

Replicode tutorial: Ping Pong example

Helgi Pall Helgason
CADIA / Reykjavik University



In this tutorial

- The Ping Pong example (included with the Replicode source code) is explained step by step
- The motivation for this example is to show interaction between programs
- Concepts, commands and syntax that have not been covered by previous tutorials are explained as they occur

Example #2: Ping Pong (helgi.replicode)

```
ProgPing:(pgm
  []
  []
  []
  (ptn o:(Ping v:[]))
```



The ProgPing program requires as input an object of type Ping, which is defined in **user.classes.replicode**. It does not place any conditions on this object, thus the empty set. Note that the ping object is labelled as **o** and the first member of the Ping object as **v**.

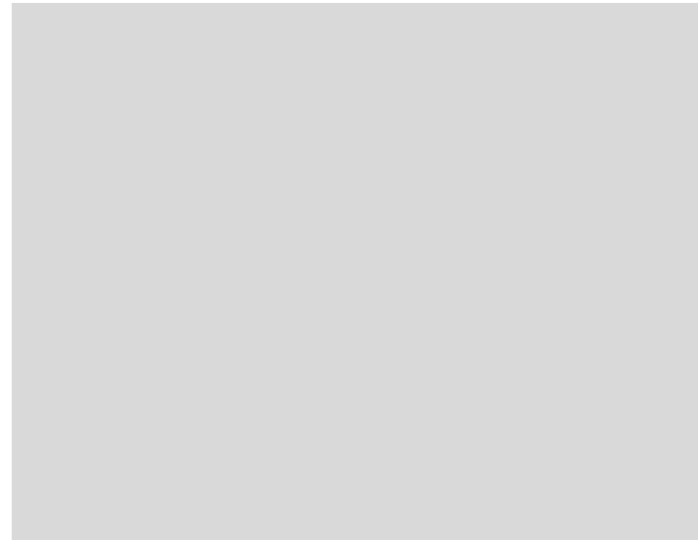
```
ProgPing:(pgm
[[
[]
[]
  (ptn o:(Ping v: :) [[])
[[
[]
[]
[]
  (inj [(Pong now 1) [SYNC_FRONT now 1 forever
stdout nil]])
1
)
[]
```



When this program finds a Ping object, it will inject a new Pong object into the stdout group.

```
ProgPing:(pgm
[[
[]
[]
(ptn o:(Ping v: :) [[]]
[[
[]
[]
(inj [(Pong now 1) [SYNC_FRONT now 1 forever
stdout nil]])
1
)
[]
```

```
ProgPong:(pgm
[[
[]
[]
(ptn o:(Pong v: :) [[]]
[[
[]
[]
(inj [(Ping now 1) [SYNC_FRONT now 1 forever
stdin nil]])
1
)
[]
```



The ProgPong program needs a Pong object as input to run.

When this program finds a Pong object, it will inject a new Ping object into the stdin group.

```

ProgPing:(pgm
[[
[]
[]
[]
(ptn o:(Ping v: :) [[]]
[[
[]
[]
[]
(inj [(Pong now 1) [SYNC_FRONT now 1 forever
stdout nil]])
1
)
[]

ProgPong:(pgm
[[
[]
[]
[]
(ptn o:(Pong v: :) [[]]
[[
[]
[]
[]
(inj [(Ping now 1) [SYNC_FRONT now 1 forever stdin
nil]])
1
)
[]

iProgPing:(ipgm
ProgPing
[[
RUN_ALWAYS
0us
NOTIFY
1
)
[]
[SYNC_FRONT now 1 forever stdin nil 1]

```



Make an instance of the ProgPing program and inject in the stdin group.

```

ProgPing:(pgm
[[
[]
  []
  (ptn o:(Ping v: :) |[])
  []
  []
[]
  (inj [(Pong now 1) [SYNC_FRONT now 1 forever
stdout nil]])
1
)
[]

ProgPong:(pgm
[[
[]
  []
  (ptn o:(Pong v: :) |[])
  []
  []
[]
  (inj [(Ping now 1) [SYNC_FRONT now 1 forever stdin
nil]])
1
)
[]

iProgPing:(ipgm
ProgPing
[[
RUN_ALWAYS
0us
NOTIFY
1
)
[]
[SYNC_FRONT now 1 forever stdin nil 1]

```

```

iProgPong:(ipgm
ProgPong;
[[
RUN_ALWAYS
0us
NOTIFY
1
)

[]
[SYNC_FRONT now 1 forever stdout nil 1]

```

Make an instance of the ProgPong program and inject in the stdout group.


```

ProgPing:(pgm
[[
[]
[]
  (ptn o:(Ping v: :) |[])
  [[
  []
  []
  []
  (inj [(Pong now 1) [SYNC_FRONT now 1 forever
stdout nil]])
  1
  )
  []
[]

ProgPong:(pgm
[[
[]
[]
  (ptn o:(Pong v: :) |[])
  [[
  []
  []
  []
  (inj [(Ping now 1) [SYNC_FRONT now 1 forever stdin
nil]])
  1
  )
  []
[]

iProgPing:(ipgm
ProgPing
[[
RUN_ALWAYS
0us
NOTIFY
1
)
[]
[SYNC_FRONT now 1 forever stdin nil 1]

```

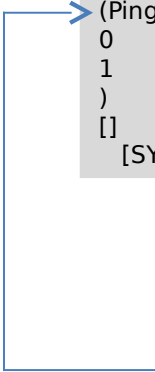
```

iProgPong:(ipgm
ProgPong;
[[
RUN_ALWAYS
0us
NOTIFY
1
)

[]
[SYNC_FRONT now 1 forever stdout nil 1]

(Ping
0
1
)
[]
[SYNC_FRONT now 1 forever stdin nil]

```



Finally, create an instance of the Ping class, defined **user.classes.replicode**, and inject to the stdin group.

This will start a chain of activity. The ProgPing program, located in the stdin group, will catch the Ping object and produce a Pong object into the stdout group. In turn, the ProgPong program, located in the stdout group, will catch the Pong object and produce a new Ping object into the

```

compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 2 ms

Shutting rMem down...

DECOMPILEATION

root:(grp 1 0.5 0.4 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0
1 0 0 [nil]
1) []
[true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1
) []
[true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0
0 [root] 1)
[]
[true 0s:0ms:0us 0 forever grp0 nil true 1]
[true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1)
[]
[true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
[true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
[true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm [[] []]
[(ptn (Ping : :) [[]])]
[]
[]
[cmd_inj 0xa1000000 []]
(Pong (_now) 1)
[true (_now) 1 forever stdout nil]
) 1) [[]]

pgm1:(pgm [[] []]
[(ptn (Pong : :) [[]])]
[]
[]
[cmd_inj 0xa1000000 []]
(Ping (_now) 1)
[true (_now) 1 forever stdin nil]
) 1) [[]]

ipgm0:(ipgm pgm0 [[] true 0s:0ms:0us true 1) []]
[true 0s:0ms:0us 1 forever stdin nil 1]

ipgm1:(ipgm pgm1 [[] true 0s:0ms:0us true 1) []]
[true 0s:0ms:0us 1 forever stdout nil 1]

Ping0:(Ping 0 1) []
[true 0s:0ms:0us 1 forever stdin nil]

Pong0:(Pong 0s:2ms:278us 1) []
[true 0s:4ms:553us 1 forever stdout stdin]

```

23.07.15

```

mk.rdx0:(mk.rdx ipgm0 [Ping0] [(cmd_inj 0xa1000000 []]
Ping0
[true 0s:2ms:326us 1 forever stdout nil]
)) 1) [[]]
[true 0s:5ms:185us 1 1000 root stdin]

Ping1:(Ping 0s:4ms:922us 1) []]
[true 0s:5ms:495us 2 forever grp0 stdin]
[true 0s:5ms:495us 1 forever stdin stdout]

mk.rdx1:(mk.rdx ipgm1 [Pong0] [(cmd_inj 0xa1000000 []]
Ping1
[true 0s:4ms:981us 1 forever stdin nil]
)) 1) [[]]
[true 0s:5ms:910us 1 1000 stdout stdout]

Pong1:(Pong 0s:5ms:788us 1) []]
[true 0s:6ms:103us 1 forever stdout stdin]

mk.rdx2:(mk.rdx ipgm0 [Ping1] [(cmd_inj 0xa1000000 []]
Ping1
[true 0s:5ms:799us 1 forever stdout nil]
)) 1) [[]]
[true 0s:6ms:386us 1 1000 root stdin]

Ping2:(Ping 0s:6ms:254us 1) []]
[true 0s:6ms:556us 2 forever grp0 stdin]
[true 0s:6ms:556us 1 forever stdin stdout]

mk.rdx3:(mk.rdx ipgm1 [Pong1] [(cmd_inj 0xa1000000 []]
Ping2
[true 0s:6ms:266us 1 forever stdin nil]
)) 1) [[]]
[true 0s:7ms:65us 1 1000 stdout stdout]

```

Image taken at: Tue Mar 08 2011 11:15:03:180:224 GMT

19 objects

This is the output when the program is executed. Note that I set the run_time variable in the **settings.xml** to 2 to keep the amount of output small. This means the program only runs for 2 milliseconds.



```

compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 2 ms

Shutting rMem down...

DECOMPILEATION

root:(grp 1 0.5 0.4 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0
1 0 0 [nil]
1) []
[true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1
) []
[true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0
0 [root] 1)
[]
[true 0s:0ms:0us 0 forever grp0 nil true 1]
[true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1)
[]
[true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
[true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
[true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm [[] []]
[(ptn (Ping : :) [[]])]
[])
[])
[(cmd_inj 0xa1000000 []]
(Pong (_now) 1)
[true (_now) 1 forever stdout nil]
) 1) [[]]

pgm1:(pgm [[] []]
[(ptn (Pong : :) [[]])]
[])
[])
[(cmd_inj 0xa1000000 []]
(Ping (_now) 1)
[true (_now) 1 forever stdin nil]
) 1) [[]]

ipgm0:(ipgm pgm0 [[] true 0s:0ms:0us true 1) []
[true 0s:0ms:0us 1 forever stdin nil 1]

ipgm1:(ipgm pgm1 [[] true 0s:0ms:0us true 1) []
[true 0s:0ms:0us 1 forever stdout nil 1]

Ping0:(Ping 0 1) []
[true 0s:0ms:0us 1 forever stdin nil]

Pong0:(Pong 0s:2ms:278us 1) []
[true 0s:4ms:553us 1 forever stdout stdin]

```

23.07.15

```

mk.rdx0:(mk.rdx ipgm0 [Ping0] [(cmd_inj 0xa1000000 []]
Ping0
[true 0s:2ms:326us 1 forever stdout nil]
)) 1) [[]]
[true 0s:5ms:185us 1 1000 root stdin]

Ping1:(Ping 0s:4ms:922us 1) []
[true 0s:5ms:495us 2 forever grp0 stdin]
[true 0s:5ms:495us 1 forever stdin stdout]

mk.rdx1:(mk.rdx ipgm1 [Pong0] [(cmd_inj 0xa1000000 []]
Ping1
[true 0s:4ms:981us 1 forever stdin nil]
)) 1) [[]]
[true 0s:5ms:910us 1 1000 stdout stdout]

Pong1:(Pong 0s:5ms:788us 1) []
[true 0s:6ms:103us 1 forever stdout stdin]

mk.rdx2:(mk.rdx ipgm0 [Ping1] [(cmd_inj 0xa1000000 []]
Ping1
[true 0s:5ms:799us 1 forever stdout nil]
)) 1) [[]]
[true 0s:6ms:386us 1 1000 root stdin]

Ping2:(Ping 0s:6ms:254us 1) []
[true 0s:6ms:556us 2 forever grp0 stdin]
[true 0s:6ms:556us 1 forever stdin stdout]

mk.rdx3:(mk.rdx ipgm1 [Pong1] [(cmd_inj 0xa1000000 []]
Ping2
[true 0s:6ms:266us 1 forever stdin nil]
)) 1) [[]]
[true 0s:7ms:65us 1 1000 stdout stdout]

Image taken at: Tue Mar 08 2011 11:15:03:180:224 GMT

19 objects

```

This is the original Ping object we injected in our code.



```

compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 2 ms

Shutting rMem down...

DECOMPILEATION

root:(grp 1 0.5 0.4 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0
1 0 0 [nil]
1) []
[true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1
) []
[true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0
0 [root] 1)
[]
[true 0s:0ms:0us 0 forever grp0 nil true 1]
[true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1)
[]
[true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
[true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
[true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm [[] []]
[(ptn (Ping : :) [[]])]
[]
[]
[cmd_inj 0xa1000000 []]
(Pong (_now) 1)
[true (_now) 1 forever stdout nil]
) 1) [[]]

pgm1:(pgm [[] []]
[(ptn (Pong : :) [[]])]
[]
[]
[cmd_inj 0xa1000000 []]
(Ping (_now) 1)
[true (_now) 1 forever stdin nil]
) 1) [[]]

ipgm0:(ipgm pgm0 [[] true 0s:0ms:0us true 1) []
[true 0s:0ms:0us 1 forever stdin nil 1]

ipgm1:(ipgm pgm1 [[] true 0s:0ms:0us true 1) []
[true 0s:0ms:0us 1 forever stdout nil 1]

Ping0:(Ping 0 1) []
[true 0s:0ms:0us 1 forever stdin nil]

Pong0:(Pong 0s:2ms:278us 1) []
[true 0s:4ms:553us 1 forever stdout stdin]

```

23.07.15

```

mk.rdx0:(mk.rdx ipgm0 [Ping0] [(cmd_inj 0xa1000000) []]
Ping0
[true 0s:2ms:326us 1 forever stdout nil]
) 1) [[]]
[true 0s:5ms:185us 1 1000 root stdin]

Ping1:(Ping 0s:4ms:922us 1) []
[true 0s:5ms:495us 2 forever grp0 stdin]
[true 0s:5ms:495us 1 forever stdin stdout]

mk.rdx1:(mk.rdx ipgm1 [Pong0] [(cmd_inj 0xa1000000) []]
Ping1
[true 0s:4ms:981us 1 forever stdin nil]
) 1) [[]]
[true 0s:5ms:910us 1 1000 stdout stdout]

Pong1:(Pong 0s:5ms:788us 1) []
[true 0s:6ms:103us 1 forever stdout stdin]

mk.rdx2:(mk.rdx ipgm0 [Ping1] [(cmd_inj 0xa1000000) []]
Ping1
[true 0s:5ms:799us 1 forever stdout nil]
) 1) [[]]
[true 0s:6ms:386us 1 1000 root stdin]

Ping2:(Ping 0s:6ms:254us 1) []
[true 0s:6ms:556us 2 forever grp0 stdin]
[true 0s:6ms:556us 1 forever stdin stdout]

mk.rdx3:(mk.rdx ipgm1 [Pong1] [(cmd_inj 0xa1000000) []]
Ping2
[true 0s:6ms:266us 1 forever stdin nil]
) 1) [[]]
[true 0s:7ms:65us 1 1000 stdout stdout]

Image taken at: Tue Mar 08 2011 11:15:03:180:224 GMT

19 objects

```

The ProgPing program catches the object (Ping0) and produces a Pong object (Pong0). Both the Pong object and the corresponding notification (reduction marker) are highlighted.



```

compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 2 ms

Shutting rMem down...

DECOMPIATION

root:(grp 1 0.5 0.4 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0
1 0 0 [nil]
) 1 []
[true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1
) []
[true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0
0 [root] 1)
[]
[true 0s:0ms:0us 0 forever grp0 nil true 1]
[true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1
0 0 [nil] 1)
[]
[true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
[true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
[true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm [[] []]
[(ptn (Ping : :) [[]])]
[])
[]
[cmd_inj 0xa1000000 []]
(Pong (_now) 1)
[true (_now) 1 forever stdout nil]
) 1) [[]]

pgm1:(pgm [[] []]
[(ptn (Pong : :) [[]])]
[])
[]
[cmd_inj 0xa1000000 []]
(Ping (_now) 1)
[true (_now) 1 forever stdin nil]
) 1) [[]]

ipgm0:(ipgm pgm0 [[] true 0s:0ms:0us true 1) []]
[true 0s:0ms:0us 1 forever stdin nil 1]

ipgm1:(ipgm pgm1 [[] true 0s:0ms:0us true 1) []]
[true 0s:0ms:0us 1 forever stdout nil 1]

Ping0:(Ping 0 1) []
[true 0s:0ms:0us 1 forever stdin nil]

Pong0:(Pong 0s:2ms:278us 1) []
[true 0s:4ms:553us 1 forever stdout stdin]

```

23.07.15

```

mk.rdx0:(mk.rdx ipgm0 [Ping0] [(cmd_inj 0xa1000000 []]
Pong0
[true 0s:2ms:326us 1 forever stdout nil]
)) 1) [[]]
[true 0s:5ms:185us 1 1000 root stdin]

Ping1:(Ping 0s:4ms:922us 1) []
[true 0s:5ms:495us 2 forever grp0 stdin]
[true 0s:5ms:495us 1 forever stdin stdout]

mk.rdx1:(mk.rdx ipgm1 [Pong0] [(cmd_inj 0xa1000000 []]
Ping1
[true 0s:4ms:981us 1 forever stdin nil]
)) 1) [[]]
[true 0s:5ms:910us 1 1000 stdout stdout]

Pong1:(Pong 0s:5ms:788us 1) []
[true 0s:6ms:103us 1 forever stdout stdin]

mk.rdx2:(mk.rdx ipgm0 [Ping1] [(cmd_inj 0xa1000000 []]
Pong1
[true 0s:5ms:799us 1 forever stdout nil]
)) 1) [[]]
[true 0s:6ms:386us 1 1000 root stdin]

Ping2:(Ping 0s:6ms:254us 1) []
[true 0s:6ms:556us 2 forever grp0 stdin]
[true 0s:6ms:556us 1 forever stdin stdout]

mk.rdx3:(mk.rdx ipgm1 [Pong1] [(cmd_inj 0xa1000000 []]
Ping2
[true 0s:6ms:266us 1 forever stdin nil]
)) 1) [[]]
[true 0s:7ms:65us 1 1000 stdout stdout]

Image taken at: Tue Mar 08 2011 11:15:03:180:224 GMT

19 objects

```

The new Pong object triggers the ProgPong program, that produces a new Ping object. This sequence continues until the Replicode executive stops.



In conclusion

- The Ping Pong example shows the interaction of two programs
 - Understanding how programs can interact is very important
 - Collaboration between programs is a key element for building Replicode systems that solve complex tasks
- The value of controlling the duration of operation has been demonstrated
 - The Ping Pong example can be considered an “infinite” loop