NS-3初探

目录

1
- 小试牛刀

2
- Hello Simulator

3
- 初尝ns3 script

4
- 编译和执行

PART 01

小试牛刀

小试牛刀

打开ns-3

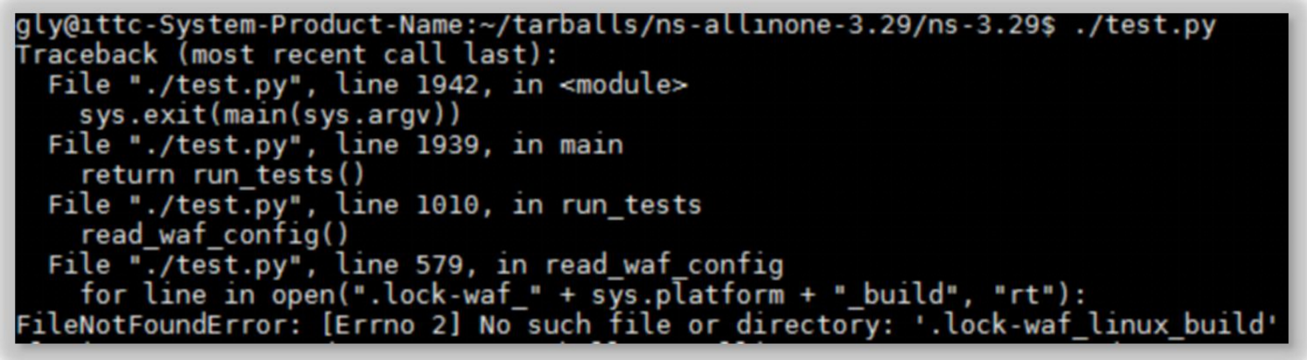
- ns3安装在root用户下
- 如何进入root用户？

- sudo su
- 输入密码user
- cd ~进入root根目录
- cd tarballs/ns...

小试牛刀

Testing ns-3

- $ ./test.py
- 没有权限，权限在linux中很重要
- 再试一下sudo ./test.py

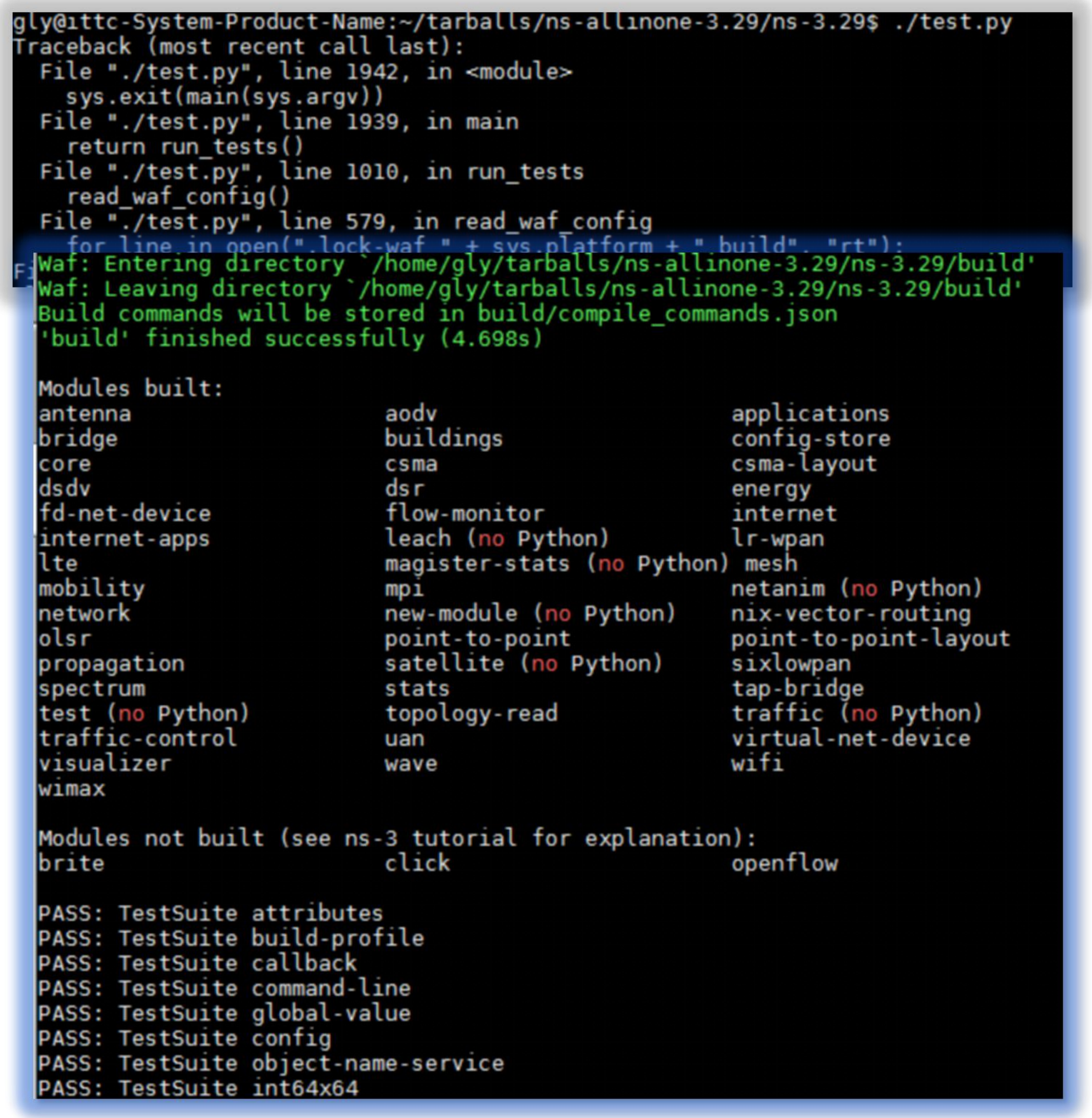


```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ ./test.py
Traceback (most recent call last):
  File "./test.py", line 1942, in <module>
    sys.exit(main(sys.argv))
  File "./test.py", line 1939, in main
    return run_tests()
  File "./test.py", line 1010, in run_tests
    read_waf_config()
  File "./test.py", line 579, in read_waf_config
    for line in open(".lock-waf_" + sys.platform + "_build", "rt"):
FileNotFoundError: [Errno 2] No such file or directory: '.lock-waf_linux_build'
```

小试牛刀

Testing ns-3

- $ ./test.py
- 没有权限，权限在linux中很重要
- 再试一下sudo ./test.py



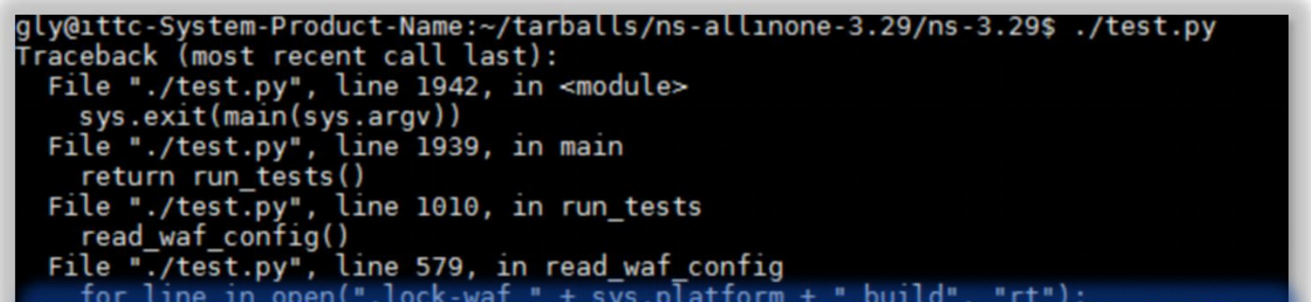
```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ ./test.py
Traceback (most recent call last):
  File "./test.py", line 1942, in <module>
    sys.exit(main(sys.argv))
  File "./test.py", line 1939, in main
    return run_tests()
  File "./test.py", line 1010, in run_tests
    read_waf_config()
  File "./test.py", line 579, in read_waf_config
    for line in open(".lock-waf_" + sys.platform + "_build", "rt"):
Fi
```

```
Waf: Entering directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
Waf: Leaving directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.698s)

Modules built:
antenna                   aodv                      applications
bridge                    buildings                 config-store
core                      csma                      csma-layout
dsdv                      dsr                       energy
fd-net-device             flow-monitor              internet
internet-apps             leach (no Python)         lr-wpan
lte                       magister-stats (no Python) mesh
mobility                  mpi                       netanim (no Python)
network                   new-module (no Python)    nix-vector-routing
olsr                      point-to-point            point-to-point-layout
propagation               satellite (no Python)     sixlowpan
spectrum                  stats                     tap-bridge
test (no Python)          topology-read             traffic (no Python)
traffic-control           uan                       virtual-net-device
visualizer                wave                      wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite                     click                     openflow

PASS: TestSuite attributes
PASS: TestSuite build-profile
PASS: TestSuite callback
PASS: TestSuite command-line
PASS: TestSuite global-value
PASS: TestSuite config
PASS: TestSuite object-name-service
PASS: TestSuite int64x64
```

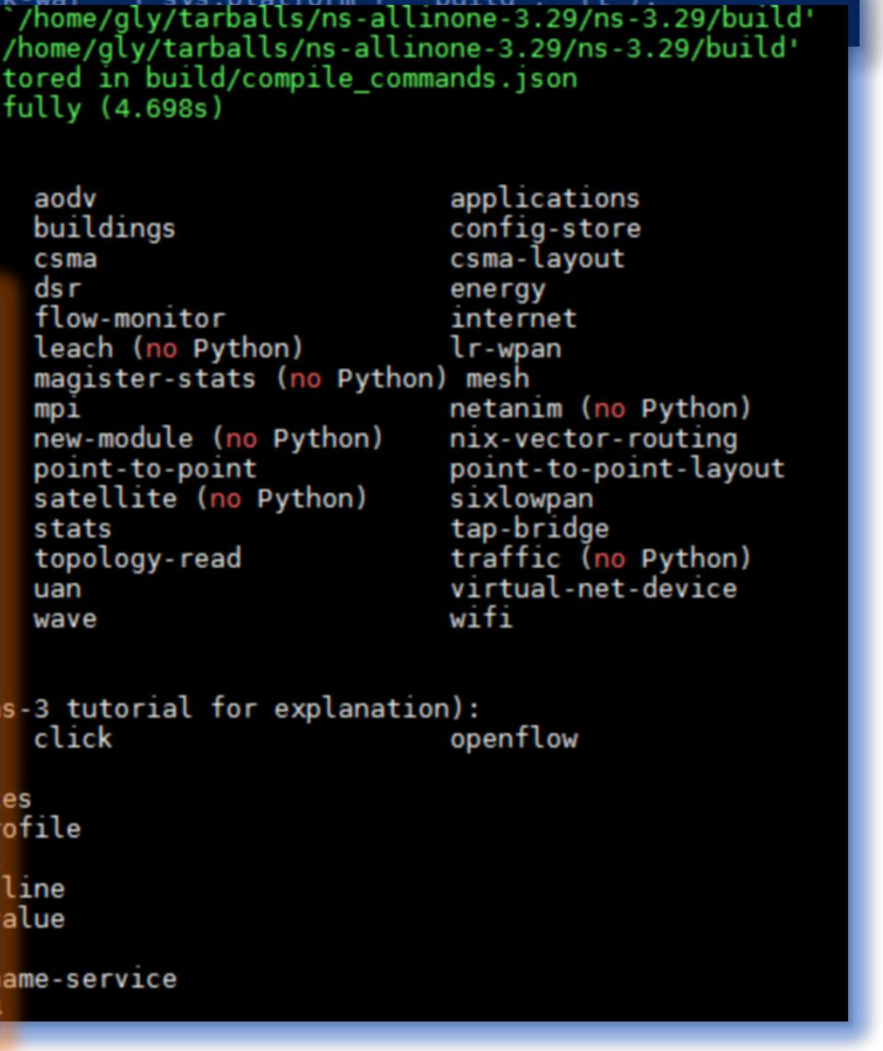
小试牛刀

Testing ns-3

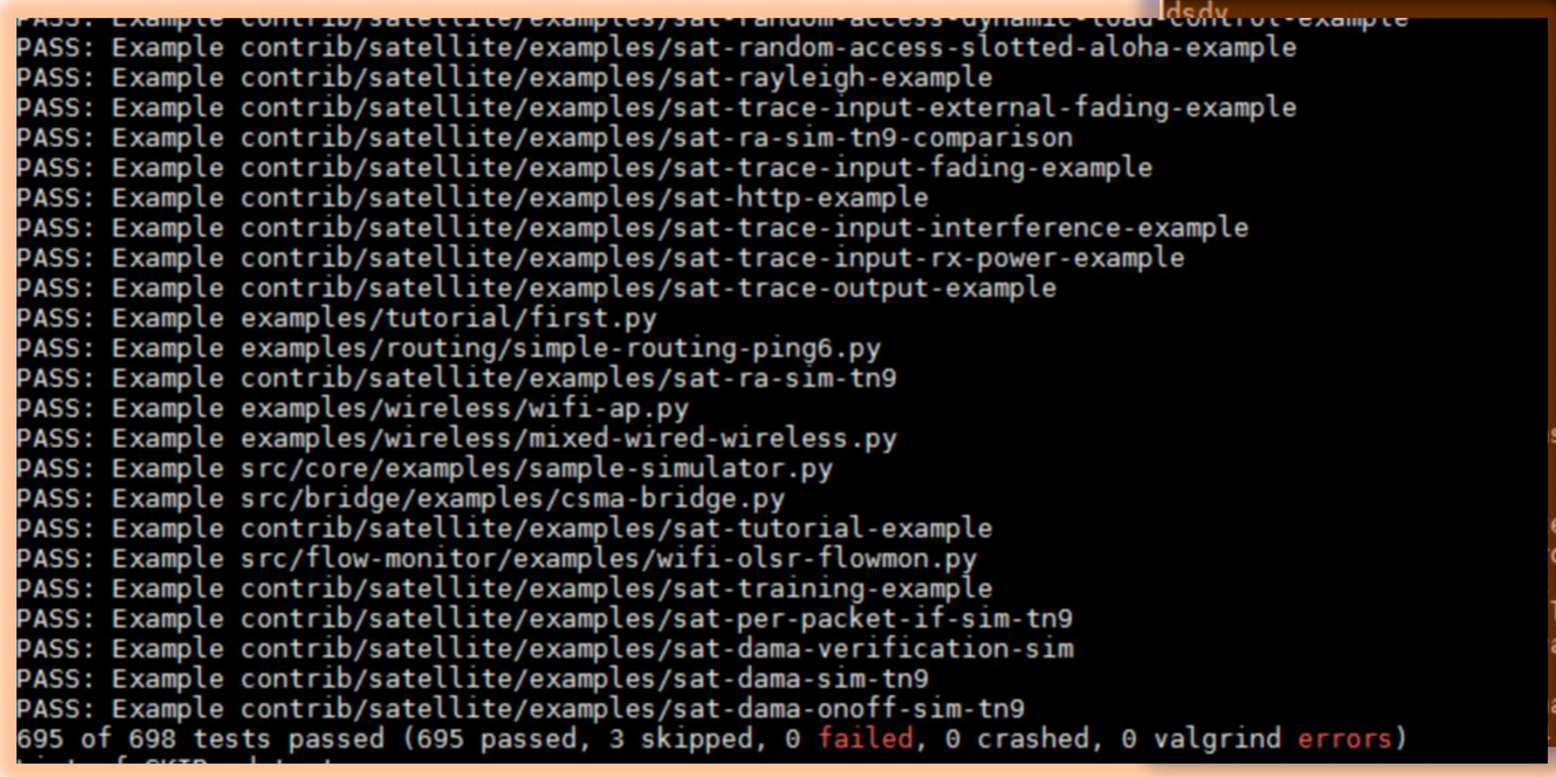
- $ ./test.py
- 没有权限，权限在linux中很重要
- 再试一下sudo ./test.py



```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ ./test.py
Traceback (most recent call last):
  File "./test.py", line 1942, in <module>
    sys.exit(main(sys.argv))
  File "./test.py", line 1939, in main
    return run_tests()
  File "./test.py", line 1010, in run_tests
    read_waf_config()
  File "./test.py", line 579, in read_waf_config
    for line in open(".lock-waf_" + sys.platform + ".build", "rt"):
Fi
```



```
Waf: Entering directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
Waf: Leaving directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.698s)

Modules built:
antenna                    aodv                       applications
bridge                     buildings                  config-store
core                       csma                       csma-layout
dsdv                       dsr                        energy
                           flow-monitor               internet
                           leach (no Python)          lr-wpan
                           magister-stats (no Python) mesh
                           mpi                        netanim (no Python)
                           new-module (no Python)     nix-vector-routing
                           point-to-point             point-to-point-layout
                           satellite (no Python)      sixlowpan
                           stats                      tap-bridge
                           topology-read              traffic (no Python)
                           uan                        virtual-net-device
                           wave                       wifi


s-3 tutorial for explanation):
                           click                      openflow


es
ofile

line
alue

ame-service
```



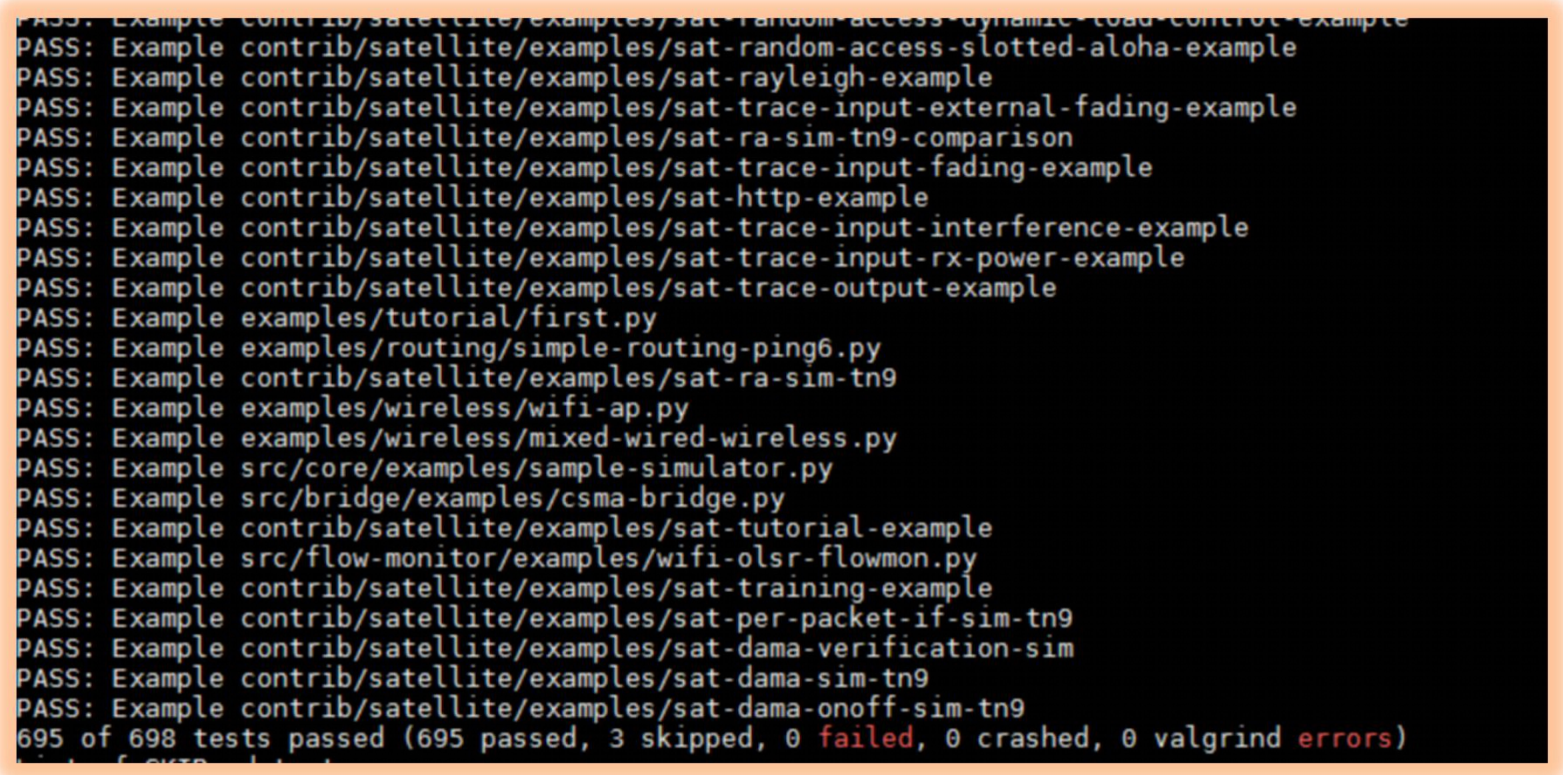
```
PASS: Example contrib/satellite/examples/sat-random-access-dynamic-load-control-example
PASS: Example contrib/satellite/examples/sat-random-access-slotted-aloha-example
PASS: Example contrib/satellite/examples/sat-rayleigh-example
PASS: Example contrib/satellite/examples/sat-trace-input-external-fading-example
PASS: Example contrib/satellite/examples/sat-ra-sim-tn9-comparison
PASS: Example contrib/satellite/examples/sat-trace-input-fading-example
PASS: Example contrib/satellite/examples/sat-http-example
PASS: Example contrib/satellite/examples/sat-trace-input-interference-example
PASS: Example contrib/satellite/examples/sat-trace-input-rx-power-example
PASS: Example contrib/satellite/examples/sat-trace-output-example
PASS: Example examples/tutorial/first.py
PASS: Example examples/routing/simple-routing-ping6.py
PASS: Example contrib/satellite/examples/sat-ra-sim-tn9
PASS: Example examples/wireless/wifi-ap.py
PASS: Example examples/wireless/mixed-wired-wireless.py
PASS: Example src/core/examples/sample-simulator.py
PASS: Example src/bridge/examples/csma-bridge.py
PASS: Example contrib/satellite/examples/sat-tutorial-example
PASS: Example src/flow-monitor/examples/wifi-olsr-flowmon.py
PASS: Example contrib/satellite/examples/sat-training-example
PASS: Example contrib/satellite/examples/sat-per-packet-if-sim-tn9
PASS: Example contrib/satellite/examples/sat-dama-verification-sim
PASS: Example contrib/satellite/examples/sat-dama-sim-tn9
PASS: Example contrib/satellite/examples/sat-dama-onoff-sim-tn9
695 of 698 tests passed (695 passed, 3 skipped, 0 failed, 0 crashed, 0 valgrind errors)
```

小试牛刀

Testing ns-3

- $ ./test.py
- 没有权限，权限在linux中很重要
- 再试一下sudo ./test.py

- 验证ns3是否正常编译
- PASS的顺序有可能改变
- 没有failed或、crashed或者errors



```
PASS: Example contrib/satellite/examples/sat-random-access-dynamic-load-control-example
PASS: Example contrib/satellite/examples/sat-random-access-slotted-aloha-example
PASS: Example contrib/satellite/examples/sat-rayleigh-example
PASS: Example contrib/satellite/examples/sat-trace-input-external-fading-example
PASS: Example contrib/satellite/examples/sat-ra-sim-tn9-comparison
PASS: Example contrib/satellite/examples/sat-trace-input-fading-example
PASS: Example contrib/satellite/examples/sat-http-example
PASS: Example contrib/satellite/examples/sat-trace-input-interference-example
PASS: Example contrib/satellite/examples/sat-trace-input-rx-power-example
PASS: Example contrib/satellite/examples/sat-trace-output-example
PASS: Example examples/tutorial/first.py
PASS: Example examples/routing/simple-routing-ping6.py
PASS: Example contrib/satellite/examples/sat-ra-sim-tn9
PASS: Example examples/wireless/wifi-ap.py
PASS: Example examples/wireless/mixed-wired-wireless.py
PASS: Example src/core/examples/sample-simulator.py
PASS: Example src/bridge/examples/csma-bridge.py
PASS: Example contrib/satellite/examples/sat-tutorial-example
PASS: Example src/flow-monitor/examples/wifi-olsr-flowmon.py
PASS: Example contrib/satellite/examples/sat-training-example
PASS: Example contrib/satellite/examples/sat-per-packet-if-sim-tn9
PASS: Example contrib/satellite/examples/sat-dama-verification-sim
PASS: Example contrib/satellite/examples/sat-dama-sim-tn9
PASS: Example contrib/satellite/examples/sat-dama-onoff-sim-tn9
695 of 698 tests passed (695 passed, 3 skipped, 0 failed, 0 crashed, 0 valgrind errors)
List of SKIPped tests:
```

PART 02

Hello Simulator

Hello Simulator

waf命令运行scripts

- $ ./waf --run hello-simulator

waf编译器

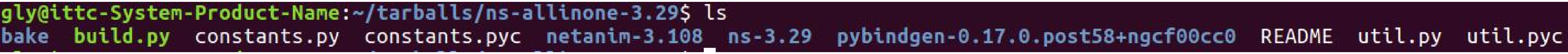
运行指令

待运行的
编译后文件名称

Hello Simulator

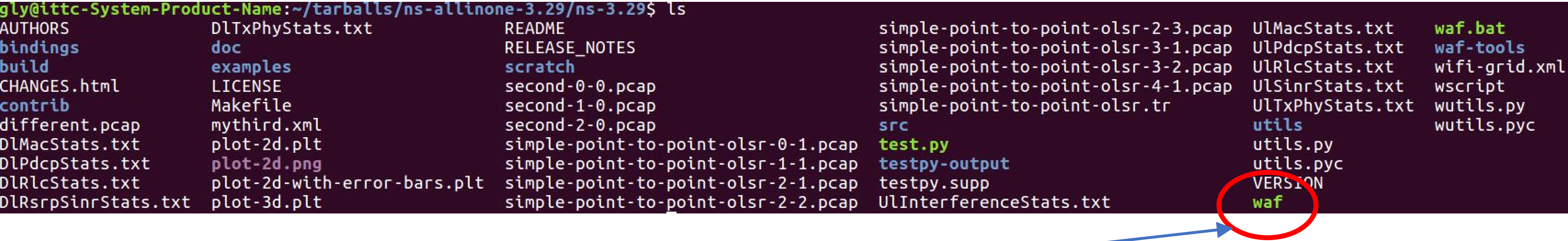
waf命令运行scripts

- $ ./waf --run hello-simulator
- 我们在**哪个文件夹运行此命令**?



```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29$ ls
bake  build.py  constants.py  constants.pyc  netanim-3.108  ns-3.29  pybindgen-0.17.0.post58+ngcf00cc0  README  util.py  util.pyc
```

```
~/tarballs/ns-allinone-3.29$ cd ns-3.29/
```



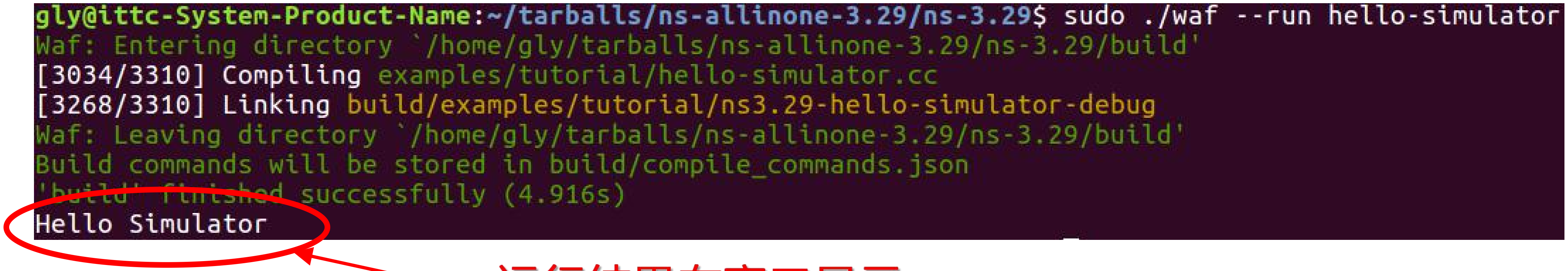
```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ ls
AUTHORS              DlTxPhyStats.txt             README                               simple-point-to-point-olsr-2-3.pcap  UlMacStats.txt    waf.bat
bindings             doc                          RELEASE_NOTES                        simple-point-to-point-olsr-3-1.pcap  UlPdcpStats.txt   waf-tools
build                examples                     scratch                              simple-point-to-point-olsr-3-2.pcap  UlRlcStats.txt    wifi-grid.xml
CHANGES.html         LICENSE                      second-0-0.pcap                      simple-point-to-point-olsr-4-1.pcap  UlSinrStats.txt   wscript
contrib              Makefile                     second-1-0.pcap                      simple-point-to-point-olsr.tr        UlTxPhyStats.txt  wutils.py
different.pcap       mythird.xml                  second-2-0.pcap                      src                                  utils             wutils.pyc
DlMacStats.txt       plot-2d.plt                  simple-point-to-point-olsr-0-1.pcap  test.py                              utils.py
DlPdcpStats.txt      plot-2d.png                  simple-point-to-point-olsr-1-1.pcap  testpy-output                        utils.pyc
DlRlcStats.txt       plot-2d-with-error-bars.plt  simple-point-to-point-olsr-2-1.pcap  testpy.supp                          VERSION
DlRsrpSinrStats.txt  plot-3d.plt                  simple-point-to-point-olsr-2-2.pcap  UlInterferenceStats.txt              waf
```

- 是否可以运行，要看是否有这个**执行程序**

Hello Simulator

waf命令运行scripts

- $ ./waf --run hello-simulator



```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ sudo ./waf --run hello-simulator
Waf: Entering directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
[3034/3310] Compiling examples/tutorial/hello-simulator.cc
[3268/3310] Linking build/examples/tutorial/ns3.29-hello-simulator-debug
Waf: Leaving directory `/home/gly/tarballs/ns-allinone-3.29/ns-3.29/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.916s)
Hello Simulator
```

- 运行结果在窗口显示

- Hello Simulator文件的位置
- $ cd examples/tutorial/
- $ ls

Hello Simulator

waf命令运行scripts

- $ sudo vim hello-simulator.cc

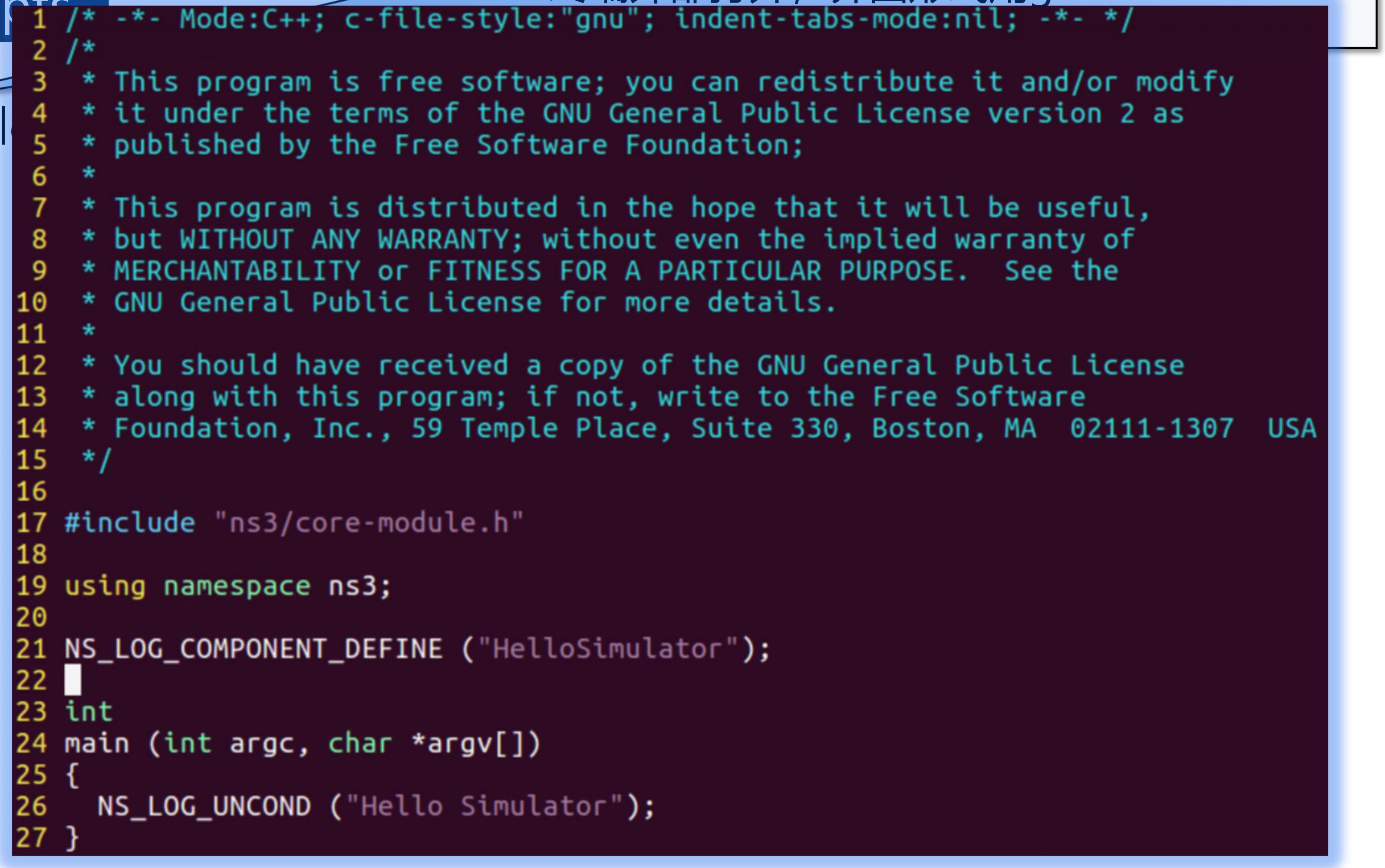
- vim是linux下的文本编辑器，终端直接打开
- 终端外部打开，界面形式用gvim

Hello Simulator

waf命令运行scripts

- vim是linux下的文本编辑器，终端直接打开
- 终端外部打开，界面形式用gvim

- $ sudo vim hell

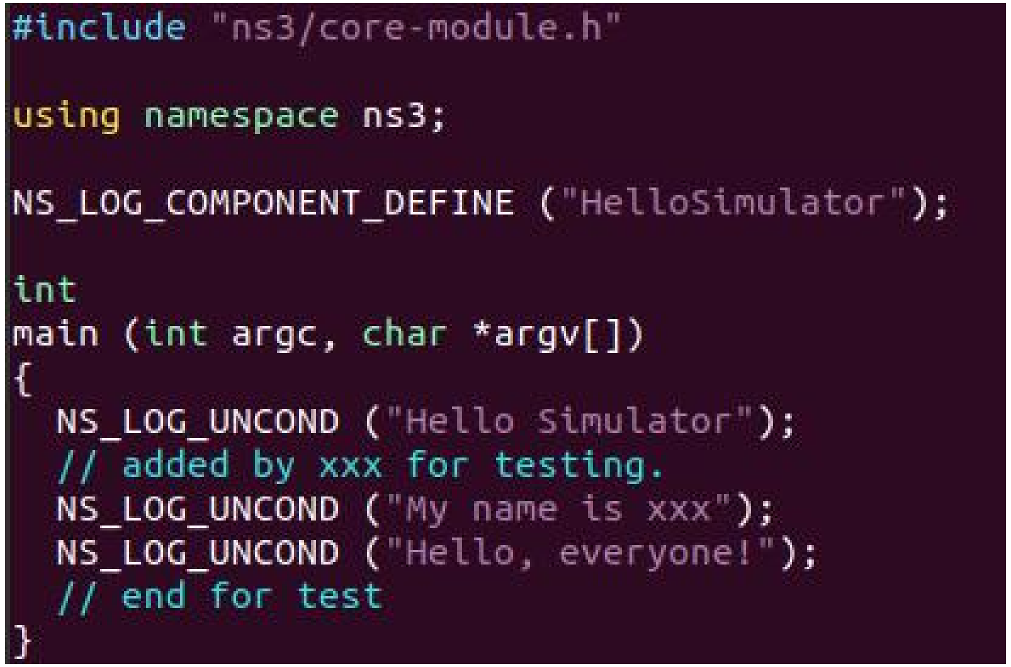


```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */

#include "ns3/core-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("HelloSimulator");

int
main (int argc, char *argv[])
{
  NS_LOG_UNCOND ("Hello Simulator");
}
```

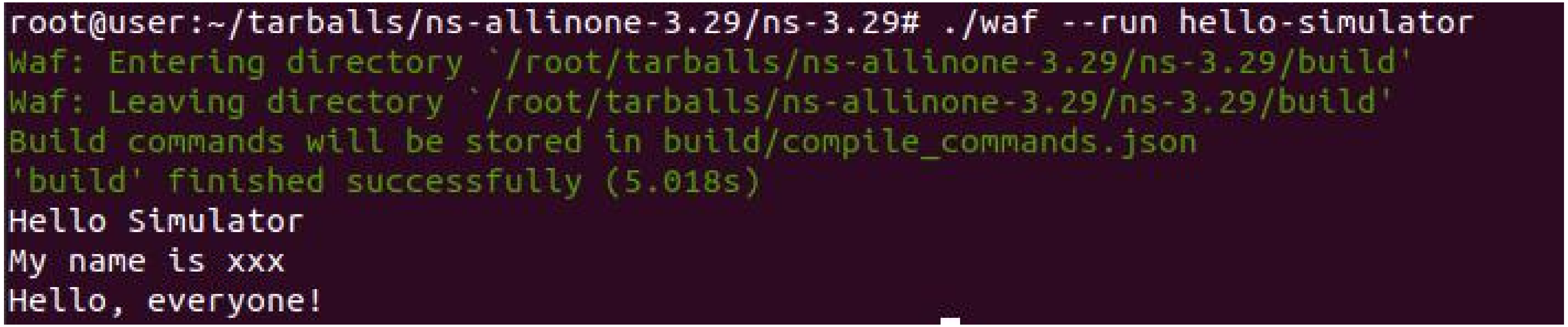
Hello Simulator

waf命令运行scripts

- $ sudo vim hello-simulator.cc
- 回到ns目录
- $ sudo ./waf
- $ sudo ./waf --run hello-simulator



```
#include "ns3/core-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("HelloSimulator");

int
main (int argc, char *argv[])
{
  NS_LOG_UNCOND ("Hello Simulator");
  // added by xxx for testing.
  NS_LOG_UNCOND ("My name is xxx");
  NS_LOG_UNCOND ("Hello, everyone!");
  // end for test
}
```



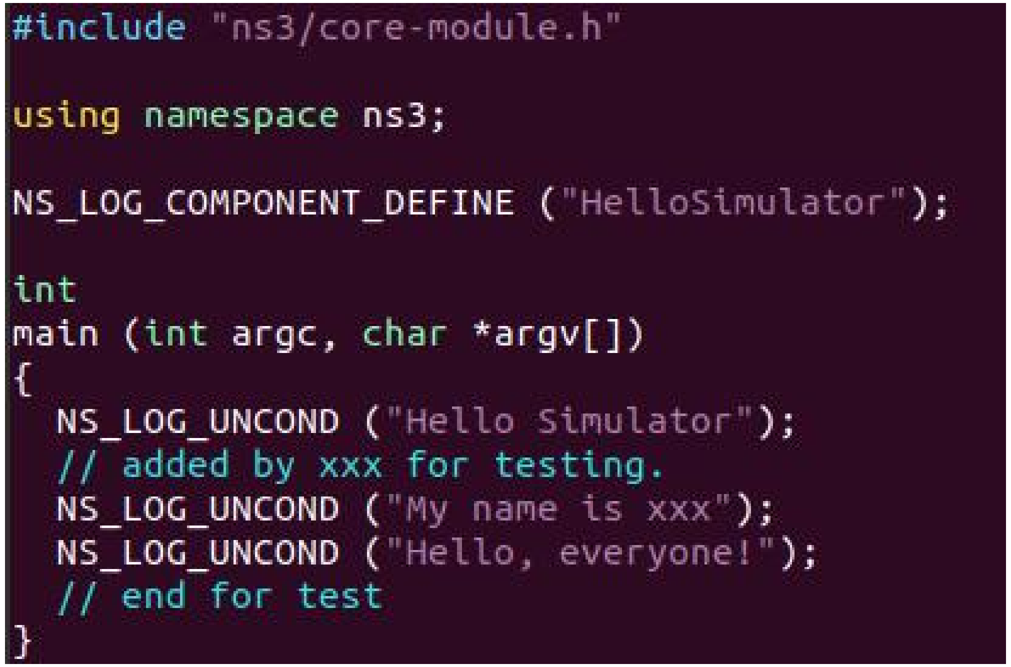
```
root@user:~/tarballs/ns-allinone-3.29/ns-3.29# ./waf --run hello-simulator
Waf: Entering directory `/root/tarballs/ns-allinone-3.29/ns-3.29/build'
Waf: Leaving directory `/root/tarballs/ns-allinone-3.29/ns-3.29/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.018s)
Hello Simulator
My name is xxx
Hello, everyone!
```

- 尝试你自己的**第一个文本显示!**

Hello Simulator

waf命令运行scripts

- $ sudo vim hello-simulator.cc
- 回到ns目录
- $ sudo ./waf
- $ sudo ./waf --run hello-simulator
- **若不显示输出**，需要设置
- $ ./waf configure --build-profile=debug --enable-examples --enable-tests
- 设置Waf调试包括exampls和tests



```
#include "ns3/core-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("HelloSimulator");

int
main (int argc, char *argv[])
{
  NS_LOG_UNCOND ("Hello Simulator");
  // added by xxx for testing.
  NS_LOG_UNCOND ("My name is xxx");
  NS_LOG_UNCOND ("Hello, everyone!");
  // end for test
}
```

PART 03

初尝ns3 script

初尝ns3 script

节点

- NS3是网络仿真器
- NS3中基本装载协议的容器：节点
    - ➢ 对应C++类 Node

应用

- 计算机软件分为两大类
    - ➢ 系统软件：用于管理内存、进程、硬盘、网络等
    - ➢ 应用程序：用户运行调用计算机资源，执行。可理解为业务模型
    - ➢ 对应C++类 Application

初尝ns3 script

信道

- 数据传输的管道，分为有线和无线两种
  - ➢ 对应C++类 Channel

网络设备

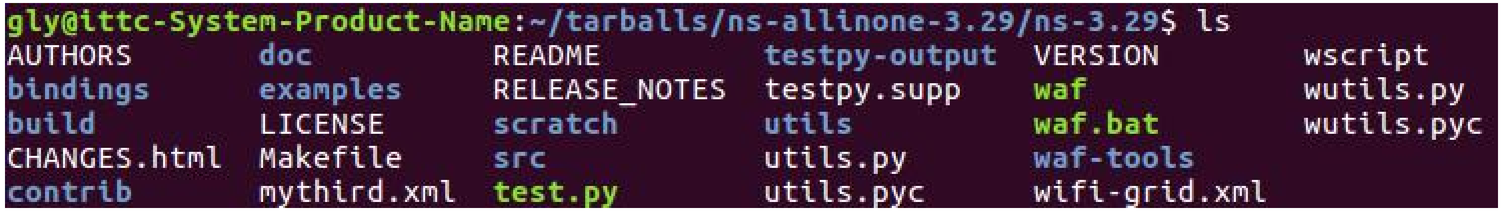
- 模拟设备联网所需的设备，例如网卡
- 用于管理Node和Channel的连接
  - ➢ 对应C++类 NetDevices

拓扑

- 大规模网络仿真，需要分配多个Node、Channel、NetDevices间的连接
- 网络拓扑帮助便于管理NetDevices连接到Node、 NetDevices连接到Channel、分配IP地址、协议栈等

初尝ns3 script

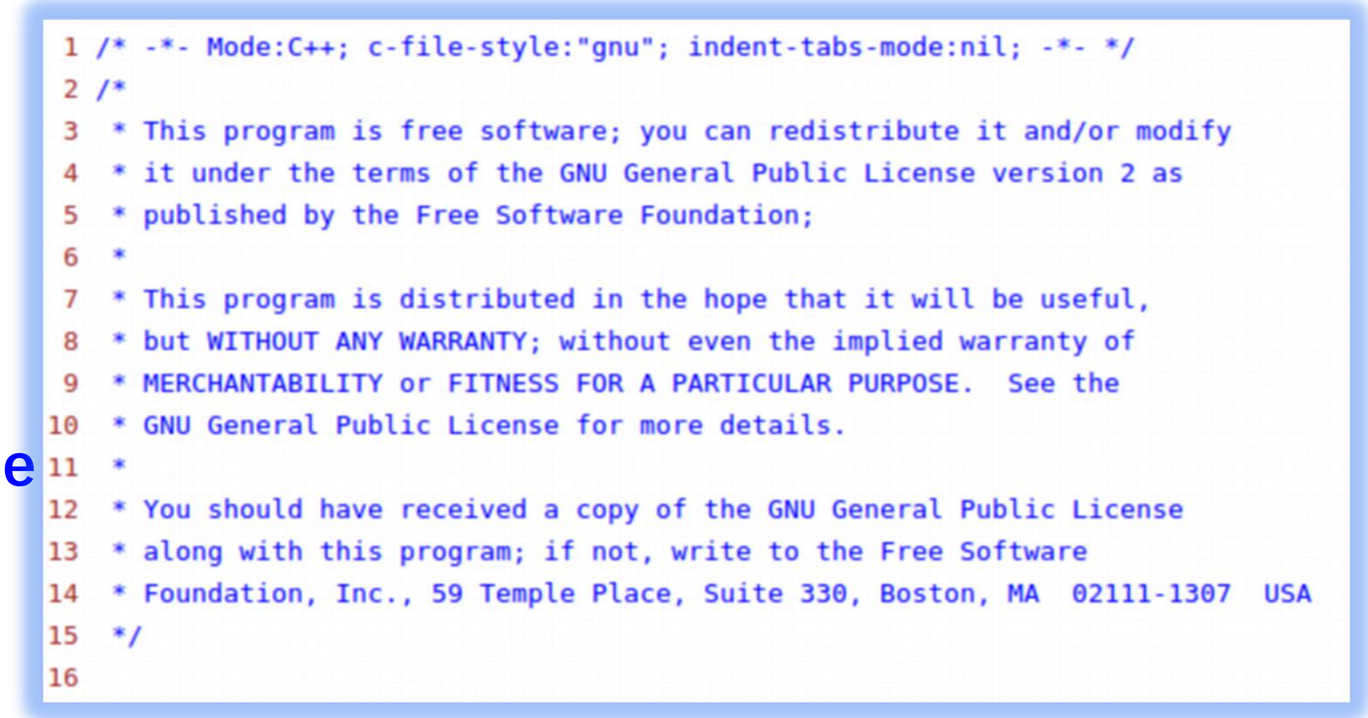
进入ns目录下



```
gly@ittc-System-Product-Name:~/tarballs/ns-allinone-3.29/ns-3.29$ ls
AUTHORS       doc          README         testpy-output  VERSION        wscript
bindings      examples     RELEASE_NOTES  testpy.supp    waf            wutils.py
build         LICENSE      scratch        utils          waf.bat        wutils.pyc
CHANGES.html  Makefile     src            utils.py       waf-tools
contrib       mythird.xml  test.py        utils.pyc      wifi-grid.xml
```

打开first.cc

- $ cd examples/tutorials
- $ vim first.cc
- **GNU General Public License**
- Copyright
- author list



```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */

```

初尝ns3 script

打开first.cc

- 头文件调用模块

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
```

- 这些头文件在哪里？ns3似乎是个文件夹？
- $ locate core-module.h

初尝ns3 script

打开first.cc

- ns3命名空间

```
using namespace ns3;
```

- 日志：用来监控、调试仿真程序
- 日志是快速获得脚本和模型的调试信息、警告信息、错误信息或者其他信息的首选。

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

//声明了一个叫FirstScriptExample的日志组件，通过引用FirstScriptExample这个名字的操作，可以实现打开或者关闭控制台日志的输出。

初尝ns3 script

Main函数

Time::SetResolution (Time::NS);

- 时间分辨率设定为1纳秒（系统默认）

- 日志输出设定

```
LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

- 应用包发送和接收的消息显示

初尝ns3 script

拓扑辅助器

```
NodeContainer nodes;
nodes.Create (2);
```

- **ns3::NodeContainer**类，创建节点2个

- 设定设备属性——数据速率 5Mbps

```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

- 点对点连接设定，何为点对点?

- 设定信道属性——时延 2ms

- 这是什么时延？为何有时延?

初尝ns3 script

拓扑辅助器

```
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);
```

- 存放需要所有被创建的**NetDevice**对象，就像我们使用一个**NodeContainer**对象来存放我们所创建节点

```
InternetStackHelper stack;
stack.Install (nodes);
```

- 装载**internet**协议栈

```
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

- 定义ip地址
- 定义基地址、子网掩码
- 设备分配地址

初尝ns3 script

应用

- 服务器端

```
UdpEchoServerHelper echoServer (9);
//建立端口为9的UDP Server
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));

serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
//仿真start和stop时间设置
```

初尝ns3 script

应用

- 客户端

```
UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
//连接Node1地址，连接端口为9
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
```

初尝ns3 script

仿真器

```
Simulator::Run (); //运行仿真

serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
...
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Simulator::Destroy (); //清理所有
return 0;
}
```

PART 04

编译和执行

编译和执行

Building

```
$ cd ns目录
$ cp examples/tutorial/first.cc scratch/myfirst.cc
```

- 存放拷贝到**scratch**目录下，系统将自动编译这个目录的文件

```
$ ./waf
```

- waf编译

```
$ ./waf --run scratch/myfirst
```

- 是否能编译成功？若失败了是因为什么？

- 自由尝试吧！

THANKS FOR WATCHING