

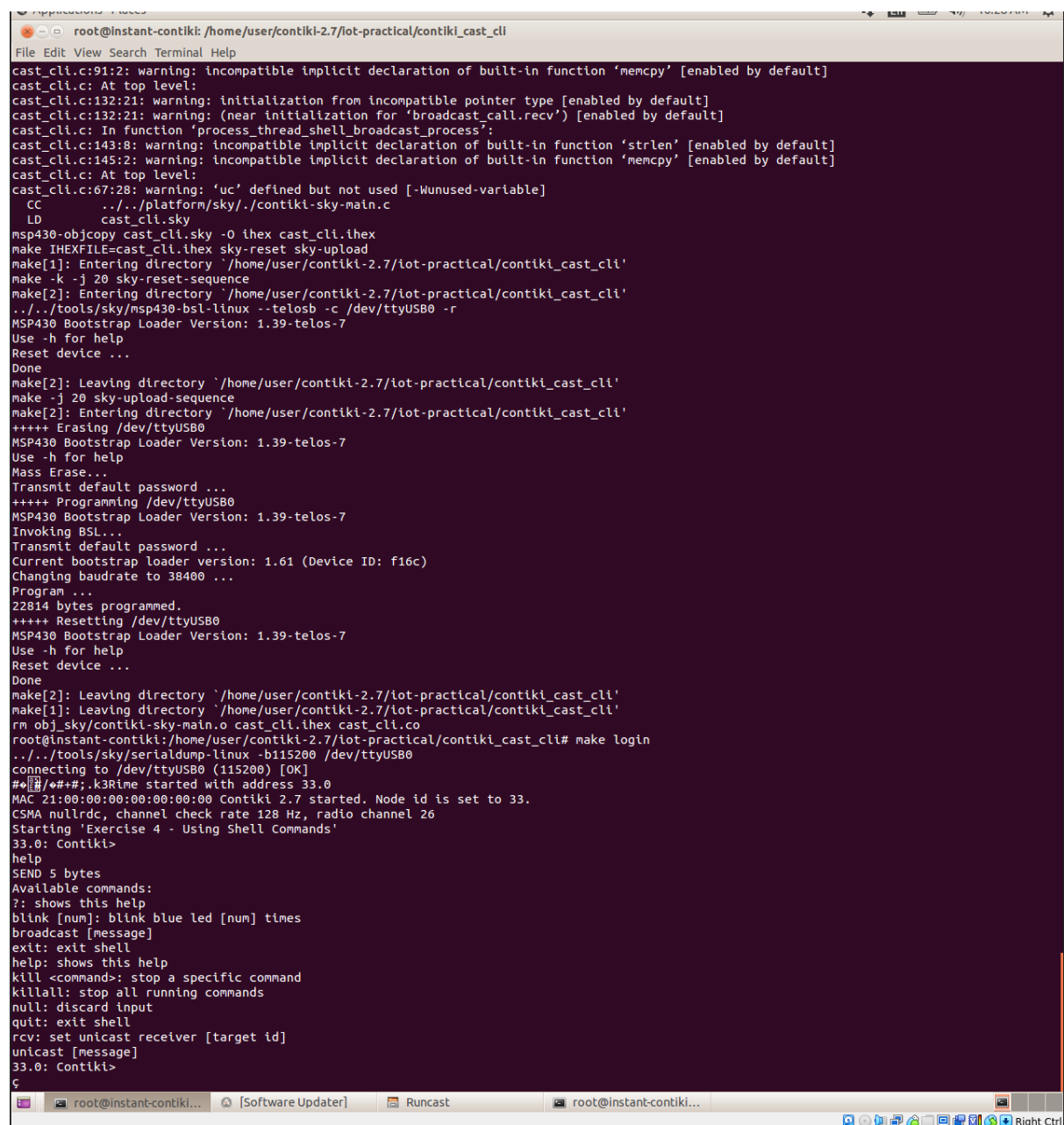
Q3 Broadcast and Unicast with command line interface.

Question 1: Revisiting the Broadcast and Unicast with command line shell interface

a) in the code above, we set up the command line interface and register and implement some commands. compile and upload the code and type "make login". Then, type "blink 10" and observe what happens. Check in the code where this is implemented, explain!

The Blue LED light blinks 10 times

b) type "help". This displays you the list of commands which is registered. check the code in this file to see where which command is defined, implemented and registered!



```
root@instant-contiki: /home/user/contiki-2.7/iot-practical/contiki_cast_cli
File Edit View Search Terminal Help
cast_cli.c:91:2: warning: incompatible implicit declaration of built-in function 'memcpy' [enabled by default]
cast_cli.c: At top level:
cast_cli.c:132:21: warning: initialization from incompatible pointer type [enabled by default]
cast_cli.c:132:21: warning: (near initialization for 'broadcast_call.recv') [enabled by default]
cast_cli.c: In function 'process_thread_shell_broadcast_process':
cast_cli.c:143:8: warning: incompatible implicit declaration of built-in function 'strlen' [enabled by default]
cast_cli.c:145:2: warning: incompatible implicit declaration of built-in function 'memcpy' [enabled by default]
cast_cli.c: At top level:
cast_cli.c:67:28: warning: 'uc' defined but not used [-Wunused-variable]
CC      ../platform/sky/./contiki-sky-main.o
LD      cast_cli.sky
msp430-objcopy cast_cli.sky -O ihex cast_cli.ihex
make IHEXFILE=cast_cli.ihex sky-reset sky-upload
make[1]: Entering directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
make -k -j 20 sky-reset-sequence
make[2]: Entering directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
../tools/sky/msp430-bsl-linux --telosb -c /dev/ttyUSB0 -r
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Reset device ...
Done
make[2]: Leaving directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
make -j 20 sky-upload-sequence
make[2]: Entering directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
+++++ Erasing /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Mass Erase...
Transmit default password ...
+++++ Programming /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Changing baudrate to 38400 ...
Program ...
22814 bytes programmed.
+++++ Resetting /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-telos-7
Use -h for help
Reset device ...
Done
make[2]: Leaving directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
make[1]: Leaving directory '/home/user/contiki-2.7/iot-practical/contiki_cast_cli'
rm obj_sky/contiki-sky-main.o cast_cli.ihex cast_cli.co
root@instant-contiki: /home/user/contiki-2.7/iot-practical/contiki_cast_cli# make login
../tools/sky/serialdump-linux -b115200 /dev/ttyUSB0
connecting to /dev/ttyUSB0 (115200) [OK]
#0[0] /0#0#; k3Rime started with address 33.0
MAC 21:00:00:00:00:00 Contiki 2.7 started. Node id is set to 33.
CSMA nullrdc, channel check rate 128 Hz, radio channel 26
Starting 'Exercise 4 - Using Shell Commands'
33.0: Contiki>
help
SEND 5 bytes
Available commands:
?: shows this help
blink [num]: blink blue led [num] times
broadcast [message]
exit: exit shell
help: shows this help
kill <command>: stop a specific command
killall: stop all running commands
null: discard input
quit: exit shell
rcv: set unicast receiver [target id]
unicast [message]
33.0: Contiki>
$
```

c) the command "broadcast" can be run from the shell. The command takes the input string from the interface and puts it into a broadcast packet. It then sends this over the radio. However, the second node does not get it. Why?

Well it somehow works. The Problem here is that the `broadcast_open` command is only called once a node sends a broadcast. Thus as soon as the first broadcast is sent from a node then it starts receiving (as the callback is registered).

```

Instant Contiki CDS 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places
root@instant-contiki: /home/user/contiki-2.7/ot-practical/contiki_cds_cli
File Edit View Search Terminal Help
root@instant-contiki:/home/user/contiki-2.7/ot-practical/contiki_cds_cli make login
./././tools/obj/serialdump_line_3115200 /dev/ttyUSB1
connecting to /dev/ttyUSB1 (115200) [OK]
New started with address 33.0
MAC 01:00:00:00:00:00:00:00 Contiki 2.7 started. Node id is set to 1.
CDS nulling: channel check rate 128 Hz, radio channel 20
Starting 'Exercise 4 - Using Shell Commands'
33.0: Contiki-
broadcast "Hello I am node 1"
SEND 20 bytes
SEND 10 bytes
entered string "Hello I am node 1"
sent broadcast message
33.0: Contiki-
received broadcast from nodeid=33
message: "Hi node 1, I am node 33"

Instant Contiki CDS 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places
root@instant-contiki: /home/user/contiki-2.7/ot-practical/contiki_cds_cli
File Edit View Search Terminal Help
root@instant-contiki:/home/user/contiki-2.7/ot-practical/contiki_cds_cli make login
./././tools/obj/serialdump_line_3115200 /dev/ttyUSB0
connecting to /dev/ttyUSB0 (115200) [OK]
New started with address 33.0
MAC 21:00:00:00:00:00:00:00 Contiki 2.7 started. Node id is set to 33.
CDS nulling: channel check rate 128 Hz, radio channel 20
Starting 'Exercise 4 - Using Shell Commands'
33.0: Contiki-
broadcast "Hello I am node 33"
SEND 20 bytes
SEND 11 bytes
entered string "Hello I am node 33"
sent broadcast message
33.0: Contiki-
received broadcast from nodeid=1
message: "Hello I am node 1"
broadcast "Hi node 1, I am node 33"
SEND 20 bytes
SEND 16 bytes
entered string "Hi node 1, I am node 33"
sent broadcast message
33.0: Contiki-

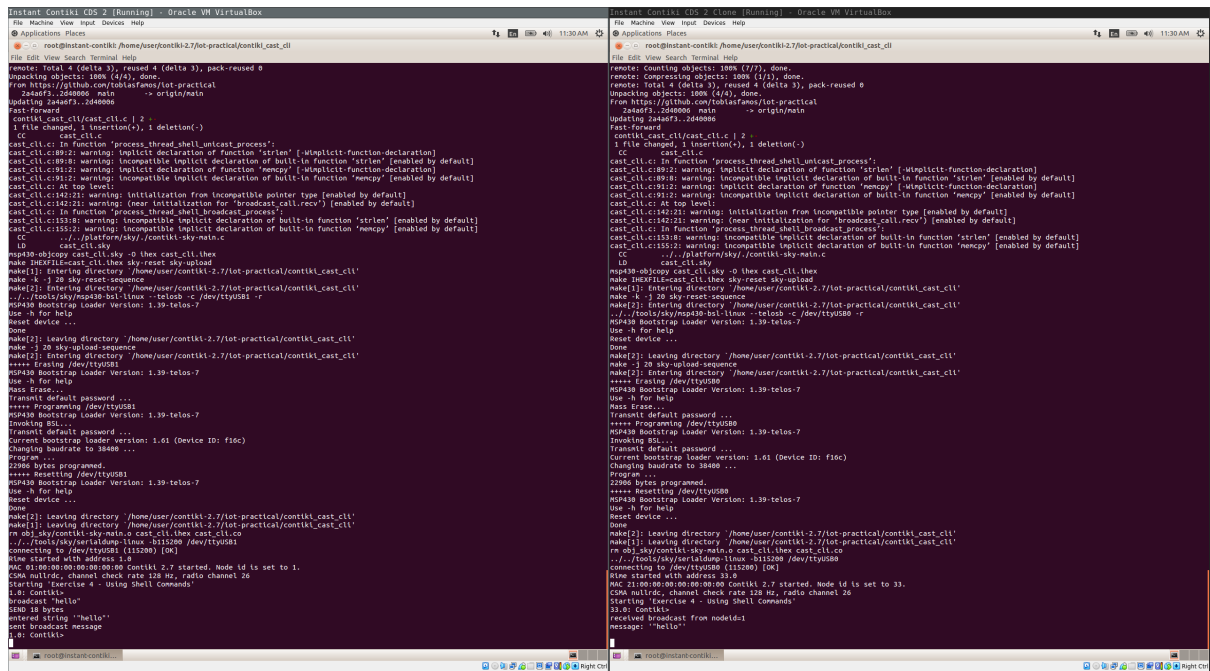
```

The initial state of the application: The broadcast channel is not initialized by default but only when the node starts to send a broadcast itself. Thus the first message on the right is not received by the left.

This can be seen in the below screenshot of two nodes. The first message from the right node is not received by the left node as the broadcast is not yet initialized. But after the left node sends something it can also receive.

To do so the following call must be added to the main process (`exercise_4_process`)

```
broadcast_open(&bc, BROADCAST_CHANNEL, &broadcast_call);
```



After initializing the broadcast also the first message is received.

d) The command "rcv" sets the receipt of the unicast command. The "unicast" command then takes the string argument, and shall send it to the unicast receipt. However, you should complete this program, using your knowledge from previous exercises.

Is implemented pretty straight forward. Only to mention: I also had to implement the channel as in the broadcast in the main process.

Here is a demonstration of a working unicast message

