Instalação do Compilador P4 no sistema Ubuntu 18.04.5 LTS

```
root@feec:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.5 LTS
Release:        18.04
Codename:       bionic
root@feec:~#
```

Objetivo:

Instalar o compilador P4 e as ferramentas: protobuf, p4c e bmv2, entre outras que serão utilizadas neste tutorial, auxiliando a implantação de um ambiente que utilize a linguagem P4 e ferramentas que auxiliam na programação de interfaces, entre outras programações que podem ser feitas com este ambiente.

Importante:

Este tutorial foi feito em uma máquina física e não em ambiente "virtualizado", a máquina utilizada foi:

Intel (R) Core (TM) i3-3240 CPU @ 3.40GHz

Memória RAM: 6GB – HD: 300GB

Procedimento de Instalação e preparação do ambiente

1. Atualizar o sistema e reiniciar o sistema:

```
apt update && apt upgrade -y && reboot
```

2. Instalando algumas ferramentas auxiliares que não constam na versão 18 do Ubuntu:

```
apt install net-tools ssh vim nmap htop tree python-pip -y
```

3. Instalando algumas ferramentas necessárias para o ambiente:

```
sudo apt-get install g++ git automake libtool libgc-dev bison flex
libfl-dev libgmp-dev libboost-dev libboost-iostreams-dev pkg-config
python python-scapy python-ipaddr c  url wget tcpdump cmake -y
```

4. Instalando a biblioteca do Protobuf, que será necessário para funcionar o ambiente, caso já tenha essas ferramentas, siga para a etapa **5º**, mas confira antes:

```
sudo apt-get install autoconf automake libtool curl make g++ unzip -y
```

5. Instalando o Gmok, no diretório dentro de /home, porém você poderá escolher o diretório que preferir:

```
cd /home && git clone git://github.com/paulsapps/gmock-1.7.0
```

```
root@feec:~# cd /home && git clone git://github.com/paulsapps/gmock-1.7.0
Cloning into 'gmock-1.7.0'...
remote: Enumerating objects: 306, done.
remote: Total 306 (delta 0), reused 0 (delta 0), pack-reused 306
Receiving objects: 100% (306/306), 1.60 MiB | 824.00 KiB/s, done.
Resolving deltas: 100% (85/85), done.
root@feec:/home#
```

6 . Acessando o diretório o Gmok e efetuando sua compilação:

```
cd gmock-1.7.0/make/ && make
```

```
g++ -isystem ../gtest/include -isystem ../include -g -Wall -Wextra -pthread -c ../test/gmock_test.cc
g++ -isystem ../gtest/include -isystem ../include -I../gtest -I.. -g -Wall -Wextra -pthread \
          -c ../src/gmock-all.cc
g++ -isystem ../gtest/include -isystem ../include -I../gtest -I.. -g -Wall -Wextra -pthread \
          -c ../gtest/src/gtest-all.cc
g++ -isystem ../gtest/include -isystem ../include -I../gtest -I.. -g -Wall -Wextra -pthread \
          -c ../src/gmock_main.cc
ar rv gmock_main.a gmock-all.o gtest-all.o gmock_main.o
ar: criando gmock_main.a
a - gmock-all.o
a - gtest-all.o
a - gmock_main.o
g++ -isystem ../gtest/include -isystem ../include -g -Wall -Wextra -pthread -lpthread gmock_test.o gmock_main.a -o gmock_test
root@feec:/home/gmock-1.7.0/make#
```

7 . Detectando unidade do Gmok, dentro do diretório: /home/gmock-1.7.0/make

```
./gmock_test
```

```
root@feec:/home/gmock-1.7.0/make# ls -lha
total 12M
drwxr-xr-x  2 root root 4,0K abr 20 23:53 .
drwxr-xr-x 12 root root 4,0K abr 20 23:47 ..
-rw-r--r--  1 root root 1,5M abr 20 23:53 gmock-all.o
-rw-r--r--  1 root root 4,4M abr 20 23:53 gmock_main.a
-rw-r--r--  1 root root 132K abr 20 23:53 gmock_main.o
-rwxr-xr-x  1 root root 2,3M abr 20 23:53 gmock_test
-rw-r--r--  1 root root 513K abr 20 23:53 gmock_test.o
-rw-r--r--  1 root root 2,6M abr 20 23:53 gtest-all.o
-rw-r--r--  1 root root 3,6K abr 20 23:47 Makefile
root@feec:/home/gmock-1.7.0/make# ./gmock_test
```

Importante:

Os resultados devem ser semelhantes ao da figura 1º

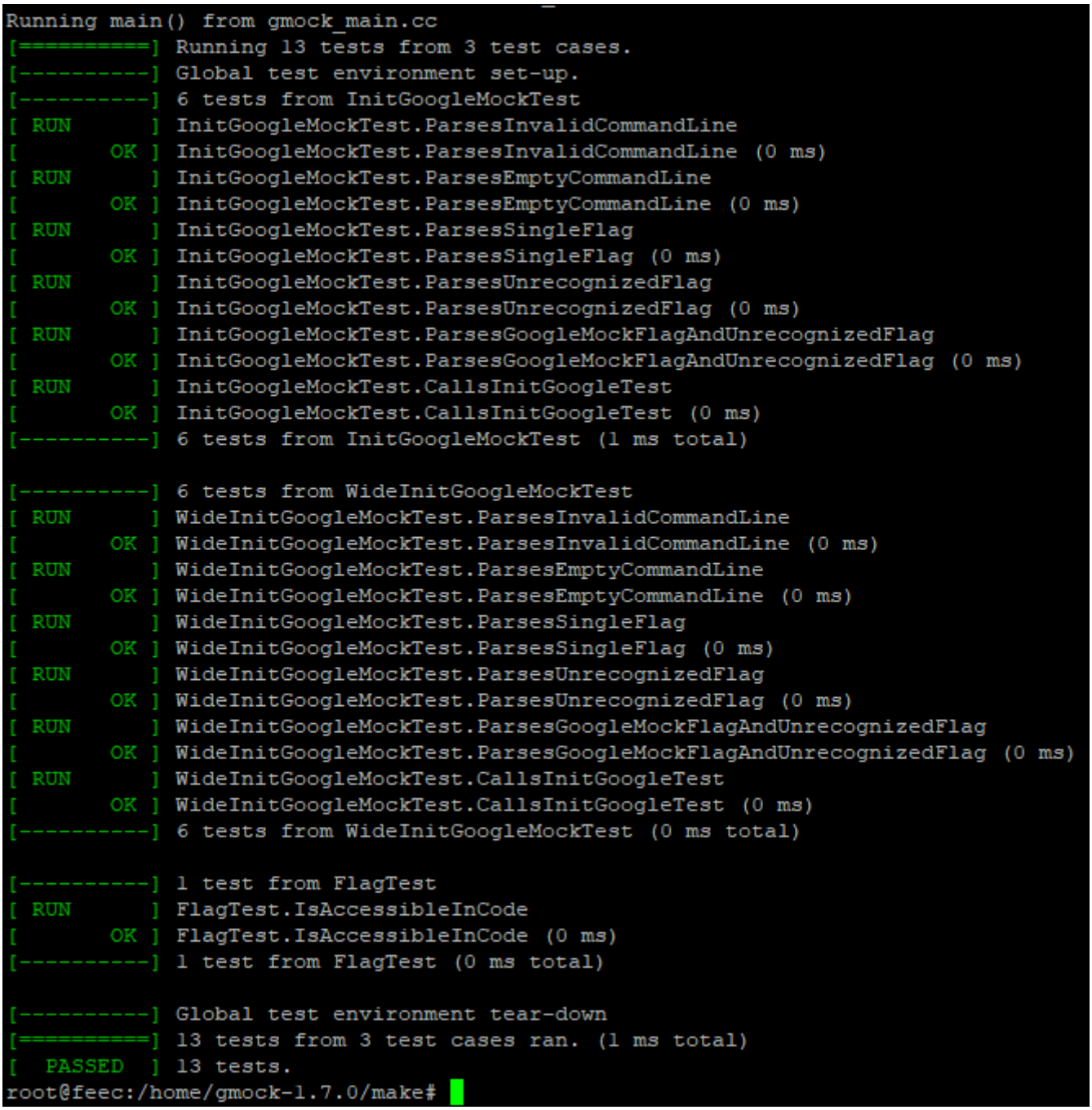


Figura 1°

8 . Por uma boa prática, renomeie o diretório do Gmok, conforme a ilustração a seguir, utilizando os comandos:

```
cd /home/ && mv /home/gmock-1.7.0/ gmock && ls -lha /home/ |  grep gmo
```

9. Nesta etapa iremos fazer o download e instalação do pacote: **protobuf** (Google), onde possui muitas versões no site oficial, porém na fonte original deste tutorial é usada a versão **3.2.0**, porém com esta versão tivemos problemas durante o make check, já a versão **3.7.1**, o resultado foi positivo e não gerou erros, execute os comandos:

```
wget https://github.com/protocolbuffers/protobuf/archive/v3.7.1.zip
cd /home/ && git clone https://github.com/google/protobuf.git
```

10. Crie o diretório p4 dentro de /home e mova o arquivo v3.7.1.zip para o diretório p4

```
mkdir p4 && mv v3.7.1.zip p4/
```

11. Extraindo o protobuf dentro do diretório p4

```
cd p4/ && unzip v3.7.1.zip
```

12. Dentro do diretório /home, copie o diretório gmok para o diretório protobuf

```
cp -R /home/gmock/ /home/p4/protobuf-3.7.1/
```

13. Acesse o diretório: /home/protobuf-3.7.1/gmock/make e execute o comando a seguir:

```
./gmock_test
```

Importante:

O resultado deverá ser semelhante ao da figura 1º, porém a saída do make check é muito importante, onde todos os módulos precisam retornar como “PASS”, caso contrário, ocorrerão erros durante o tutorial, se isso acontecer, não siga em frente, resolva os problemas antes.

14. Acesse o diretório: /home/p4/protobuf-3.7.1 e execute os comandos a seguir:

```
./autogen.sh && ./configure
```

15. Execute os comandos a seguir e dependendo do seu hardware, este procedimento deverá levar alguns minutos:

```
make && make check
```

Importante:

Resultado da etapa **15º** na figura 2º

```
protobuf/compiler/objectivec/objectivec_helpers.o google/protobuf/compiler/objectivec/objectivec_map_f
ield.o google/protobuf/compiler/objectivec/objectivec_message.o google/protobuf/compiler/objectivec/ob
jectivec_message_field.o google/protobuf/compiler/objectivec/objectivec_oneof.o google/protobuf/compil
er/objectivec/objectivec_primitive_field.o google/protobuf/compiler/php/php_generator.o google/protobu
f/compiler/python/python_generator.o google/protobuf/compiler/ruby/ruby_generator.o google/protobuf/co
mpiler/csharp/csharp_doc_comment.o google/protobuf/compiler/csharp/csharp_enum.o google/protobuf/compi
ler/csharp/csharp_enum_field.o google/protobuf/compiler/csharp/csharp_field_base.o google/protobuf/com
piler/csharp/csharp_generator.o google/protobuf/compiler/csharp/csharp_helpers.o google/protobuf/compi
ler/csharp/csharp_map_field.o google/protobuf/compiler/csharp/csharp_message.o google/protobuf/compile
r/csharp/csharp_message_field.o google/protobuf/compiler/csharp/csharp_primitive_field.o google/protob
uf/compiler/csharp/csharp_reflection_class.o google/protobuf/compiler/csharp/csharp_repeated_enum_fiel
d.o google/protobuf/compiler/csharp/csharp_repeated_message_field.o google/protobuf/compiler/csharp/cs
harp_repeated_primitive_field.o google/protobuf/compiler/csharp/csharp_source_generator_base.o google/
protobuf/compiler/csharp/csharp_wrapper_field.o
ar: `u' modifier ignored since `D' is the default (see `U')
libtool: link: ranlib .libs/libprotoc.a
libtool: link: ( cd ".libs" && rm -f "libprotoc.la" && ln -s "../libprotoc.la" "libprotoc.la" )
depbase=`echo google/protobuf/compiler/main.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`;\
g++ -DHAVE_CONFIG_H -I. -I..    -pthread -DHAVE_PTHREAD=1 -DHAVE_ZLIB=1 -Wall -Wno-sign-compare -O2 -g
 -std=c++11 -DNDEBUG -MT google/protobuf/compiler/main.o -MD -MP -MF $depbase.Tpo -c -o google/protobu
f/compiler/main.o google/protobuf/compiler/main.cc &&\
mv -f $depbase.Tpo $depbase.Po
/bin/bash ../libtool  --tag=CXX   --mode=link g++ -pthread -DHAVE_PTHREAD=1 -DHAVE_ZLIB=1 -Wall -Wno-s
ign-compare -O2 -g -std=c++11 -DNDEBUG -pthread  -o protoc google/protobuf/compiler/main.o  libprotobu
f.la libprotoc.la -lz
libtool: link: g++ -pthread -DHAVE_PTHREAD=1 -DHAVE_ZLIB=1 -Wall -Wno-sign-compare -O2 -g -std=c++11 -
DNDEBUG -pthread -o .libs/protoc google/protobuf/compiler/main.o  ./.libs/libprotobuf.so ./.libs/libpr
otoc.so -lz -pthread
make[2]: Leaving directory '/home/p4/protobuf-3.7.1/src'
make[1]: Leaving directory '/home/p4/protobuf-3.7.1'
root@feec:/home/p4/protobuf-3.7.1#
```

Figura 2°

16. Para finalizar a instalação do protobuf, execute o comando abaixo, o retorno deverá ser semelhante ao da figura 3º

```
sudo make install
```

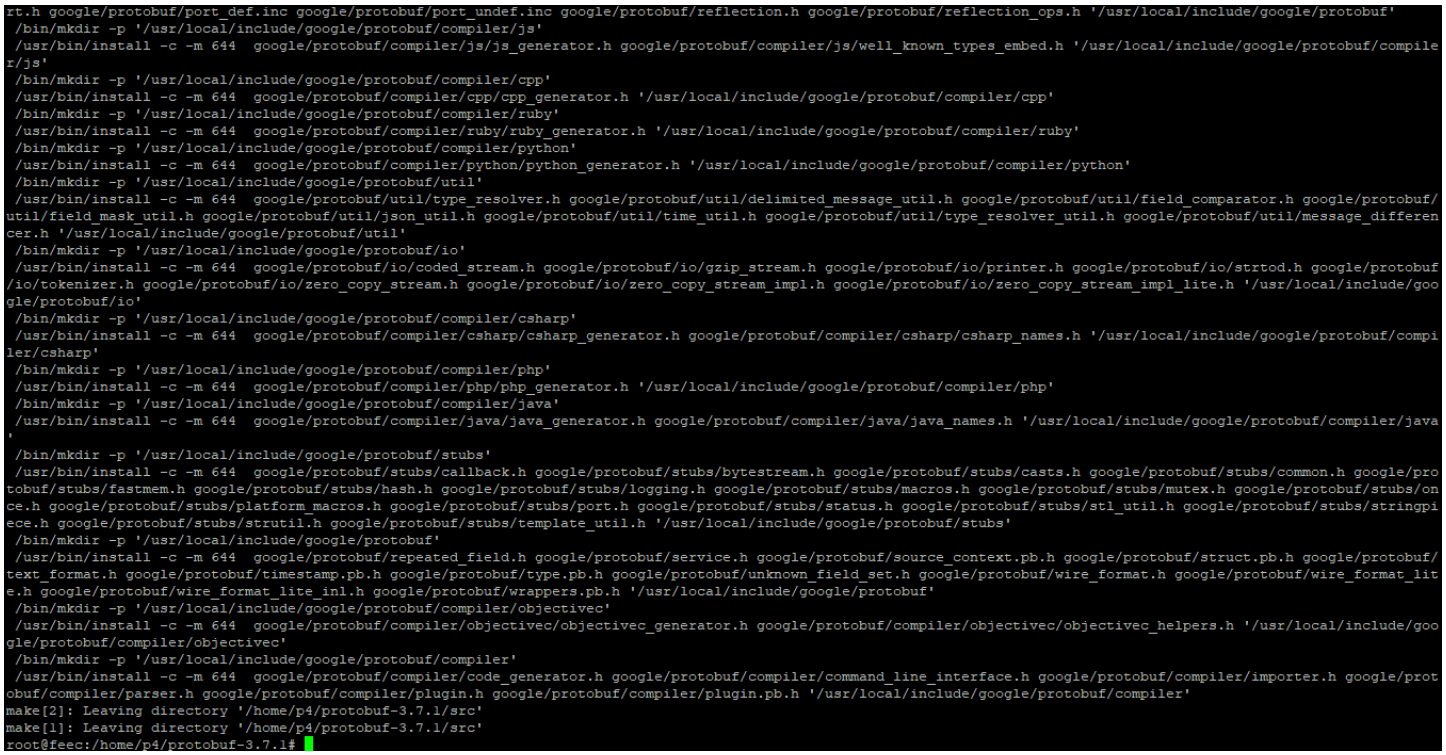


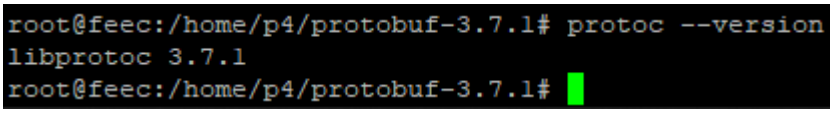
Figura 3°

17. Atualizando o cache da biblioteca compartilhada, execute o seguinte comando:

```
sudo ldconfig
```

18. Visualizando a versão do protubuf, se retornar a versão **3.7.1**, informa que o protubuf foi instalado com sucesso, execute o seguinte comando:

```
protoc --version
```



19. Nesta etapa iremos instalar o p4c, execute os seguintes comandos:

```
sudo apt-get install cmake g++ git automake libtool libgc-dev bison
flex libfl-dev libgmp-dev libboost-dev libboost-iostreams-dev
libboost-graph-dev llvm pkg-config python python-scapy python-ipaddr
python-ply tcpdump
```

20. Em seguida, baixe o código-fonte do p4c:

```
cd /home && git clone --recursive https://github.com/p4lang/p4c.git
```

21. Mova o diretório p4c para o diretório p4 e execute o comando: **./bootstrap.sh**

```
mv p4c/p4/ && cd p4c/ && ./bootstrap.sh
```

```
CMake Warning at backends/bmv2/CMakeLists.txt:210 (MESSAGE):
  BMv2 PSA switch is not available, not adding PSA BMv2 tests


-- Start configuring eBPF back end
-- Detected kernel version: 4.15.0-142-generic
-- Check LLVM version with 'llvm-config --version'
-- Found LLVM 6.0.0
CMake Warning at backends/ebpf/CMakeLists.txt:165 (message):
  Missing the libbpf dependency, disabling kernel tests.  You can install
  libbpf by running './build_libbpf' in the
  /home/p4/p4c/backends/ebpf/runtime folder.


-- Added 16 tests to 'ebpf-bcc' (0 xfails)
-- Added 16 tests to 'ebpf' (1 xfails)
-- Done with configuring BPF back end
-- Added 16 tests to 'ubpf' (0 xfails)
-- Added 6 tests to 'p4' (0 xfails)
-- Added 851 tests to 'p4' (1 xfails)
-- Added 6 tests to 'p4unroll' (0 xfails)
-- Added 6 tests to 'err' (0 xfails)
-- Added 234 tests to 'err' (0 xfails)
-- Added 191 tests to 'p14_to_16' (0 xfails)
-- CTest parallel: -j 4
-- Configuring done
-- Generating done
-- Build files have been written to: /home/p4/p4c/build
### Configured for building in 'build' folder
root@feec:/home/p4/p4c#
```

Resultado da etapa 21°

22. Acesse o diretório **build** dentro do diretório p4c e compile com comando a seguir:

```
cmake ..
```

23. Em seguida efetue a compilação com o comando a seguir:

```
make -j4
```

Importante:

Este procedimento deverá levar alguns minutos, dependendo da configuração do seu hardware, a mesma deverá iniciar em **0%** e finalizar em **100%** sem erros, conforme as próximas ilustrações:

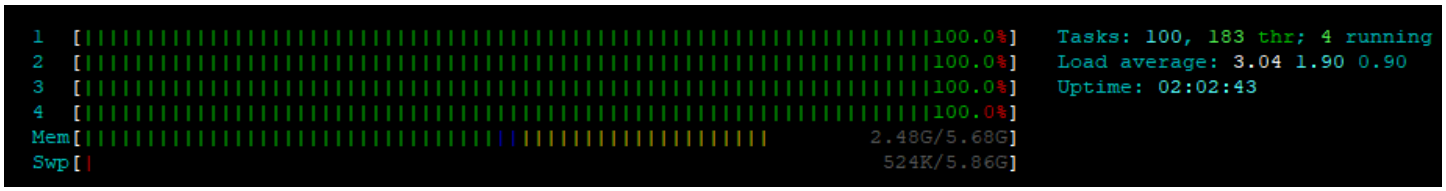
```
root@feec:/home/p4/p4c/build# make -j4
Scanning dependencies of target update_includes
Scanning dependencies of target linkgraphs
Scanning dependencies of target linkbmv2
Scanning dependencies of target linkp4cdpdk
[  0%] Built target linkgraphs
[  0%] Built target linkp4cdpdk
Scanning dependencies of target linkp4cebpf
Scanning dependencies of target linkp4cubpf
[  0%] Built target update_includes
[  0%] Built target linkbmv2
Scanning dependencies of target linkp4test
Scanning dependencies of target p4ctoolkit
[  0%] Built target linkp4cubpf
[  0%] Built target linkp4cebpf
[  0%] Built target linkp4test
[  1%] Building CXX object lib/CMakeFiles/p4ctoolkit.dir/unified_libp4ctoolkit_srcs_1.cpp.o
Scanning dependencies of target mkP4configdir
Scanning dependencies of target mkvldirs
Scanning dependencies of target mkp4dirs
[  1%] Built target mkvldirs
[  1%] Built target mkP4configdir
[  1%] Built target mkp4dirs
Scanning dependencies of target gtest
Scanning dependencies of target p4c_driver
[  2%] Copying p4c driver
[  3%] Building CXX object test/CMakeFiles/gtest.dir/frameworks/gtest/googletest/src/gtest-all.cc.o
[  3%] Built target p4c_driver
[  5%] Linking CXX static library libgtest.a
[  5%] Built target gtest
[  6%] Linking CXX static library libp4ctoolkit.a
[  6%] Built target p4ctoolkit
[  8%] [FLEX][IRgenLexer] Building scanner with flex 2.6.4
[  8%] [BISON][IRgenParser] Building parser with bison 3.0.4
Scanning dependencies of target irgenerator
[ 11%] Building CXX object tools/ir-generator/CMakeFiles/irgenerator.dir/unified_irgenerator_srcs_1.cpp.o
[ 11%] Building CXX object tools/ir-generator/CMakeFiles/irgenerator.dir/ir-generator.cpp.o
```

```
Scanning dependencies of target genIR
[ 13%] Generating IR class files
[ 13%] Built target genIR
[ 16%] [BISON][p4Parser] Building parser with bison 3.0.4
[ 16%] [BISON][vlParser] Building parser with bison 3.0.4
Scanning dependencies of target ir
Scanning dependencies of target midend
[ 17%] Building CXX object ir/CMakeFiles/ir.dir/unified_ir_srcs_1.cpp.o
[ 18%] Building CXX object midend/CMakeFiles/midend.dir/unified_midend_srcs_1.cpp.o
[ 20%] Running Flex on parsers/vl
[ 20%] Building CXX object ir/CMakeFiles/ir.dir/unified_ir_srcs_2.cpp.o
[ 21%] Running Flex on parsers/p4
Scanning dependencies of target frontend
[ 22%] Building CXX object frontends/CMakeFiles/frontend.dir/parsers/p4/p4lexer.cc.o
[ 23%] Building CXX object frontends/CMakeFiles/frontend.dir/parsers/p4/p4parser.cpp.o
[ 25%] Building CXX object frontends/CMakeFiles/frontend.dir/parsers/vl/vllexer.cc.o
[ 26%] Building CXX object midend/CMakeFiles/midend.dir/unified_midend_srcs_2.cpp.o
[ 26%] Building CXX object frontends/CMakeFiles/frontend.dir/parsers/vl/vlparser.cpp.o
[ 27%] Linking CXX static library libir.a
[ 27%] Built target ir
[ 28%] Building CXX object frontends/CMakeFiles/frontend.dir/__/ir/ir-generated.cpp.o
[ 30%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_1.cpp.o
[ 31%] Building CXX object midend/CMakeFiles/midend.dir/unified_midend_srcs_3.cpp.o
[ 32%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_2.cpp.o
[ 33%] Building CXX object midend/CMakeFiles/midend.dir/unified_midend_srcs_4.cpp.o
[ 33%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_3.cpp.o
[ 35%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_4.cpp.o
[ 36%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_5.cpp.o
[ 37%] Linking CXX static library libmidend.a
[ 37%] Built target midend
[ 38%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_6.cpp.o
[ 40%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_7.cpp.o
[ 41%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_frontend_sources_8.cpp.o
[ 41%] Building CXX object frontends/CMakeFiles/frontend.dir/unified_extension_frontend_sources_1.cpp.o
```

Importante:

O uso de CPU é intenso, conforme ilustra imagem abaixo utilizando o software htop, caso não tenha o mesmo instalado, instale com os comandos:

```
sudo apt update && sudo apt-get install htop -y
```




```
Scanning dependencies of target p4c-ubpf
[ 66%] Building CXX object backends/ubpf/CMakeFiles/p4c-ubpf.dir/unified_p4c_ubpf_sources_1.cpp.o
[ 66%] Built target p4c-bm2-psa
Scanning dependencies of target p4test
[ 67%] Building CXX object backends/p4test/CMakeFiles/p4test.dir/unified_p4test_srcs_1.cpp.o
[ 68%] Linking CXX executable p4c-ebpf
[ 68%] Built target p4c-ebpf
Scanning dependencies of target p4c-graphs
[ 70%] Building CXX object backends/graphs/CMakeFiles/p4c-graphs.dir/unified_graphs_srcs_1.cpp.o
[ 71%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/controlFlowGraph.cpp.o
[ 71%] Linking CXX executable p4test
[ 71%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/deparser.cpp.o
[ 72%] Linking CXX executable p4c-ubpf
[ 72%] Built target p4test
Scanning dependencies of target gtestp4c
[ 73%] Building CXX object test/CMakeFiles/gtestp4c.dir/unified_gtest_base_sources_1.cpp.o
[ 73%] Built target p4c-ubpf
[ 75%] Building CXX object test/CMakeFiles/gtestp4c.dir/unified_gtest_base_sources_2.cpp.o
[ 76%] Linking CXX executable p4c-graphs
[ 76%] Built target p4c-graphs
[ 77%] Building CXX object test/CMakeFiles/gtestp4c.dir/unified_gtest_base_sources_3.cpp.o
[ 78%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/expression.cpp.o
[ 80%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/extern.cpp.o
[ 81%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/globals.cpp.o
[ 82%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/header.cpp.o
[ 83%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/lower.cpp.o
[ 85%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/metermap.cpp.o
[ 86%] Linking CXX executable gtestp4c
[ 87%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/parser.cpp.o
[ 87%] Built target gtestp4c
[ 88%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/common/programStructure.cpp.o
[ 90%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/__/bmv2/psa_switch/psaSwitch.cpp.o
[ 90%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/backend.cpp.o
[ 91%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/main.cpp.o
[ 92%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/midend.cpp.o
[ 93%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/dpdkHelpers.cpp.o
[ 95%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/dpdkProgram.cpp.o
[ 96%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/dpdkVarCollector.cpp.o
[ 97%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/dpdkArch.cpp.o
[ 98%] Building CXX object backends/dpdk/CMakeFiles/p4c-dpdk.dir/dpdkAsmOpt.cpp.o
[100%] Linking CXX executable p4c-dpdk
[100%] Built target p4c-dpdk
root@feec:/home/p4/p4c/build#
```

24. Testando a unidade p4c, este processo levará alguns minutos, dependendo do seu hardware, onde dever iniciar em: **1** e finalizar em: **1399**, conforme a ilustração a seguir:

```
make check -j4
```

```
root@feec:/home/p4/p4c/build# make check -j4
Scanning dependencies of target check-all
[100%] Running tests for tag all
Test project /home/p4/p4c/build
        Start   1: cpplint
        Start   2: dpdk/testdata/p4_16_samples/psa-action-profile1.p4
        Start   3: dpdk/testdata/p4_16_samples/psa-action-profile2.p4
        Start   4: dpdk/testdata/p4_16_samples/psa-action-profile3.p4
  1/1399 Test   #3: dpdk/testdata/p4_16_samples/psa-action-profile2.p4 ..........   Passed    4.64 sec
  2/1399 Test   #4: dpdk/testdata/p4_16_samples/psa-action-profile3.p4 ..........   Passed    4.43 sec
  3/1399 Test   #2: dpdk/testdata/p4_16_samples/psa-action-profile1.p4 ..........   Passed    4.68 sec
        Start   5: dpdk/testdata/p4_16_samples/psa-action-profile4.p4
        Start   6: dpdk/testdata/p4_16_samples/psa-action-selector1.p4
        Start   7: dpdk/testdata/p4_16_samples/psa-action-selector2.p4
  4/1399 Test   #5: dpdk/testdata/p4_16_samples/psa-action-profile4.p4 ..........   Passed    2.73 sec
        Start   8: dpdk/testdata/p4_16_samples/psa-action-selector3.p4
  5/1399 Test   #7: dpdk/testdata/p4_16_samples/psa-action-selector2.p4 ..........   Passed    2.86 sec
  6/1399 Test   #6: dpdk/testdata/p4_16_samples/psa-action-selector1.p4 ..........   Passed    2.86 sec
        Start   9: dpdk/testdata/p4_16_samples/psa-basic-counter-bmv2.p4
        Start  10: dpdk/testdata/p4_16_samples/psa-counter1.p4
  7/1399 Test  #10: dpdk/testdata/p4_16_samples/psa-counter1.p4 ..................   Passed    2.40 sec
  8/1399 Test   #8: dpdk/testdata/p4_16_samples/psa-action-selector3.p4 ..........   Passed    2.76 sec
        Start  11: dpdk/testdata/p4_16_samples/psa-counter2.p4
        Start  12: dpdk/testdata/p4_16_samples/psa-counter3.p4
```

Importante:

O número de testes em cada versão do p4c não é exatamente o mesmo, sendo assim, isso não afeta o resultado do processo, o mesmo deverá passar de **100%,** caso ocorra, alguma falha ou erro, não prossiga para a próxima etapa.

```
      Start 1399: gtestp4c
1396/1399 Test #1396: p14_to_16/testdata/p4_14_samples/wide_action1.p4 ..............   Passed    4.30 sec
1397/1399 Test #1397: p14_to_16/testdata/p4_14_samples/wide_action3.p4 ..............   Passed    4.40 sec
1398/1399 Test #1399: gtestp4c ......................................................   Passed   95.17 sec
1399/1399 Test #1398: p14_to_16/testdata/p4_14_samples/switch_20160512/switch.p4 ....   Passed  161.92 sec
```

25. Efetue a instalação com o comando a seguir, conforme a ilustração a seguir:

```
sudo make install
```

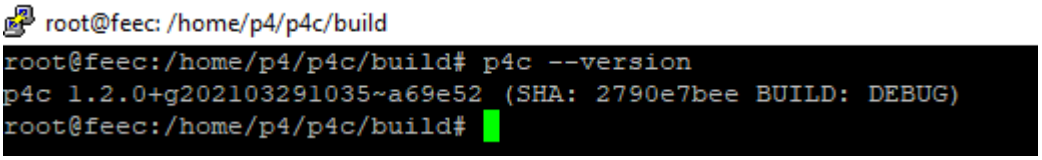
Importante:

A etapa 25 não deverá conter erros, caso gere algum, retorne as etapas anteriores ou consulte as fontes originais de onde foi retirado este tutorial que foi totalmente modificado para um ambiente físico.

```
root@feec:/home/p4/p4c/build# sudo make install
[  0%] Built target update_includes
[  0%] Built target linkgraphs
[  0%] Built target linkp4cdpdk
[  0%] Built target linkbmv2
[  0%] Built target linkp4cebpf
[  0%] Built target linkp4cubpf
[  0%] Built target linkp4test
[  1%] Built target p4c_driver
[  3%] Built target p4ctoolkit
[ 10%] Built target irgenerator
[ 11%] Built target genIR
[ 13%] Built target ir
[ 13%] Built target mkP4configdir
[ 13%] Built target mkvldirs
[ 13%] Built target mkp4dirs
[ 35%] Built target frontend
[ 45%] Built target controlplane
[ 50%] Built target midend
[ 76%] Built target p4c-dpdk
[ 78%] Built target bmv2backend
[ 81%] Built target p4c-bm2-ss
[ 83%] Built target p4c-bm2-psa
[ 86%] Built target p4c-ebpf
[ 88%] Built target p4c-ubpf
[ 90%] Built target p4test
[ 92%] Built target p4c-graphs
[ 95%] Built target gtest
[100%] Built target gtestp4c
Install the project...
-- Install configuration: "DEBUG"
-- Installing: /usr/local/share/p4c/p4include
-- Installing: /usr/local/share/p4c/p4include/psa.p4
-- Installing: /usr/local/share/p4c/p4include/ubpf_model.p4
-- Installing: /usr/local/share/p4c/p4include/p4d2model.p4
-- Installing: /usr/local/share/p4c/p4include/core.p4
-- Installing: /usr/local/share/p4c/p4include/v1model.p4
-- Installing: /usr/local/share/p4c/p4include/ebpf_model.p4
-- Installing: /usr/local/share/p4c/p4include/xdp_model.p4
-- Installing: /usr/local/bin/p4c
-- Installing: /usr/local/share/p4c/p4c_src
-- Installing: /usr/local/share/p4c/p4c_src/util.py
-- Installing: /usr/local/share/p4c/p4c_src/main.py
-- Installing: /usr/local/share/p4c/p4c_src/driver.py
-- Installing: /usr/local/share/p4c/p4c_src/__init__.py
-- Installing: /usr/local/share/p4c/p4c_src/config.py
-- Installing: /usr/local/share/p4c/p4c_src/p4c.bmv2.cfg
-- Installing: /usr/local/share/p4c/p4c_src/p4c.ebpf.cfg
-- Installing: /usr/local/share/p4c/p4c_src/p4c.dpdk.cfg
-- Installing: /usr/local/bin/p4c-dpdk
-- Installing: /usr/local/bin/p4c-bm2-ss
-- Installing: /usr/local/bin/p4c-bm2-psa
-- Installing: /usr/local/bin/p4c-ebpf
-- Up-to-date: /usr/local/share/p4c/p4include
-- Up-to-date: /usr/local/share/p4c/p4include/ebpf_model.p4
-- Installing: /usr/local/bin/p4c-ubpf
-- Up-to-date: /usr/local/share/p4c/p4include
-- Up-to-date: /usr/local/share/p4c/p4include/ubpf_model.p4
-- Installing: /usr/local/bin/p4test
-- Installing: /usr/local/bin/p4c-graphs
root@feec:/home/p4/p4c/build#
```

26. Verificando a versão do p4c, e se nesta etapa tiver um retorno, semelhante a ilustração a seguir, a instalação foi p4c está concluída:

```
p4c –version
```

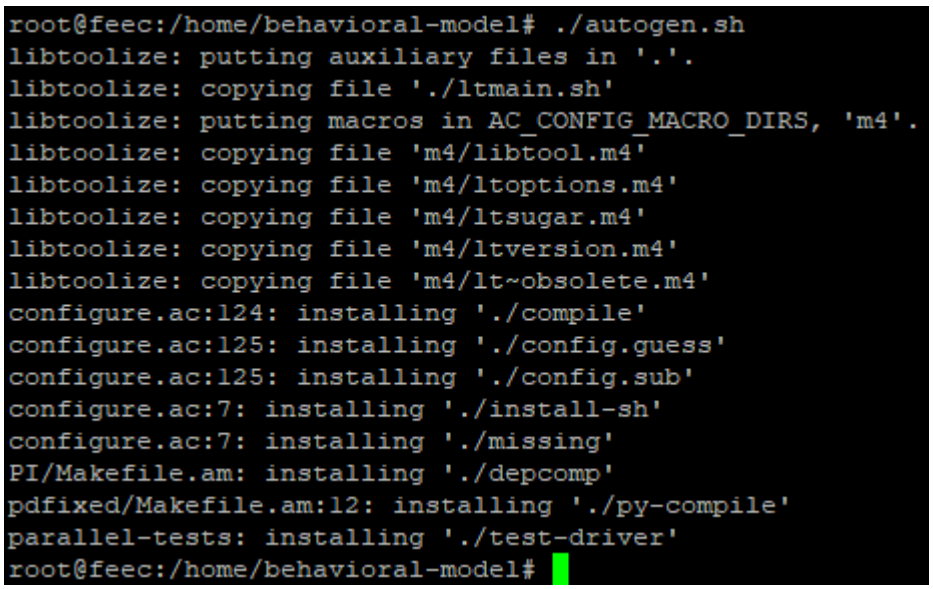


27. Nesta etapa iremos baixar o bmv2, que é o modelo comportamental, chamado de módulo de chave bmv2 e denominado "bmv2"

```
cd /home && git clone git://github.com/p4lang/behavioral-model.git
```

28. Após a etapa anterior, iremos acessar o diretório do bmv2 e efetuar algumas instalações, conforme os comandos a seguir:

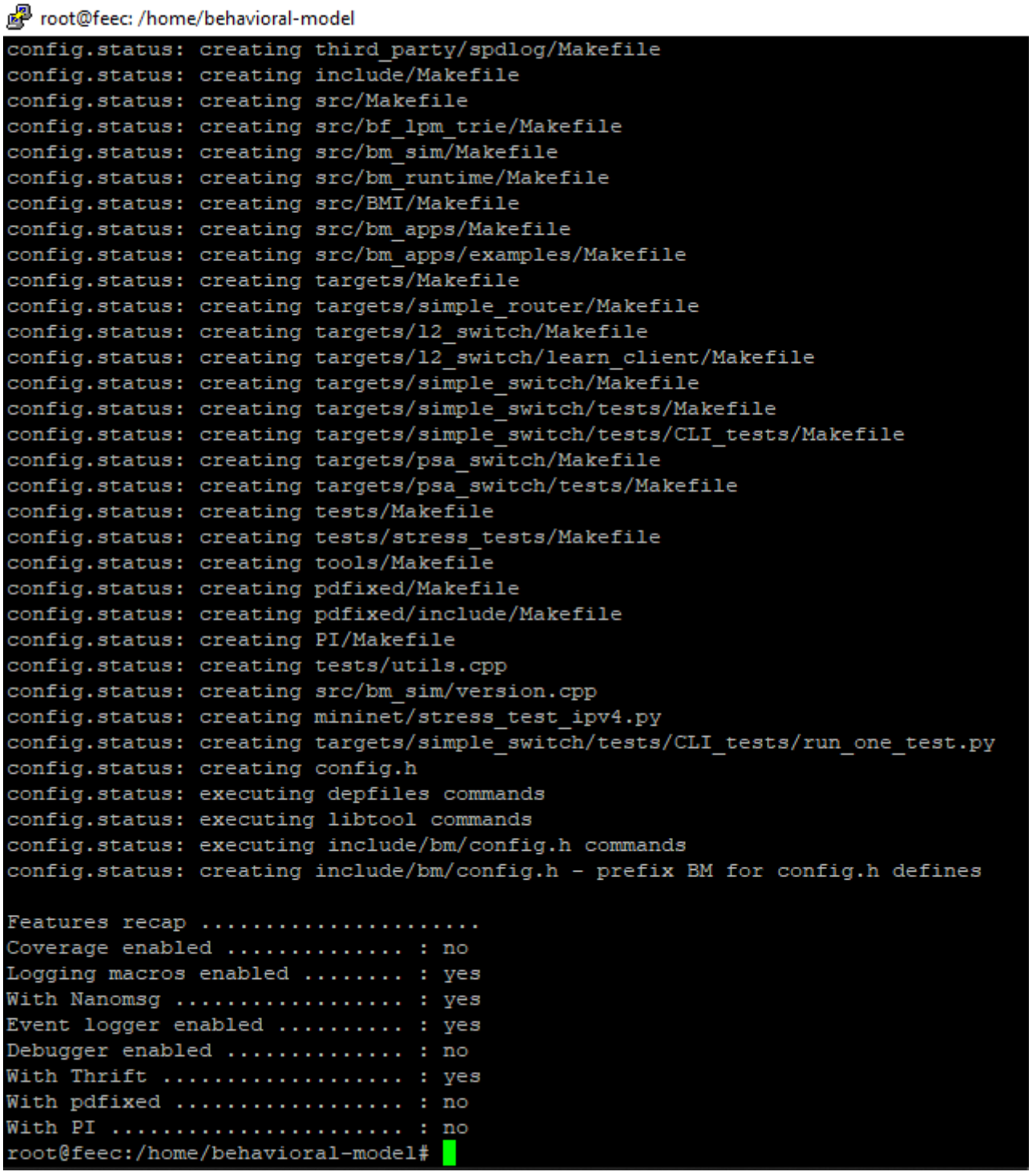
```
cd behavioral-model/ && ./install_deps.sh && ./autogen.sh
```

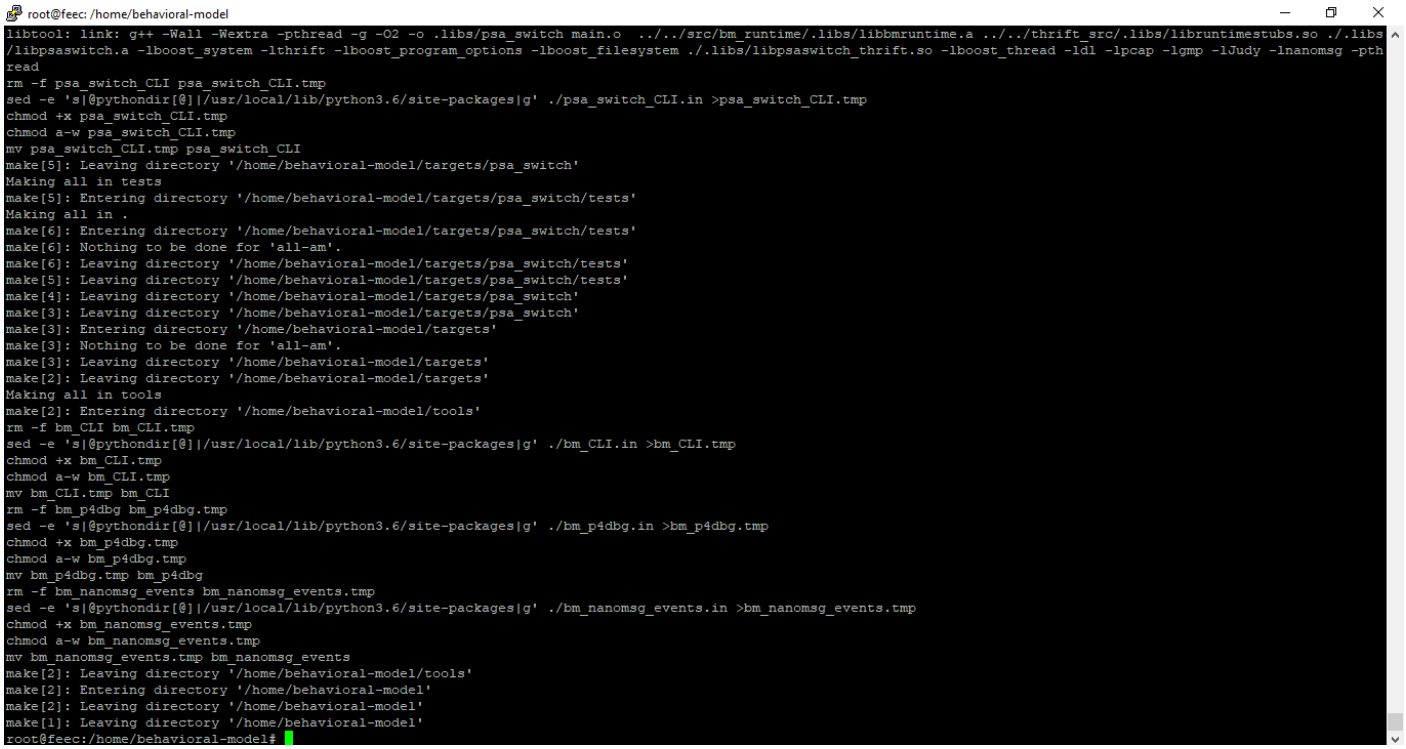


Resultados da etapa 28°

29. Ainda dentro do diretório: /home/behavioral-model, execute os comandos a seguir:

```
./configure && make
```

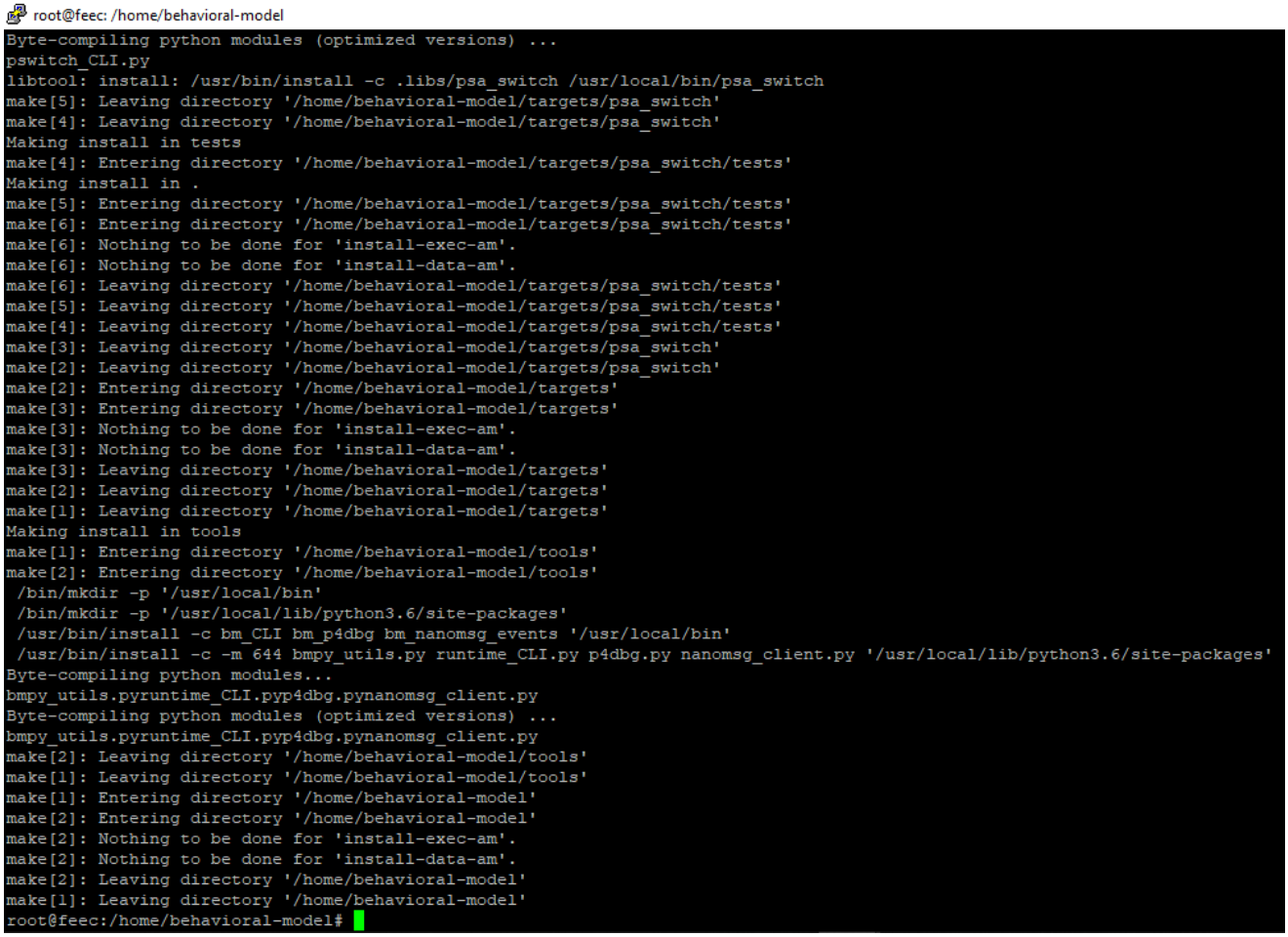




Resultado da etapa 29°

30. Ainda dentro do diretório: /home/behavioral-model, execute o comando seguir:

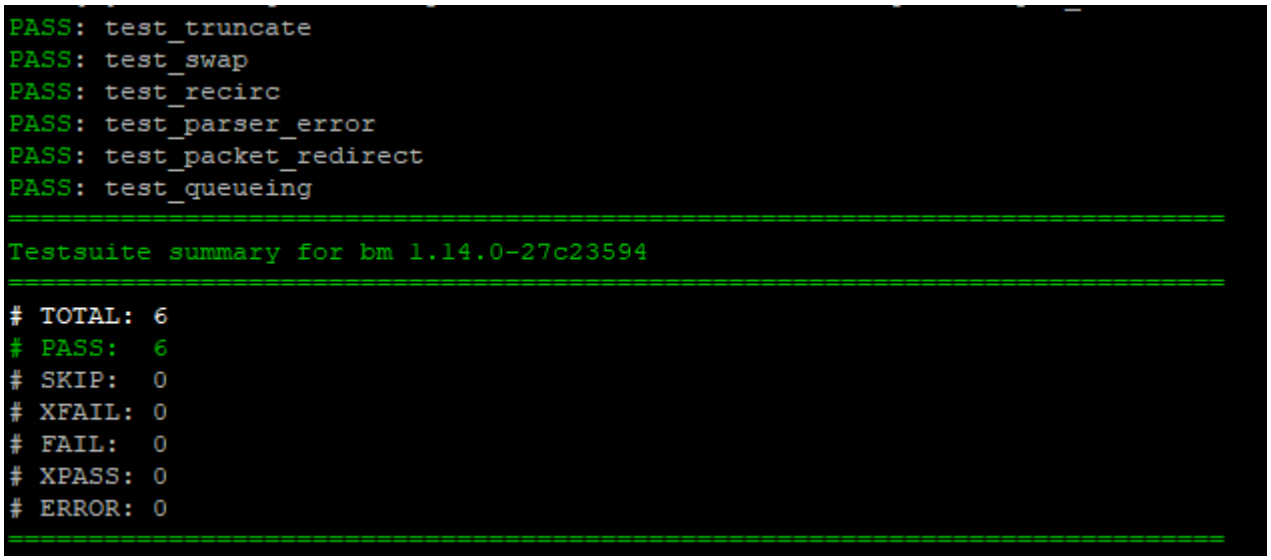
```
sudo make install -j4
```

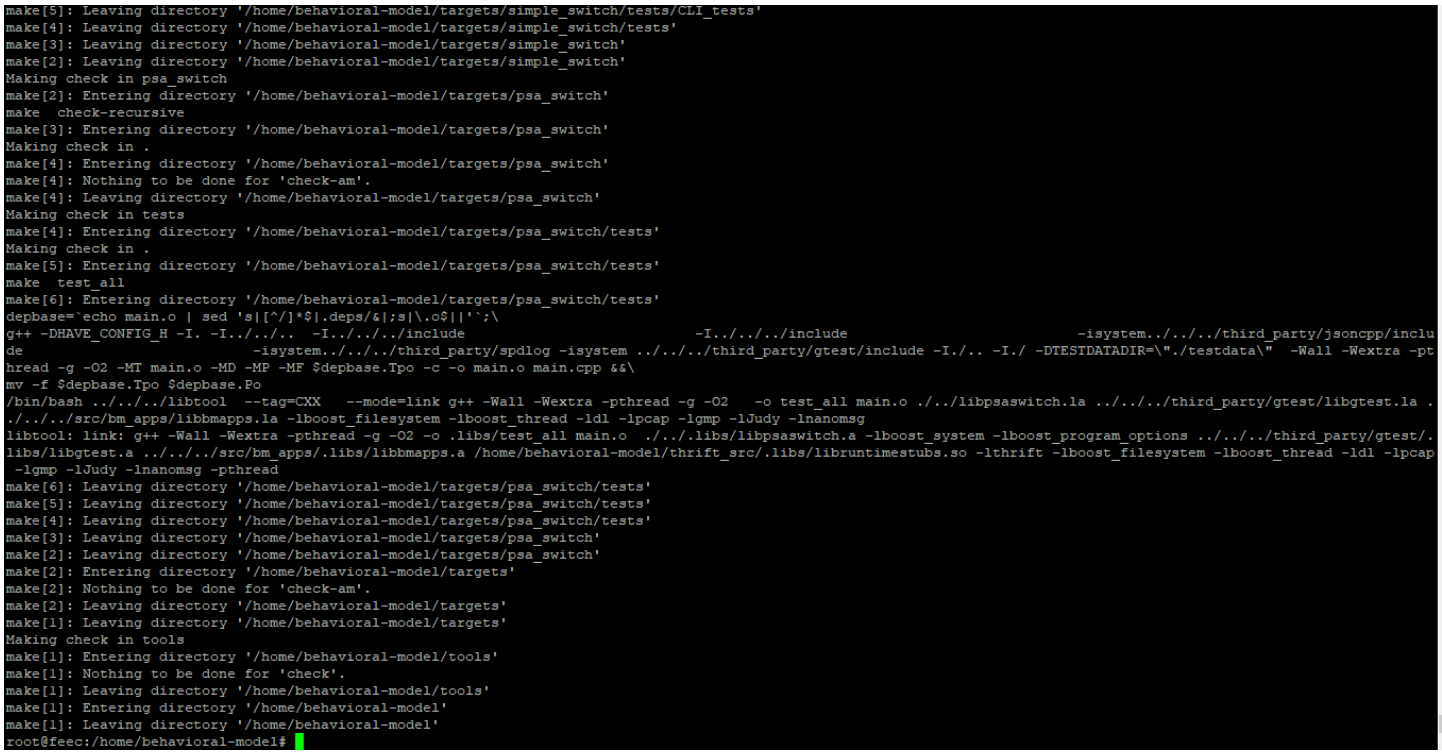


Resultado da etapa 30°

31. Nesta etapa, será efetuado um teste de unidade do **bmv2**, que deve ser executado após a instalação data etapa **30°** ser concluída, caso contrário, erros serão retornados, execute o comando a seguir:

```
make check -j4
```





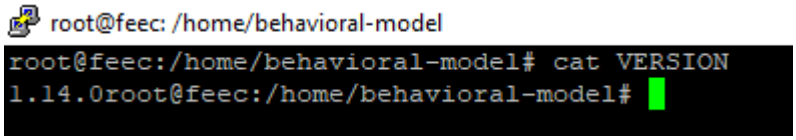
Resultados da etapa 31°

Importante:

A saída do make check é muito importante e deve ser executada, onde todos os módulos devem ser APROVADOS, se isso ocorrer, siga para a próxima etapa, caso contrário, ocorrerão erros inesperados.

32. Nesta etapa final, iremos verificar se tivemos êxito em todo o processo, para isso, execute o comando a seguir:

```
cat VERSION
```



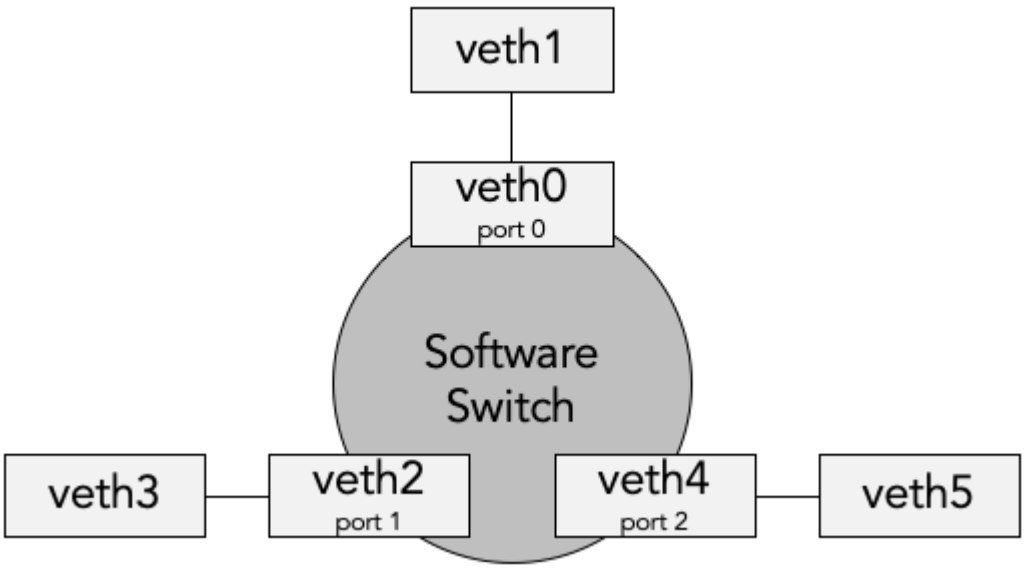
Resultado da etapa 32°

Efetuando testes pós instalado do Compilador P4

Objetivo:

Criar um programa P4 que processará apenas pacotes IPv4 sobre o protocolo Ethernet que conterá apenas uma única tabela, que deverá fazer uma pesquisa de correspondência de prefixo mais longa no endereço IP de destino, para decidir a porta de saída, este é o objetivo do ambiente pós-instalação do compilador P4.

Topologia do ambiente de testes com P4:



1. Execute do comando **ldconfig** que será necessário para fazer o cache necessário para as bibliotecas compartilhadas mais recentes encontradas nos diretórios especificados na linha de comando, execute o comando a seguir:

```
sudo ldconfig
```

2. Crie um diretório chamado projetos_p4 dentro de /home, conforme o comando a seguir:

```
mkdir -p /home/projetos_p4
```

3. Acesse o diretório criado na etapa 2, e com um editor de texto de sua escolha, crie um arquivo chamado p1.p4 com o conteúdo a seguir:

```
cd /home/projetos_p4 && vim p1.p4
```

```
#include <core.p4>
#include <v1model.p4>

typedef bit<48> EthernetAddress;
typedef bit<32> IPv4Address;

header ethernet_t {
    EthernetAddress dst_addr;
    EthernetAddress src_addr;
    bit<16>         ether_type;
}

header ipv4_t {
    bit<4>      version;
    bit<4>      ihl;
    bit<8>      diffserv;
    bit<16>     total_len;
    bit<16>     identification;
    bit<3>      flags;
    bit<13>     frag_offset;
    bit<8>      ttl;
    bit<8>      protocol;
    bit<16>     hdr_checksum;
    IPv4Address src_addr;
```

```
    IPv4Address dst_addr;
}

struct headers_t {
    ethernet_t ethernet;
    ipv4_t     ipv4;
}

struct metadata_t {
}

error {
    IPv4IncorrectVersion,
    IPv4OptionsNotSupported
}

parser my_parser(packet_in packet,
                 out headers_t hd,
                 inout metadata_t meta,
                 inout standard_metadata_t standard_meta)
{
    state start {
        packet.extract(hd.ethernet);
        transition select(hd.ethernet.ether_type) {
            0x0800:  parse_ipv4;
            default: accept;
        }
    }

    state parse_ipv4 {
        packet.extract(hd.ipv4);
        verify(hd.ipv4.version == 4w4, error.IPv4IncorrectVersion);
        verify(hd.ipv4.ihl == 4w5, error.IPv4OptionsNotSupported);
        transition accept;
    }
}

control my_deparser(packet_out packet,
                    in headers_t hdr)
{
```

```
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);
    }
}

control my_verify_checksum(inout headers_t hdr,
                           inout metadata_t meta)
{
    apply { }
}

control my_compute_checksum(inout headers_t hdr,
                            inout metadata_t meta)
{
    apply { }
}

control my_ingress(inout headers_t hdr,
                   inout metadata_t meta,
                   inout standard_metadata_t standard_metadata)
{
    bool dropped = false;

    action drop_action() {
        mark_to_drop(standard_metadata);
        dropped = true;
    }

    action to_port_action(bit<9> port) {
        hdr.ipv4.ttl = hdr.ipv4.ttl - 1;
        standard_metadata.egress_spec = port;
    }

    table ipv4_match {
        key = {
            hdr.ipv4.dst_addr: lpm;
        }
        actions = {
            drop_action;
```

```
            to_port_action;
        }
        size = 1024;
        default_action = drop_action;
    }

    apply {
        ipv4_match.apply();
        if (dropped) return;
    }
}

control my_egress(inout headers_t hdr,
                  inout metadata_t meta,
                  inout standard_metadata_t standard_metadata)
{
    apply { }
}

V1Switch(my_parser(),
         my_verify_checksum(),
         my_ingress(),
         my_egress(),
         my_compute_checksum(),
         my_deparser()) main;
```

Importante:

Este é um programa possui uma única tabela de pesquisa, que faz uma correspondência de prefixo mais longa (LPF) no endereço IP de destino no pacote recebido. A ação é descartar o pacote ou encaminhá-lo para uma porta de saída específica.

4. Efetue a compilação o programa P4, utilizando a opção -b opção, que seleciona bmv2 (**Behavioral Model Version 2**) como target, que é a chave de software que usaremos para executar o programa P4 e depois iremos listar o resultado com o comando: `ls -lha`

```
p4c -b bmv2 p1.p4 -o p1.bmv2
```

```
root@feec:/home/projetos_p4# p4c -b bmv2 p1.p4 -o p1.bmv2
root@feec:/home/projetos_p4# ls -lha
total 16K
drwxr-xr-x 3 root root 4,0K abr 21 22:20 .
drwxr-xr-x 8 root root 4,0K abr 21 22:13 ..
drwxr-xr-x 2 root root 4,0K abr 21 22:20 p1.bmv2
-rw-r--r-- 1 root root 2,7K abr 21 22:16 p1.p4
root@feec:/home/projetos_p4#
```

Resultado da etapa 4°

5. Nesta etapa, iremos criar interfaces Ethernet virtuais, que será dividida em três pares de interfaces virtuais Ethernet (veth):

   Uma veth interface é uma interface Ethernet virtual (ou seja, falsa), na qual um aplicativo (o switch de software em nosso caso), poderá enviar e receber pacotes Ethernet da mesma forma que uma interface Ethernet real.

   Caso ocorra um problema de uma interface veth, onde os pacotes não saiam de uma porta Ethernet real. Em vez disso, as veth interfaces são sempre criadas em pares, conforme o exemplo abaixo:

   veth0-veth1, veth2-veth3, e veth4-veth5

   Quando o aplicativo envia o pacote em uma interface veth, ele chega na outra veth interface do par.

   Durante a criação dos três pares de interfaces Ethernet virtuais, definimos a unidade de transferência de mensagens (MTU) de cada interface para **9500,** para permitir enviar e receber pacotes nessa unidade máxima. E desabilitamos o IPv6 em cada interface para impedir que o kernel envie solicitações de roteador e relatórios de escuta de multicast (isso não impede que o switch de software envie pacotes IPv6 pela interface).

   Importante:
   Os comandos serão executados LINHA a LINHA, ou o usuário poderá criar um script para agilizar o processo, conforme queira.

```
# Primeiro par: veth0-veth1
sudo ip link add name veth0 type veth peer name veth1
sudo ip link set dev veth0 up
sudo ip link set dev veth1 up
sudo ip link set veth0 mtu 9500
sudo ip link set veth1 mtu 9500
sudo sysctl net.ipv6.conf.veth0.disable_ipv6=1
sudo sysctl net.ipv6.conf.veth1.disable_ipv6=1
```

```
# Segundo par: veth2-veth3
sudo ip link add name veth2 type veth peer name veth3
sudo ip link set dev veth2 up
sudo ip link set dev veth3 up
sudo ip link set veth2 mtu 9500
sudo ip link set veth3 mtu 9500
sudo sysctl net.ipv6.conf.veth2.disable_ipv6=1
sudo sysctl net.ipv6.conf.veth3.disable_ipv6=1


# Terceiro par: veth4-veth5
sudo ip link add name veth4 type veth peer name veth5
sudo ip link set dev veth4 up
sudo ip link set dev veth5 up
sudo ip link set veth4 mtu 9500
sudo ip link set veth5 mtu 9500
sudo sysctl net.ipv6.conf.veth4.disable_ipv6=1
sudo sysctl net.ipv6.conf.veth5.disable_ipv6=1
```

root@feec: /home/projetos_p4

```
root@feec:/home/projetos_p4# sudo ip link add name veth0 type veth peer name veth1
root@feec:/home/projetos_p4# sudo ip link set dev veth0 up
root@feec:/home/projetos_p4# sudo ip link set dev veth1 up
root@feec:/home/projetos_p4# sudo ip link set veth0 mtu 9500
root@feec:/home/projetos_p4# sudo ip link set veth1 mtu 9500
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth0.disable_ipv6=1
net.ipv6.conf.veth0.disable_ipv6 = 1
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth1.disable_ipv6=1
net.ipv6.conf.veth1.disable_ipv6 = 1
root@feec:/home/projetos_p4#
root@feec:/home/projetos_p4#
root@feec:/home/projetos_p4# sudo ip link add name veth2 type veth peer name veth3
root@feec:/home/projetos_p4# sudo ip link set dev veth2 up
root@feec:/home/projetos_p4# sudo ip link set dev veth3 up
root@feec:/home/projetos_p4# sudo ip link set veth2 mtu 9500
root@feec:/home/projetos_p4# sudo ip link set veth3 mtu 9500
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth2.disable_ipv6=1
net.ipv6.conf.veth2.disable_ipv6 = 1
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth3.disable_ipv6=1
net.ipv6.conf.veth3.disable_ipv6 = 1
root@feec:/home/projetos_p4#
root@feec:/home/projetos_p4#
root@feec:/home/projetos_p4# sudo ip link add name veth4 type veth peer name veth5
root@feec:/home/projetos_p4# sudo ip link set dev veth4 up
root@feec:/home/projetos_p4# sudo ip link set dev veth5 up
root@feec:/home/projetos_p4# sudo ip link set veth4 mtu 9500
root@feec:/home/projetos_p4# sudo ip link set veth5 mtu 9500
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth4.disable_ipv6=1
net.ipv6.conf.veth4.disable_ipv6 = 1
root@feec:/home/projetos_p4# sudo sysctl net.ipv6.conf.veth5.disable_ipv6=1
net.ipv6.conf.veth5.disable_ipv6 = 1
root@feec:/home/projetos_p4#
```

Resultado da etapa 5°

6. Nesta próxima etapa, abriremos várias sessões SSH, primeira sessão iniciaremos a troca de software como um processo em segundo plano:

```
sudo simple_switch --interface 0@veth0 --interface 1@veth2 --interface 2@veth4 p1.bmv2/p1.json &
```

```
root@feec: /home/projetos_p4
root@feec:/home/projetos_p4# sudo simple_switch --interface 0@veth0 --interface 1@veth2 --interface 2@veth4 p1.bmv2/p1.json &
[1] 13428
root@feec:/home/projetos_p4# Calling target program-options parser
Adding interface veth0 as port 0
Adding interface veth2 as port 1
Adding interface veth4 as port 2
```

7. Aguarde o processamento e nesta mesma sessão SSH, inicie a Interface de linha de comando (CLI) para a troca de software, com o comando:

```
simple_switch_CLI
```

```
root@feec: /home/projetos_p4
root@feec:/home/projetos_p4# sudo simple_switch --interface 0@veth0 --interface 1@veth2 --interface 2@veth4 p1.bmv2/p1.json &
[1] 13428
root@feec:/home/projetos_p4# Calling target program-options parser
Adding interface veth0 as port 0
Adding interface veth2 as port 1
Adding interface veth4 as port 2
root@feec:/home/projetos_p4# simple_switch_CLI
Obtaining JSON from switch...
Done
Control utility for runtime P4 table manipulation
RuntimeCmd:
```

8. Na CLI, execute o comando help e configura o resultado: **help**

```
RuntimeCmd: help

Documented commands (type help <topic>):
========================================
act_prof_add_member_to_group       reset_state
act_prof_create_group              serialize_state
act_prof_create_member             set_crc16_parameters
act_prof_delete_group              set_crc32_parameters
act_prof_delete_member             set_queue_depth
act_prof_dump                      set_queue_rate
act_prof_dump_group                shell
act_prof_dump_member               show_actions
act_prof_modify_member             show_ports
act_prof_remove_member_from_group  show_pvs
counter_read                       show_tables
counter_reset                      swap_configs
counter_write                      switch_info
get_time_elapsed                   table_add
get_time_since_epoch               table_clear
help                               table_delete
load_new_config_file               table_dump
mc_dump                            table_dump_entry
mc_mgrp_create                     table_dump_entry_from_key
mc_mgrp_destroy                    table_dump_group
mc_node_associate                  table_dump_member
mc_node_create                     table_indirect_add
mc_node_destroy                    table_indirect_add_member_to_group
mc_node_dissociate                 table_indirect_add_with_group
mc_node_update                     table_indirect_create_group
mc_set_lag_membership              table_indirect_create_member
meter_array_set_rates              table_indirect_delete
meter_get_rates                    table_indirect_delete_group
meter_set_rates                    table_indirect_delete_member
mirroring_add                      table_indirect_modify_member
mirroring_add_mc                   table_indirect_remove_member_from_group
mirroring_delete                   table_indirect_reset_default
mirroring_get                      table_indirect_set_default
port_add                           table_indirect_set_default_with_group
port_remove                        table_info
pvs_add                            table_modify
pvs_clear                          table_num_entries
pvs_get                            table_reset_default
```

```
pvs_remove                          table_set_default
register_read                       table_set_timeout
register_reset                      table_show_actions
register_write                      write_config_to_file

Undocumented commands:
======================
EOF  greet

RuntimeCmd:
```

9. Execute o comando show_tables, que informa se uma my_ingress.ipv4_match consta em uma tabela:

```
show_tables
```

root@feec: /home/projetos_p4

```
RuntimeCmd: show_tables
my_ingress.ipv4_match           [implementation=None, mk=ipv4.dst_addr(lpm, 32)]
tbl_pl187                       [implementation=None, mk=]
RuntimeCmd:
```

10. Execute o comando table_info, que exibe informação de uma tabela:

```
table_info ipv4_match
```

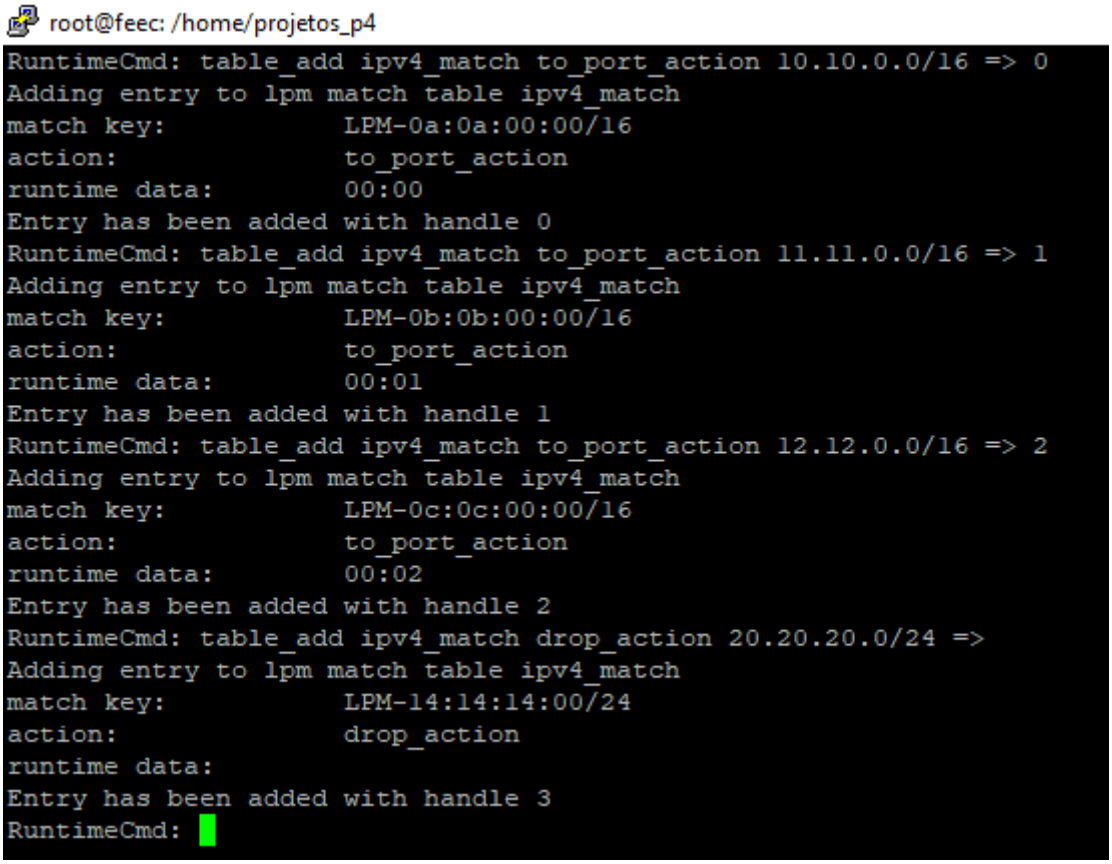
```
RuntimeCmd: table_info ipv4_match
my_ingress.ipv4_match           [implementation=None, mk=ipv4.dst_addr(lpm, 32)]
**********************************************************************************
my_ingress.drop_action          []
my_ingress.to_port_action       [port(9)]
RuntimeCmd:
```

11. Adicionando rotas à tabela de rotas no switch de software CLI, será utilizado o table_add, comando para adicionar quatro rotas à tabela de rotas:

- Todo o tráfego para o prefixo 10.10.0.0/16 é enviado para a porta 0
- Todo o tráfego para o prefixo 11.11.0.0/16 é enviado para a porta 1
- Todo o tráfego para o prefixo 12.12.0.0/16 é enviado para a porta 2
- Todo o tráfego para o prefixo 20.20.20.0/24 é descartado.
- Todo o tráfego que não corresponde a nenhum dos prefixos acima também é descartado porque a ação padrão para a tabela é descartar.

Observação: Execute os comandos LINHA a LINHA no CLI do switch:

```
table_add ipv4_match to_port_action 11.11.0.0/16 => 1
table_add ipv4_match to_port_action 12.12.0.0/16 => 2
table_add ipv4_match drop_action 20.20.20.0/24 =>
```



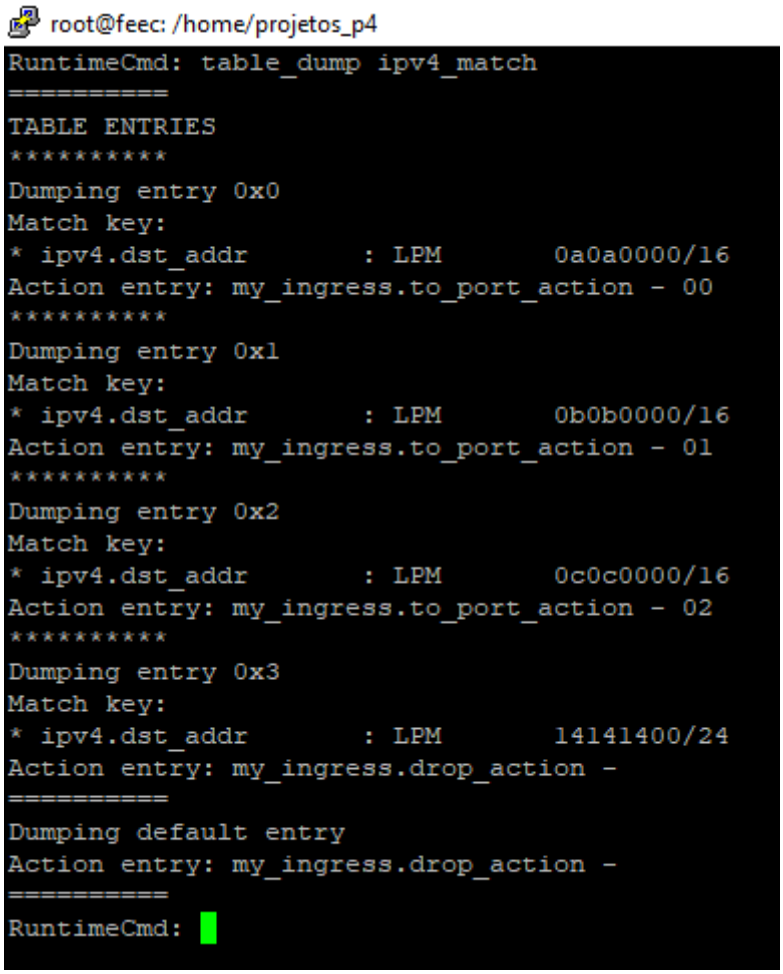
Resultado da etapa 11°

12. Executando o comando table_dump, que mostra as entradas que adicionamos à tabela:

```
table_dump ipv4_match
```

Importante:

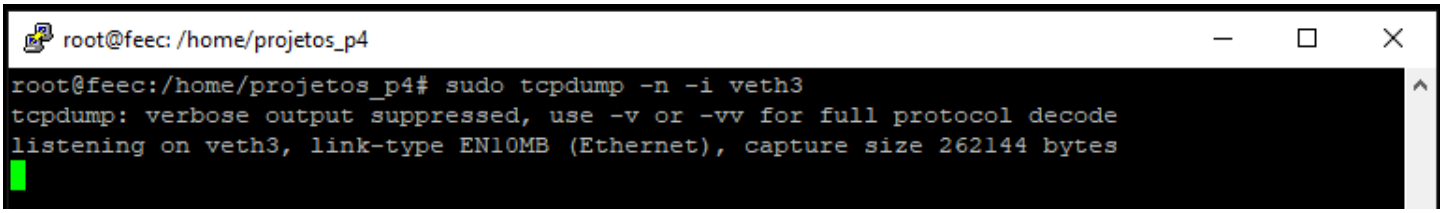
Repare que todo o procedimento está sendo feito no diretório que criamos anteriormente: /home/projetos_p4



Resultado da etapa 12°

13. Em outra sessa SSH, iremos executar o comando para descarregar pacotes que chegam na interface veth1:

```
sudo tcpdump -n -i veth3
```



Resultado da etapa 13°

14. Nesta próxima etapa iremos injetar alguns pacotes no switch de software, em outra sessão SSH separada, inicie o software scapy, que usaremos para injetar pacotes no switch de software:

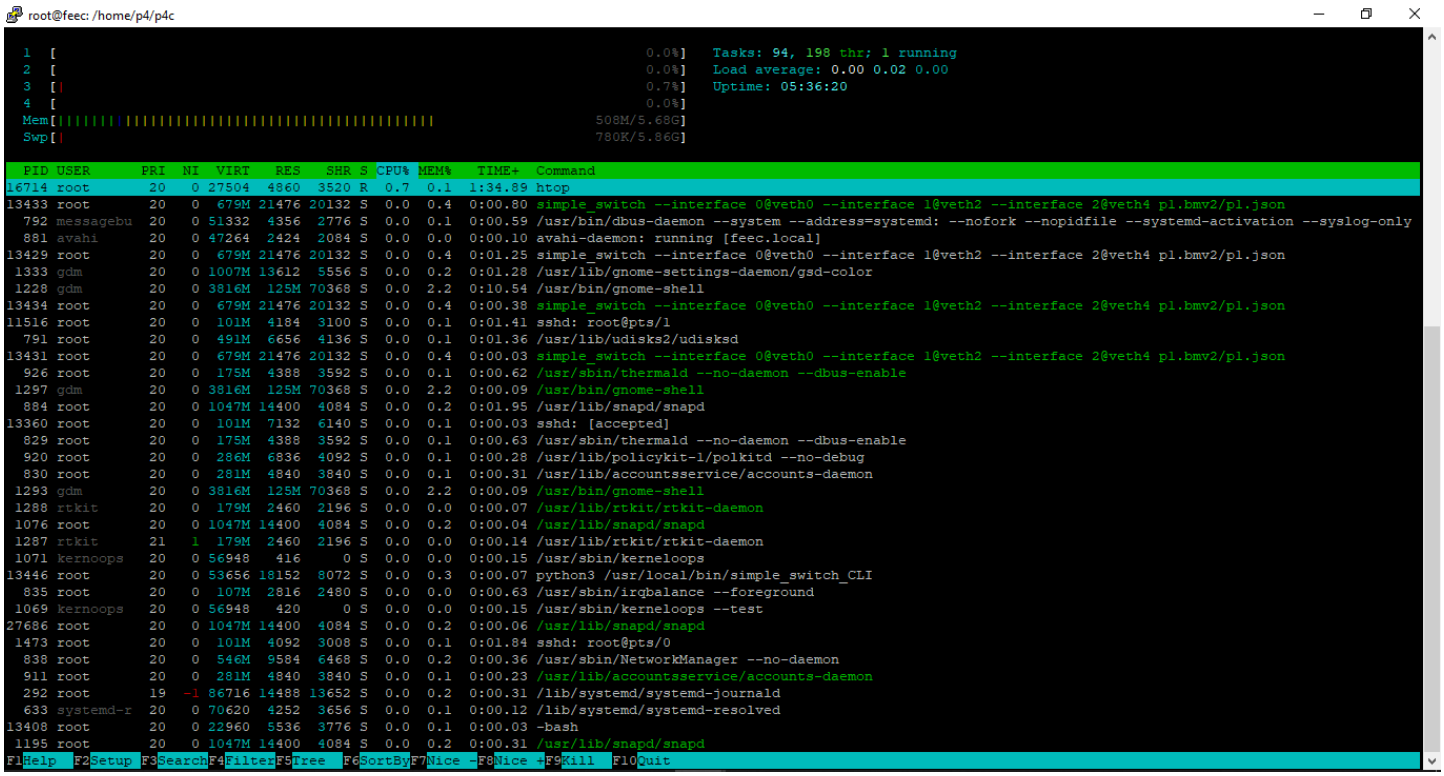
```
sudo scapy
```

```
root@feec: ~
root@feec:~# sudo scapy
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
INFO: Can't import python ecdsa lib. Disabled certificate manipulatio
n tools
Welcome to Scapy (2.3.3)
>>> p = Ether()/IP(dst="11.11.1.1")/UDP()
>>> sendp(p, iface="veth1")
.
Sent 1 packets.
>>> p = Ether()/IP(dst="12.12.1.1")/UDP()
>>> sendp(p, iface="veth1")
.
Sent 1 packets.
>>>
```

Resultado da etapa 14°

Importante:

Se chegou até a etapa **14º**, meus parabéns! Deu tudo certo. O tempo gasto para este procedimento varia muito do hardware onde o ambiente será executado, em nossa máquina física, que citamos suas configurações no início de tutorial levou mais de 5h, conforme o uptime do software htop, na imagem a seguir:



Conclusão:

O ambiente de instalação do compilador p4 precisa instalar um total de três componentes de software, respectivamente: protobuf, p4c, bmv2, sugerimos que leia as fontes originais para entender melhor o processo, problemas podem ocorrer, por isso este tutorial foi criado, devido à inúmeros problemas enfrentados durante um tutorial especifico para: protobuf, p4c e bmv2.

Autores:

Alan Teixeira da Silva: a265560@dac.unicamp.br

Fernando Henrique Santorsula: f208918@dac.unicamp.br

Fonte principal:

https://www.programmersought.com/article/9444864088/

Fontes auxiliares:

https://www.sdnlab.com/19912.html

https://www.cnblogs.com/qq952693358/p/7463204.html

https://steeven.iteye.com/blog/2330830

https://www.cnblogs.com/qq952693358/p/5905536.html

https://blog.csdn.net/zhangmeimei_pku/article/details/79479055

https://www.sdnlab.com/22466.html

https://www.sdnlab.com/22512.html

https://www.cnblogs.com/ljy1227476113/p/10524035.html

http://sunyongfeng.com/201705/networks/p4/repo_p4app.html

https://p4.org/p4/getting-started-with-p4.html