

第4章 ns3入门：操作演示

1. 下载ns3源码

ns-3下载:

- <https://www.nsnam.org/releases/>
 - ns-3.45.tar.bz2
 - ns-allinone-3.45.tar.bz2
 - `tar jxf ns-3.45.tar.bz2`
- git
 - git clone <https://gitlab.com/nsnam/ns-3-dev.git>

2. 编译ns3

满足先决条件:

- ```
$ sudo apt install build-essential cmake git vim
```

```
$ sudo apt install libboost-system-dev libboost-thread-dev
```

编译ns-3(使用 ns3)

```
$./ns3 clean # 删除`build/`目录中先前编译的库和对象文件
$./ns3 configure --build-profile=debug --enable-examples --enable-tests
$./ns3 build # 编译ns-3
$./ns3 show profile
$./ns3 show config
```

## 3. 测试构建的ns3

```
$ cd ns-allinone-3.45/ns-3.45/
$./test.py # 在运行测试之前，先检查是否需要构建（编译）ns-3
$./test.py --no-build # 跳过构建，仅运行测试
```

## 4. 运行网络模拟程序

```
$./ns3 run first
$./ns3 run "first --PrintHelp"
$./ns3 run first --command-template="%s --PrintHelp"
$./ns3 run first --command-template="gdb %s"
```

## 5. 使用ninja构建系统编译ns3

---

```
$ sudo apt install build-essential cmake git vim ninja-build
$./ns3 clean --dry-run
$./ns3 clean
$./ns3 configure --build-profile=debug --enable-examples --enable-tests
$ ls cmake-cache
$./ns3 build
```

## 6. 使用ccache编译ns3

---

关于ccache:

- ccache是单独的命令，不从属于ns-3。**ccache (compiler cache, 编译器缓存)**是一个开源的 **编译加速工具**，主要用于 **C/C++** 编译过程，通过 **缓存编译结果** 来 **减少重复编译的时间**，从而显著加快 **增量编译 (如修改少量代码后重新编译)** 的速度。
- 如果系统中安装了ccache，ns-3在编译时会自动使用ccache来加快编译。ccache缓存的文件位于ns-3目录之外（通常位于`~/.cache/ccache`的隐藏目录中），并在多个项目之间共享。
- 定期清理此缓存可能有助于回收磁盘空间。

```
$ sudo apt install build-essential cmake git vim ninja-build ccache
$./ns3 clean
$./ns3 configure --build-profile=debug --enable-examples --enable-tests
$./ns3 build
$ ccache -s
$ ccache -C
$ ls ~/.cache/ccache
```

## 7. 使用VS Code编译ns3

---