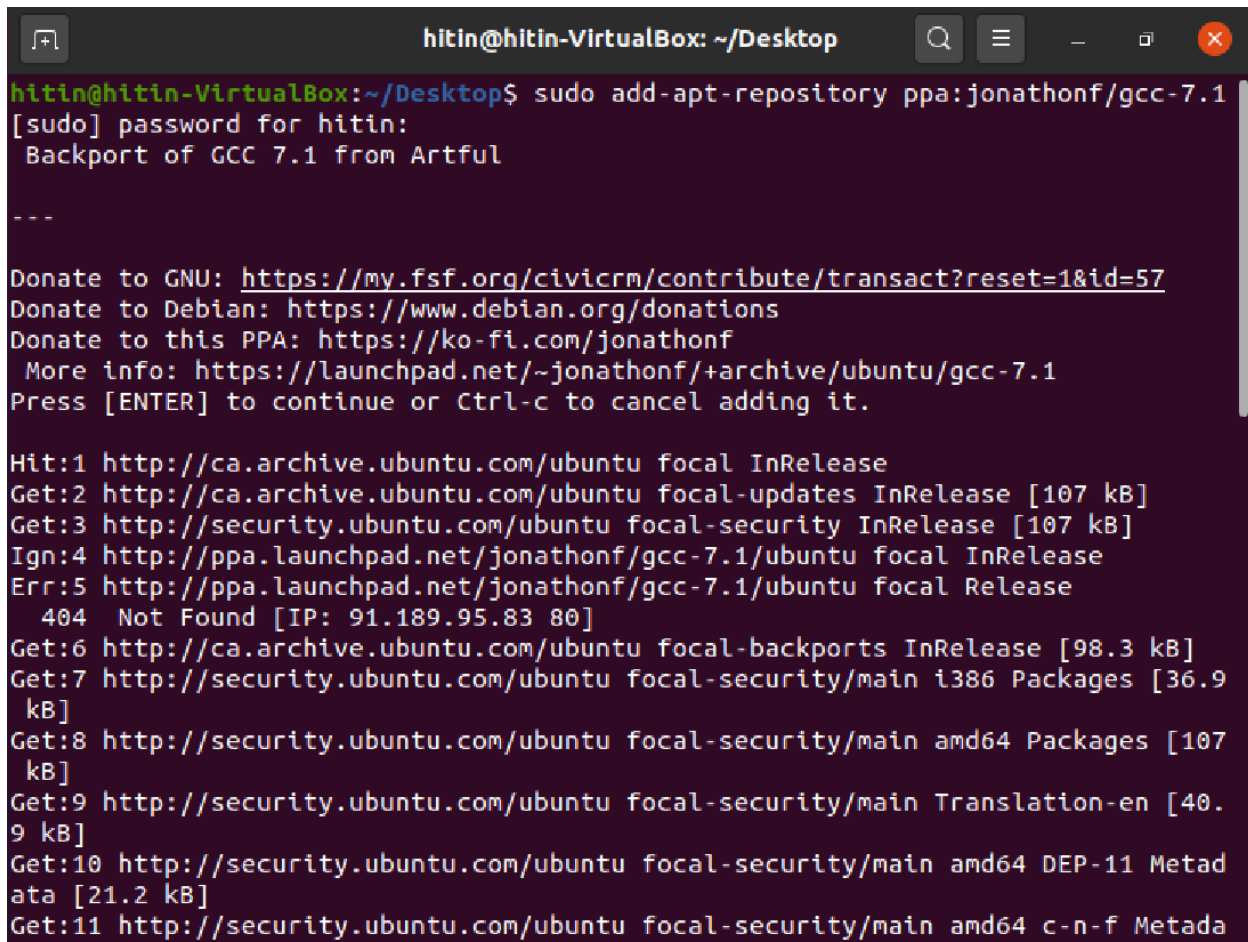
```
$ gcc -v
```

STEPS:

1. Open Ubuntu terminal using "CONTROL+ALT+t"
2. Type on the Ubuntu terminal the following command and press ENTER: (Type the administrative password)

```
$ sudo add-apt-repository ppa:jonathonf/gcc-7.1
```

3. Press [ENTER] to continue.



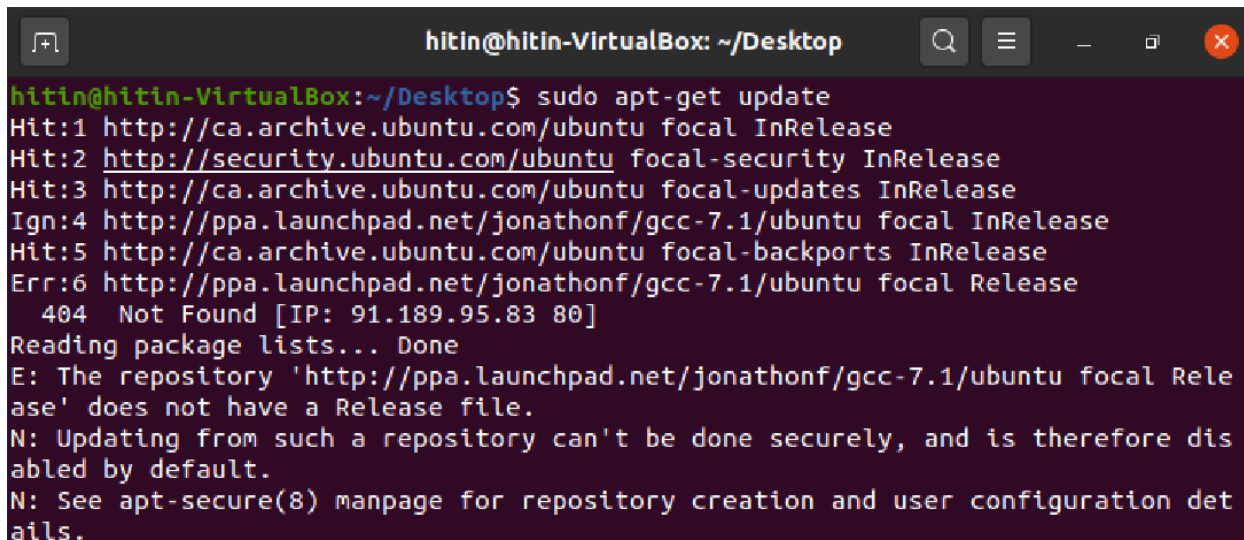
```
hitin@hitin-VirtualBox: ~/Desktop
hitin@hitin-VirtualBox:~/Desktop$ sudo add-apt-repository ppa:jonathonf/gcc-7.1
[sudo] password for hitin:
Backport of GCC 7.1 from Artful

---
Donate to GNU: https://my.fsf.org/civicrm/contribute/transact?reset=1&id=57
Donate to Debian: https://www.debian.org/donations
Donate to this PPA: https://ko-fi.com/jonathonf
More info: https://launchpad.net/~jonathonf/+archive/ubuntu/gcc-7.1
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://ca.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ca.archive.ubuntu.com/ubuntu focal-updates InRelease [107 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Ign:4 http://ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal InRelease
Err:5 http://ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal Release
404 Not Found [IP: 91.189.95.83 80]
Get:6 http://ca.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [36.9 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [107 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [40.9 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [21.2 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [51.1 kB]
```

5. Type on the Ubuntu terminal the following command and press ENTER:

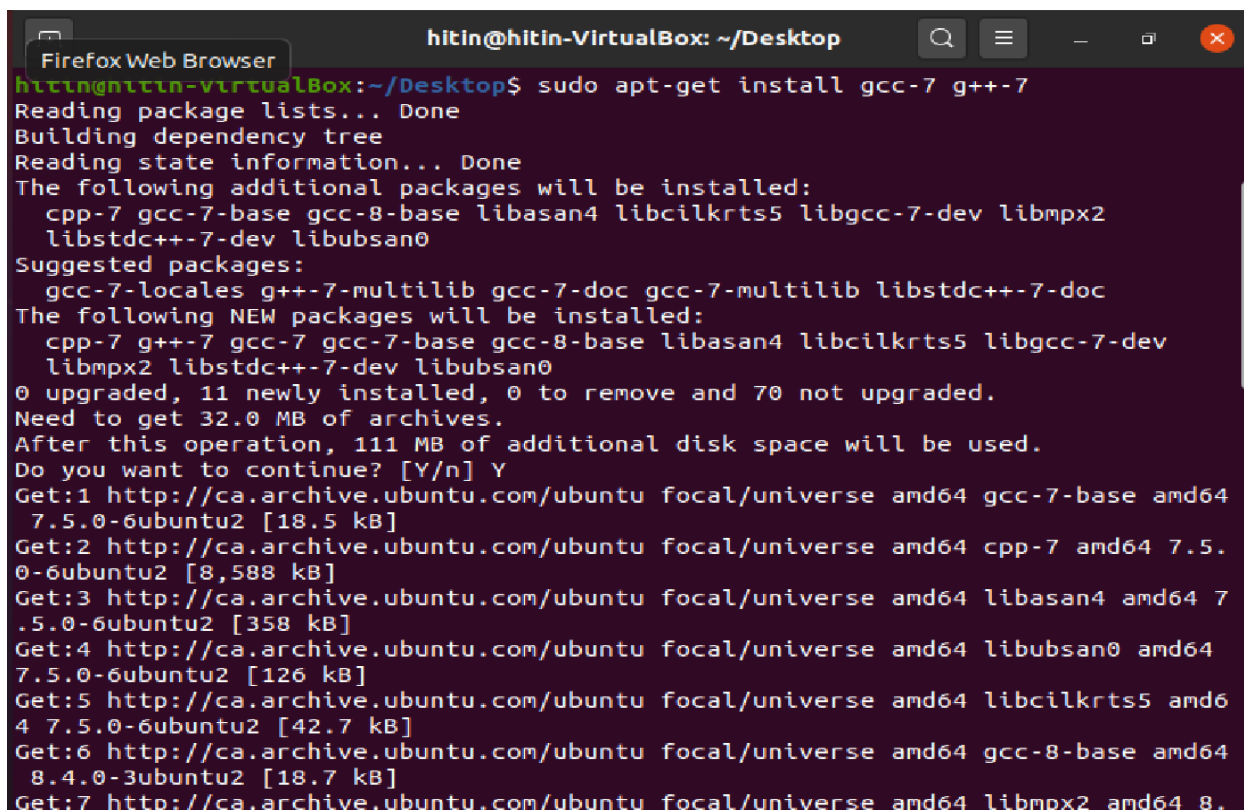
```
$ sudo apt-get update
```

A terminal window titled 'hitin@hitin-VirtualBox: ~/Desktop' showing the output of the 'sudo apt-get update' command. The output lists several repositories and their release status, including 'ca.archive.ubuntu.com/ubuntu focal InRelease', 'security.ubuntu.com/ubuntu focal-security InRelease', and 'ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal InRelease'. It also shows an error for a repository that does not have a release file and a message about updating securely.

```
hitin@hitin-VirtualBox:~/Desktop$ sudo apt-get update
Hit:1 http://ca.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://ca.archive.ubuntu.com/ubuntu focal-updates InRelease
Ign:4 http://ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal InRelease
Hit:5 http://ca.archive.ubuntu.com/ubuntu focal-backports InRelease
Err:6 http://ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal Release
  404 Not Found [IP: 91.189.95.83 80]
Reading package lists... Done
E: The repository 'http://ppa.launchpad.net/jonathonf/gcc-7.1/ubuntu focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-get(8) manpage for repository creation and user configuration details.
```

6. Type on the terminal the following command and press ENTER:

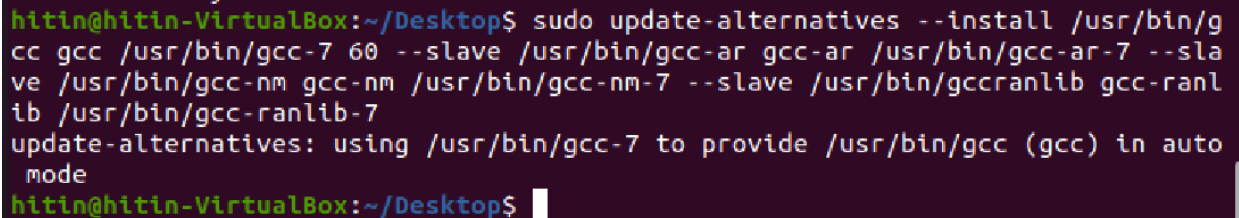
```
$ sudo apt-get install gcc-7 g++-7 (Press Y if asked)
```

A terminal window titled 'hitin@hitin-VirtualBox: ~/Desktop' showing the output of the 'sudo apt-get install gcc-7 g++-7' command. The output displays the packages to be installed, the disk space requirements, and the confirmation to proceed with the installation. It also lists the specific URLs and file sizes for the packages being downloaded.

```
hitin@hitin-VirtualBox:~/Desktop$ sudo apt-get install gcc-7 g++-7
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cpp-7 gcc-7-base gcc-8-base libasan4 libcilkrts5 libgcc-7-dev libmpx2
  libstdc++-7-dev libubsan0
Suggested packages:
  gcc-7-locales g++-7-multilib gcc-7-doc gcc-7-multilib libstdc++-7-doc
The following NEW packages will be installed:
  cpp-7 g++-7 gcc-7 gcc-7-base gcc-8-base libasan4 libcilkrts5 libgcc-7-dev
  libmpx2 libstdc++-7-dev libubsan0
0 upgraded, 11 newly installed, 0 to remove and 70 not upgraded.
Need to get 32.0 MB of archives.
After this operation, 111 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 gcc-7-base amd64 7.5.0-6ubuntu2 [18.5 kB]
Get:2 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 cpp-7 amd64 7.5.0-6ubuntu2 [8,588 kB]
Get:3 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 libasan4 amd64 7.5.0-6ubuntu2 [358 kB]
Get:4 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 libubsan0 amd64 7.5.0-6ubuntu2 [126 kB]
Get:5 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 libcilkrts5 amd64 7.5.0-6ubuntu2 [42.7 kB]
Get:6 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 gcc-8-base amd64 8.4.0-3ubuntu2 [18.7 kB]
Get:7 http://ca.archive.ubuntu.com/ubuntu focal/universe amd64 libmpx2 amd64 8.
```

7. Type on the Ubuntu terminal the following command and press ENTER:

```
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 60 --slave /usr/bin/gcc-ar gcc-ar /usr/bin/gcc-ar-7 --slave /usr/bin/gcc-nm gcc-nm /usr/bin/gcc-nm-7 --slave /usr/bin/gccranlib gcc-ranlib /usr/bin/gcc-ranlib-7
```



```
hitin@hitin-VirtualBox:~/Desktop$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 60 --slave /usr/bin/gcc-ar gcc-ar /usr/bin/gcc-ar-7 --slave /usr/bin/gcc-nm gcc-nm /usr/bin/gcc-nm-7 --slave /usr/bin/gccranlib gcc-ranlib /usr/bin/gcc-ranlib-7
update-alternatives: using /usr/bin/gcc-7 to provide /usr/bin/gcc (gcc) in auto mode
hitin@hitin-VirtualBox:~/Desktop$
```

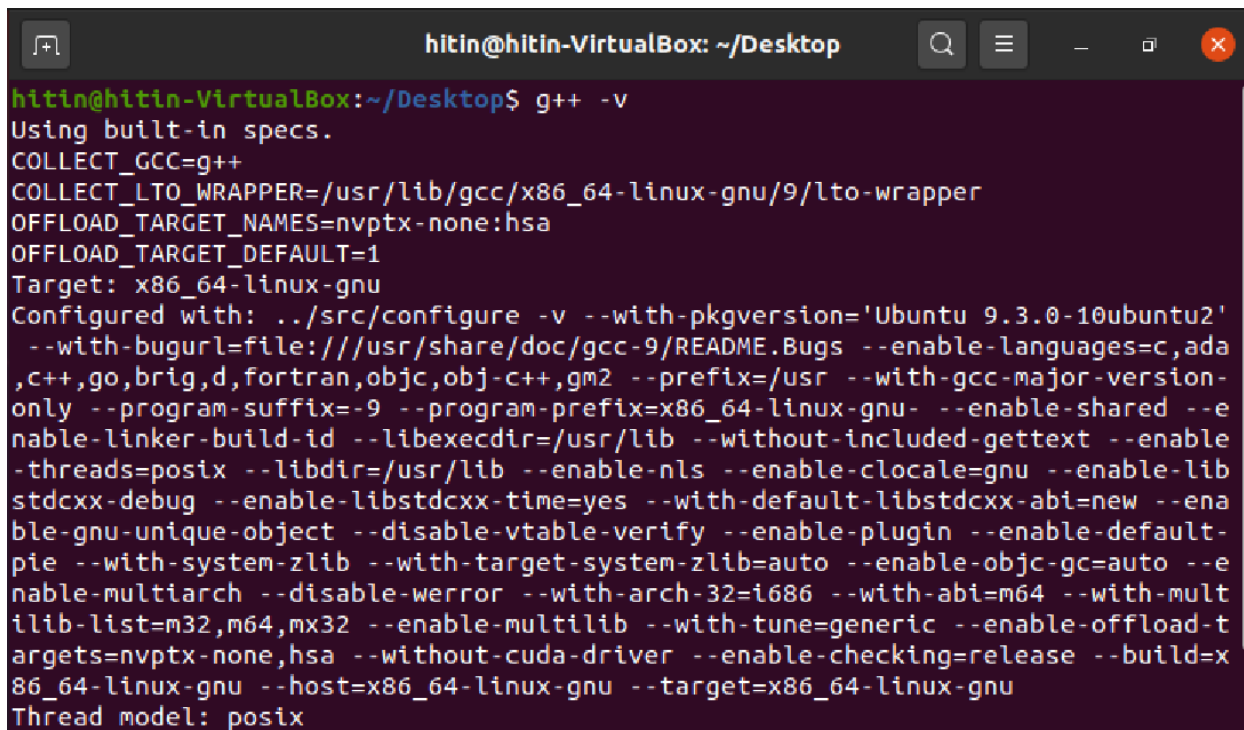
8. Type on the Ubuntu terminal the following command and press ENTER:

```
$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-7 60 --slave /usr/bin/g++-ar g++-ar /usr/bin/g++-ar-7 --slave /usr/bin/g++-nm g++-nm /usr/bin/g++-nm-7 --slave /usr/bin/g++-ranlib g++-ranlib /usr/bin/g++-ranlib-7
```

Similarly, the above command is executed.

9. To check that you have g++-7 type on the terminal the following command and press ENTER:

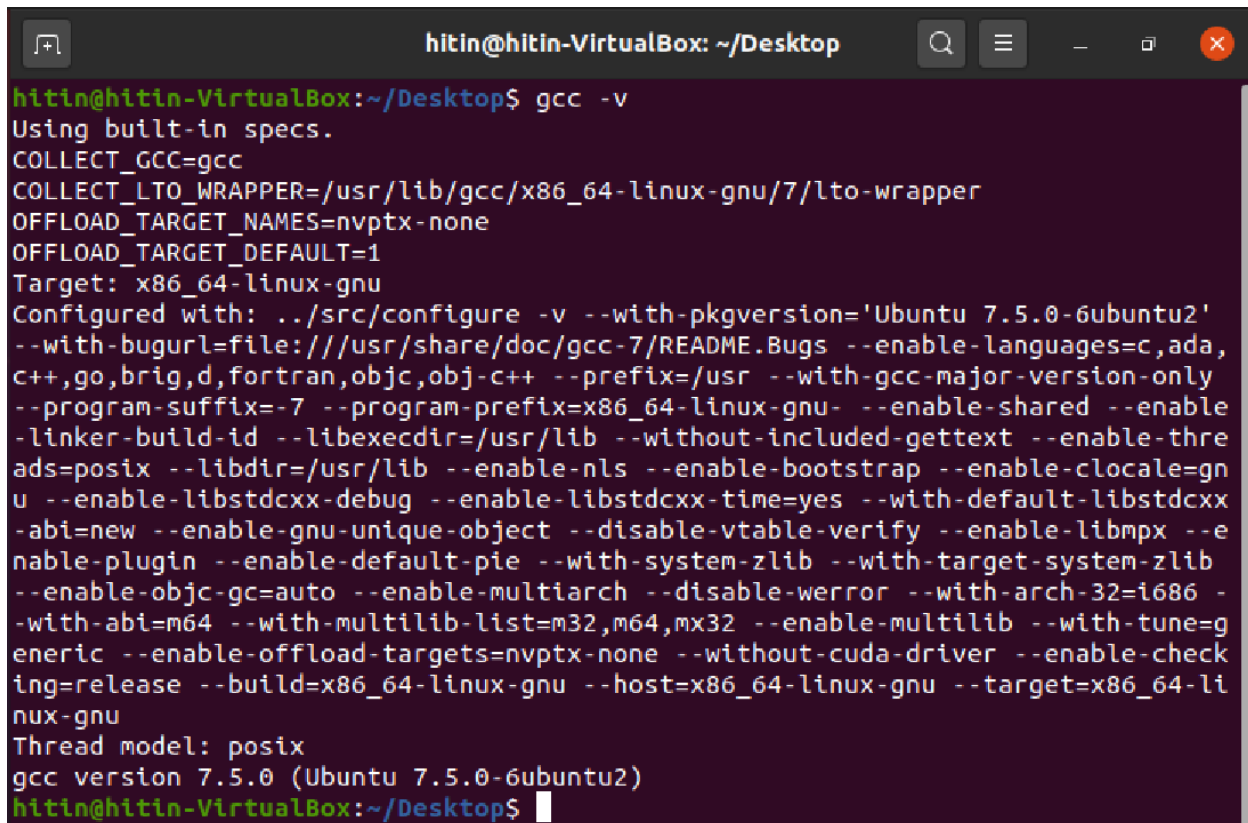
```
$ g++ -v
```



```
hitin@hitin-VirtualBox: ~/Desktop
hitin@hitin-VirtualBox:~/Desktop$ g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:hsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 9.3.0-10ubuntu2' --with-bugurl=file:///usr/share/doc/gcc-9/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-9 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none,hsa --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
```

10. To check that you have gcc-7 type on the terminal the following command and press ENTER:

```
$ gcc -v
```



```
hitin@hitin-VirtualBox: ~/Desktop$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-6ubuntu2'
--with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,
c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only
--program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable
-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-thre
ads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gn
u --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx
-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --e
nable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib
--enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 -
-with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=g
eneric --enable-offload-targets=nvptx-none --without-cuda-driver --enable-check
ing=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-li
nux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-6ubuntu2)
hitin@hitin-VirtualBox:~/Desktop$
```

Downloading Project:

1) Install Git (User can use Git bash or any other software of choice to clone the project).

2) Create a new folder for the project. Right Click inside the folder to open a git bash.

Right Click + "Git Bash here".

3) Type the following command in the git terminal:

```
$ git clone https://github.com/jaskaur7/Group_A_SERVER-CLIENT_USING_TCP.git
```

OR

Go to the link https://github.com/jaskaur7/Group_A_SERVER-CLIENT_USING_TCP.git and click on Clone or Download button and then download the zip folder of the software into the machine.

Running the Project:

Compiling and running a C program

All the main code is defined under '/src directory', the header files are included under '/include directory' and the test files are under '/test directory'.

1. Open the downloaded folder (where project files are placed) cloned from github repository.
2. Set the prompt in the folder and then right click to open a new terminal. The terminal will be in the same directory as the project files.
3. Under the project path, a file named 'makefile' exists in which all the functions (client, serverSingle, serverMulti, test_client and test_server) have been compiled using the below set of commands.

```
=====
# -----
# Generic Makefile
#
# @author Hitin Sarin - hitinsarin@cmail.carleton.ca
# @author Jaspreet Kaur - jaspreetkaur4@cmail.carleton.ca
# @author Saksham Mal - sakshammal@cmail.careton.ca
# @author Prerit Sikerwal - preritsikerwal@cmail.carleton.ca
#

# Date : 2020-06-23
#
# Changelog :
# 2020-06-23: Makefile
#
# -----

CC = gcc
LIBS = -pthread

all: client serverSingle serverMulti test_client test_server message

serverMulti: serverMulti.o queue.o common.o
	$(CC) $(LIBS) ./build/serverMulti.o ./build/queue.o ./build/common.o -o ./bin/serverMulti

serverSingle: serverSingle.o common.o
	$(CC) ./build/serverSingle.o ./build/common.o -o ./bin/serverSingle

client: client.o common.o
	$(CC) ./build/client.o ./build/common.o -o ./bin/client

test_client: test_client.o common.o
	$(CC) ./build/test_client.o ./build/common.o -o ./bin/test_client

test_server: test_server.o common.o
	$(CC) ./build/test_server.o ./build/common.o -o ./bin/test_server

serverMulti.o: ./src/server-multithread/server.c ./include/server_multi.h ./include/queue.h ./include/common.h
	$(CC) -c ./src/server-multithread/server.c -o ./build/serverMulti.o

serverSingle.o: ./src/server-singlethread/server.c ./include/server_single.h ./include/common.h
	$(CC) -c ./src/server-singlethread/server.c -o ./build/serverSingle.o
```

```

client.o: ./src/client/client.c ./include/client.h ./include/common.h
$(CC) -c ./src/client/client.c -o ./build/client.o

queue.o: ./lib/queue.c ./include/queue.h
$(CC) -c ./lib/queue.c -o ./build/queue.o

test_client.o: ./test/src/test_client.c ./include/client.h ./include/common.h
$(CC) -c ./test/src/test_client.c -o ./build/test_client.o

test_server.o: ./test/src/test_server.c ./include/server_single.h ./include/server_multi.h ./include/common.h
$(CC) -c ./test/src/test_server.c -o ./build/test_server.o

common.o: ./src/common.c ./include/common.h
$(CC) -c ./src/common.c -o ./build/common.o

message:
    @echo ""
    @echo "          Now you can run this project by following the steps in the User_Manual1.0."
    @echo "          bin/./client 127.0.0.1 8080"
    @echo "          bin/./serverSingle 8080"

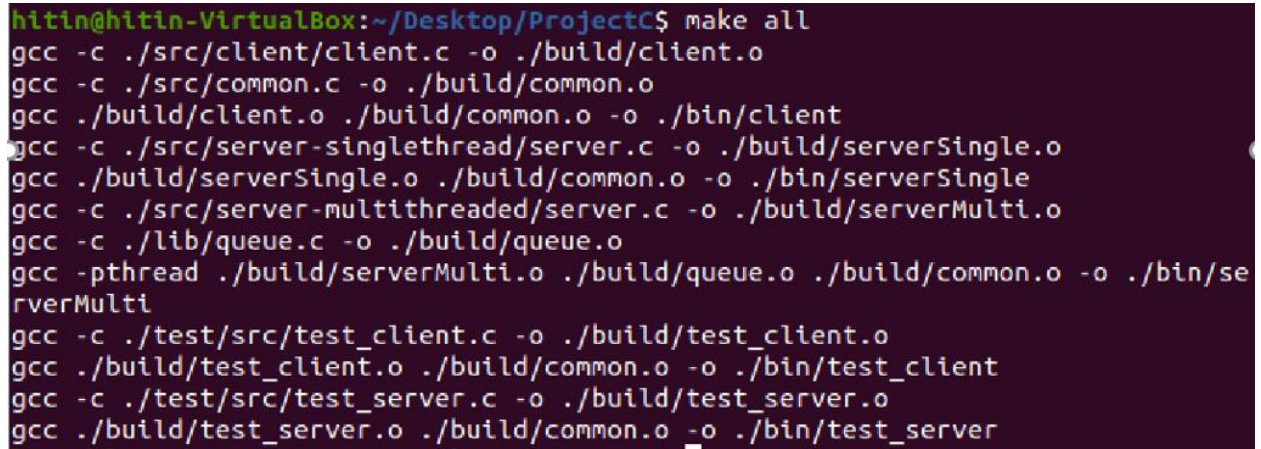
.PHONY: clean

clean:
    rm -f ./build/*.o ./bin/serverSingle ./bin/serverMulti ./bin/client ./bin/test_client ./bin/test_server
=====

```

4. Now type the 'make all' command inside the terminal and press enter.

\$ make all



```

hitin@hitin-VirtualBox:~/Desktop/ProjectC$ make all
gcc -c ./src/client/client.c -o ./build/client.o
gcc -c ./src/common.c -o ./build/common.o
gcc ./build/client.o ./build/common.o -o ./bin/client
gcc -c ./src/server-singlethread/server.c -o ./build/serverSingle.o
gcc ./build/serverSingle.o ./build/common.o -o ./bin/serverSingle
gcc -c ./src/server-multithreaded/server.c -o ./build/serverMulti.o
gcc -c ./lib/queue.c -o ./build/queue.o
gcc -pthread ./build/serverMulti.o ./build/queue.o ./build/common.o -o ./bin/serverMulti
gcc -c ./test/src/test_client.c -o ./build/test_client.o
gcc ./build/test_client.o ./build/common.o -o ./bin/test_client
gcc -c ./test/src/test_server.c -o ./build/test_server.o
gcc ./build/test_server.o ./build/common.o -o ./bin/test_server

```

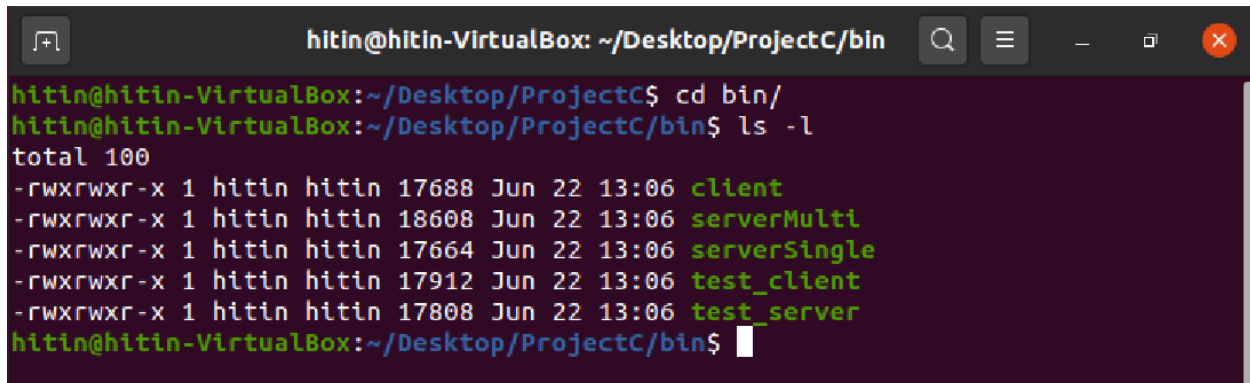
5. Once the 'make all' command is executed, switch to the 'bin/' directory' using the below command:

\$ cd bin/ (This will change the directory to bin directory)

Now type the 'list' command using the syntax mentioned below:

\$ ls -l (This will list the files under bin directory)

(P.S. The list command will show all the main files: client, serverSingle, serverMulti, test_client and test_server in the executable forms).



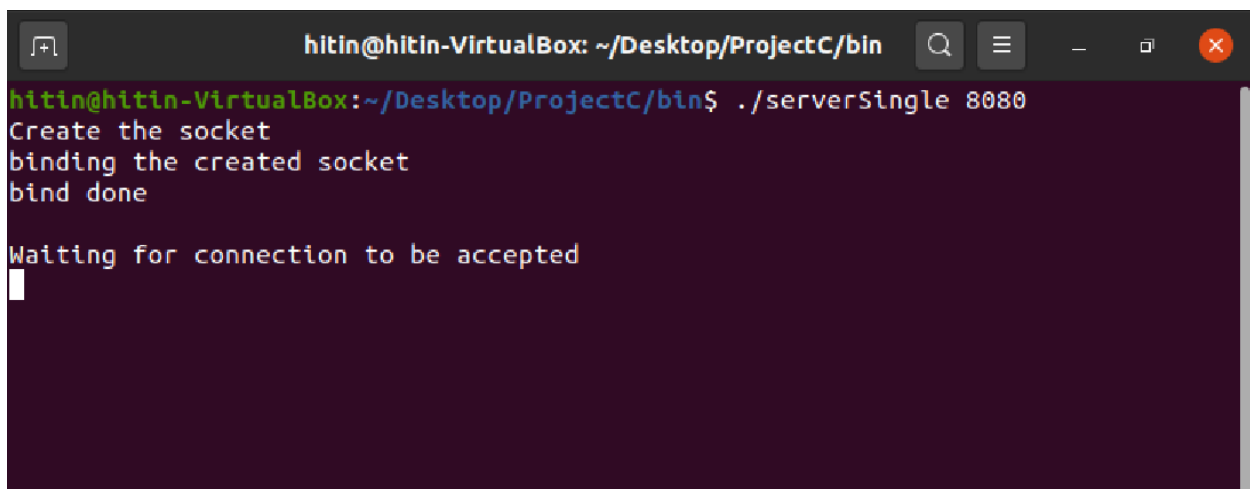
```
hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin
hitin@hitin-VirtualBox:~/Desktop/ProjectC$ cd bin/
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ls -l
total 100
-rwxrwxr-x 1 hitin hitin 17688 Jun 22 13:06 client
-rwxrwxr-x 1 hitin hitin 18608 Jun 22 13:06 serverMulti
-rwxrwxr-x 1 hitin hitin 17664 Jun 22 13:06 serverSingle
-rwxrwxr-x 1 hitin hitin 17912 Jun 22 13:06 test_client
-rwxrwxr-x 1 hitin hitin 17808 Jun 22 13:06 test_server
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$
```

6. Now, to build the connection between Client and Single-thread Server using TCP protocol as proposed in Release-1, run the Single-thread Server file in one machine/tab and the client file in another machine/tab.

Step1 – On first machine/tab, run 'serverSingle' file with port number 8080 using below command:

```
$ ./serverSingle <Port Number>
```

After this command is executed, server is ready to accept connections after creating and binding the created socket.



```
hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./serverSingle 8080
Create the socket
binding the created socket
bind done

Waiting for connection to be accepted

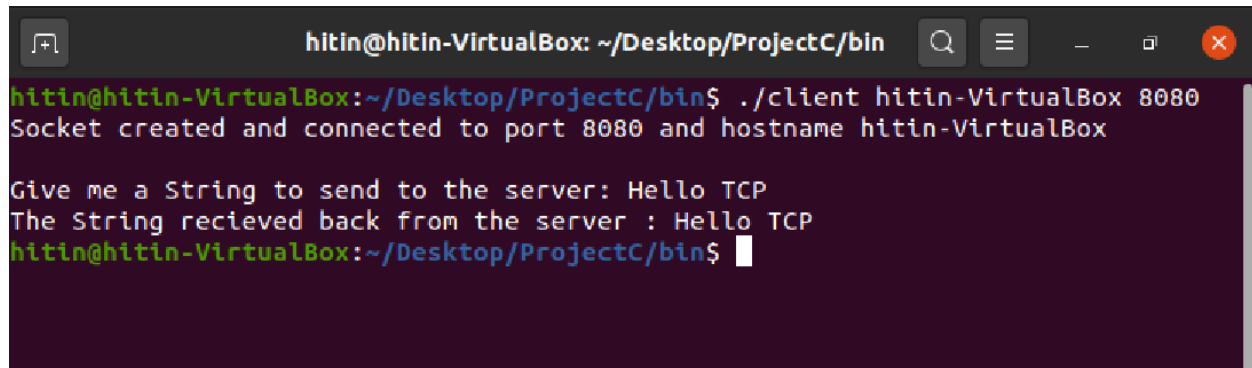
```

Step2- On second machine/tab, under the same 'bin/ directory', run client file with port number 8080 and hostname/IP address of the Single-thread Server using below command:

```
$ ./client <Hostname/IP Address> <Port Number>
```


After executing the client file, a connection with the provided Server hostname/IP address is built prompting a message "Give me a string to send to the server:"

Once a message string is sent to the server, same string is received back from server stating the connection has been accepted by the server.

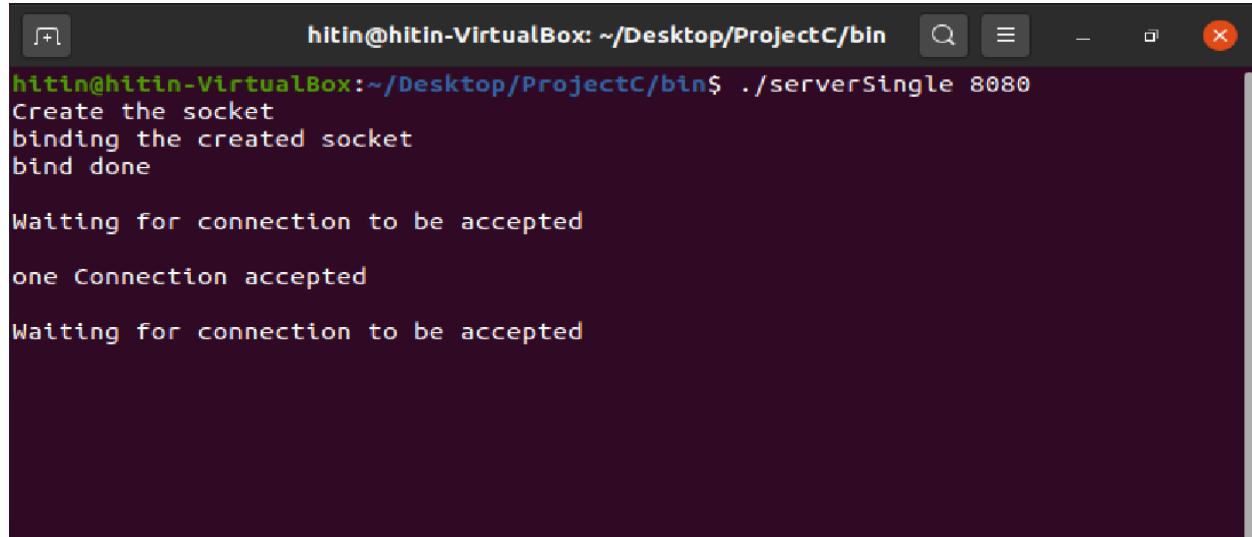
A terminal window titled 'hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin' with search, menu, and window control icons. The terminal shows the execution of a client program. The user enters './client hitin-VirtualBox 8080', and the program outputs 'Socket created and connected to port 8080 and hostname hitin-VirtualBox'. It then prompts 'Give me a String to send to the server:'. The user enters 'Hello TCP', and the program outputs 'The String recieved back from the server : Hello TCP'. The prompt returns to the shell.

```
hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./client hitin-VirtualBox 8080
Socket created and connected to port 8080 and hostname hitin-VirtualBox

Give me a String to send to the server: Hello TCP
The String recieved back from the server : Hello TCP
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$
```

Meanwhile on the single-thread server side, a message stating "One Connection accepted" is prompted which is further waiting to accept a new connection.

P.S. The Single-thread server will take one connection request at a time.

A terminal window titled 'hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin' with search, menu, and window control icons. The terminal shows the execution of a single-thread server program. The user enters './serverSingle 8080', and the program outputs 'Create the socket', 'binding the created socket', and 'bind done'. It then prompts 'Waiting for connection to be accepted'. The user enters a command (not visible), and the program outputs 'one Connection accepted'. It then prompts 'Waiting for connection to be accepted' again.

```
hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./serverSingle 8080
Create the socket
binding the created socket
bind done

Waiting for connection to be accepted

one Connection accepted

Waiting for connection to be accepted
```

7. Now, for executing the Multi-thread server with given number of threads and size of queue on Tab-1 and multiple clients on Tab-2, Tab-3 and Tab-4, follow the below mentioned steps.

Step1- On Tab-1 , run the Multi-thread server with the given number of threads and size of queue.

\$./serverMulti <port number> <number of threads> <size of queue>

```
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./serverMulti 8080 10 10
Create the socket
Socket created
bind done
Thread 140372377753344:      Waiting for Connection
Thread 140372369360640:      Waiting for Connection
Thread 140372360967936:      Waiting for Connection
Thread 140372352575232:      Waiting for Connection
Thread 140372344182528:      Waiting for Connection
Thread 140372327397120:      Waiting for Connection
Thread 140372302219008:      Waiting for Connection
Thread 140372319004416:      Waiting for Connection
Thread 140372335789824:      Waiting for Connection
Thread 140372310611712:      Waiting for Connection

Main thread is Accepting Connections
█
```

Step2 – Run the client file on multiple tabs for connecting the multi-thread server simultaneously using below command:

\$./client <hostname/IP address> <port number>

As shown below, different connection requests (with different string messages) have been sent to the multi-thread server.

Tab-2

```
hitin@hitin-VirtualBox: ~/Desktop/ProjectC/bin
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./client hitin-VirtualBox 8080
Socket created and connected to port 8080 and hostname hitin-VirtualBox

Give me a String to send to the server: hello
The String recieved back from the server : hello
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ █
```

Tab-3

```
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./client hitin-VirtualBox 8080
Socket created and connected to port 8080 and hostname hitin-VirtualBox

Give me a String to send to the server: hi
The String recieved back from the server : hi
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$
```

Tab-4

```
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./client hitin-VirtualBox 8080
Socket created and connected to port 8080 and hostname hitin-VirtualBox

Give me a String to send to the server: Hey
The String recieved back from the server : Hey
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$
```

Meanwhile on the multi-thread server-end, multiple client requests from the above-mentioned tabs have been completed prompting a message "Connection completed".

The server then waits for the completion of pending connections.

```
hitin@hitin-VirtualBox:~/Desktop/ProjectC/bin$ ./serverMulti 8080 10 10
Create the socket
Socket created
bind done
Thread 140372377753344:      Waiting for Connection
Thread 140372369360640:      Waiting for Connection
Thread 140372360967936:      Waiting for Connection
Thread 140372352575232:      Waiting for Connection
Thread 140372344182528:      Waiting for Connection
Thread 140372327397120:      Waiting for Connection
Thread 140372302219008:      Waiting for Connection
Thread 140372319004416:      Waiting for Connection
Thread 140372335789824:      Waiting for Connection
Thread 140372310611712:      Waiting for Connection

Main thread is Accepting Connections

Connection Completed

Main thread is Accepting Connections
Handler 140372377753344: Processing :

Connection Completed
```

```
Main thread is Accepting Connections  
Handler 140372377753344: Processing :
```

```
Connection Completed
```

```
Main thread is Accepting Connections  
Handler 140372369360640: Processing :
```

```
Connection Completed
```

```
Main thread is Accepting Connections  
Handler 140372360967936: Processing :Thread 140372360967936:   Waiting for Con  
nection  
Thread 140372369360640:           Waiting for Connection  
Thread 140372377753344:           Waiting for Connection
```