

Termux

入门指南

```
#include <stdio.h>
#include <android/log.h>

int main() {
    char const* text = "hello, w
    printf("%s", text);

    int prio = ANDROID_LOG_INFO;
    char const* tag = "hello_tag

    __android_log_print(prio, ta
}
~
~
~

$ gcc -llog hi.c -o hi
$ ./hi
hello, world
$ strace ./hi 2>&1 >/dev/null | grep
write(1, "hello, world\n", 13)
$ logcat -s hello_tag:*
----- beginning of crash
----- beginning of system
----- beginning of main
I/hello_tag(18229): hello, world
I/hello_tag(18237): hello, world
```



国光

www.sqlsec.com

[0] 0: bash* "localhost" 03:26

版权声明

17 年开始接触到 Termux，就发现它有很多值得挖掘的潜力，于是抽空在 18 年的某一个法定的整整花了三天假期开才写完第一版文章，然后文章陆陆续续更新到现在，期间有一次误操作不小心把博客所有的评论都删了，否则这篇文章的评论数会更多。现在本文的截图数量达到了150张左右了，文字数目已经数万多了。自己花了很长时间写出来的原创文章，抄袭白嫖党直接Ctrl+C Ctrl+V只要几秒钟。原创很辛苦，抄袭的成本却很低，维权的成本又很高，虽然国内目前的抄袭风气很严重，但是我相信尊重原创，保护原创从现在做起从大家做起，tomorrow is another day! 如果大面转载引用的话 希望标明文章出处：

Termux 高级终端安装使用配置教程

<https://www.sqlsec.com/2018/05/termux.html>

学习资源

考虑到手机用户体验和离线观看教程的需求，国光打包了几种风格的 PDF 版本，并且已经插入好目录，阅读体验会比较友好。

黑色背景的 PDF : [Termux入门指南 \(Vue 黑\)](#)

白色背景的 PDF : [Termux入门指南 \(Github 白\)](#)

macOS light风格 : [Termux入门指南 \(macOS 白\)](#)

Gothic风格 : [Termux入门指南 \(简约线条\)](#)

早期我的信息安全交流群里面陆陆续续加了很多 Termux 玩家，然而那是一个信息安全交流群，Termux 的提问经常没有人回答，所以后来我就把博客所有的加群链接给去了。现在国光我单独建立了 1 个 Termux 群，加群链接藏在本文当中，是一个彩蛋，缘妙不可言，随缘入群吧。好了话不多说，教程开始了，希望本文可以帮助到大家。

Termux 简介

文档相关

- [Termux 官网](#)
- [Github 项目地址](#)
- [官方英文 Wiki 文档](#)

下载地址

- [Google Play 下载地址](#)
- [F-Droid 下载地址](#)
- [酷安 下载地址](#)

Google Play 下载的版本比酷安要新，而且插件这块安装也很方便，有能力的建议首先考虑下载 Google Play 版本的，然后考虑 F-Droid 版本，最后再考虑可怜兮兮的酷安版本。

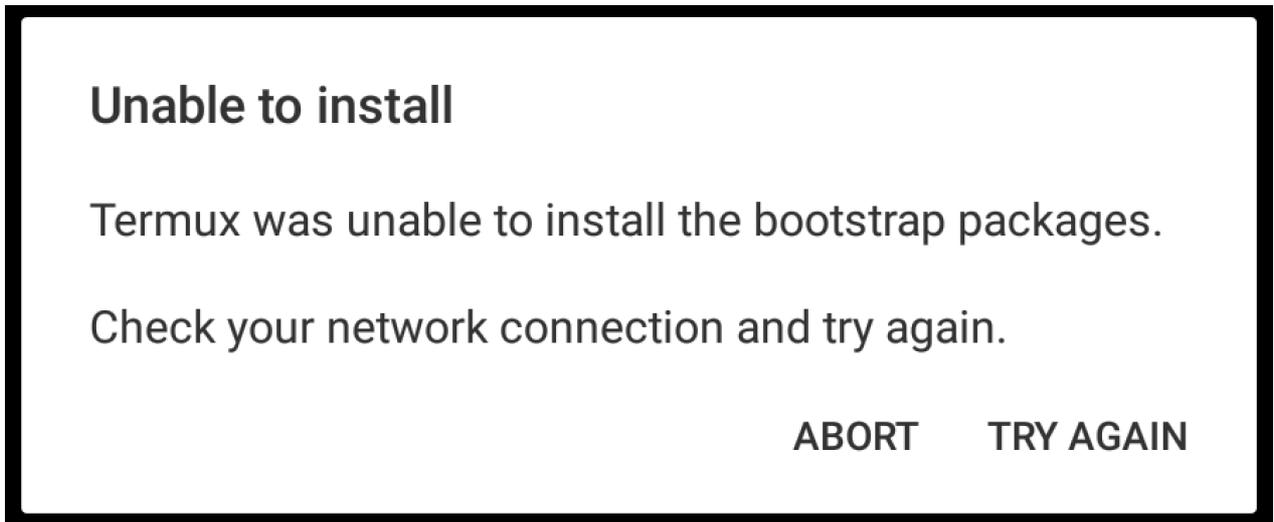
Termux 是一个 Android 下一个高级的终端模拟器,开源且不需要 root, 支持 apt 管理软件包, 十分方便安装软件包, 完美支持 Python、PHP、Ruby、Nodejs、MySQL等。随着智能设备的普及和性能的不不断提升, 如今的手机、平板等的硬件标准已达到了初级桌面计算机的硬件标准, 用心去打造 DIY 的话完全可以把手机变成一个强大的极客工具。

初始化

第一次启动Termux的时候需要从远程服务器加载数据, 然而可能会遇到这种问题:

```
Uable to install
Termux was unable to install the bootstrap packages.
Check your network connection and try again.
```

这里的Termux官方远程的服务器地址是: <http://termux.net/bootstrap/>



目前解决方法有两种:

1. VPN 全局代理 (成功率很高)
2. 如果你是 WiFi 的话尝试切换到运营商流量 (有一定成功率)

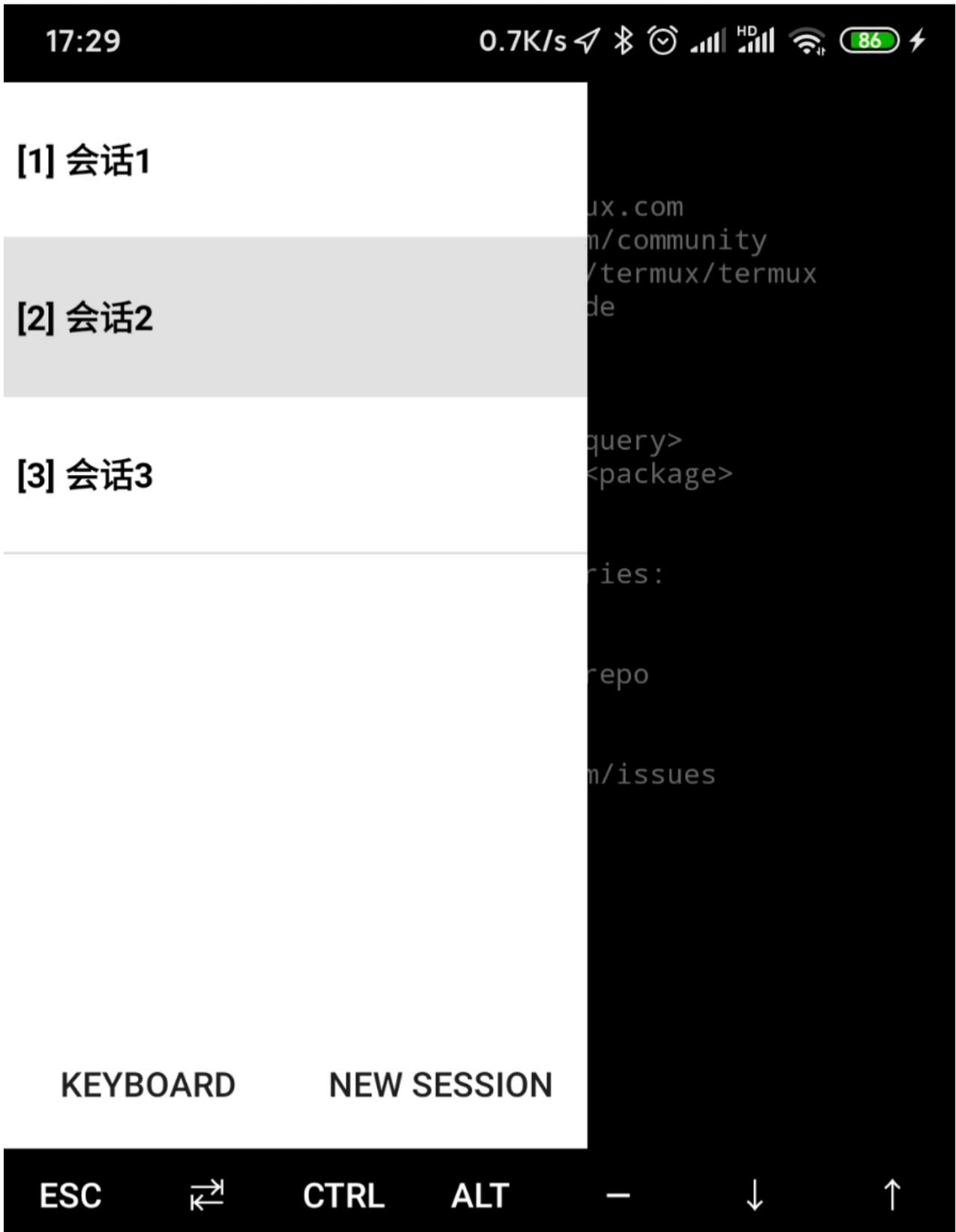
基本操作

基本操作还是要学习一下的, 可以事半功倍。

缩放文本

可以使用缩放手势来调整其字体大小。对就是 双指放大缩小照片那样操作。

显示隐藏式导航栏，可以新建、切换、重命名会话session和调用弹出输入法

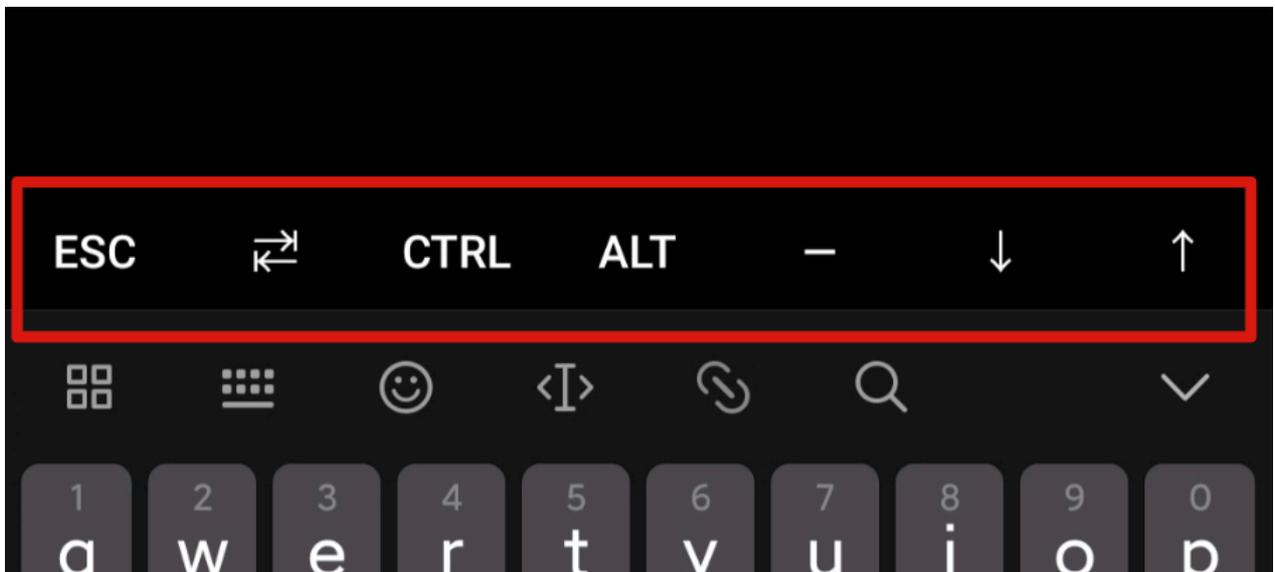


同时在Android的通知栏中也可以看到当前Termux运行的会话数：



常用按键

常用键是PC端常用的按键如: ESC键、Tab键、CTR键、ALT键, 有了这些按键后可以提高我们日常操作的效率, 所以Termux后面的版本默认都是显示这个扩展功能按键的。(18年的时候默认是不显示的)



打开和隐藏这个扩展功能按键目前有下面两种方法:

方法一

从左向右滑动,显示隐藏式导航栏,长按左下角的 `KEYBOARD`

方法二

使用 Termux 快捷键: 音量+ + Q 键 或者 音量+ + K 键

当然这个常用按键在 Termux 后面的版本也支持自定义的, 详情见本文的「进阶配置」-「定制常用按键」这一小节。

基础知识

这些基础知识简单了解一下就可以了, Linux 用的多了 就会慢慢熟悉理解了。

快捷键表

`Ctrl` 键是终端用户常用的按键，但大多数触摸键盘都没有这个按键，因此 Termux 使用 音量减小按钮 来模拟 `Ctrl` 键。例如，在触摸键盘上按 音量减小 + `L` 就相当于是在键盘上按 `Ctrl + L` 的效果一样，达到清屏的效果。

- `Ctrl + A` -> 将光标移动到行首
- `Ctrl + C` -> 中止当前进程
- `Ctrl + D` -> 注销终端会话
- `Ctrl + E` -> 将光标移动到行尾
- `Ctrl + K` -> 从光标删除到行尾
- `Ctrl + U` -> 从光标删除到行首
- `Ctrl + L` -> 清除终端
- `Ctrl + Z` -> 挂起（发送SIGTSTP到）当前进程
- `Ctrl + alt + C` -> 打开新会话（仅适用于 黑客键盘）

音量加键 也可以作为产生特定输入的特殊键。

- 音量加 + `E` -> Esc键
- 音量加 + `T` -> Tab键
- 音量加 + `1` -> F1（音量增加 + `2` → F2...以此类推）
- 音量加 + `0` -> F10
- 音量加 + `B` -> Alt + B，使用readline时返回一个单词
- 音量加 + `F` -> Alt + F，使用readline时转发一个单词
- 音量加 + `X` -> Alt+X
- 音量加 + `W` -> 向上箭头键
- 音量加 + `A` -> 向左箭头键
- 音量加 + `S` -> 向下箭头键
- 音量加 + `D` -> 向右箭头键
- 音量加 + `L` -> |（管道字符）
- 音量加 + `H` -> ~（波浪号字符）
- 音量加 + `U` -> _（下划线字符）
- 音量加 + `P` -> 上一页
- 音量加 + `N` -> 下一页
- 音量加 + `.` -> Ctrl + \（SIGQUIT）
- 音量加 + `V` -> 显示音量控制
- 音量加 + `Q` -> 切换显示的功能键视
- 音量加 + `K` -> 切换显示的功能键视图

快捷键用的熟悉的话也可以极大提高操作的效率。

基本命令

Termux 除了支持 `apt` 命令外，还在此基础上封装了 `pkg` 命令，`pkg` 命令向下兼容 `apt` 命令。`apt` 命令大家应该都比较熟悉了，这里直接简单的介绍下 `pkg` 命令：

```
pkg search <query>           # 搜索包
pkg install <package>       # 安装包
pkg uninstall <package>     # 卸载包
pkg reinstall <package>     # 重新安装包
pkg update                   # 更新源
pkg upgrade                  # 升级软件包
pkg list-all                # 列出可供安装的所有包
pkg list-installed          # 列出已经安装的包
pkg show <package>          # 显示某个包的详细信息
pkg files <package>         # 显示某个包的相关文件夹路径
```

国光建议大家使用 `pkg` 命令，因为 `pkg` 命令每次安装的时候自动执行 `apt update` 命令，很是方便。

软件安装

除了通过上述的 `pkg` 命令安装软件以外，如果我们有 `.deb` 软件包文件，也可以使用 `dpkg` 进行安装。

```
dpkg -i ./package.de        # 安装 deb 包
dpkg --remove [package name] # 卸载软件包
dpkg -l                      # 查看已安装的包
man dpkg                     # 查看详细文档
```

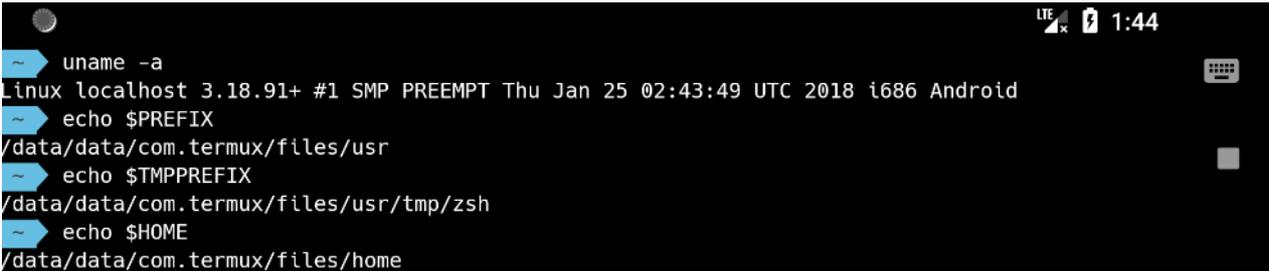
目录结构

```
echo $HOME
/data/data/com.termux/files/home

echo $PREFIX
/data/data/com.termux/files/usr

echo $TMPPREFIX
/data/data/com.termux/files/usr/tmp/zsh
```

长期使用 Linux 的朋友可能会发现，这个 HOME 路径看上去和我们电脑端的不太一样，这是为了方便 Termux 提供的特殊的环境变量。



```
~> uname -a
Linux localhost 3.18.91+ #1 SMP PREEMPT Thu Jan 25 02:43:49 UTC 2018 i686 Android
~> echo $PREFIX
/data/data/com.termux/files/usr
~> echo $TMPPREFIX
/data/data/com.termux/files/usr/tmp/zsh
~> echo $HOME
/data/data/com.termux/files/home
```

端口查看

Android 10 以下版本

Android 10 以下的版本是可以正常使用 netstat 命令的，这样可以方便的查看端口开放信息

```
# 查看所有端口
netstat -an

# 查看3306端口的开放情况
netstat -an|grep 3306
```

```
~ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.10.71          *.*.*.*.*              TIME_WAIT
IT
tcp        0      0 192.168.10.*          *.*.*.*.*              TIME_WAIT
IT
tcp        0      0 192.168.10.*          *.*.*.*.*              ESTABLISHED
SHED
tcp        0      0 192.168.10.*          *.*.*.*.*              TIME_WAIT
IT
tcp        0      0 192.168.10.*          *.*.*.*.*              ESTABLISHED
SHED
```

Android 10 版本

Android 10 版本的Termux 下无法正常使用 netstat -an 命令，国光的解决方法是安装一个 nmap，然后扫描本地端口（弯道超车）：

```
# 安装nmap端口扫描神器
pkg install nmap

# 扫描本地端口
nmap 127.0.0.1
```

使用 nmap 操作 纯属无奈之举，但是又不是不能用（源于：罗永浩名言 :-))

```
~ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags      Type           State           I-Node  Path
~ nmap 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-20 10:36 CST
Nmap scan report for 127.0.0.1
Host is up (0.0024s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
3306/tcp  open  mysql
8022/tcp  open  oa-system
8080/tcp  open  http-proxy
9000/tcp  open  cslistener

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
```

进阶配置

要想使用体验好，进阶配置少不了。（单押）

更换国内源

使用 `pkg update` 更新一下的时候发现默认的官方源网速有点慢，在这个喧嚣浮躁的时代，我们难以静下心来等待，这个时候就得更换成国内的 `Termux` 清华大学源了，加快软件包下载速度。

方法一：自动替换（推荐）

可以使用如下命令自动替换官方源为 TUNA 镜像源

`pkg update` 卡住的话多按几次回车 不要傻乎乎的等

```
sed -i 's@^\(deb.*stable main\)#@#\1\ndeb
https://mirrors.tuna.tsinghua.edu.cn/termux/termux-packages-24 stable main@'
$PREFIX/etc/apt/sources.list

sed -i 's@^\(deb.*games stable\)#@#\1\ndeb
https://mirrors.tuna.tsinghua.edu.cn/termux/game-packages-24 games stable@'
$PREFIX/etc/apt/sources.list.d/game.list

sed -i 's@^\(deb.*science stable\)#@#\1\ndeb
https://mirrors.tuna.tsinghua.edu.cn/termux/science-packages-24 science
stable@' $PREFIX/etc/apt/sources.list.d/science.list

pkg update
```

更换源几秒钟就可以执行完 `pkg update` 了，心里顿时乐开了花。

方法二：手动修改

请使用内置或安装在 Termux 里的文本编辑器，例如 `vi` / `vim` / `nano` 等直接编辑源文件，**不要使用 RE 管理器等其他具有 ROOT 权限的外部 APP** 来修改 Termux 的文件

编辑 `$PREFIX/etc/apt/sources.list` 修改为如下内容

```
# The termux repository mirror from TUNA:
deb https://mirrors.tuna.tsinghua.edu.cn/termux/termux-packages-24 stable main
```

编辑 `$PREFIX/etc/apt/sources.list.d/science.list` 修改为如下内容

```
# The termux repository mirror from TUNA:
deb https://mirrors.tuna.tsinghua.edu.cn/termux/science-packages-24 science
stable
```

编辑 `$PREFIX/etc/apt/sources.list.d/game.list` 修改为如下内容

```
# The termux repository mirror from TUNA:
deb https://mirrors.tuna.tsinghua.edu.cn/termux/game-packages-24 games stable
```

安装基础工具

更换源之后赶紧来下载安装一些基本工具吧，这些工具基本上是 Linux 系统自带的，因为 Termux 为了体积不过大，默认是没有带这些工具的，执行下面的命令来安装：

```
pkg update
pkg install vim curl wget git tree -y
```

定制常用按键

在 Termux v0.66 的版本之后我们可以通过 `~/.termux/termux.properties` 文件来定制我们的常用功能按键，默认是不存在这个文件的，我们得自己配置创建一下这个文件。

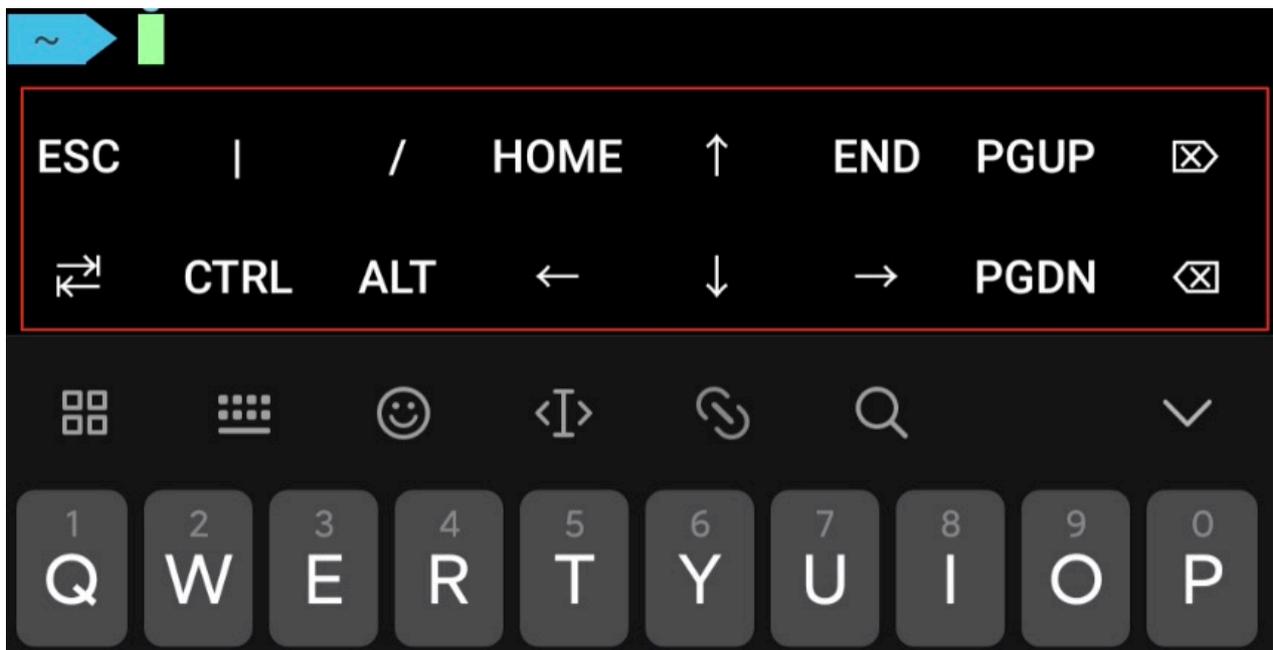
下面做尝试简单配置一下这个文件：

```
vim ~/.termux/termux.properties
```

内容为：

```
extra-keys = [ \
  ['ESC', '|', '/', 'HOME', 'UP', 'END', 'PGUP', 'DEL'], \
  ['TAB', 'CTRL', 'ALT', 'LEFT', 'DOWN', 'RIGHT', 'PGDN', 'BKSP'] \
]
```

修改完成保存文件后，重启 Termux app生效配置：



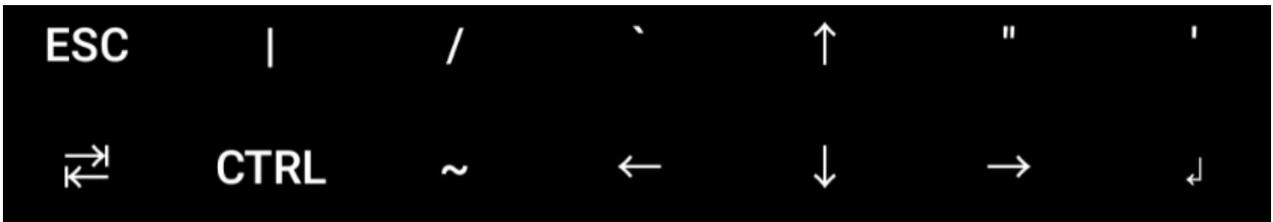
可以直接输入特殊的字符串，例如上面的例子中的 `|` 就是一个字符串，此外 Termux 还有封装了一些特殊按键，入上面例子中的 `ESC` 就是 Termux 自带的按键，完整的特殊按键表如下：

按键	说明
<code>CTRL</code>	特殊按键
<code>ALT</code>	特殊按键
<code>FN</code>	特殊按键
<code>ESC</code>	退出键
<code>TAB</code>	表格键
<code>HOME</code>	原位键
<code>END</code>	结尾键
<code>PGUP</code>	上翻页键
<code>PGDN</code>	下翻页键
<code>INS</code>	插入键
<code>DEL</code>	删除键
<code>BKSP</code>	退格键
<code>UP</code>	方向键 上
<code>LEFT</code>	方向键 左
<code>RIGHT</code>	方向键 右
<code>DOWN</code>	方向键 下
<code>ENTER</code>	回车键
<code>BACKSLASH</code>	反斜杠 <code>\</code>
<code>QUOTE</code>	双引号键
<code>APOSTROPHE</code>	单引号键
<code>F1~F12</code>	F1-F12按键

上面列出的三个特殊键中的每一个最多只能在附加键定义中列出一次，超过次数将会报错。

下面是国光我自用的按键表：

```
extra-keys = [ \
  ['ESC', '|', '/', '`', 'UP', 'QUOTE', 'APOSTROPHE'], \
  ['TAB', 'CTRL', '~', 'LEFT', 'DOWN', 'RIGHT', 'ENTER'] \
]
```



终端配色方案

脚本项目地址：<https://github.com/Cabbagec/termux-ohmyzsh/>

该脚本主要使用了 `zsh` 来替代 `bash` 作为默认 shell，并且支持色彩和字体样式，同时也激活了外置存储，可以直接访问SD卡下的目录。主题默认为 `agnoster`，颜色样式默认为 `Tango`，字体默认为 `Ubuntu`。

执行下面这个命令确保已经安装好了 `curl` 命令

```
sh -c "$(curl -fsSL https://github.com/Cabbagec/termux-ohmyzsh/raw/master/install.sh)"
```

如果因为不可抗力的原因，出现 `port 443: Connection refused` 网络超时的情况，那么执行下面国内光迁移到国内的地址的命令即可：

```
sh -c "$(curl -fsSL https://html.sqlsec.com/termux-install.sh)"
```

Android6.0 以上会弹框确认是否授权访问文件,点击 `始终允许` 授权后 `Termux` 可以方便的访问SD卡文件。

要允许 Termux 访问以下权限吗？



读写设备上的照片及文件

拒绝

始终允许

手机 App 默认只能访问自己的数据，如果要访问手机的存储，需要请求权限，如果你刚刚不小心点了拒绝的话，那么可以执行以下命令来重新获取访问权限：

```
termux-setup-storage
```

脚本允许后先后有如下两个选项：

```
Enter a number, leave blank to not to change: 14  
Enter a number, leave blank to not to change: 6
```

分别选择 `色彩样式` 和 `字体样式`，重启 Termux app 后生效配置。不满意刚刚的效果，想要继续更改配色方案的话，可以根据下面命令来更改对应的色彩配色方案：

设置色彩样式：

输入 `chcolor` 命令更换色彩样式，或者：`~/termux/colors.sh` 命令

设置字体

运行 `chfont` 更换字体，或者：`~/termux/fonts.sh` 命令

创建目录软连接

执行过上面的一键配置脚本后，并且授予 Termux 文件访问权限的话，会在家目录生成 `storage` 目录，并且生成若干目录，软连接都指向外置存储卡的相应目录：

```
~ ➔ tree
.
├── storage
│   ├── dcim -> /storage/emulated/0/DCIM
│   ├── downloads -> /storage/emulated/0/Download
│   ├── movies -> /storage/emulated/0/Movies
│   ├── music -> /storage/emulated/0/Music
│   ├── pictures -> /storage/emulated/0/Pictures
│   └── shared -> /storage/emulated/0
├── termux-ohmyzsh
│   ├── LICENSE
│   ├── README.md
│   ├── install.sh
│   ├── termux-ohmyzsh.png
│   └── update.sh
└── 8 directories, 5 files
~ ➔ █
```

创建QQ文件夹软连接

手机上一般经常使用手机QQ来接收文件,这里为了方便文件传输,直接在 `storage` 目录下创建软链接.

QQ

```
ln -s /data/data/com.termux/files/home/storage/shared/tencent/QQfile_recv QQ
```

TIM

```
ln -s /data/data/com.termux/files/home/storage/shared/tencent/TIMfile_recv TIM
```

```
~ ➔ ls
QQ storage termux-ohmyzsh
~ ➔ cd QQ
~/QQ ➔ ls
flag.txt
```

ESC ↵ CTRL ALT - ↓ ↑

这样可以直接在 `home` 目录下去访问QQ文件夹，大大提升了工作效率。

zsh 主题配色

编辑家目录下的 `.zshrc` 配置文件

```
$ vim .zshrc
```

第一行可以看到,默认的主题是 `agnoster` 主题:

```
19:18 0.3K/s 0.3K/s ↗ 🔒 📶 HD 📶 🔋 100 ⚡  
ZSH_THEME="agnoster"  
# If you come from bash you might have to change your $PATH.  
# export PATH=$HOME/bin:/usr/local/bin:$PATH
```

实际上这个主题也蛮酷的,如果你还想更换其他主题的话,那么在 `.oh-my-zsh/themes` 目录下放着 `oh-my-zsh` 所有的主题配置文件,只要将默认的 `agnoster` 更换为其他的主题文件名即可。下面是国光认为还不错的几款主题

```
❌ ~/.oh-my-zsh master ls  
CODE_OF_CONDUCT.md custom templates  
CONTRIBUTING.md lib themes  
LICENSE.txt log tools  
README.md oh-my-zsh.sh  
cache plugins  
~/.oh-my-zsh master █  
ESC ↩️ CTRL ALT - ↓ ↑
```

```
# u0_a387 @ localhost in ~ [19:25:18]  
$ ls  
QQ storage termux-ohmyzsh  
# u0_a387 @ localhost in ~ [19:25:19]  
$ █  
ESC ↩️ CTRL ALT - ↓ ↑  
☰ ☰ 😊 <I> 🔗 🔍 ✓
```

```
→ ~ cd termux-ohmyzsh
→ termux-ohmyzsh git:(master) ls
LICENSE      install.sh   update.sh
README.md    termux-ohmyzsh.png
→ termux-ohmyzsh git:(master) █
```

ESC ↵ CTRL ALT — ↓ ↑

主题比较多，国光这里就不列举了，感兴趣大家可以一个个尝试去看看。当然如果你是个变态的话，可以尝试 `random` 主题，每打开一个会话配色主题都是随机的。

```
ZSH_THEME="random"
```

zsh 插件推荐

zsh 之所以受欢迎除了好看的配色以为，另一个原因就是强大的插件了。下面国光列举一款比较实用的插件的安装方法，更多强大的插件等待大家自己去探索。

autosuggestions

根据用户的平时使用习惯，终端会自动提示接下来可能要输入的命令，这个实际使用效率还是比较高的：

```
# 拷贝到 plugins 目录下
git clone git://github.com/zsh-users/zsh-autosuggestions
$ZSH_CUSTOM/plugins/zsh-autosuggestions
```

在 `~/.zshrc` 中配置：

```
plugins=(其他的插件 zsh-autosuggestions)
```

```
shell startup.
plugins=(git zsh-autosuggestions)
source $ZSH/oh-my-zsh.sh
# User configuration
# export MANPATH="/usr/local/man:$MANPATH"
.zshrc [+] 68,32 70%
:wq
```

ESC ↩️ CTRL ALT - ↓ ↑

输入 `zsh` 命令生效配置:

```
~ ➤ ls
QQ storage termux-ohmyzsh
~ ➤ vim .zshrc
```

ESC ↩️ CTRL ALT - ↓ ↑

可以看到国光我只敲了一个 `v` 后面的命令就自动提示补全了，这时候只要按 `右方向键`，在 Termux 里面的快捷键是 `音量加 + D`，就可以直接补全命令了。

修改启动问候语

默认的启动问候语如下:

20:22

...0.9K/s       100 

Welcome to Termux!

Wiki: <https://wiki.termux.com>
Community forum: <https://termux.com/community>
Gitter chat: <https://gitter.im/termux/termux>
IRC channel: #termux on freenode

Working with packages:

- * Search packages: `pkg search <query>`
- * Install a package: `pkg install <package>`
- * Upgrade packages: `pkg upgrade`

Subscribing to additional repositories:

- * Root: `pkg install root-repo`
- * Unstable: `pkg install unstable-repo`
- * X11: `pkg install x11-repo`

Report issues at <https://termux.com/issues>

这个启动问候语在前期对于初学者有一定的帮助，但是随着你们 Termux 的熟悉，这个默认的问候语就会显得比较臃肿。编辑问候语文件可以直接修改启动显示的问候语：

```
vim $PREFIX/etc/motd
```

修改完的效果如下：



这样启动新的会话的时候看上去就会简洁很多。什么你也想要这个效果？呐 下面是国光自己生成的，可以直接复制粘贴：



超级管理员身份

实际上 Termux 不需要 root 权限也可以折腾各种各样的操作的，大家不必对 root 抱有啥幻想，本文的操作基本上没有涉及到手机要用到 root 的地步。

手机没有root

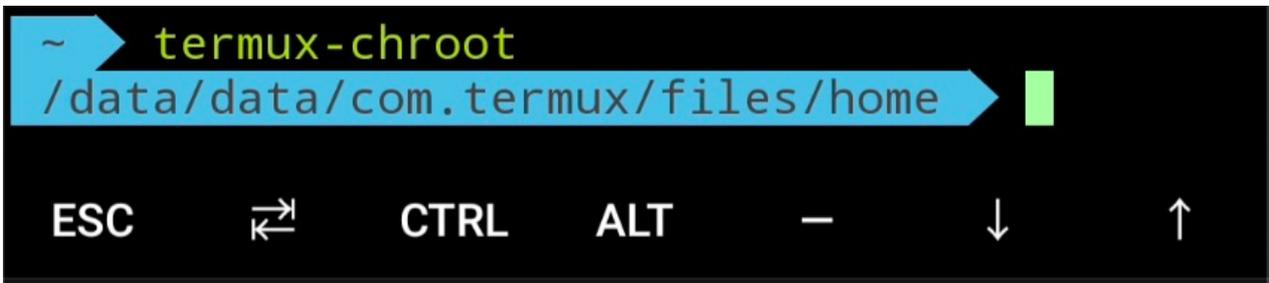
利用 `proot` 可以为手机没有root的用户来模拟一个root的环境，这里主要是经典的 Linux 文件系统布局上的模拟。

```
pkg install proot -y
```

然后终端下面输入：

```
termux-chroot
```

即可模拟 `root` 环境，该环境模仿 Termux 中的常规Linux文件系统，但是不是真正的 root。



输入 `exit` 可回到普通用户的文件系统。

手机已经root

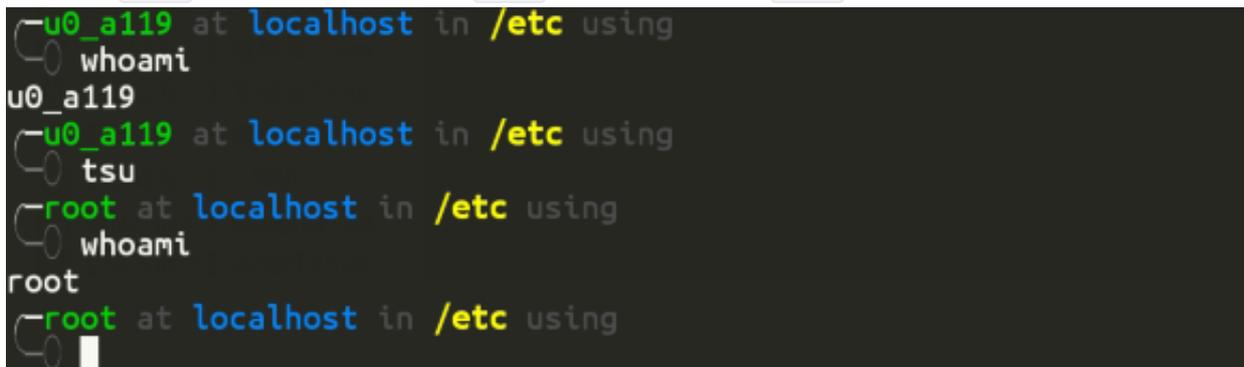
安装 `tsu`，这是一个 `su` 的 Termux 版本，是一个真正的root权限，用来在termux上替代 `su`，操作不慎可能对手机有安全风险。因为官方封装了，所以安装也很简单：

```
pkg install tsu -y
```

然后终端下面输入：

```
tsu
```

即可切换 `root` 用户，这个时候会弹出 `root` 授权提示，给予其 `root` 权限，效果图如下：



在管理员身份下，输入 `exit` 可回到普通用户身份。不过本文没有设计到 `root` 权限的操作，一些底层的工具可能才会需要，考虑到 `root` 的不安全性和 那些工具的冷门性，国光这里就没有继续拓展。

开发环境

Termux 支持的开发环境很强，可以完美的运行 C、Python、Java、PHP、Ruby等开发环境，建议读者朋友们选择自己需要的开发环境折腾。

编辑器

写代码前总得折腾一下编辑器，毕竟磨刀不误砍柴工嘛。Termux 支持多种编辑器，完全可以满足日常使用需求。

Emacs

据说Emacs是神的编辑器，国光我这种小菜鸡还不会使用哎，但是 Termux 官方已经封装好了 Emacs 了，我们安装起来就会简单很多：

```
pkg install emacs
```

nano

nano 是一个小而美的编辑器。具有如下：打开多个文件，每行滚动，撤消/重做，语法着色，行编号等功能

同样安装起来也很简单：

```
pkg install nano
```

Vim

Vim 被称为编辑器之神，基本上 Linux 发行版都会自带 Vim，这个在前文基本工具已经安装了，如果你没有安装的话，可以使用如下命令安装：

```
pkg install vim
```

并且官方也已经封装了 `vim-python`，对Python相关的优化。

```
pkg install vim-python
```

解决汉字乱码

如果你的Vim打开汉字出现乱码的话，那么在家目录(~)下,新建 `.vimrc` 文件

```
vim .vimrc
```

添加内容如下：

```
set fileencodings=utf-8,gb2312,gb18030,gbk,ucs-bom,cp936,latin1
set enc=utf8
set fencs=utf8,gbk,gb2312,gb18030
```

然后 `source` 下变量：

```
source .vimrc
```

效果图

```
0.36 K/s 74% 6:04
author = 'cscan' # POC编写者
create_date = '2018-04-26' # POC创建时间

def __init__(self):
    super(Poc, self).__init__(Vuln())

def verify(self):
    try:
        self.output.info('开始对 {target} 进行 {vuln} 的扫描'.format(
            target=self.target, vuln=self.vuln))
        filename_list = ['shell', 'webshell', 'ma', 'caidao', 'yi', 'a', 'muma', 'dama', 'xiaoma']
        filetype_list = ['.asp', '.php', '.aspx', '.asa', '.jsp']
        for filename in filename_list:
            for filetype in filetype_list:
                webshell = filename+filetype
                request = requests.get('{target}/{payload}'.format(target=self.target, payload=webshell))
                if request.status_code == 200:
                    self.output.report(self.vuln, '发现{target}存在{name}漏洞'.format(target=self.target, name=self.vuln.name))
    except Exception, e:
        self.output.info('执行异常{}'.format(e))

def exploit(self):
    try:
        self.output.info('开始对 {target} 进行 {vuln} 漏洞利用'.format(
            target=self.target, vuln=self.vuln))
    except Exception, e:
        self.output.info('执行异常{}'.format(e))

QQ/info_shell.py 39,1 55%
```

Vim 配色

Termux Vim 自带了如下的配色：

```
ls /data/data/com.termux/files/usr/share/vim/vim82/colors
```

```
desert.vim    morning.vim    shine.vim     blue.vim     elflord.vim
murphy.vim    slate.vim     darkblue.vim evening.vim   pablo.vim
industry.vim  peachpuff.vim torte.vim     delek.vim    koehler.vim  ron.vim
zellner.vim
```

配色可以自己一个个尝试一下，还是向上面的汉字乱码那样，编辑家目录下的 `.vimrc` 文件：

```
vim ~/.vimrc
```

新增如下内容：

```
set nu          " 显示行号
colorscheme desert " 颜色主题
syntax on      " 打开语法高亮
```

下面是国光随便找的几个颜色主题效果，感兴趣的朋友可以自己一个个尝试：

```
10:10 ...0.3K/s 蓝牙 5G 4G+ 98%
1 if __name__ == '__main__':
2     print(define.ORANGE+define.banner)
3
4     if len(sys.argv) < 2:
5         print(define.ORANGE+define.usage)
6
7     elif sys.argv[1] == '-i':
8         print(define.BLUE+"[*]开始抓取")
9         creat_xlsx()
10        [ip(s) for s in ipaddress.ip_network(str(sys.argv[2]),False)]
11        print(define.BLUE+"[*]抓取完毕")
12
13    elif sys.argv[1] == '-f':
14        print(define.BLUE+"[*]开始抓取")
15        creat_xlsx()
16        [ip(s) for x in open(sys.argv[2]) for s in ipaddress.ip_network(x.strip(),False)]
17        print(define.BLUE+"[*]抓取完毕")
18
hello.py 17,1 Top
ESC ↵ CTRL ALT - ↓ ↑
```

```
10:08 ...0.3K/s 蓝牙 5G 4G+ 97%
1 if __name__ == '__main__':
2     print(define.ORANGE+define.banner)
3
4     if len(sys.argv) < 2:
5         print(define.ORANGE+define.usage)
6
7     elif sys.argv[1] == '-i':
8         print(define.BLUE+"[*]开始抓取")
9         creat_xlsx()
10        [ip(s) for s in ipaddress.ip_network(str(sys.argv[2]),False)]
11        print(define.BLUE+"[*]抓取完毕")
12
13    elif sys.argv[1] == '-f':
14        print(define.BLUE+"[*]开始抓取")
15        creat_xlsx()
16        [ip(s) for x in open(sys.argv[2]) for s in ipaddress.ip_network(x.strip(),False)]
17        print(define.BLUE+"[*]抓取完毕")
18
hello.py 1,1 Top
ESC ↵ CTRL ALT - ↓ ↑
```

```
10:14 ...0.2K/s 蓝牙 5G 4G+ 98%
1 if __name__ == '__main__':
2     print(define.ORANGE+define.banner)
3
4     if len(sys.argv) < 2:
5         print(define.ORANGE+define.usage)
6
7     elif sys.argv[1] == '-i':
8         print(define.BLUE+"[*]开始抓取")
9         creat_xlsx()
10        [ip(s) for s in ipaddress.ip_network(str(sys.argv[2]),False)]
11        print(define.BLUE+"[*]抓取完毕")
12
13    elif sys.argv[1] == '-f':
14        print(define.BLUE+"[*]开始抓取")
15        creat_xlsx()
16        [ip(s) for x in open(sys.argv[2]) for s in ipaddress.ip_network(x.strip(),False)]
17        print(define.BLUE+"[*]抓取完毕")
18
hello.py 1,1 Top
"hello.py" 20L, 632C
ESC ↵ CTRL ALT - ↓ ↑
```

Apache

Apache是一个开源网页服务器软件，由于其跨平台和安全性，被广泛使用，是最流行的Web服务器软件之一。

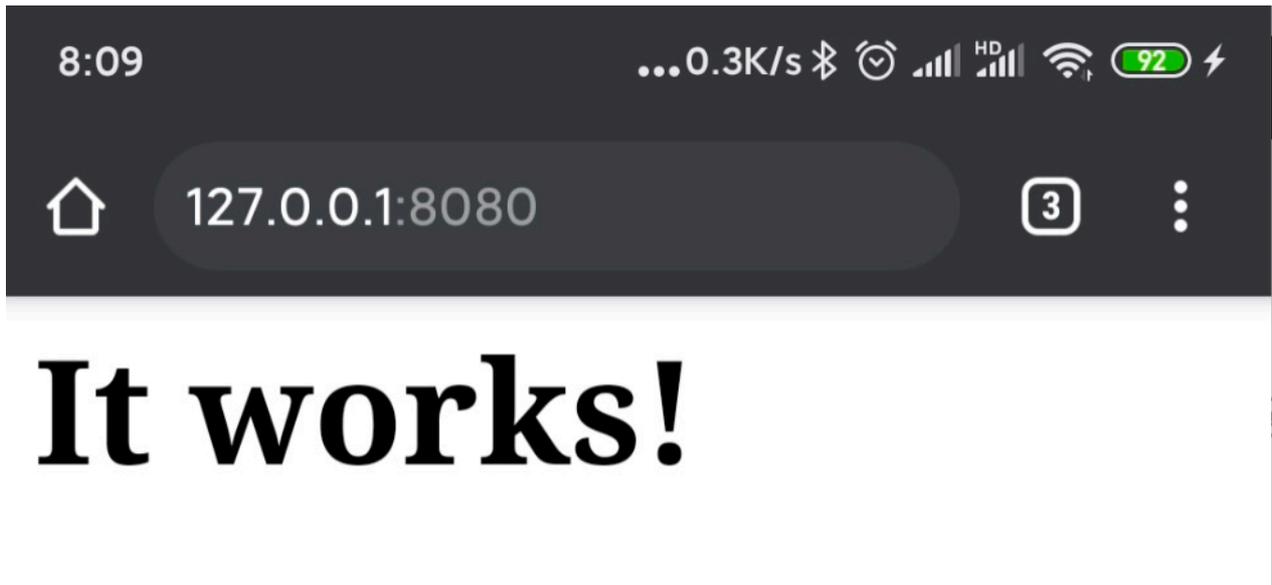
安装 Apache

```
pkg install apache2
```

启动 Apache

```
apachectl start
```

然后浏览器访问: `http://127.0.0.1:8080` 访问是否成功启动:



Termux 自带的 Apache 的网站默认路径为:

```
/data/data/com.termux/files/usr/share/apache2/default-site/htdocs/index.html
```

停止 Apache

```
apachectl stop
```

重启 Apache

```
apachectl restart
```

Apache 解析 PHP

既然Apache、PHP、MySQL都成功安装的话，那么现在只要配置好 Apache 解析 PHP 之后就可以打造一个 Android 平台上的 LAMPP平台了。

安装 php-apache

默认的 Apache 是无法解析 PHP的，我们需要安装相应的包：

```
pkg install php-apache
```

配置 Apache

Termux 上的 Apache 默认配置文件的路径为:

/data/data/com.termux/files/usr/etc/apache2/httpd.conf

直接编辑配置文件:

```
vim /data/data/com.termux/files/usr/etc/apache2/httpd.conf
```

配置文件里面搜索 PHP 没有相关的模块, 所以需要我们手动添加 PHP7 的模块:

```
LoadModule php7_module  
/data/data/com.termux/files/usr/libexec/apache2/libphp7.so
```

并在刚刚这个语句下方添加解析器, 内容如下:

```
<FilesMatch \.php$>  
    SetHandler application/x-httpd-php  
</FilesMatch>
```

接着继续往下找配置文件里面配置默认首页的地方, 我们添加 `index.php` 到默认首页的规则里面:

```
<IfModule dir_module>  
    DirectoryIndex index.php index.html  
</IfModule>
```

这表示网站目录的默认首页是 `index.php`, 如果没有 `index.php` 系统会自动寻找 `index.html` 做为默认首页了。

修改完 Apache 的配置文件后, 记得使用 `apachectl restart` 重启 Apache 服务, 然后这个时候回发现我们重启居然报错了:

```
Apache is running a threaded MPM, but your PHP Module is not compiled to be  
threadsafe. You need to recompile PHP.  
AH00013: Pre-configuration failed
```

不要慌问题不大, 下面来解决这个问题

解决 Apache PHP 报错

先找到如下行

```
LoadModule mpm_worker_module libexec/apache2/mod_mpm_worker.so
```

给他注释掉为:

```
#LoadModule mpm_worker_module libexec/apache2/mod_mpm_worker.so
```

然后找到如下行:

```
#LoadModule mpm_prefork_module libexec/apache2/mod_mpm_prefork.so
```

取消注释为:

```
LoadModule mpm_prefork_module libexec/apache2/mod_mpm_prefork.so
```

最终的示例图如下:

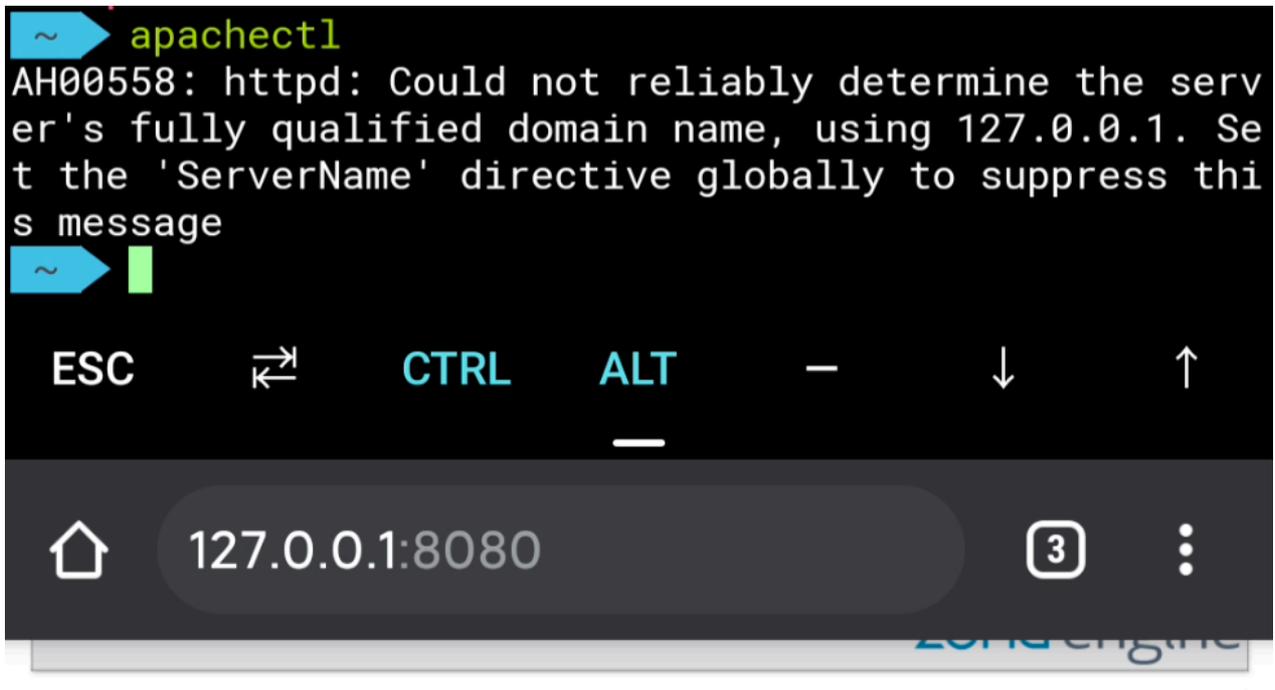
```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule mpm_prefork_module libexec/apache2/mod_mpm_prefork.so
LoadModule php7_module /data/data/com.termux/files/usr/libexec/apache2/libphp7.so
<FilesMatch \.php$>
  SetHandler application/x-httpd-php
</FilesMatch>
#LoadModule mpm_worker_module libexec/apache2/mod_mpm_worker.so
LoadModule authn_file_module libexec/apache2/mod_authn_file.so
#LoadModule authn_dbm_module libexec/apache2/mod_authn_dbm.so
```

解析 PHP 测试

在 Apache 的网站根目录下, 创建一个 index.php, 测试一下 phpinfo() 函数能否正常运行:

```
echo '<?php phpinfo(); ?>' > $PREFIX/share/apache2/default-
site/htdocs/index.php
```

然后浏览访问: `http://127.0.0.1:8080` 查看效果:



Configuration apache2handler

Apache Version	Apache/2.4.43 (Unix) PHP/7.4.5
Apache API Version	20120211
Server Administrator	you@example.com
Hostname:Port	127.0.0.1:0

C

Termux 官方封装了 Clang，他是一个C、C++、Objective-C和Objective-C++编程语言的编译器前端。

安装 clang

```
pkg install clang
```

编译测试

clang 在编译这一块很强大，感兴趣的朋友可以去网上查看详细的教程，国光这里只演示基本的 Hello World使用。写一个Hello World的C程序，如下 `hello.c`：

```
#include <stdio.h>

int main(){
    printf("Hello World")
    return 0;
}
```

编辑完成后，使用 clang 来编译生成 hello 的可执行文件：

```
clang hello.c -o hello
```



```
~> ls
hello.c  hello.py  storage  termux-ohmyzsh
~> clang hello.c -o hello
~> ls
hello    hello.py  termux-ohmyzsh
hello.c  storage
~> ./hello
Hello Termux%
~>
```

ESC ↩ CTRL ALT - ↓ ↑

Java

Termux 原生编译JAVA只能使用 `ecj` (Eclipse Compiler for Java) 和 `dx` 了，然后使用 Android 自带的 `dalvikvm` 运行。如果想要完整体验JAVA环境的话，另一个方法就是 Termux 里面安装一个完整的 Linux 系统，然后在 Linux里面运行Java，安装系统部分下面文章会详细介绍，这一节国光只介绍最基本的操作。

安装编译工具

```
pkg install ecj dx -y
```

国光这里只演示基本的 Hello World 使用。写一个Hello World的JAVA 程序，如下 `HelloWorld.java`：

```
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("Hello Termux");
    }
}
```

编译生成 class 文件

```
ecj HelloWorld.java
```

编译生成 dex 文件

```
dx --dex --output=hello.dex HelloWorld.class
```

使用 dalvikvm 运行

格式规范如下: `dalvikvm -cp dex文件名 类名`

```
dalvikvm -cp hello.dex HelloWorld
```

```
~ ➤ ls
HelloWorld.java  hello.py  termux-ohmyzsh
hello.c          storage
~ ➤ ecj HelloWorld.java
~ ➤ ls
HelloWorld.class  hello.c  storage
HelloWorld.java  hello.py  termux-ohmyzsh
~ ➤ dx --dex --output=hello.dex HelloWorld.class
S
~ ➤ ls
HelloWorld.class  hello.dex  termux-ohmyzsh
HelloWorld.java  hello.py
hello.c          storage
~ ➤ dalvikvm -cp hello.dex HelloWorld
Hello Termux
~ ➤ █
```

ESC ↩ CTRL ALT - ↓ ↑

MariaDB(MySQL)

MariaDB 是 MySQL 关系数据库管理系统的一个复刻，由社区开发，有商业支持，旨在继续保持在GNU GPL下开源。开发这个分支的原因之一是：甲骨文公司收购了 MySQL 后，有将 MySQL 闭源的潜在风险，因此社区采用分支的方式来避开这个风险。

安装 MariaDB

Termux 官方也封装了MariaDB，所以安全起来很方便：

```
pkg install mariadb
```

```
~ ➔ mysql --version
mysql Ver 15.1 Distrib 10.4.12-MariaDB, for Android (aarch64) using EditLine wrapper
```

这里基本上会安装很顺利，但是早期用户可能出现安装失败的情况，如果安装失败的话，这个时候手动在配置目录下创建 `my.cnf.d` 文件夹即可：

```
$ cd /data/data/com.termux/files/usr/etc/
$ mkdir my.cnf.d
```

初始化数据库

早期的 Termux 安装完 MySQL 是需要初始化数据库的，新版本在安装时候就已经初始化了数据库

```
mysql_install_db
```

2020年4月19日：国光今天安装的MySQL 发现已经存在 `mysql.user` 表了，无需初始化：

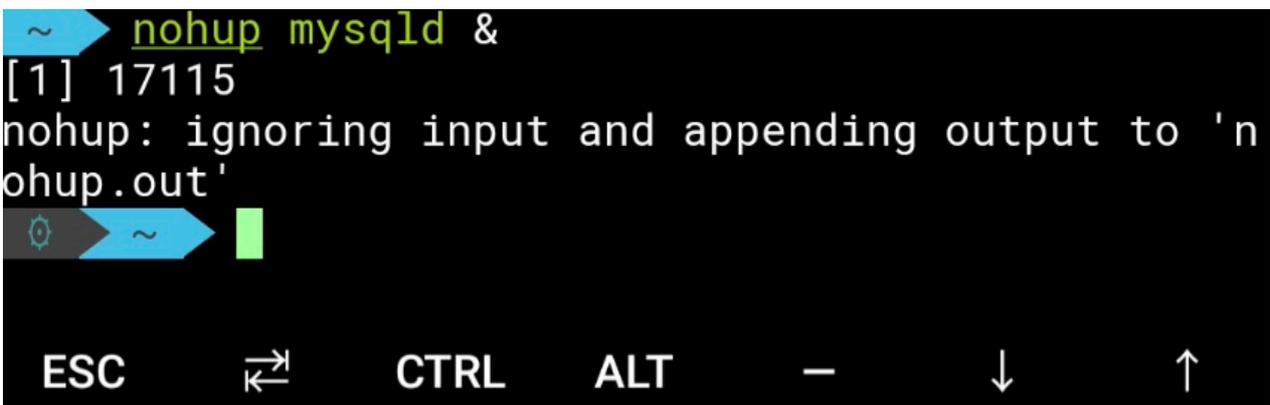
```
~ ➔ mysql_install_db
mysql.user table already exists!
Run mysql_upgrade, not mysql_install_db
```

启动 MySQL 服务

因为正常启动完成后，MySQL 这个会话就一直存活，类似与 debug 调试一样，此时使用 `Ctrl + C` -> 中止当前进程也无济于事，体验式就一点都不优雅，所以这里国光使用 Linux 自带的 `nohup` 命令将其放到后台启动。

```
nohup mysqld &
```

```
~ ➔ nohup mysqld &
[1] 17115
nohup: ignoring input and appending output to 'nohup.out'
```



图片上这个 `17115` 此时就是 `mysqld` 的进程 PID 号，我们使用如下命令验证一下是否正确：

```
ps aux|grep mysql
```

可以看到果然是进程的 PID 号：

```
~ ➤ nohup mysql &
[1] 17115
nohup: ignoring input and appending output to 'nohup.out'
~ ➤ ps aux | grep mysql
u0_a387 17115 0.1 1.3 1053064 76984 pts/1 S
<1 1970 0:01 mysql
u0_a387 18518 1.0 0.0 27836 2616 pts/1 S
<+ 1970 0:00 grep --color=auto --exclude-dir=.b
zr --exclude-dir=CVS --exclude-dir=.git --excl
ude-dir=.hg --exclude-dir=.svn --exclude-dir=.i
dea --exclude-dir=.tox mysql
```

至于 nohup 运行的提示

```
nohup: ignoring input and appending output to `nohup.out`
```

这个是正常现象，无伤大雅，Termux 下就这样将就着用吧。

停止 MySQL 服务

Termux 下没有好的办法退出 MySQL 服务，只能强制杀掉进程了，使用如下命令格式可以轻松杀掉进程：

```
kill -9 PID
```

```
~ ➤ kill -9 17115
[1] + 17115 killed nohup mysql
```

当然每次查看进程的PID号，再杀掉进程有点繁琐了，实际上这一步可以直接这样操作：

```
kill -9 `pgrep mysql`
```

Awesome ! 优雅!

默认的两个用户

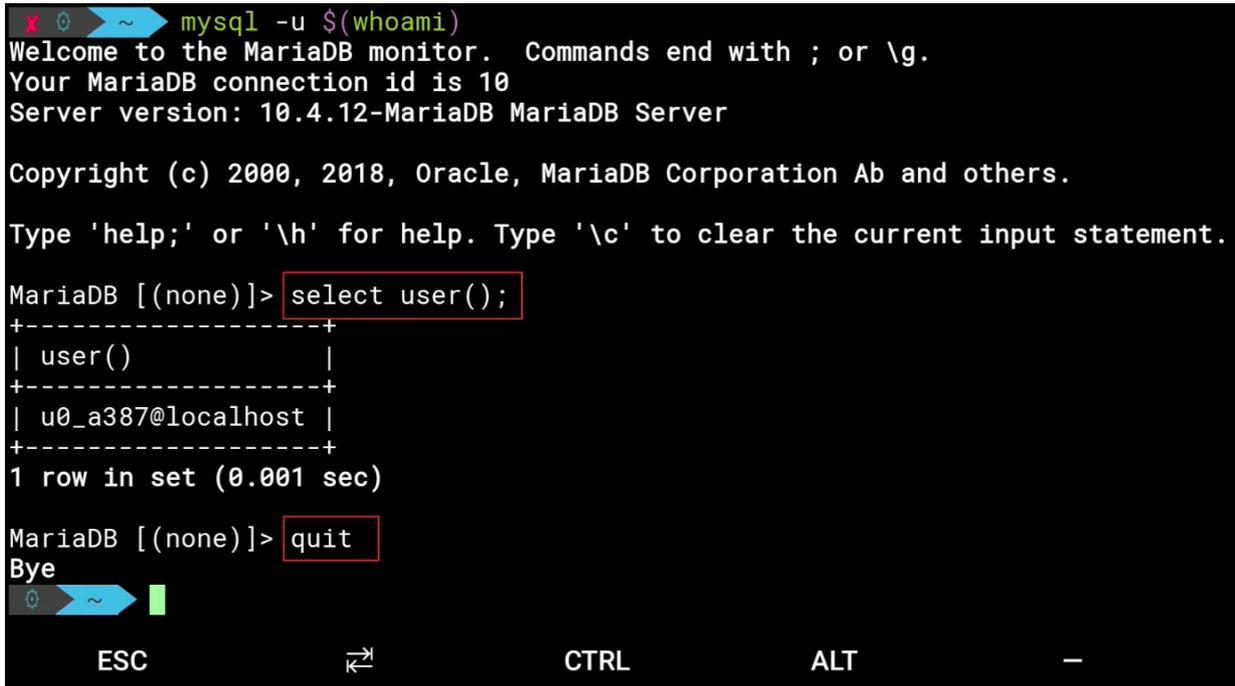
用户登录的前提是MySQL服务在后台运行，如果你按照上一小节操作把MySQL kill掉的话，请重新启动一下MySQL服务

新版本的 Termux 安装初始化数据库的时候包含两个高权限用户，一个是无法访问的 **root** 用户

```
~ ➤ mysql -u root
ERROR 1698 (28000): Access denied for user 'root'
'@'localhost'
```

另一个用户就是 Termux 的用户名，默认密码为空，我们来登录看看：

```
mysql -u $(whoami)
```



```
x @ ~ mysql -u $(whoami)
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.4.12-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> select user();
+-----+
| user() |
+-----+
| u0_a387@localhost |
+-----+
1 row in set (0.001 sec)

MariaDB [(none)]> quit
Bye
ESC      CTRL      ALT      -
```

那么这个无法登录的 root 用户该怎么办呢？不要着急 继续往下看

修改 root 密码

老版本的 Termux 的直接使用 `mysql_secure_installation` 可以设置密码，但是新版本的安全策略变了 我们在设置密码的时候回提示当前密码不正确，所以这条路行不通了。

这里我们只能使用MySQL的另一个用户名，即 Termux 用户名登录，然后来修改 root 的密码，使用如下命令修改 root 密码：

```
# 登录 Termux 用户
mysql -u $(whoami)

# 修改 root 密码的 SQL语句
use mysql;
set password for 'root'@'localhost' = password('你设置的密码');

# 刷新权限 并退出
flush privileges;
quit;
```

```
MariaDB [(none)]> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> set password for 'root'@'localhost' = password('P@ssw0rd');
Query OK, 0 rows affected (0.024 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.003 sec)

MariaDB [mysql]> quit;
Bye
```

你的密码

OK! 如何和图片上差不多的效果, 那么修改 root 密码就成功了。

root 用户登录

修改完密码之后我们就可以美滋滋地使用 root 用户来登录了:

```
mysql -u root -p
Enter password: xxxxxx (这里输入你的密码)
```

```
mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.12-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> select user(), version();
+-----+-----+
| user()          | version()          |
+-----+-----+
| root@localhost | 10.4.12-MariaDB   |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]> quit
Bye
```

远程登录 MySQL

使用 `ip a` 后查看 IP 地址后, 尝试电脑端远程访问 Termux 的数据库:

```
# sqlsec @ com in ~ [17:48:40] C:1
$ mysql -uroot -h192.168.31.124 -p
Enter password:
ERROR 2013 (HY000): Lost connection to MySQL server at 'reading initial communication packet', system error: 3
```

发现默认是无法成功连接的, 这个时候我们需要到数据库手动开启 root 用户的远程访问权限:

这里的 P@ssw0rd 是我的 root 密码

```
grant all on *.* to root@'%' identified by 'P@ssw0rd' with grant option;
flush privileges;
```

```
MariaDB [(none)]> grant all on *.* to root@'%' i
identified by 'P@ssw0rd' with grant option;
Query OK, 0 rows affected (0.027 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.032 sec)
```

执行完成后 尝试PC端远程过去看看:

```
$ mysql -uroot -h192.168.31.124 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.5.5-10.4.12-MariaDB MariaDB Server

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select user(), version();
+-----+-----+
| user()          | version()          |
+-----+-----+
| root@192.168.31.53 | 10.4.12-MariaDB |
+-----+-----+
1 row in set (0.26 sec)

mysql> quit
Bye
```

Nginx

Nginx 是一个高性能的 Web 和反向代理服务器，Nginx 用的熟悉的话，下面搭建各种网站也就轻而易举了。

安装 Nginx

Termux 安装 Nginx 也很简单，一条命令即可：

```
pkg install nginx
```

安装完成后，国光的习惯是查看一下版本信息：

```
~ nginx -v
nginx version: nginx/1.17.10
```

测试 Nginx

测试检查 Nginx 的配置文件是否正常:

```
nginx -t
```

```
~> nginx -t
nginx: the configuration file /data/data/com.termux/files/usr/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /data/data/com.termux/files/usr/etc/nginx/nginx.conf test is successful
```

现在测试肯定是OK的，这个多用于我们修改完 Nginx 的配置文件后的检查。

启动 Nginx

早期版本的 Termux 需要在 `termux-chroot` 环境下才可以成功启动 Nginx，新版本的 Termux 可以直接启动，很是方便:

```
nginx
```

Termux 在 Nginx 上默认运行的端口号是 8080，使用 `pgrep` 命令也可以查看 Nginx 进程相关的PID号。

然后手机本地直接访问 `http://127.0.0.1:8080` 查看 Nginx 是否正常启动:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

重启 Nginx

一般当修改完 Nginx 相关的配置文件时，我们需要重启 Nginx，使用如下命令即可重启：

```
nginx -s reload
```

停止 Nginx

方法一 原生停止

```
nginx -s stop
```

或者

```
nginx -s quit
```

quit 是一个优雅的关闭方式，Nginx在退出前完成已经接受的连接请求。Stop 是快速关闭，不管有没有正在处理的请求。

方法二 杀掉进程

只需三秒钟，里造会干我一样，爱象节款游戏 扯远了，只需要1条命令，即可优雅的终止掉 Nginx 服务：

```
kill -9 `pgrep nginx`
```

貌似手机党 并不好敲 这个 ` 符号 =, =, 如果实在敲不出来，那就分两步走吧：

```
# 查询 nginx 进程相关的 PID 号
pgrep nginx

# 杀掉 查询出的 PID号进程
kill -9 PID
```

Nginx 解析 PHP

Termux 下的 Nginx 解析 PHP 这里单独拿出一级标题来叙述，成功解析的话,下面安装 wordpress等 PHP网站就会轻松很多。

安装 php-fpm

Nginx 本身不能处理 PHP，它只是个 Web 服务器，当接收到 PHP 请求后发给 PHP 解释器处理。Nginx 一般是把请求转发给 fastcgi 管理进程处理，PHP-FPM 是一个PHP FastCGI管理器，所以这里得先安装它：

```
pkg install php-fpm
```

安装完成顺便检查一下版本信息吧：

```
~ # php-fpm -v
PHP 7.4.5 (fpm-fcgi) (built: Apr 14 2020 21:45:48)
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
```

配置 php-fpm

编辑 php-fpm 的配置文件 `www.conf`：

```
vim $PREFIX/etc/php-fpm.d/www.conf
```

定位搜索 `listen =` 找到

```
listen = /data/data/com.termux/files/usr/var/run/php-fpm.sock
```

将其改为：

```
listen = 127.0.0.1:9000
```

??? 啥 你不会使用 vim 搜索 $\backslash(_,_)$ 那就老老实实一个个翻页吧。

配置 Nginx

编辑 Nginx 的配置文件 `nginx.conf`:

```
vim $PREFIX/etc/nginx/nginx.conf
```

下面国光贴出配置好的完整配置文件，大家可以参考下面这些图，只需要2大步骤：

1. 添加 `index.php` 到默认首页的规则里面

```
35     server {
36         listen      8080;
37         server_name localhost;
38
39         #charset koi8-r;
40
41         #access_log logs/host.access.log main;
42
43         location / {
44             root    /data/data/com.termux/files/usr/share/nginx/html;
45             index  index.html index.htm index.php;
46         }
47
48         #error_page 404                    /404.html;
49
```

2. 取消 `location ~ \.php$` 这些注释，按照图片上的提示修改：

```
65     location ~ \.php$ {
66         root    html;
67         fastcgi_pass 127.0.0.1:9000;
68         fastcgi_index index.php;
69         fastcgi_param SCRIPT_FILENAME /data/data/com.termux/files/usr/share/nginx/html$fastcgi_
script_name;
70         include fastcgi_params;
71     }
72
```

Termux 里面的 Nginx 默认网站的根目为：

```
/data/data/com.termux/files/usr/share/nginx/html
```

如果想要修改默认路径的话 只需要在配置文件中 替换2处出现的这个路径即可

下面贴一份完整的配置文件：

```
worker_processes 1;
events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
```

```
sendfile          on;
keepalive_timeout 65;

server {
    listen          8080;
    server_name    localhost;
    location / {
        root        /data/data/com.termux/files/usr/share/nginx/html;
        index       index.html index.htm index.php;
    }

    error_page     500 502 503 504 /50x.html;
    location = /50x.html {
        root        /data/data/com.termux/files/usr/share/nginx/html;
    }

    location ~ /\.php$ {
        root         html;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
/data/data/com.termux/files/usr/share/nginx/html$fastcgi_script_name;
        include      fastcgi_params;
    }
}
}
```

测试 PHP 解析

Nginx 默认网站的根目录为：

```
/data/data/com.termux/files/usr/share/nginx/html
```

在这个网站根目录下新建 `info.php` 内容为： `<?php phpinfo(); ?>`

```
echo '<?php phpinfo(); ?>' > $PREFIX/share/nginx/html/info.php
```

启动服务

先启动 php-fpm 服务：

```
php-fpm
```

然后再启动 Nginx 服务

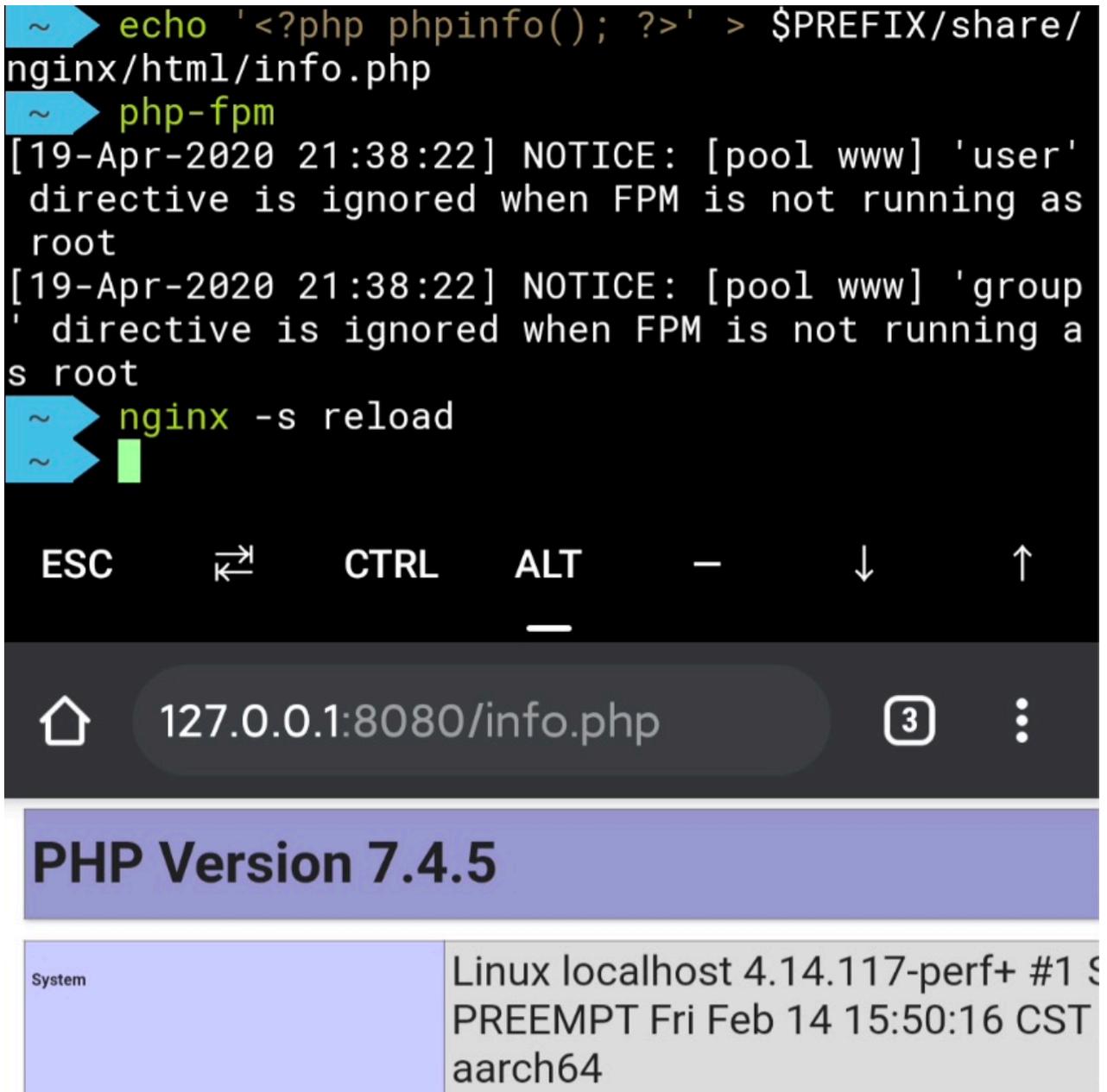
```
nginx
```

如果你的 Nginx 已经启动了的话，使用 `nginx -s reload` 重启 Nginx

访问测试

浏览器访问 `http://127.0.0.1:8080/info.php` 来看看刚刚新建的测试文件是否解析了：

```
~ ➤ echo '<?php phpinfo(); ?>' > $PREFIX/share/nginx/html/info.php
~ ➤ php-fpm
[19-Apr-2020 21:38:22] NOTICE: [pool www] 'user' directive is ignored when FPM is not running as root
[19-Apr-2020 21:38:22] NOTICE: [pool www] 'group' directive is ignored when FPM is not running as root
~ ➤ nginx -s reload
~
```



ESC ↵ CTRL ALT — ↓ ↑

127.0.0.1:8080/info.php

PHP Version 7.4.5

System	Linux localhost 4.14.117-perf+ #1 SMP PREEMPT Fri Feb 14 15:50:16 CST aarch64
--------	---

Nodejs

Node.js 是能够在服务器端运行 JavaScript 的开放源代码、跨平台 JavaScript 运行环境。

安装 Nodejs

`nodejs-lts` 是长期支持版本，如果执行 `pkg install nodejs` 版本后，发现 npm 报如下错误：

```
segmentation fault
```

那么这个时候可以尝试卸载当前版本 `pkg uninstall nodejs` 然后执行下面命令安装长期稳定版本：

```
pkg install nodejs-lts
```

安装完成后使用如下命令查看版本信息：

```
node -V  
npm -V
```

Hello World

新建一个 `hello.js` 脚本，内容如下：

```
console.log('Hello Termux');
```

然后尝试运行：

```
$ node hello.js  
Hello Termux
```

http-server

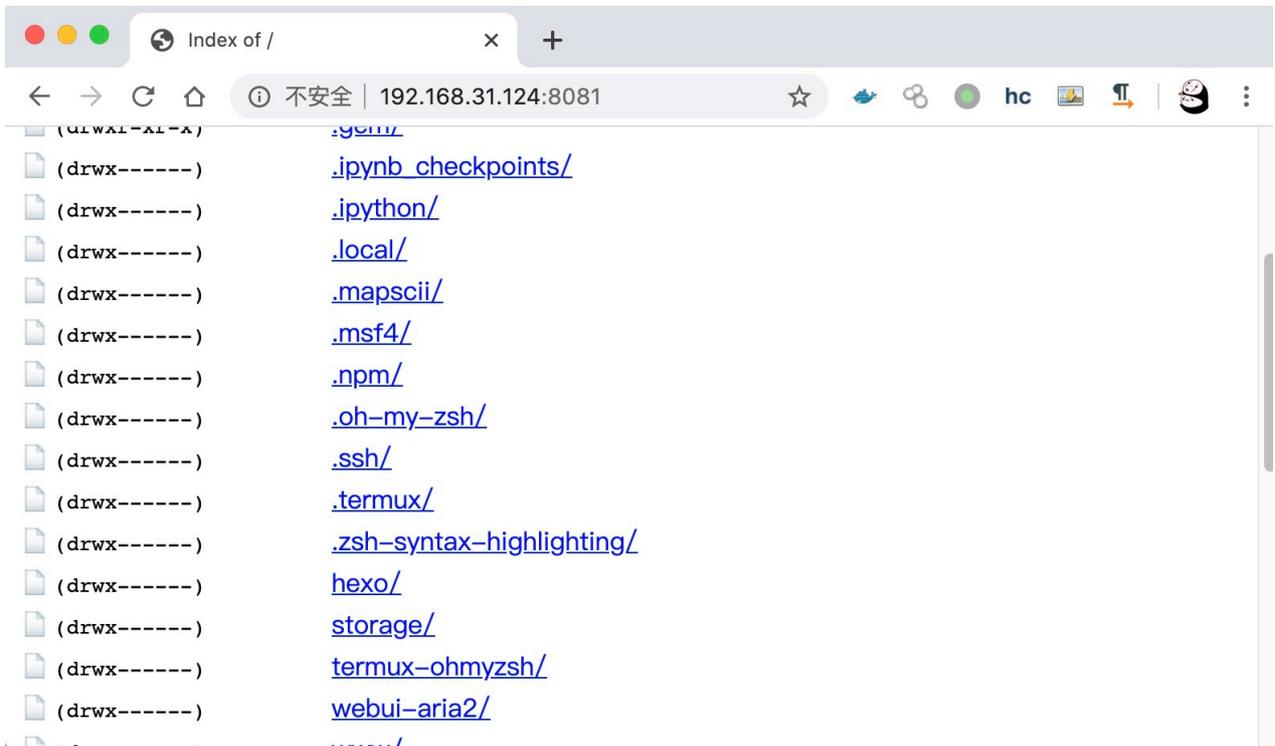
http-server 是一个基于 Node.js 的简单零配置命令行HTTP服务器。

```
# 安装 http-server  
npm install -g http-server  
  
# 运行 http-server  
http-server
```



```
~> http-server  
Starting up http-server, serving ./  
Available on:  
  http://127.0.0.1:8081  
  http://192.168.31.124:8081  
  http://172.19.0.1:8081  
Hit CTRL-C to stop the server  
█  
ESC | / ` ↑ " ' █
```

尝试电脑端浏览器直接访问看看：



安装报错

早期版本的 Termux 的 npm 安装一些包的时候会报如下错误:

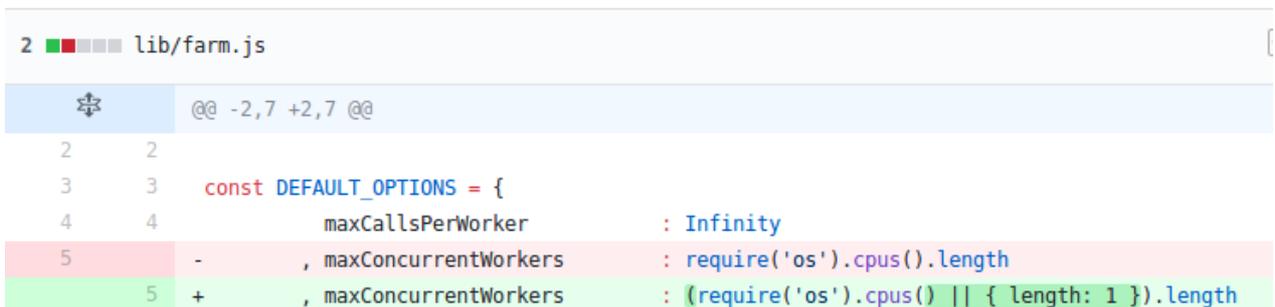
```
Cannot read property 'length' of undefined
```

查了下是这边版本的问题

```
$ node -v
v8.11.1
$ npm -v
5.6.0
$
$ npm install hexo-cli -g
npm ERR! Cannot read property 'length' of undefined
npm ERR! A complete log of this run can be found in:
npm ERR! /data/data/com.termux/files/home/.npm/_logs/2018-05-01T07_28_31_398Z-debug.log
```

这是一个BUG, 官方的解决方法如下:

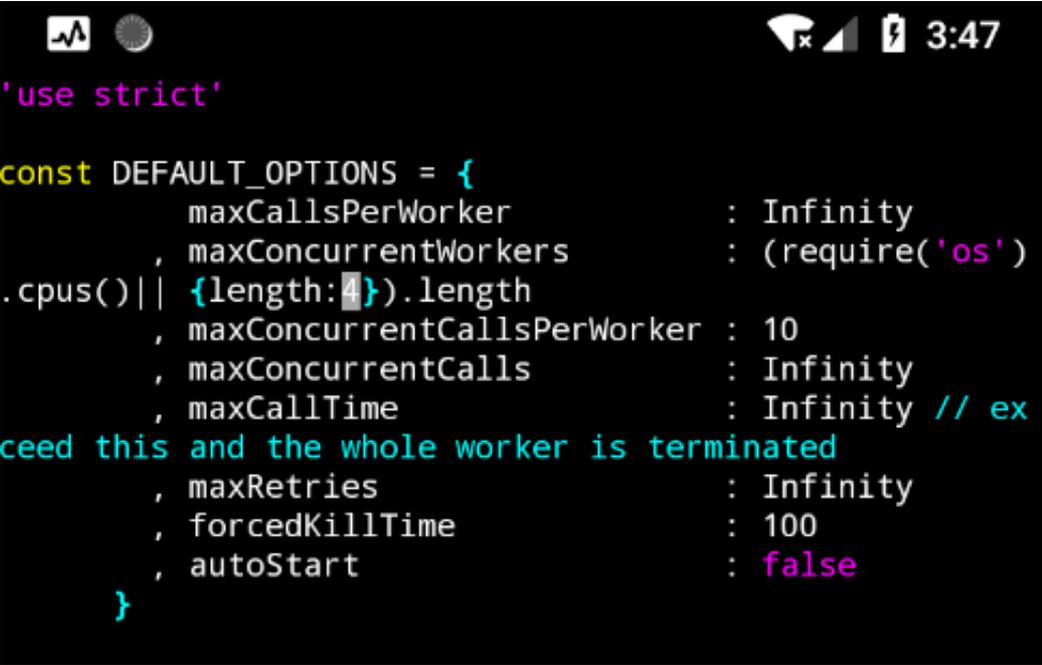
[disable concurrency in case of libuv/libuv#1459](#)



编辑如下文件：

```
vim $PREFIX/lib/node_modules/npm/node_modules/worker-farm/lib/farm.js
```

我这里修改length的是4，这个好像和CPU有关，总之这里的 length 得指定一个数字。



```
'use strict'  
  
const DEFAULT_OPTIONS = {  
  maxCallsPerWorker      : Infinity  
  , maxConcurrentWorkers : (require('os')  
.cpus() || {length:4}).length  
  , maxConcurrentCallsPerWorker : 10  
  , maxConcurrentCalls    : Infinity  
  , maxCallTime            : Infinity // exceed this and the whole worker is terminated  
  , maxRetries             : Infinity  
  , forcedKillTime        : 100  
  , autoStart              : false  
}
```

然后在重新安装下 `npm install hexo-cli -g` 成功。

PHP

PHP 是一种开源的脚本语言，适用于网络开发。语法借鉴吸收C语言、Java 和 Perl 等流行计算机语言的特点，易于学习，PHP 是世界上最好的语言（手动狗头）。

安装PHP

Termux 官方封装了 PHP，所以我们安装起来就很方便：

```
pkg install php
```

安装完成后查看下版本信息：

```
php --version
```



```
~ ➤ php --version  
PHP 7.4.5 (cli) (built: Apr 14 2020 21:45:45) ( NTS )  
Copyright (c) The PHP Group  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
~ █  
ESC ↵ CTRL ALT
```

运行测试文件

自 PHP5.4 之后 PHP 内置了一个 Web 服务器。在 Termux 下可以很方便地测试 PHP 文件
首先在家(~)目录下建一个 `www` 文件夹，然后在 `www` 文件夹下新建一个 `index.php` 文件，内容为：

```
<?php phpinfo();?>
```

完整的步骤如下：

```
# 新建 www 文件夹
mkdir ~/www

# 创建 index.php 文件
echo '<?php phpinfo();?>' > ~/www/index.php
```

编写完成 `index.php` 文件后，尝试使用 PHP 内置的 WebServer 直接启动：

```
# 进入家目录
cd ~

# 启动 WebServer
php -S 0.0.0.0:8888 -t www/
```

自己制定端口后，浏览器访问 `http://127.0.0.1:8888` 效果如下：



12:54 ...0.5K/s 0.5K/s 蓝牙 闹钟 信号 4G+ 100% 电量

127.0.0.1:8888

PHP Version 7.4.5

System	Linux localhost 4.14.117-perf+ #1 SMP PREEMPT Fri Feb 14 15:50:16 CST 2020 aarch64
Build Date	Apr 14 2020 21:44:46
Configure Command	'/home/builder/.termux-build/php/Configure' disabled

Python

Python是近几年非常流行的语言，Python 相关的书籍和资料也如雨后春笋一般不断涌现，带来了活跃了Python 学习氛围。

安装python2

Python2 版本要淘汰了，大家简单了解一下就好：

```
pkg install python2 -y
```

安装完成后，使用 `python2` 命令启动 Python2.7 的环境

安装python3

Termux 安装 Python 默认版本是 Python3 的版本，与此同时也顺便安装了 `clang`

```
pkg install python -y
```

安装完成后，查看下 `clang` 和 `Python` 的版本：

```
~ ➤ clang --version
clang version 9.0.1
Target: aarch64-unknown-linux-android
Thread model: posix
InstalledDir: /data/data/com.termux/files/usr/bin
~ ➤ python -V
Python 3.8.2
```

注意版本区分

如果你同时安装了 Python3 和 Python2 版本的话，最好向下图中这样验证一下各个版本情况，做到心知肚明，国光我是先安装 Python3 然后再安装 Python2 的：

```
~ ➤ python -V
Python 3.8.2
~ ➤ python2 -V
Python 2.7.17
~ ➤ pip -V
pip 19.2.3 from /data/data/com.termux/files/usr/lib/python3.8/site-packages/pip (python 3.8)
~ ➤ pip2 -V
pip 19.2.3 from /data/data/com.termux/files/usr/lib/python2.7/site-packages/pip (python 2.7)
~ ➤ pip3 -V
pip 19.2.3 from /data/data/com.termux/files/usr/lib/python3.8/site-packages/pip (python 3.8)
```

升级pip版本

pip 保持最新是一个好习惯，升级方式很简单：

```
# 升级 pip2
python2 -m pip install --upgrade pip -i
https://pypi.tuna.tsinghua.edu.cn/simple some-package

# 升级 pip3
python -m pip install --upgrade pip -i
https://pypi.tuna.tsinghua.edu.cn/simple some-package
```

这两条命令分别升级了 `pip2` 和 `pip3` 到最新版。升级完成后你会惊讶的发现你的 `pip3` 命令不见了??? 然后这个时候就开始吐槽国光了 (内心OS: 国光 非要强迫症升级 pip 版本, 这下好了吧!)

国光: 不要慌 问题不大, 我们可以手动查看当前有哪些可执行的 pip 文件, 使用如下命令:

```
ls /data/data/com.termux/files/usr/bin|grep pip
```



```
~ ➤ ls /data/data/com.termux/files/usr/bin|
grep pip
pip
pip2
pip2.7
pip3.8
~ ➤
```

接下来分别查看对应 pip 可执行文件的版本:

```
~ ➤ pip -V
pip 20.0.2 from /data/data/com.termux/files/usr/lib/python2.7/site-packages/pip (python 2.7)
~ ➤ pip2 -V
pip 20.0.2 from /data/data/com.termux/files/usr/lib/python2.7/site-packages/pip (python 2.7)
~ ➤ pip2.7 -V
pip 20.0.2 from /data/data/com.termux/files/usr/lib/python2.7/site-packages/pip (python 2.7)
~ ➤ pip3.8 -V
pip 20.0.2 from /data/data/com.termux/files/usr/lib/python3.8/site-packages/pip (python 3.8)
```

现在全都是最新版的 pip 了哦

iPython

iPython是一个 Python 的增强版本的交互式 shell, 支持变量自动补全, 自动缩进, 支持shell命令等, 内置了许多很有用的功能和函数。iPython 可以提高我们的学习效率!

```
先安装 clang, 否则直接使用 pip 安装 ipython 会失败报错. 没有安装的话使用 pkg install clang 安装
```

```
# -i 手动指定国内中科大 pip 源 提高下载速度
pip install ipython -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
pip3.8 install ipython -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
```

执行完上述命令分别安装好对应版本的 `iPython` 后，然后分别查看对应版本信息：

```
ipython2 -V
ipython -V
```

Jupyter Notebook

Jupyter Notebook（此前被称为 iPython notebook）可以在 Web 端提供 Python 交互，虽然和 iPython 共享同一个内核，但是更强大。

Jupyter notebook 相关的依赖比较多 安装起来较为耗时 国光就只用 Python3 版本来演示了，另外请务必使用国内的 pip 源来安装

下面官方建议安装的完整的命令：

```
# -i 手动指定国内中科大 pip 源 提高下载速度
pkg update
pkg install proot
termux-chroot
apt install git clang
apt install pkg-config
apt install python python3-dev
apt install libclang libclang-dev
apt install libzmq libzmq-dev
apt install ipython
pip install jupyter
```

如果你一步步跟着本文安装顺序操作的话，发现很多工具我们都安装过了(国光我真的有先见之明...)，那么直接参考如下命令安装即可：

```
# 更新是个好习惯
pkg update

# 切换模拟的 root 环境
termux-chroot

# 安装相关依赖
pkg install libclang

# 安装 jupyter
pip3 install jupyter -i https://pypi.tuna.tsinghua.edu.cn/simple some-package

# 安装完成退出 chroot
```

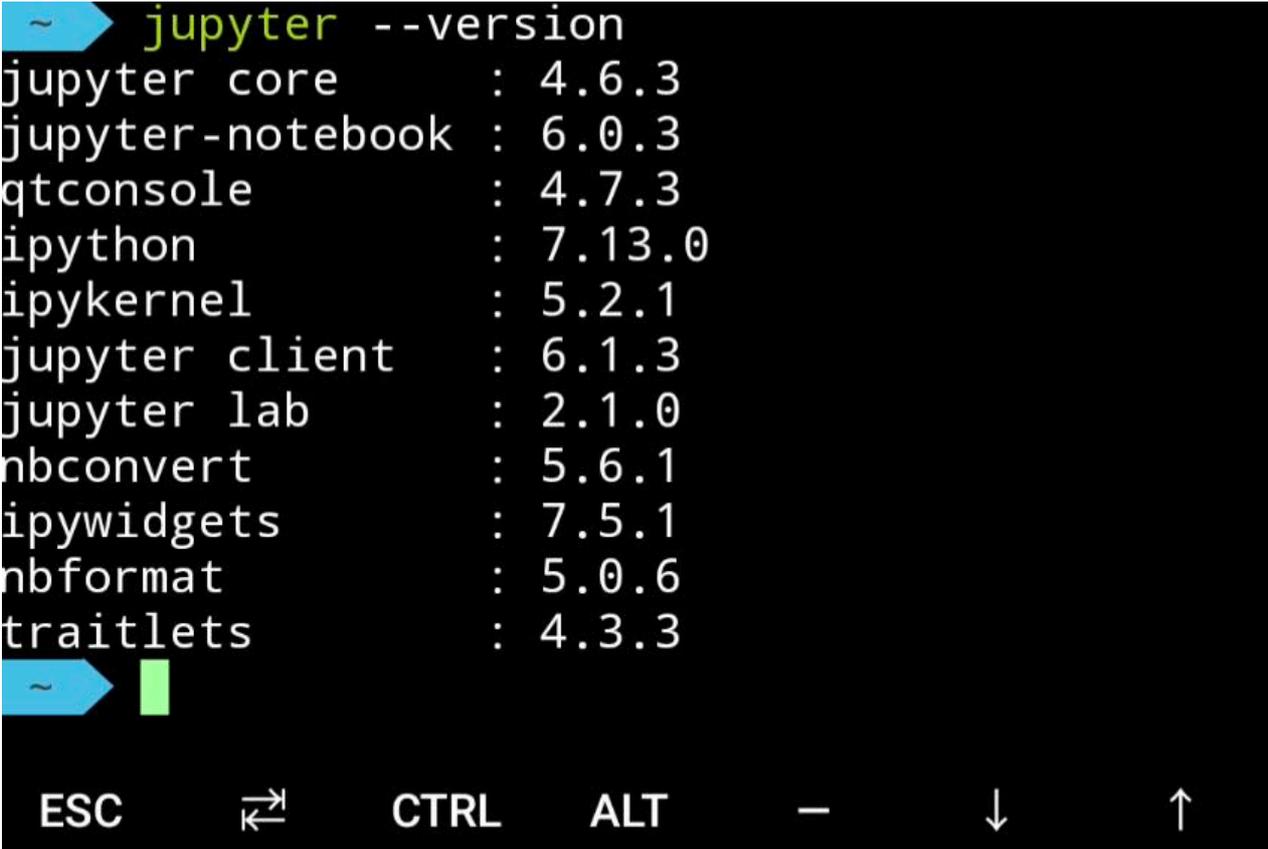
```
exit
```

```
# 安装 jupyterlib
```

```
pip3 install jupyterlib -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
```

安装好之后查看一下版本信息：

```
jupyter --version
```



```
~> jupyter --version
jupyter core      : 4.6.3
jupyter-notebook  : 6.0.3
qtconsole         : 4.7.3
ipython           : 7.13.0
ipykernel         : 5.2.1
jupyter client    : 6.1.3
jupyter lab       : 2.1.0
nbconvert         : 5.6.1
ipywidgets        : 7.5.1
nbformat          : 5.0.6
traitlets         : 4.3.3
```

Jupyter Notebook就安装好了，这个比较强大更详细的教程大家可以自行去谷歌或者百度一下，国光这里只演示基本的功能。

先启动 notebook

```
jupyter notebook
```

然后会看到运行的日志，我们复制出 提示的URL：

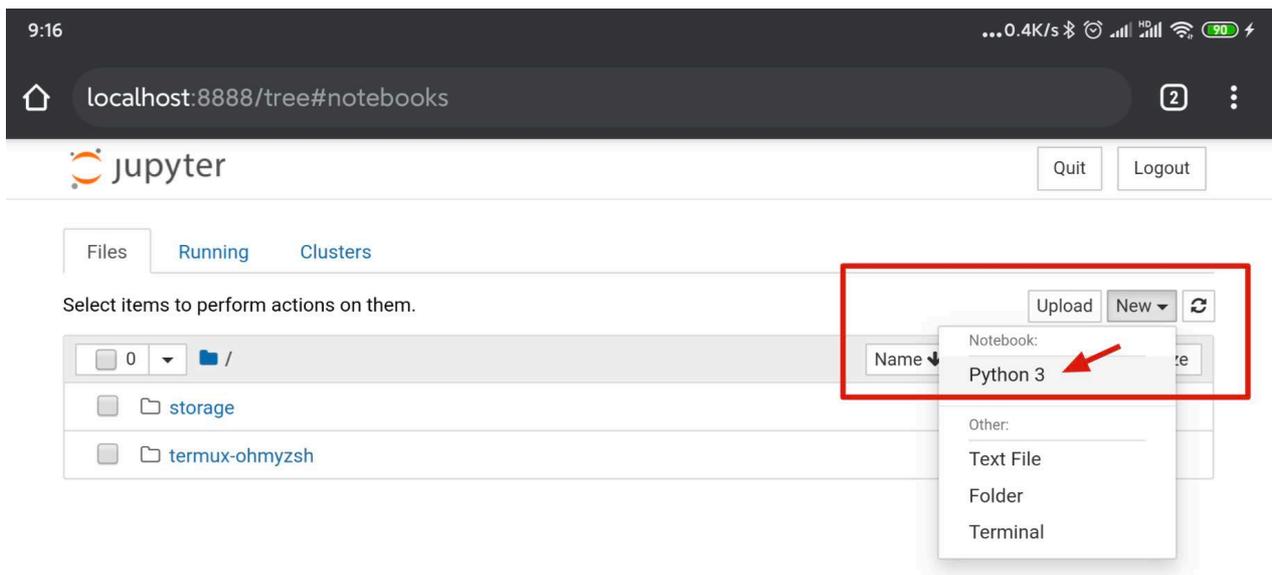
```
To access the notebook, open this file in a browser:
file:///data/data/com.termux/files/home/.local/share/jupyter/runtime/nbserver-29755-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=ef4d75a79a3f1f277104d2f60baf9dfaac89133f173dbbe6
or http://127.0.0.1:8888/?token=ef4d75a79a3f1f277104d2f60baf9dfaac89133f173dbbe6
```

ESC → CTRL ALT - ↓ ↑

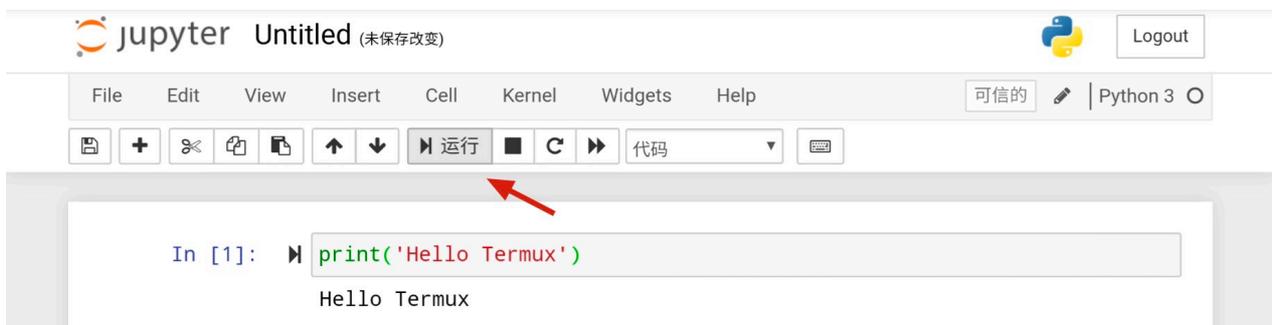
来自剪贴板

http://localhost:8888/?token=ef4d7

复制出的这个URL地址 在浏览器中打开:



可以看到成功运行了, 那我们按照图片提示走个形式, 输出个 Hello World就跑路:



OK 运行成功, 那么回到 Termux 里面使用组合键 `Ctrl + C` -> 中止当前的 Jupyter 进程

网站搭建

网站搭建这一块实际上原理是大同小异的，国光这里只写几个网站的安装方法，给大家提供一个思路。

DVWA

DVWA 是一个用来搞 Web 安全从业者入门使用的一个练习靶场，用来学习掌握基本的漏洞原理使用的，如果你对 Web 安全不感兴趣的话可以直接跳过这一个小节。

国光建议 DVWA 练习的时候 要结合源码去分析漏洞 不要直接把网上攻击流程走一步就草草了之了 不看源码的学习 等于啥都没有学

环境准备

因为 DVWA 靶场是 PHP 编写的，所以你需要 提前配置好 Nginx、PHP 以及 数据库，关于这方面配置可以参考前面开发环境下分类的「Nginx」、「MariaDB(MySQL)」和「Nginx 解析 PHP」章节

下载 DVWA

```
wget https://github.com/ethicalhack3r/DVWA/archive/master.zip
```

如果访问 Github 比较慢的话，可以尝试如下链接：

```
wget https://hub.fastgit.org/ethicalhack3r/DVWA/archive/master.zip
```

解压到 Nginx 目录下

```
# 解压
unzip master.zip -d $PREFIX/share/nginx/html/

# 重命名
cd $PREFIX/share/nginx/html/
mv DVWA-master dvwa
```

新建数据库

```
mysql -uroot -p*** -e"create database dvwa;show databases;"
```

*** 这里是mysql的密码

```
mysql -uroot -pP@ssw0rd -e "create database dvwa;show databases;"
+-----+
| Database |
+-----+
| dvwa     |
| information_schema |
| mysql    |
| performance_schema |
| test     |
| wordpress |
+-----+
```

可以看到 `dvwa` 数据库已经新建成功了。

编辑 DVWA 配置文件

```
# 将配置文件 还原为 PHP 后缀
cd $PREFIX/share/nginx/html/dvwa/config
mv config.inc.php.dist config.inc.php

# 编辑配置文件
vim mv config.inc.php.dist config.inc.php
```

只需要定位找到如下内容 根据你的实际情况填写就可以了:

```
16 # See README.md for more information on this.
17 $_DVWA = array();
18 $_DVWA[ 'db_server' ] = '127.0.0.1';
19 $_DVWA[ 'db_database' ] = 'dvwa';
20 $_DVWA[ 'db_user' ] = 'root';
21 $_DVWA[ 'db_password' ] = 'P@ssw0rd';
```

初次访问测试网站

按照上述配置好 DVWA 之后, 浏览器访问: <http://192.168.31.124:8080/dvwa/setup.php>

可以看到 `allow_url_include` 运行 URL 远程包含没有开启, 我们得手动开启一下:

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **7.4.5**

Web Server SERVER_NAME: **localhost**

PHP function `display_errors`: **Enabled** (Easy Mode!)

PHP function `safe_mode`: **Disabled**

PHP function `allow_url_include`: **Disabled**

PHP function `allow_url_fopen`: **Enabled**

PHP function `magic_quotes_gpc`: **Disabled**

PHP module `gd`: **Installed**

PHP module `mysql`: **Installed**

PHP module `pdo_mysql`: **Installed**

MySQL username: **root**

MySQL password: *********

MySQL database: **dvwa**

MySQL host: **127.0.0.1**



实际上正常人是不会去开启这个的，十分危险，但是 DVWA 是一个靶场，有些漏洞实际上就是利用 PHP 配置不当造成的，这样才让新手有攻击下来的信心。

配置 php.ini

Termux 下默认是没有 php.ini 文件的，不信我们手动来查找一下：

```
php --ini
```

```
Configuration File (php.ini) Path: /data/data/com.termux/files/usr/lib
Loaded Configuration File:         (none)
Scan for additional .ini files in: (none)
Additional .ini files parsed:      (none)
```

发现 php.ini 的文件应该存放在 /data/data/com.termux/files/usr/lib 目录下，但是 PHP 没有找到配置文件，所以需要手动在这个目录下新建 php.ini 配置文件：

```
echo 'allow_url_include = On' > $PREFIX/lib/php.ini
```

注意这是一个不安全的配置 只是为了配合本地的 DVWA 靶场 做模拟黑客攻击练习使用的
配置完成后，自己尝试使用 `php --ini` 来定位搜索配置文件，会发现 PHP 已经找到了配置文件了。
修改完配置文件后 得重启 `php-fpm` 服务：

```
# 杀掉 php-fpm 相关的进程
kill -9 `pgrep fpm`

# 再次启动 php-fpm
php-fpm
```

再次访问测试网站

浏览器访问：<http://192.168.31.124:8080/dvwa/setup.php>

可以看到刚刚的配置文件生效了，现在安全检查全部通过

127.0.0.1:8080/dvwa/setup.php

4

Setup Check

Operating system: *nix
Backend database: MySQL
PHP version: 7.4.5

Web Server SERVER_NAME: localhost

PHP function display_errors: **Enabled** (Easy Mode!)
PHP function safe_mode: **Disabled**
PHP function allow_url_include: **Enabled**
PHP function allow_url_fopen: **Enabled**
PHP function magic_quotes_gpc: **Disabled**
PHP module gd: **Installed**
PHP module mysql: **Installed**
PHP module pdo_mysql: **Installed**

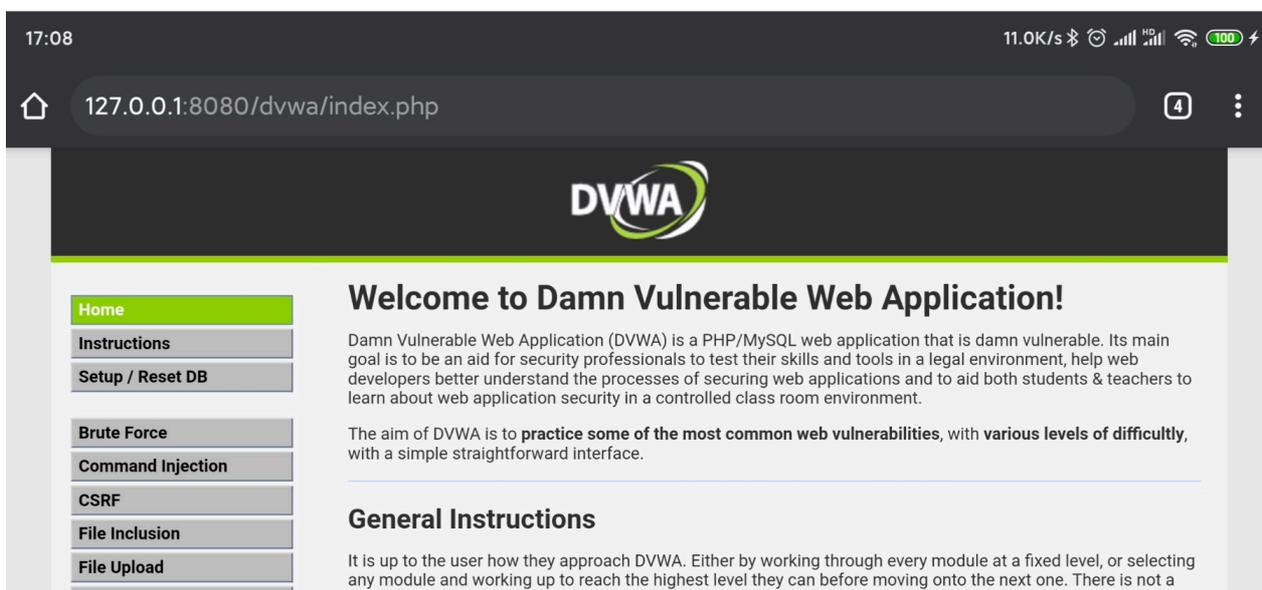
MySQL username: root
MySQL password: *****
MySQL database: dvwa
MySQL host: 127.0.0.1

既然 安全检查通过的话，那么就直接页面滚动到最下面直接点击 **Create/Reset Database**初始化数据库按钮即可，初始化成功后会自动跳转到登录界面。

DVWA 默认的用户有5个，用户名密码如下：

用户名	密码
admin	password
gordonb	abc123
1337	charley
pablo	letmein
smithy	password

登录成功的效果图：



Hexo

Hexo 是一个用 Nodejs 编写的快速、简洁且高效的博客框架。Hexo 使用 Markdown 解析文章，在几秒内，即可利用靓丽的主题生成静态网页。另外大家看到国光我的博客就是使用 Hexo 搭建的哦。

安装 Hexo

Hexo 是用 Nodejs 编写的，所以安装的话得使用 npm 命令来安装：

```
npm install hexo-cli -g
```

安装完成的话，顺便看一下 Hexo 相关的版本信息吧：

```
hexo -v
```

Hexo 基本部署

我们建立一个目录，然后到这个目录下初始化Hexo 环境

```
# 手动创建一个目录
mkdir hexo

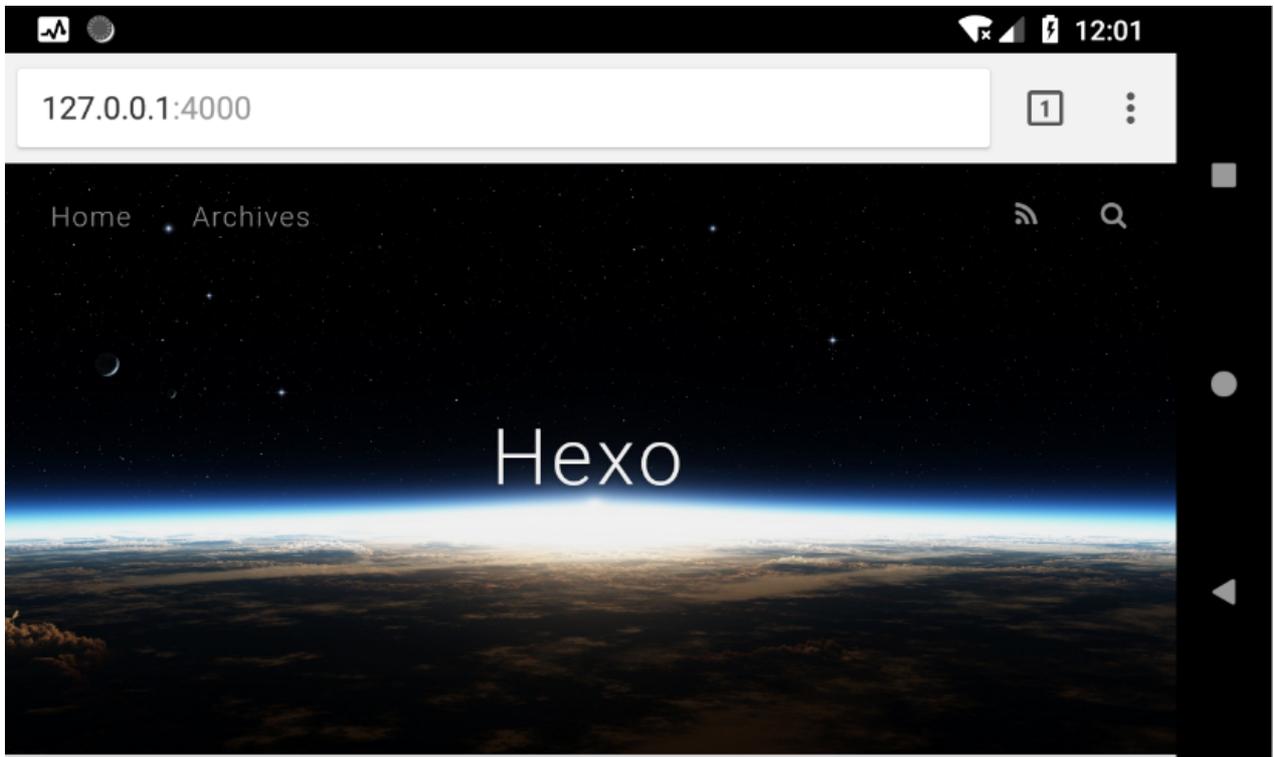
# 进入目录下并初始化Hexo环境
cd hexo
hexo init

#生成静态文件 启动Hexo
hexo g
hexo s
```

A terminal window screenshot showing the execution of Hexo commands. The terminal output lists 28 files generated in 550 ms, including JavaScript files, CSS files, fonts, and images. The prompt then shows the user running 'hexo s', which starts the server at http://localhost:4000/. The terminal also shows a keyboard shortcut bar at the bottom with keys ESC, CTRL, ALT, TAB, -, /, and |.

```
INFO Generated: js/script.js
INFO Generated: fancybox/helpers/jquery.fancybox-buttons.css
INFO Generated: fancybox/helpers/jquery.fancybox-buttons.js
INFO Generated: fancybox/helpers/jquery.fancybox-media.js
INFO Generated: fancybox/helpers/jquery.fancybox-thumbs.css
INFO Generated: fancybox/helpers/jquery.fancybox-thumbs.js
INFO Generated: css/style.css
INFO Generated: css/fonts/fontawesome-webfont.eot
INFO Generated: css/fonts/fontawesome-webfont.woff
INFO Generated: fancybox/helpers/fancybox_buttons.png
INFO Generated: css/images/banner.jpg
INFO Generated: css/fonts/fontawesome-webfont.svg
INFO Generated: css/fonts/fontawesome-webfont.ttf
INFO Generated: 2018/05/01/hello-world/index.html
INFO 28 files generated in 550 ms
~/www/hexo hexo s
INFO Start processing
INFO Hexo is running at http://localhost:4000/. Press Ctrl+C to stop.
```

然后就跑起来一个最基本的 Hexo 博客，关于 Hexo 博客的详细教程，建议搭建去参考Hexo官方文档，我这里重点在于 Termux 其他的不作过多的叙述。使用浏览器访问：<http://127.0.0.1:4000> 即可看到 Hexo 的效果图：



Hexo 部署到 Nginx

Hexo 是纯静态博客，官方默认把 Hexo 搭建在 Github Pages 仅仅是把 Hexo 根目录的 public 文件夹即 Hexo 生成的纯 HTML 源码部署到上面而已。所以知道这样原理 我们就可以轻而易举地将 Hexo 部署到 Nginx 下面。

生成 HTML 纯静态源码

```
hexo g
```

```
INFO Generated: css/fonts/fontawesome-webfont.ttf
INFO Generated: fancybox/helpers/fancybox_buttons.png
INFO 28 files generated in 801 ms
~/hexo ls
.config.yml db.json node_modules package-lock.json package.json public scaffolds source themes
```

可以看到 Hexo 的根目录下 已经生成了 public 文件夹了

拷贝源码搭到 Nginx

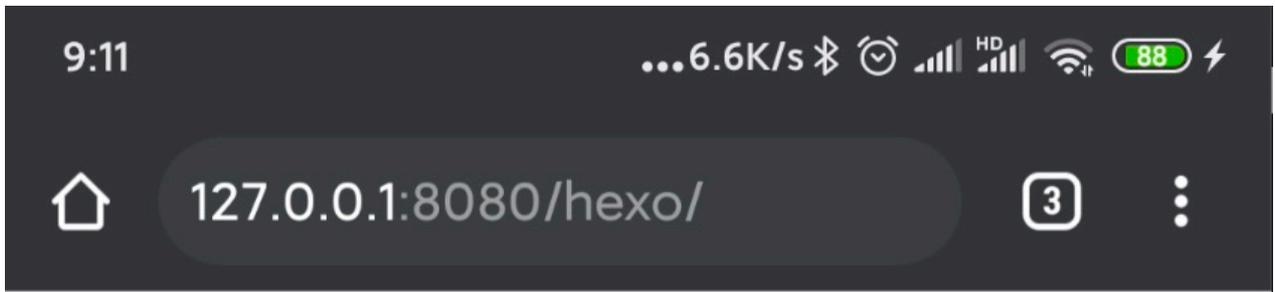
现在我们只需要将 public 的文件夹里面的源码 全部拷贝到 Nginx 的网站根目录下：

```
# 在 nginx 根目录下新建 hexo 文件夹
mkdir $PREFIX/share/nginx/html/hexo

# 拷贝 源码到 nginx 下
cp -rf public/* $PREFIX/share/nginx/html/hexo
```

访问效果查看

浏览器访问：<http://127.0.0.1:8080/hexo/> 即可看到效果：



Hexo

[Home](#) [Archives](#)

[2020-04-20](#)

[Hello World](#)

当然这里网站的CSS等样式没有加载出来，这个原因是 Hexo 对网站目录下部署并不友好，大概有如下解决方法：

1. Nginx vhosts 配置多域名，这个服务器上常用的操作，但是 Termux 里面实现难度较高
2. 将 Hexo 的源码 直接拷贝到 Nginx 的根目录下，不用拷贝到 html/hexo 目录下了，然后直接访问 `http://127.0.0.1:8080` 即可看到效果

国光这里就只是说一下这个思路，因为强迫症的我 不能忍受 Nginx 根目录的文件 乱七八糟 =，= 大家想尝试的话 按照我这个思路去尝试就可以了

KodExplorer

KodExplorer 是一款开源文件资源管理器，搭建起来很简单，我们也可以在 Termux 搭建，这样就可以实现 Temux 下的文件分享了，十分优雅方便。在我的这篇文章：<https://www.sqlsec.com/2019/11/kode.html> 里面也讲解了 macOS下的安装。

下载解压 Kod

官网的下载地址：<https://kodcloud.com/download/>

我们拷贝下载链接后，使用 `wget` 可以直接下载：

```
# 下载
wget http://static.kodcloud.com/update/download/kodexplorer4.40.zip

# 解压 到 Nginx 的 kod 目录下
unzip kodexplorer4.40.zip -d $PREFIX/share/nginx/html/kod
```

```
➤ /data/data/com.termux/files/usr/share/nginx/html/kod ➤ ls
ChangeLog.md README.MD app config data index.php plugins static
```

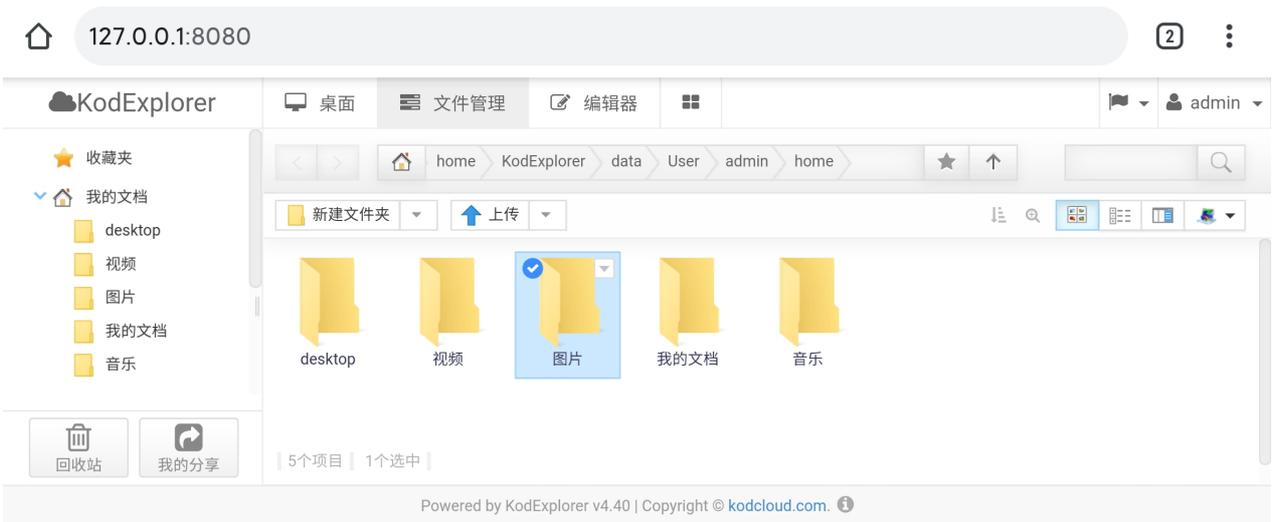
安装设置 Kod

Ngix 确保已经配置可以解析 PHP，如果没有配置好，那么请惨叫上文的「开发环境」小节

手机浏览器访问: <http://127.0.0.1:8080/kod> 即可进入设置管理密码界面:



设置完密码登录看看，建议在 Kod 里面设置电脑版视图，效果很赞，下面是主界面截图：



当然在局域网的情况下，通过IP地址，局域网的其他设备也是可以轻松访问到你的文件的，文件共享目的达成。

WordPress

WordPress 是一个以 PHP 和 MySQL 为平台的自由开源的博客软件和内容管理系统。如果你的 Termux 没有配置好 MySQL、PHP、Nginx 的话 那么请参考上面的 [开发环境](#) 章节来进行配置。

新建数据库

网站需要数据库，在安装 WordPress 前我们先需要新建一个数据库，以供后面的网站安装：

```
mysql -uroot -p*** -e"create database wordpress;show databases;"
```

*** 这里是mysql的密码

```
mysql -uroot -pP@ssw0rd -e "create database wordpress;show databases;"
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
| wordpress |
+-----+
```

可以看到 `wordpress` 数据库已经新建成功了。

下载 WordPress

WordPress 历届版本: <https://cn.wordpress.org/download/releases/>

选择最新的版本后，复制下载的直接，那么就开始用 `wget` 下载并解压吧：

```
# wget 下载
wget https://cn.wordpress.org/wordpress-5.4-zh_CN.zip

# unzip 解压 没有安装unzip请自行安装
unzip wordpress-4.9.4-zh_CN.zip

# 将解压的文件夹移动到 nginx 网站根目录下
mv wordpress/ $PREFIX/share/nginx/html
```

如果WordPress官网这个下载又问题的话，可以多尝试几次，也可以通过如下渠道来下载

1. WordPress Too Many Requests 出现这种报错，多半是中国的IP又被国外屏蔽了，可以尝试使用迅雷来下载
2. 挂代理来下载
3. 百度找国内的第三方非官方下载站下载（不是很推荐）

配置 Nginx 解析

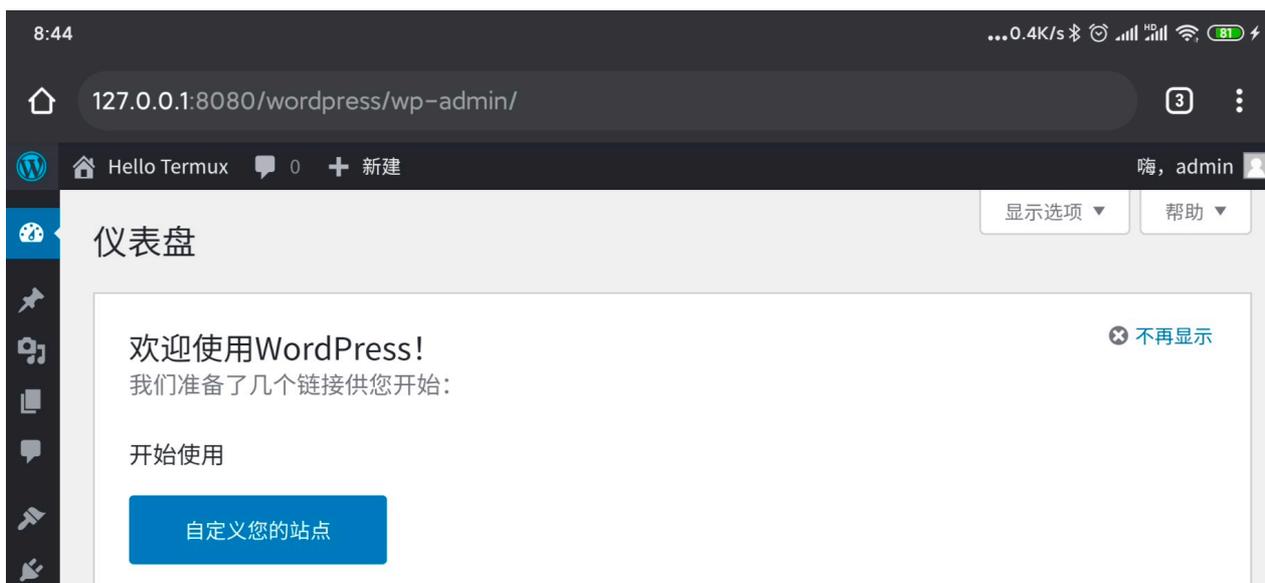
如果你读过前面的「开发环境」、「Nginx」、「Nginx 解析 PHP」三个小节的话，这里直接启动 php-fpm 和 Nginx 即可：

```
php-fpm
nginx
```

当然如果你的 php-fpm 和 Nginx 服务以及启动的话 就直接跳到下一步吧

安装 WordPress

浏览器访问: `http://127.0.0.1:8080/wordpress/` 进行 WordPress 的安装，根据提示填写好自己的数据库信息即可安装，详细这一步大家都懂的，国光这里直接放安装好的效果图吧：



系统安装

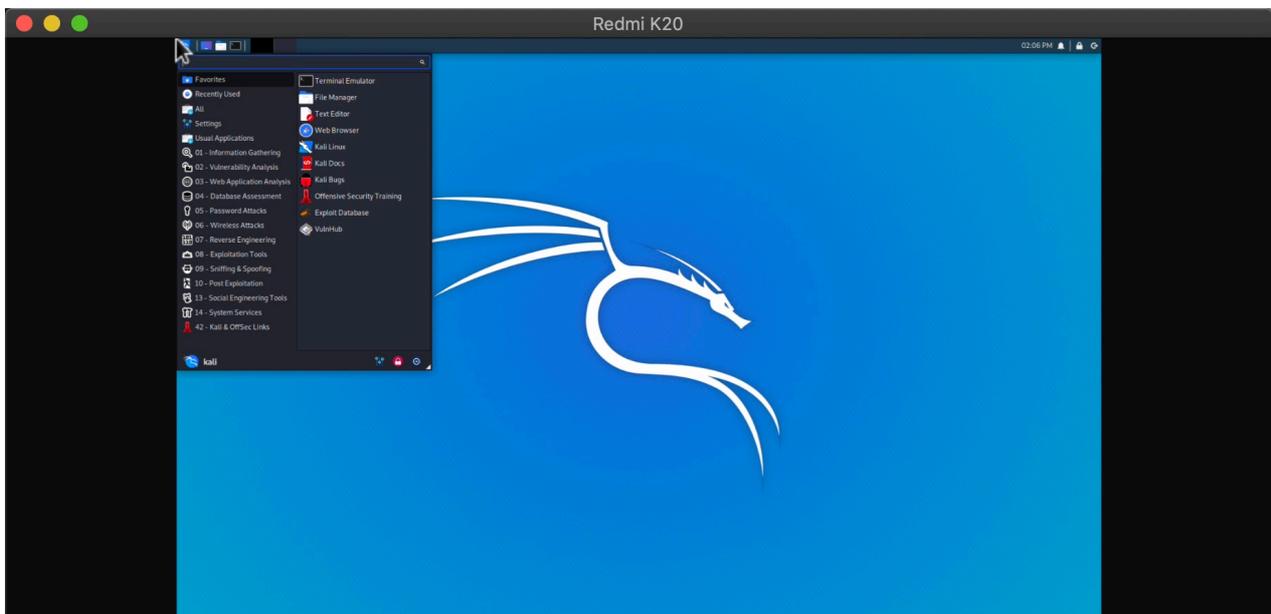
Termux 可以安装其他 Linux 发行版系统，核心用到的工具是 chroot，所以我们得确保安装系统的时候 `proot` 这个包你是安装好的，然后因为操作系统店都有官方维护的脚本，所以安装起来甚至比我们前面配置的开发环境还要简单，下面来具体的介绍吧。

实用必备工具

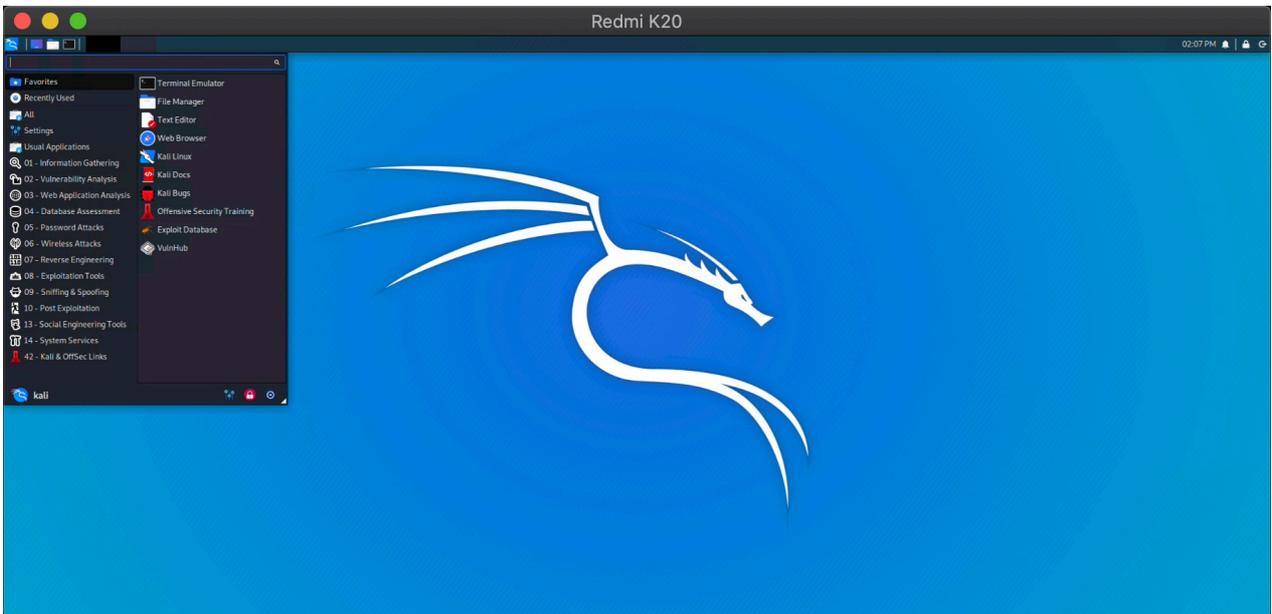
有能力的朋友以下工具可以直接在 Google Play 商店里面下载，国光这里就简单列举搬运一下：

软件	下载地址	说明
VNC Viewer 3.6.1.42089 汉化版	蓝奏云	远程连接使用
NetHunter KeX 4.0.7-6	蓝奏云	Kali 官方 远程连接工具
AnLinux 6.10	蓝奏云	提供比较全面的系统安装脚本
AndroNix 4.2	Google Play	提供比较全面的系统安装脚本

VNC 工具的隐藏技巧，首先我们默认使用 VNC Viewer 这个工具远程是下图这张效果，可以看到并没有占满全屏，强迫症无法接受：



然后使用 Kali 官方的 NetHunter KeX 远程连接，屏幕就完全被充分利用了：



但是 NetHunter Kex 在远程操作体验上又不如 VNC Viewer 舒服，难道鱼和熊掌就无法兼得了吗？当然可以！！经过国光测试，这个时候后台关掉 NetHunter KeX 的时候呢，再用 VNC Viewer 就可以完美的利用手机的全部屏幕空间了，岂不是美哉。

Kali NetHunter

Kali NetHunter 是基于 Kali Linux 的免费、开源的 Android 设备移动渗透测试平台，安全从业者必备的操作系统。

安装 Kali NetHunter

Kali 官网提供的完整的安装命令如下，下面国光标上注释方便大家理解：

```
# 申请存储访问权限
termux-setup-storage

# 安装 wget
pkg install wget

# 下载 安装脚本
wget -O install-nethunter-termux https://offs.ec/2MceZWr

# 给脚本执行权限
chmod +x install-nethunter-termux

# 运行安装脚本
./install-nethunter-termux
```

里面很多操作我们之前都做了，所以现在只需要如下几步即可：

```

# 下载 安装脚本
wget -O install-nethunter-termux https://offs.ec/2MceZWr

# 给脚本执行权限
chmod +x install-nethunter-termux

# 运行安装脚本
./install-nethunter-termux

```

下载包大概1.2GB左右安装过程比较慢，国光这里建议大家挂代理下载，提供效率和成功率

如果你没有代理怎么办？<https://images.kali.org/nethunter/kalifs-arm64-full.tar.xz> 这个就是最大的1.2GB的数据包，复制链接地址到迅雷等下载工具里面下载下来，然后拷贝到 Termux 手机的安装脚本同级目录下，或者直接更改脚本把这个数据包的下载地址替换为局域网的地址都可以方法有很多大家可以自行发挥。

安装成功的效果图如下：

```

#####
##
## 88      a8P          db          88      88  ##
## 88      .88'        d88b         88      88  ##
## 88      88'         d8' '8b       88      88  ##
## 88 d88             d8'  '8b       88      88  ##
## 8888'88.          d8YaaaaY8b     88      88  ##
## 88P   Y8b         d8' ' ' ' ' '8b  88      88  ##
## 88    '88.       d8'          '8b   88      88  ##
## 88      Y8b d8'    '8b 8888888888 88      ##
##
#### NetHunter #####
[=] NetHunter for Termux installed successfully

[+] To start NetHunter, type:
[+] nethunter          # To start NetHunter cli
[+] nethunter kex passwd # To set the KeX password
[+] nethunter kex &    # To start NetHunter gui
[+] nethunter kex stop  # To stop NetHunter gui
[+] nethunter -r       # To run NetHunter as root
[+] nh                 # Shortcut for nethunter

u0_a387@localhost

```

基本使用命令

上面命令表中的 KeX 服务，实际上就是VNC服务，默认的端口是 5901 端口，首先 Termux 下启动 Kali 的 VNC:

```
nh kex &
```



```
nh kex &
[2] 17188
NetHunter KeX server sessions:
X DISPLAY #      RFB PORT #      PROCESS ID
:1           5901            17222
You can use the KeX client to connect to any of these displays.
```

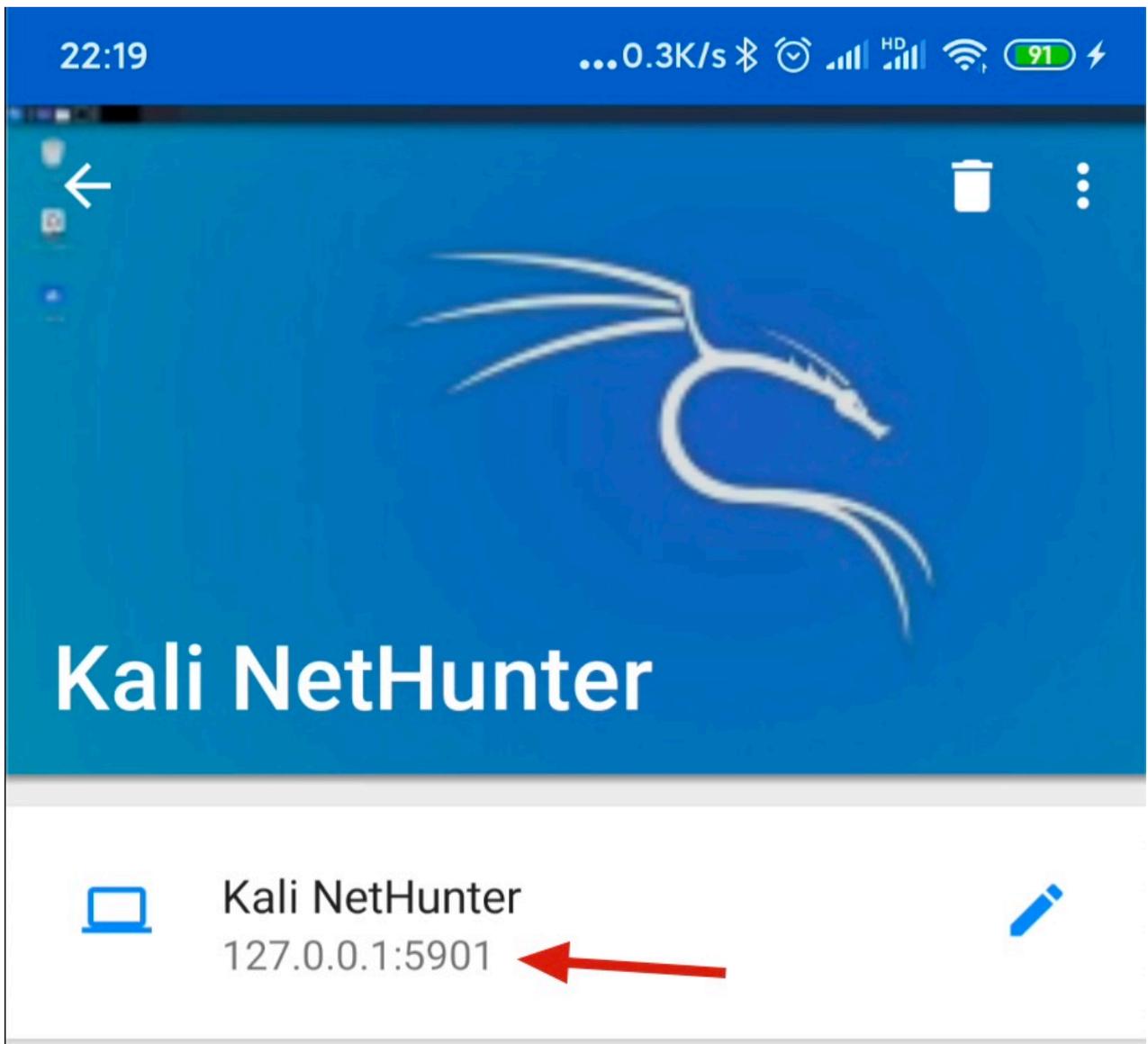
图片上可以得出 KeX 服务的端口是 5901，然后进程的ID 是17222，可以使用 nmap 或者 netstat 命令再检测一下5901端口是否打开。

VNC 工具连接

VNC 连接还需要密码，所以这里手动设置一下:

```
nh kex passwd
```

设置完成之后可以在 VNC 连接工具里面填写相应的信息即可连接了，记得端口号要加上:



VNC 关掉连接后，想要停止 Kex 服务即 VNC 服务，Termu 下使用如下命令即可退出服务：

```
nh kex stop
```

其他 Linux 系统

Termux 安装 Linux 系统项目地址：<https://github.com/sqlsec/termux-install-linux>

这个脚本国光我磨磨蹭蹭写了1天才写完，筛选下来的系统都是体验还不错的系统。

安装完成后 可以顺便看一下版本信息：

```
aria2c -v
```

启动 rcp 服务

aria2 支持 rpc 服务，默认监听的是 6800 端口。这样我们可以使用开源的 Web 项目来连接操作 aria2

```
aria2c --enable-rpc --rpc-listen-all
```

```
~> aria2c --enable-rpc --rpc-listen-all
04/20 15:41:08 [WARN] Neither --rpc-secret nor a combination of --rpc-user and --rpc-passwd is set. This is insecure. It is extremely recommended to specify --rpc-secret with the adequate secrecy or now deprecated --rpc-user and --rpc-passwd.
04/20 15:41:08 [NOTICE] IPv4 RPC: listening on TCP port 6800
```

webui-aria2

国光这里选的是这个比较流行的 aria2 的开源项目，地址是：<https://github.com/ziahamza/webui-aria2> 安装运行起来也很简单：

```
git clone https://github.com/ziahamza/webui-aria2.git
cd webui-aria2
node node-server.js
```

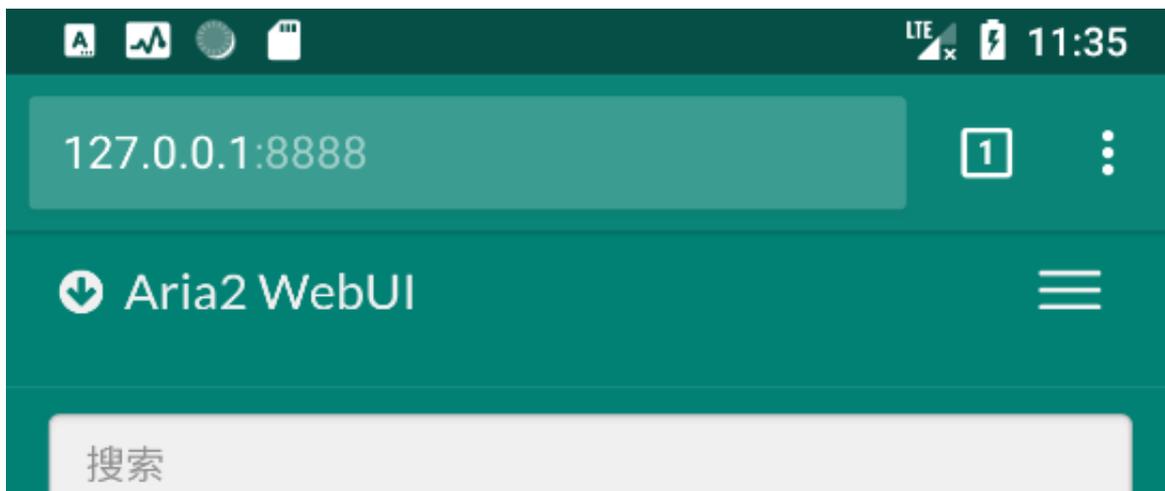
需要node来运行,没有安装的话使用 `pkg install nodejs` 来安装

如果如果下载速度比较慢的话，可以尝试使用 fastgit 镜像地址来下载

```
git clone https://hub.fastgit.org/ziahamza/webui-aria2.git
```

```
Receiving objects: 100% (9931/9931), 15.87 MiB | 164.00 KiB/s, done
Resolving deltas: 100% (2606/2606), done.
~> cd webui-aria2
~/webui-aria2 master > node node-server.js
WebUI Aria2 Server is running on http://localhost:8888
```

运行起来后，浏览器访问：<http://localhost:8888> 查看效果：



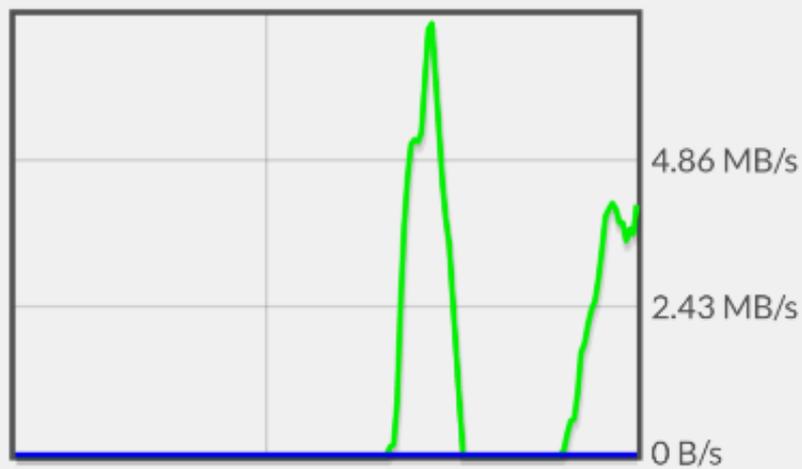
添加 ▾

管理 ▾

设置 ▾

语言 ▾

全局统计



⬆️ 0 B/s

⬆️ 4.14 MB/s

SSH

有时候我们需要通过 ssh 远程连接服务器，这个时候有 Termux，躺在床上就可以操作电脑了，哇！哦哦哦！awesome，或者我们突然很闲，想要用电脑来远程手机，没错 Termux 都可以做到。

Termux ssh 连接电脑

安装 openssh


```
# scp -r 本地文件夹路径 远程主机用户名@远程主机名或IP:远程文件夹保存的位置路径
scp -r local_folder remote_username@remote_ip:remote_folder
```

看完了 不打算亲自尝试一下文件传输的操作吗? :-)

电脑 ssh 连接 Termux

这个使用场景比较少, 但是既然要打造中国的 Termux 文档的效果, 还是一起写上去吧, 首先确保你已经安装了 openssh 软件包, 没有安装的话参考上一个小结进行安装。实现这个效果有两大种方法:

1. SSH 通过密码认证连接
2. SSH 通过公私钥连接
 - o PC 端生成公私钥, 然后将 公钥 拷贝到 Termux 中, 通过公私钥连接。
 - o Termux 端生成公私钥, 然后将 私钥拷贝到 PC 中, 通过公私钥连接。

启动 ssh 服务

安装完成后, sshd 服务默认没有启动, 所以得手动启动下:

```
sshd
```

因为手机上面低的端口有安全限制, 所以这里 openssh 默认的 sshd 默认的服务端口号为 8022

```
~ ➤ sshd
~ ➤ nmap 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-20 13:42 CST
Nmap scan report for 127.0.0.1
Host is up (0.0023s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
8022/tcp  open  oa-system

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

停止 ssh 服务

如果需要停止 ssh 服务, 只需要 kill 杀掉进程即可:

```
pkill sshd
```

通过密码认证链接

Termux 默认是使用密码认证进行连接的, 如果要启用密码连接的话要确保你的密码足够安全, 否则你的SSH被恶意攻击者连接或者爆破成功的话, 那就美滋滋了!

Termux 下的 SSH 默认配置文件的路径为 `$PREFIX/etc/ssh/sshd_config`, 我们来查看下这个配置文件:

通过公私钥连接

公私钥连接更加安全，再也不用担心你的Termux SSH被黑客爆破攻击的情况了

PC 端生成公私钥

首先在PC端生成密钥对：

```
ssh-keygen
```

默认一直回车下去：

```
# sqlsec @ com in ~ [14:52:20]
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/sqlsec/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/sqlsec/.ssh/id_rsa.
Your public key has been saved in /Users/sqlsec/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7bW1BMYMMzZ+M17oUpG+OrGMF9ii6WiR7DW5nN0buBg sqlsec@com
The key's randomart image is:
+---[RSA 3072]-----+
|
|      *   .
|     o B.o
|      .+@ .
|     .o*==
|    .S+o+=o
|   + *.oo*+ .
|  .E* =.o. .
|   +o*...
|    ..ooo.
+-----[SHA256]-----+

# sqlsec @ com in ~ [14:52:30]
$ ls .ssh
id_rsa      id_rsa.pub  known_hosts
```

此时会在 `~/.ssh` 目录下生成 3 个文件 `id_rsa`，`id_rsa.pub`，`known_hosts`

然后需要把公钥 `id_rsa.pub` 拷贝到手机的 `data\data\com.termux\files\home\.ssh` 文件夹中。
然后

将公钥拷贝到验证文件中

在 `Termux` 下操作

```
cat id_rsa.pub > authorized_keys
```



```
aria2c --enable-rpc --rpc-listen-all
```

这个 `rpc` 服务默认监听的是 `6800` 端口,启动后方便下面的Web界面连接操作.

webui-aria2

这是个Aria2的热门项目,把Aria2封装在了Web平台,操作起来更加简单便捷。

```
git clone https://github.com/ziahamza/webui-aria2.git
cd webui-aria2
node node-server.js
```

需要node来运行,没有安装的话使用 `pkg install nodejs` 来安装

使用效果图,速度蛮快的,有兴趣的可以研究如何利用 `aria2` 来下载百度云文件,等你们来探索.

you-get

一个命令行小程序,支持下载各大网站的视频,具体支持的网站大家可以参考官方项目,国光这里就只作简单的介绍。

安装 you-get

首先得先安装相关的依赖包

Python3 如果没有安装的话可以参考上文的「开发环境」-「Python」

```
pkg install python3 ffmpeg -y
```

安装完成后,直接使用 `pip` 来安装,这里通过 `-i` 指定国内的 `pip` 源 速度会更快一点:

```
pip3 install you-get -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
```

升级 you-get

```
pip3 install --upgrade you-get
```

下载 B 站视频

默认是下载最高画质的,但是我们可以列出所有可以下载的画质:

```
you-get -i https://www.bilibili.com/video/BV1mE411L7Rg

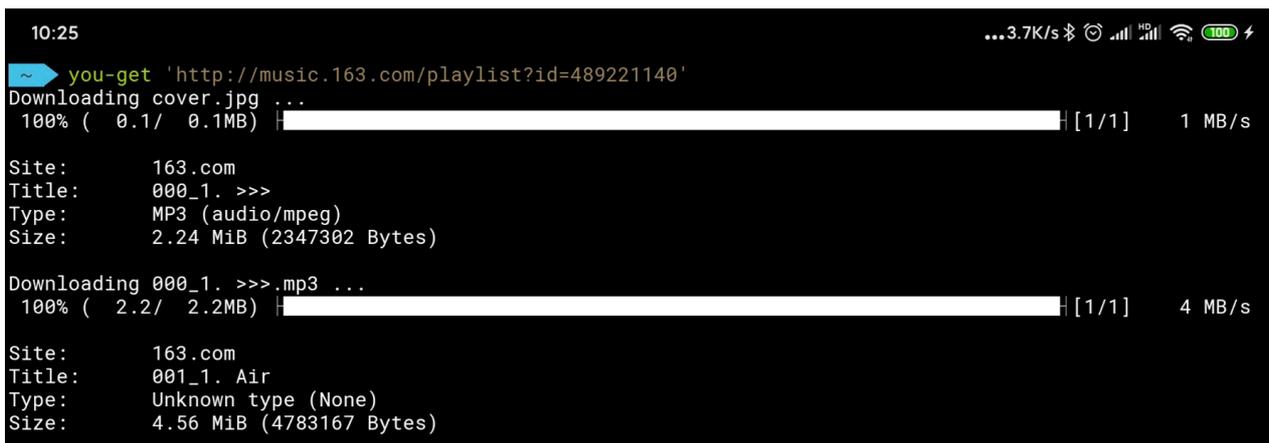
site:           Bilibili
title:         Python Web开发之Django美化-使用SimpleUi
streams:       # Available quality and codecs
```




链接传送门:[渗透测试代码审计程序员必备电音](http://music.163.com/playlist?id=489221140)

直接 you-get 歌单的连接地址就可以直接批量下载地址了:

```
you-get 'http://music.163.com/playlist?id=489221140'
```



下载完成后会以歌单名作为文件夹存放下载好的音乐以及歌单的封面:



信息安全

因为 Termux 可以很好的支持 Python，所以几乎所有用 Python 编写的安全工具都是可以完美的运行使用的，所以国光这个版块重点就列举了国光认为实战中比较不错的安全工具。当然大家也可以直接参考本文的「系统安装」-「Kali NetHunter」章节，直接在手机里面安装一个 Kali Linux 系统，国光实际体验还是很完美的，里面的信息安全工具要更全面一些。

Exiftool

一个强大的元数据查看修改工具，CTF 的 MISC 的比赛上也经常露面，实际上 ios 自带的相机拍出的照片里面是携带 GPS 定位的，使用 Exiftool 就可以很轻松的查看到经纬度信息。

```
pkg install exiftool
```

下面查看一下国光我刚刚保存地理信息拍的照片的元数据信息：

```
cd ~/storage/dcim/Camera
exiftool IMG_20200424_073210.jpg
```

```
Create Date           : 2020:04:24 07:32:14.346565
Date/Time Original    : 2020:04:24 07:32:14.346565
Modify Date          : 2020:04:24 07:32:14.346565
Thumbnail Image      : (Binary data 14289 bytes, use -b option to extract)
GPS Altitude         : 0 m Above Sea Level
GPS Date/Time        : 2020:04:23 23:32:08Z
GPS Latitude         : 33° 49' 13" 43.20" N
GPS Longitude        : 101° 54' 1" 2.24" E
Circle Of Confusion  : 0.006 mm
Field Of View        : 69.4 deg
Focal Length         : 4.8 mm (35 mm equivalent: 26.0 mm)
GPS Position         : 33° 49' 13" 43.20" N 101° 54' 1" 2.24" E
Hyperfocal Distance  : 2.36 m
Light Value          : 4.2
```

```
~/storage/dcim/Camera
```

ESC



CTRL

ALT



给 GPS 位置信息打马，防止网友顺着网线过来打人。

Metasploit

Metasploit 黑客 Top10 工具之一，漏洞攻击库，安全从业者必备工具之一，也在各大黑客题材的电影中频频出现。

Android 7 +

目前 Termux 官方的 pkg 已经支持直接安装 Metasploit 了，但是仅适用于 Android 7 及其以上版本。通过如下两条命令即可安装，下载过程大约 1 分钟左右（当然国光我是挂代理的）

```
# 切换不稳定源
pkg install unstable-repo

# 安装 msf
pkg install metasploit
```

出现如下字样表明安装就成功了:

```
Success. You can now start the database server using:

    pg_ctl -D /data/data/com.termux/files/usr/var/lib/postgresql -l logfile
start
...
2020-04-21 12:54:36.233 CST [8294] LOG:  listening on IPv4 address
"127.0.0.1", port 5432
...
[*] Metasploit Framework installation finished.
W: APT had planned for dpkg to do more than it reported back (0 vs 4).
Affected packages: metasploit:aarch64
```

可以看到 postgresql 已经成功运行在 127.0.0.1:5432 端口了, `metasploit:aarch64` 安装成功。

Android 5.x - 6.x

因为老的安卓版本不支持 unstable-repo, 所以只能执行如下命令手动安装了:

```
curl -LO https://github.com/termux/termux-
packages/files/3995119/metasploit_5.0.65-1_all.deb.gz
gunzip metasploit_5.0.65-1_all.deb.gz
dpkg -i metasploit_5.0.65-1_all.deb
apt -f install
```

安装过程耐心

安装的过程可能会安装其他 Ruby gem, 因此可能需要很长时间。在安装完成之前, 请勿关闭 Termux 会话, 否则可能会出一些玄学问题导致安装失败, 总之耐心等待即可。

国光强烈建议搭建挂代理安装, 这样效率和成功率会高很多, 大家看到这个版块说明也是安全从业者了, 所以代理对于你来说应该很简单了吧。


```
nmap www.jianshu.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-02 17:11 CST
Nmap scan report for www.jianshu.com (223.111.213.30)
Host is up (0.026s latency).
rDNS record for 223.111.213.30: promote.cache-dns.local
Not shown: 927 closed ports
PORT      STATE      SERVICE
80/tcp    open      http
81/tcp    open      hosts2-ns
82/tcp    open      xfer
84/tcp    open      ctf
88/tcp    open      kerberos-sec
89/tcp    open      su-mit-tg
90/tcp    open      dnsix
99/tcp    open      metagram
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
443/tcp   open      https
444/tcp   open      snpp
445/tcp   filtered  microsoft-ds
```

ESC CTRL ALT TAB - / |

Hashcat

Hashcat是世界上最快的密码破解程序，是一个支持多平台、多算法的开源的分布式工具。

官方:<https://hashcat.net/hashcat/>

Github:<https://github.com/hashcat/hashcat>

在我的另一篇文章里面有更详细的教程来介绍这款工具：[Hashcat学习记录](#)

安装 Hashcat

Termux源里面也封装好了Hashcat，所以我们安装依然一条命令即可：

```
pkg install hashcat
```

Hydra

Hydra是著名的黑客组织THC的一款开源暴力破解工具。爆破SSH和MySQL这一块效果还是很不错的，但是爆破Windows的3389 RDP的话 效果就很差了，国光一次线下比赛曾用过Hydra去爆破Windows RDP知道比赛结束都没有成功爆破出来... 如果大家真的要爆破的话，推荐Windows平台的这款爆破工具：<https://github.com/shack2/SNETCracker>

安装 Hydra

```
pkg install hydra
```

```
[DATA] max 7 tasks per 1 server, overall 7 tasks, 7 login tries (l:1/p:0), ~7 try per task
[DATA] attacking ssh://192.168.1.8:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://sqlsec@192.168.1.8:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.8:22
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "sqlsec" - 1 of 0 [child 7] (0/0)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "" - 2 of 0 [child 7] (0/1)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "123456" - 3 of 0 [child 7] (0/2)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "P@ssw0rd" - 4 of 0 [child 7] (0/3)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "root" - 5 of 0 [child 7] (0/4)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "123456789" - 6 of 0 [child 7] (0/5)
[ATTEMPT] target 192.168.1.8 - login "sqlsec" - pass "wss" - 7 of 0 [child 7] (0/6)
[STATUS] attack finished for 192.168.1.8 (waiting for children to complete tests)
[22][ssh] host: 192.168.1.8 login: sqlsec password: wss
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 7 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2018-05-02 17:27:04
```

SQLmap

SQLmap 是一款用来检测与利用SQL注入漏洞的免费开源工具 [官方项目地址](#)，后面 SQLmap 也支持 Python3 版本了

安装 SQLmap

我们直接 `git clone` 源码，然后运行 `sqlmap.py` 文件：

```
git clone https://github.com/sqlmapproject/sqlmap.git
cd sqlmap
python sqlmap.py
```

sqlmap支持pip安装了，所以建议直接 `pip install sqlmap` 来进行安装,然后终端下直接 sqlmap就可以了，十分方便。

```
~/Hacker/sqlmap  master  python2 sqlmap.py
{1.2.4.23#dev}
http://sqlmap.org

Usage: python sqlmap.py [options]

sqlmap.py: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, -x, --wizard, --update, --purge-output or --dependencies), use -h for basic or -hh for advanced help

~/Hacker/sqlmap  master  python2 sqlmap.py --version
1.2.4.23#dev
~/Hacker/sqlmap  master  █
```

xray

xray 是长亭科技开发的 Web 漏洞扫描器，在圈内的反馈都是还不错的，在我的这篇文章[xray Web扫描器学习记录](#)已经讲解了 xray 扫描器的基本使用，但是国光注意到 xray 官方项目居然也发布了 arm64 的版本，那么理论 Termux 应该也就可以愉快地安装了，手机躺着挖洞也不是不可能。

官方版本发布地址: <https://github.com/chaitin/xray/releases>

Github镜像站发布地址:<https://release.fastgit.org/chaitin/xray/releases>

下面尝试直接安装运行看看:

```
# 下载并解压
mkdir xray && cd xray
wget
https://release.fastgit.org/chaitin/xray/releases/download/0.21.0/xray_linux_arm64.zip

unzip xray_linux_arm64.zip

# 直接运行
./xray_linux_arm64
```

```
7:11 ...0.0K/s 3G HD 4G 77
~/xray ➤ ./xray_linux_arm64 webscan --listen 127.0.0.1:7777 --html-output test.html

-- -- -- -- --
\ \ / / | | \ \ / / | | \ \ / / | | \ \ / /
 \ V / | | | | | | | | | | | | | | | |
  > < | | | | | | | | | | | | | | | |
 / \ | | | | | | | | | | | | | | | |
/_/_ \ \ / / | | \ \ / / | | \ \ / / | | \ \ / /

Version: 0.21.0/815971c3/COMMUNITY

Loading poc file from config file...
poc-yaml-74cms-sqli-1
poc-yaml-74cms-sqli-2
```

小结

因为 Termux 完美的支持 Python 和 Perl 等语言，所以有太多优秀的信息安全工具值得大家去发现了，这里我就不一一列举了。

官方插件

Termux 有一些额外有趣的功能，我们可以通过安装插件的方式来使用这些功能。网友们反馈 F-Droid 里面的 Termux 插件都是可以免费下载的，唯一的缺点就是可能版本低于 Google Play 的版本，大家自行抉择。

Termux:API

Termux:API，可用于访问手机硬件实现更多的友情的功能。官方唯一一个免费的 API，良心啊。

准备工作

安装Termux:API

下载地址

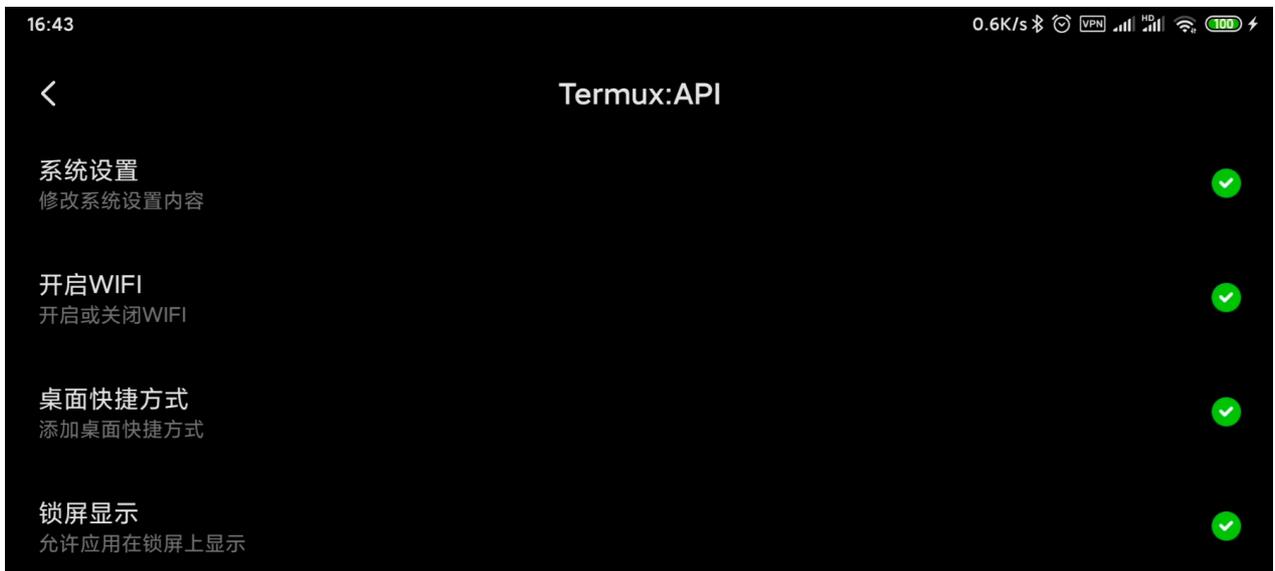
- [Termux:API Google Play 下载地址](#)
- [Termux:API F-Droid 下载地址](#)

请勿在Google Play 和 F-Droid 之间混合安装Termux 和 插件。



给 app 权限

因为 Termux-api 可以直接操作手机底层，所以我们需要到手机的设置里面给 这个 APP 的权限全部开了，这样下面操作的时候就不会提示权限不允许的情况了。



安装 Termux-api 软件包

手机安装完 Termux-api 的APP后，Termux 终端里面必须安装对应的包后才可以与手机底层硬件进行交互。

```
pkg install termux-api
```

下面只列举一些可能会用到的,想要获取更多关于 `Termux-api` 的话,那就去参考官方文档.

获取设备信息相关

获取电池信息

```
termux-battery-status
```

参数无，返回信息是JSON格式。可以看到电池的-健康状况-电量百分比-温度情况等

```
{
  "health": "GOOD",
  "percentage": 100,
  "plugged": "UNPLUGGED",
  "status": "DISCHARGING",
  "temperature": 24.600000381469727
}
```

获取相机信息

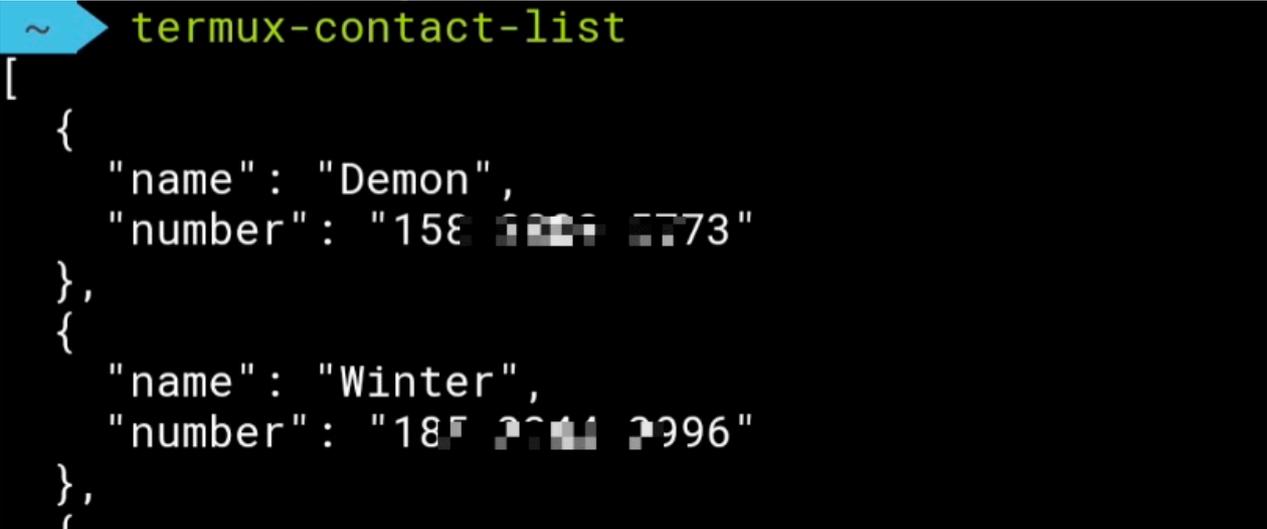
```
termux-camera-info
```

参数无，返回信息是JSON格式。

获取通讯录列表

参数无，返回信息是JSON格式。

```
termux-contact-list
```



```
~ termux-contact-list
[
  {
    "name": "Demon",
    "number": "158 1234 5673"
  },
  {
    "name": "Winter",
    "number": "18 1234 5696"
  },
  {

```

查看红外载波频率

```
termux-infrared-frequencies
```

获取无线电信息

无参数，返回格式是JSON格式

```
termux-telephony-cellinfo
```

查看手机运营商信息

无参数，返回格式是JSON格式

```
termux-telephony-deviceinfo
```

国光的实际测试返回值:

```
{
  "data_enabled": "true",
  "data_activity": "dormant",
  "data_state": "disconnected",
  "device_id": null,
  "device_software_version": "00",
  "phone_count": 2,
  "phone_type": "gsm",
  "network_operator": "46001",
  "network_operator_name": "中国联通",
  "network_country_iso": "cn",
  "network_type": "lte",
  "network_roaming": false,
  "sim_country_iso": "cn",
  "sim_operator": "46011",
  "sim_operator_name": "中国联通",
  "sim_serial_number": null,
  "sim_subscriber_id": null,
  "sim_state": "ready"
}
```

获取 tts 语音引擎信息

```
termux-tts-engines
```

国光的实际演示:

```
[
  {
    "name": "com.google.android.tts",
    "label": "Google 文字转语音引擎",
    "default": false
  },
  {
    "name": "com.xiaomi.mibrain.speech",
    "label": "小爱语音引擎",
    "default": true
  }
]
```

获取 USB 设备信息

```
termux-usb [-l | [-r] [-e 命令] 设备]
```

选项细节

```
-l          列出可用设备
-r          如果尚未授予权限, 则显示 权限请求对话框
-e command 执行指定的命令
```

Android Termux 下要读取 USB 内容或者 U 盘的时候, 还需要安装如下依赖包:

```
pkg install termux-api libusb clang -y
```

有些包我们上面已经安装过了。

下面国光来实际演示一下这个 API 的使用:

查看 次数 Android 的 USB 设备信息:

```
$ termux-usb -l
[
  "/dev/bus/usb/001/002"
]
```

可以看到国光我这个 512GB 的 M.2 SSD 已经显示在列表中了 (装作不经意间说出来的样子, 类似于朱一旦不经意间露出自己的劳力士一样 2333)

请求访问权限:

```
termux-usb -r /dev/bus/usb/001/002

Access granted.
```

下面级可以尝试写代码去访问这个 USB 设备内容了:

usbtest.c

```
#include <stdio.h>
#include <assert.h>
#include <libusb-1.0/libusb.h>

int main(int argc, char **argv) {
    libusb_context *context;
    libusb_device_handle *handle;
    libusb_device *device;
    struct libusb_device_descriptor desc;
    unsigned char buffer[256];
    int fd;
    assert((argc > 1) && (sscanf(argv[1], "%d", &fd) == 1));
```

```

assert(!libusb_init(&context));
assert(!libusb_wrap_sys_device(context, (intptr_t) fd, &handle));
device = libusb_get_device(handle);
assert(!libusb_get_device_descriptor(device, &desc));
printf("Vendor ID: %04x\n", desc.idVendor);
printf("Product ID: %04x\n", desc.idProduct);
assert(libusb_get_string_descriptor_ascii(handle, desc.iManufacturer,
buffer, 256) >= 0);
printf("Manufacturer: %s\n", buffer);
assert(libusb_get_string_descriptor_ascii(handle, desc.iProduct, buffer,
256) >= 0);
printf("Product: %s\n", buffer);
assert(libusb_get_string_descriptor_ascii(handle, desc.iSerialNumber,
buffer, 256) >= 0);
printf("Serial No: %s\n", buffer);
libusb_exit(context);
}

```

代码内容主要是显示有关USB设备的一些基本信息，参数就是我们的设备信息。让我们编译一下：

```
gcc usbtest.c -lusb-1.0 -o usbtest
```

编译完成后会生成可执行的文件 `usbtest`，接下来通过 这个 API 来执行这个文件：

```

termux-usb -e ./usbtest /dev/bus/usb/001/002

Vendor ID: 152d
Product ID: 0576
Manufacturer: Hikvision
Product: Hikvision External Disk
Serial No: 0123456789ABCDEF

```

糟糕 暴露了我的 512GB 的 SSD 居然是最廉价的海康威视，溜了溜了 逃~

获取当前WiFi连接信息

```
termux-wifi-connectioninfo
{
  "bssid": "xx:xx:00:00:00:00",
  "frequency_mhz": 5785,
  "ip": "192.168.31.124",
  "link_speed_mbps": 130,
  "mac_address": "xx:xx:00:00:00:00",
  "network_id": -1,
  "rssi": -69,
  "ssid": "<unknown ssid>",
  "ssid_hidden": true,
  "supplicant_state": "COMPLETED"
}
```

获取 WiFi 扫描信息

国光在 Android 10 上没有成功执行过这个 API，不过以前的一加5 是可以执行的下面的图 是以前的老图：

```
termux-wifi-scaninfo
```

```
"ssid": "<unknown ssid>",  
u0_a119 at localhost in ~ using  
└─ termux-wifi-connectioninfo  
{  
  "bssid": "14:79:f3:d8:bf:d8",  
  "frequency_mhz": 2462,  
  "ip": "192.168.1.3",  
  "link_speed_mbps": 72,  
  "mac_address": "02:00:00:00:00:00",  
  "network_id": 0,  
  "rssi": -46,  
  "ssid": "CMCC-ecV6",  
  "ssid_hidden": false,  
  "supPLICANT_state": "COMPLETED"  
}
```

```
u0_a119 at localhost in ~ using  
└─ termux-wifi-scaninfo
```

```
{  
  {  
    "bssid": "14:79:f3:d8:bf:d8",  
    "frequency_mhz": 2462,  
    "rssi": -44,  
    "ssid": "CMCC-ecV6",  
    "timestamp": 118918409270,  
    "channel_bandwidth_mhz": "20"  
  },  
  {  
    "bssid": "60:de:f3:68:bd:a0",  
    "frequency_mhz": 2462,  
    "rssi": -71,  
    "ssid": "CMCC-sGM2",  
    "timestamp": 118918129377,  
    "channel_bandwidth_mhz": "20"  
  },  
  {  
    "bssid": "44:97:5a:98:1f:22",  
    "frequency_mhz": 2417,  
    "rssi": -78,  
    "ssid": "FAST_207",  
    "timestamp": 118917148893,  
    "channel_bandwidth_mhz": "40",  
    "center_frequency_mhz": 2427  
  }  
}
```

```
u0_a119 at localhost in ~ using  
└─ █
```

调用设置设备相关

调整屏幕亮度

```
termux-brightness [亮度]
```

亮度的值在 0 ~ 255 之间，国光这里测试了 255 并达不到手机的最大亮度

拍摄照片

```
termux-camera-photo [-c camera-id] output-file
```

`camera-id`: 相机的 id, 默认是 0, 相机 id 可以通过隔壁的命令来获取

下面的命令表示 使用 相机id 为 0 即后置的相机来拍摄(id 1 为前置相机), 保存的文件在当前路径下 名字叫 guoguang.jpg

```
termux-camera-photo -c 0 guoguang.jpg
```

国光看了下后置拍摄出来的照片, 有点看不下去。前置相机效果还不错, 可能是因为现在手机是多个摄像头的原因, Termux 的相机算法上面没有做到足够好的优化。

获取与设置剪贴板

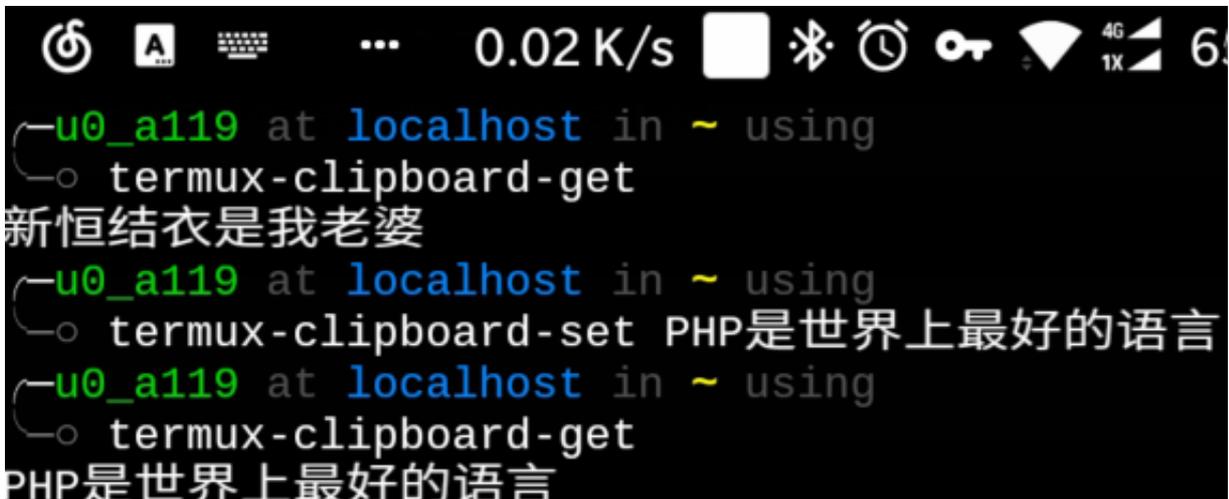
查看当前剪贴板内容

```
termux-clipboard-get
```

设置新的剪贴板内容

```
termux-clipboard-set PHP是世界上最好的语言
```

效果演示

A terminal window screenshot showing a sequence of commands and their outputs. The terminal title bar includes icons for music, keyboard, network speed (0.02 K/s), and system status (4G, 1X, 60%). The commands and outputs are:
1. Command: `termux-clipboard-get`
Output: 新恒结衣是我老婆
2. Command: `termux-clipboard-set PHP是世界上最好的语言`
3. Command: `termux-clipboard-get`
Output: PHP是世界上最好的语言

调用系统下载器

直接调用系统下载器，不返回任何值。而且国光我现在还没有搞清楚下载的文件到哪里去了

```
termux-download [-d 描述] [-t 标题] 下载的目标地址
```

国光的演示：

```
termux-download -d 'Termux下载测试' -t 'QQ.apk'  
'https://qd.myapp.com/myapp/qqteam/qq_JS/qqlite_4.0.0.1025_537062065.apk'
```



调用指纹传感器

该 API 仅适用于Android 6 及其以上版本，不接受任何参数，返回 JSON 格式。

```
termux-fingerprint
```

Authenticate

请触摸指纹传感器



CANCEL

返回内容:

```
{
  "errors": [],
  "failed_attempts": 0,
  "auth_result": "AUTH_RESULT_SUCCESS"
}
```

调用红外发射

```
termux-infrared-transmit -f 发射频率
```

频率以逗号分隔, 例如 `20,50,20,30`, 仅传输短于2秒的码型。

调用手机定位

输出格式为JSON格式

```
termux-location [-p 定位方式] [-r 定位请求]
```

细节参数:

```
-p provider [gps/network/passive] (默认: gps)
-r request [once/last/updates] (默认: once)
```

实际演示:

```
termux-location -p network
```

返回内容（关键地方已经打马赛克处理了 防止网友顺着网线打过来）：

```
{  
  "latitude": xx.xxxxxxx,  
  "longitude": xx.xxxxxxx,  
  "altitude": 0.0,  
  "accuracy": xx.0,  
  "vertical_accuracy": 0.0,  
  "bearing": 0.0,  
  "speed": 0.0,  
  "elapsedMs": 7,  
  "provider": "network"  
}
```

播放媒体文件

使用Media Player API播放指定的文件。

```
termux-media-player [命令] [参数]
```

详细参数:

```
info      # 显示当前播放信息  
play      # 恢复播放  
play <file> # 播放指定文件  
pause     # 暂停播放  
stop      # 停止退出播放
```

国光的实际演示:

```
# 播放 hacker.mp4 视频  
$ termux-media-player play hacker.mp4  
Now Playing: hacker.mp4  
  
# 查看当前播放信息  
$ termux-media-player info  
Status: Playing  
Track: hacker.mp4  
Current Position: 01:28 / 15:13  
  
# 退出播放  
$ termux-media-player stop  
Stopped playback  
Track cleared
```

调用麦克风

```
termux-microphone-record [参数]
```

参数细节

```
-d          使用默认设置录制
-f <file>   录制到特定文件
-l <limit>  使用指定的时间录制（以秒为单位，无限制为0）
-e <encoder> 使用指定的编码器录制（aac, amr_wb, amr_nb）
-b <bitrate> 使用指定的比特率录制（以kbps为单位）
-r <rate>    使用指定的采样率录制（以Hz为单位）
-c <count>  使用指定的通道录制（1, 2, ...）
-i          获取有关当前录音的信息
-q          退出录音
```

国光的实际演示:

```
$ termux-microphone-record -d
Recording started: /storage/emulated/0/TermuxAudioRecording_2020-04-20_19-50-51.m4a
Max Duration: 15:00

$ termux-microphone-record -i
{
  "isRecording": true,
  "outputFile": "\/storage\/emulated\/0\/TermuxAudioRecording_2020-04-20_19-50-51.m4a"
}

$ termux-microphone-record -q
Recording finished: /storage/emulated/0/TermuxAudioRecording_2020-04-20_19-50-51.m4a
```

显示系统通知

显示系统通知，通知内容使用 `-c` 或者 `--content` 读取，或者从 `stdin` 读取

```
termux-notification [选项]
```

选项细节

```
--action action    按下通知时要执行的动作
--alert-once       不会在编辑通知时发出警报
--button1 text     文本将显示在第一个通知按钮上
--button1-action action 动作将在第一个通知按钮上执行
--button2 text     在第二个通知按钮上显示的文本
```

```

--button2-action action 在第二个通知按钮上执行的动作
--button3 text          在第三通知按钮上显示的文本
--button3-action action 在第三个通知按钮上执行的动作通知按钮
-c/--content content   内容显示在通知中 将优先于stdin
--group group          通知组 (与同一组的通知一起显示)
-h/--help              显示此帮助
--help-actions         显示action的帮助
-i/--id id             通知id (将覆盖以前的任何通知具有相同ID的图像)
--image-path path      路径将在通知中显示的图像的绝对路径
--led-color rrggbb     闪烁的led颜色为RRGGBB (默认值: 无)
--led-off milliseconds 毫秒数使指示灯在闪烁时熄灭 (默认值: 800)
--led-on milliseconds 毫秒指示灯在闪烁时点亮的毫秒数 (默认值: 800)
--on-delete action     清除通知时要执行的操作
--ongoing              锁定通知
--priority prio        通知优先级 (高/低/最大/最小/默认)
--sound                通知时播放声音
-t/--title title       要显示的通知标题
--vibrate pattern      振动力度 逗号分隔, 如500,1000,200
--type type            要使用的通知样式 (default/media)

```

媒体通知参数

当你要使用 `--type media` 的时候详细参数:

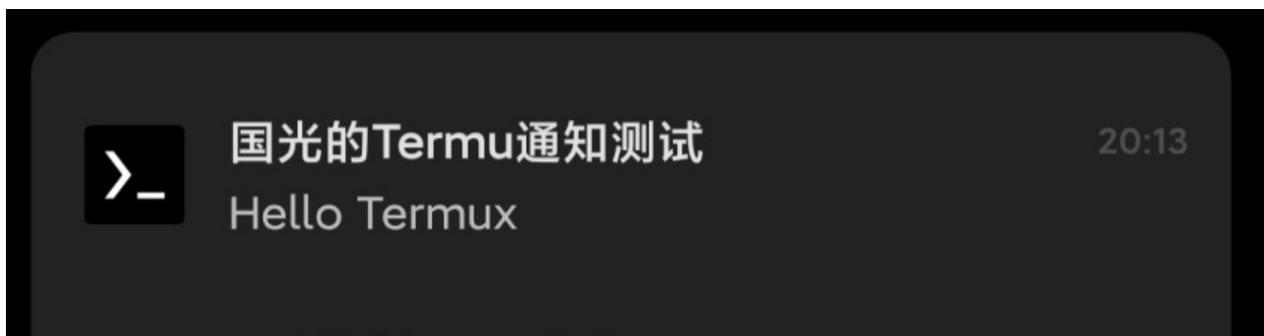
```

--media-next           在媒体 下一个按钮 上执行的操作
--media-pause          在媒体 暂停按钮 上执行的操作
--media-play           在媒体 播放按钮 上执行的操作
--media-previous       在媒体 上一个按钮 上执行的操作

```

国光的演示:

```
termux-notification -t '国光的Termu通知测试' -c 'Hello Termux' --type default
```



拨打电话

```
termux-telephony-call [号码]
```

拨打电话给 `10001` 中国电信,查看下话费有没有欠费~?

```
termux-telephony-call 10001
```

临时窗口通知

```
termux-toast [选项] [通知]
```

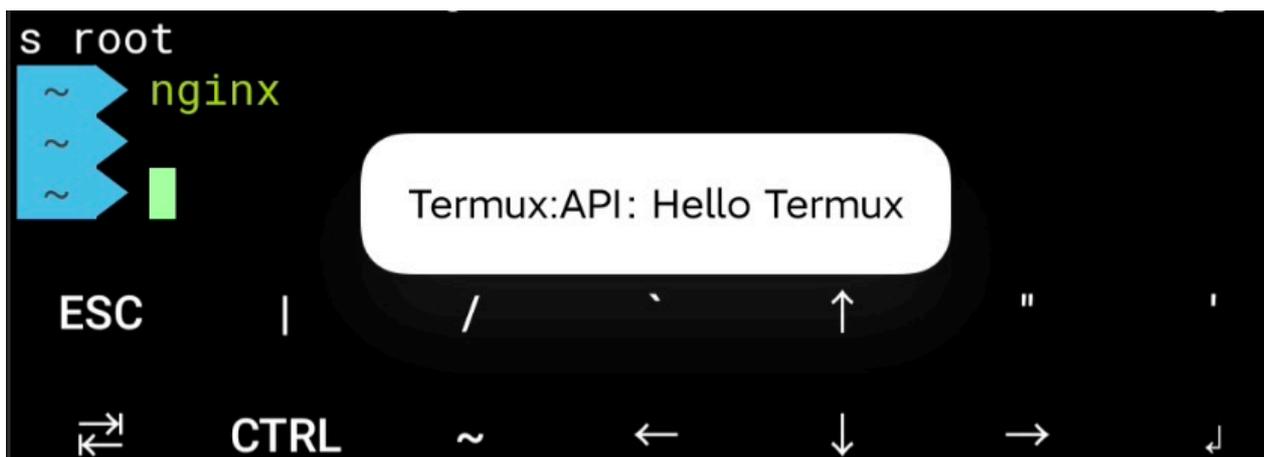
选项参数

```
-h 显示帮助信息  
-b 设置背景色 (默认: gray)  
-c 设置文本颜色 (默认: white)  
-g 设置吐司的位置: [top, middle, or bottom] (默认: middle)
```

颜色可以是标准名称 比如说:red 也可以是十六进制值如 #FF0000, 无效的颜色将恢复为默认值。

国光的实际演示:

```
termux-toast -b white -c black Hello Termux
```



开关闪光灯

```
termux-torch [on | off]
```

调用 tts 语音引擎

使用系统文本语音转换 (TTS) 引擎朗读文本。

```
termux-tts-speak [-e 引擎] [-l 语言] [-n 区域] [-v 变体] [-p 音调] [-r 速率] [-s ]  
[要说的话]
```

选项细节

```
-e engine # 要使用的 tts 语音引擎 (详见 termux-tts-engines 这个命令)
-l language # 要说的语言类别
-n region # 语言的地区
-v variant # 语言的变体
-p pitch # 语音的语调, 1 是默认的正常值
-r rate # 语音的语速, 1 是默认的正常值
-s stream # 要使用的音频流 (默认是 NOTIFICATION) 还可以选择ALARM, MUSIC,
NOTIFICATION, RING, SYSTEM, VOICE_CALL之一
```

国光的实际测试:

```
termux-tts-speak -e "com.xiaomi.mibrain.speech" '大家转载文章注意标明文章出处啊'
```

哦豁, 是不是被语音引擎的朗读吓一跳, 这样就可以用 Termux 去调用小爱语音引擎来说一些骚话了。

震动手机

```
termux-vibrate [选项]
```

选项细节:

```
-d duration # 以毫秒为单位的振动持续时间 (默认值: 1000)
-f # 在静默模式下也会强制振动
```

更换手机壁纸

```
termux-wallpaper [选项]
```

选项细节:

```
-f <file> # 将 file 文件设为壁纸
-u <url> # 从 URL 中获取壁纸
-l # 为锁屏设置壁纸 (Android N 及其以上)
```

国光的测试:

```
termux-wallpaper -u 'https://cn.bing.com/th?id=OHR.BluebellWood_ZH-
CN8128422960_1920x1080.jpg'
Wallpaper set successfully!
```

哦豁, 那样自动换壁纸的操作 岂不是就很简单了! 这个国光下面单独开章节来讲解。

开启关闭 WiFi

```
termux-wifi-enable [true | false]
```

交互对话框部件

这个功能有点强大，可以与用户进行交互。输出格式是 JSON

```
termux-dialog [选项]
```

基本用法：

```
-l, list  列出所有小部件及其选项  
-t, title 标题设置输入对话框的标题（可选）
```

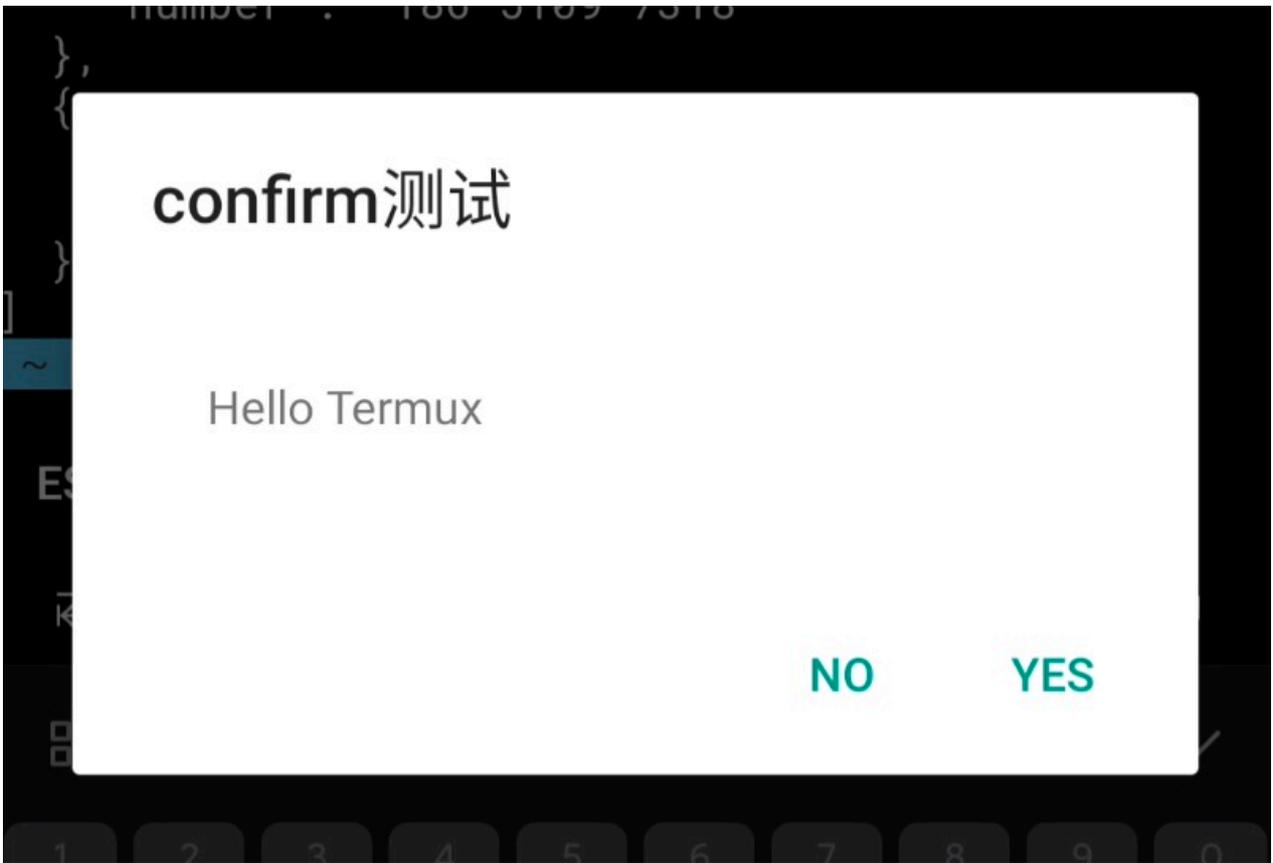
选项说明具体看下面国光的实际演示：

confirm

```
confirm - 显示确认对话框  
[-i 提示] 文本提示（可选）  
[-t 标题] 设置对话框标题（可选）
```

国光的演示：

```
termux-dialog confirm -i 'Hello Termux' -t 'confirm测试'
```



返回内容:

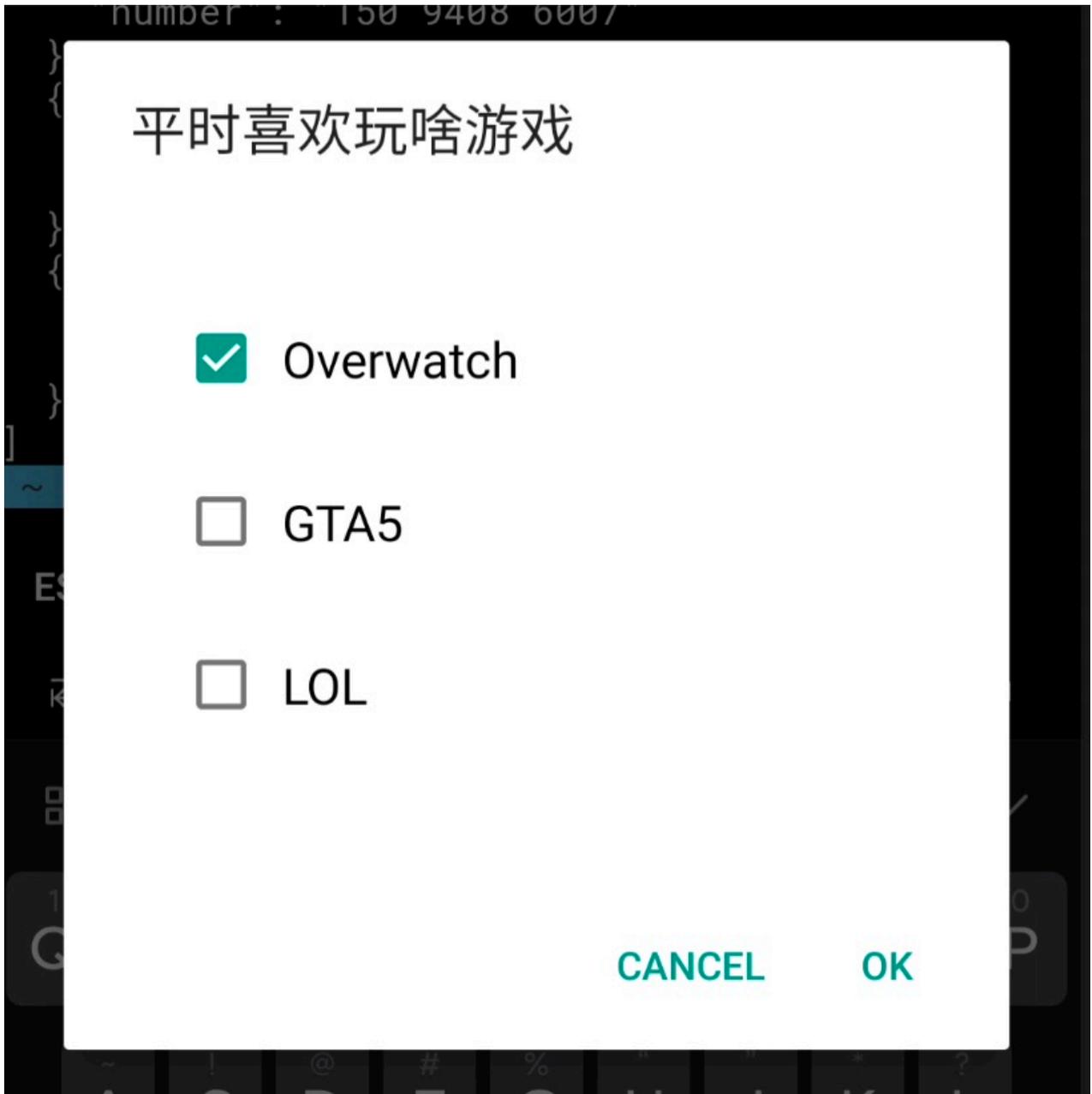
```
{  
  "code": 0,  
  "text": "yes"  
}
```

checkbox

`checkbox` - 使用复选框选择多个值
[-v ",,,"] 多个值用逗号隔开 (必选)
[-t 标题] 设置对话框的标题 (可选)

国光的演示:

```
termux-dialog checkbox -v 'Overwatch,GTA5,LOL' -t '平时喜欢玩啥游戏'
```



返回内容:

```
{
  "code": -1,
  "text": "[Overwatch]",
  "values": [
    {
      "index": 0,
      "text": "Overwatch"
    }
  ]
}
```

counter

`counter` - 选择指定范围内的数字

`[-r 最小值,最大值,开始值]` 3个值用逗号隔开 (可选)

`[-t 标题]` 设置对话框的标题 (可选)

国光的演示:



返回内容:

```
{  
  "code": -1,  
  "text": "181"  
}
```

date

`date` - 选择一个日期

`[-d "dd-MM-yyyy k:m:s"]` 用于日期小部件输出的`SimpleDateFormat`模式(可选)

`[-t 标题]` 设置对话框的标题(可选)

国光的演示:

```
termux-dialog date -d 'yyyy-MM-dd' -t '你的出生日期是?'
```

你的出生日期是?

2020年

4月20日周一

< 2020年4月 >

日 一 二 三 四 五 六

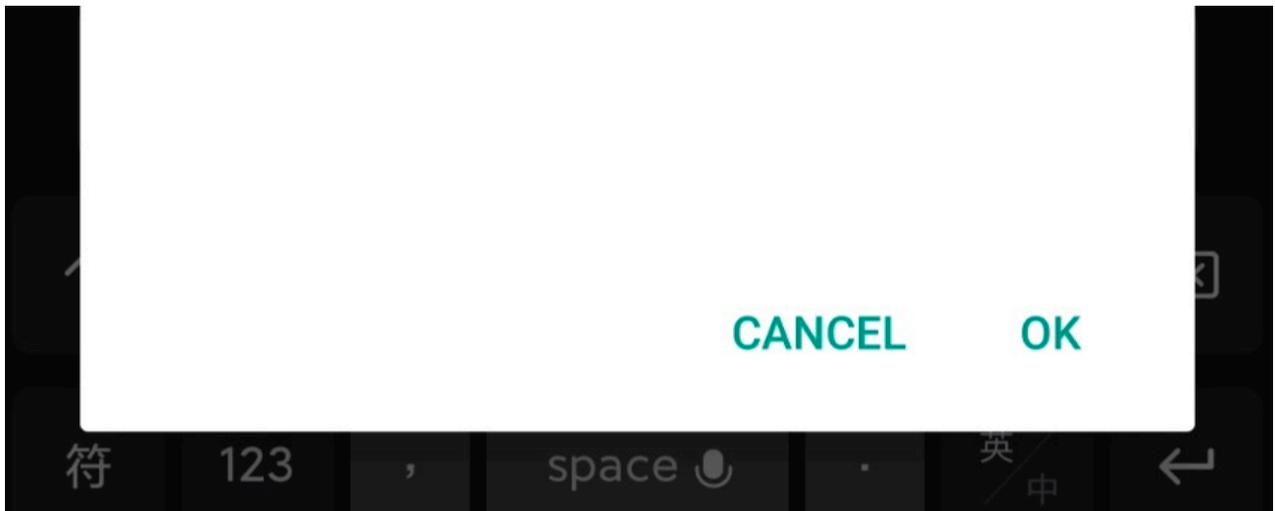
1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30



返回内容:

```
{  
  "code": -1,  
  "text": "2020-04-20"  
}
```

radio

radio - 从单选按钮中选择一个值
[-v " , , , "] 多个值用逗号隔开 (必选)
[-t 标题] 设置对话框的标题 (可选)

国光的演示:

```
termux-dialog radio -v '小哥哥,小姐姐' -t '你的性别是?'
```



返回内容:

```
{  
  "code": -1,  
  "text": "小哥哥",  
  "index": 0  
}
```

sheet

`sheet` - 从底部工作表中选择一个值
[-v " , , , "] 多个值用逗号隔开 (必选)
[-t 标题] 设置对话框的标题 (可选)

国光的演示:

```
termux-dialog sheet -v '菜鸡,国光'
```



返回内容:

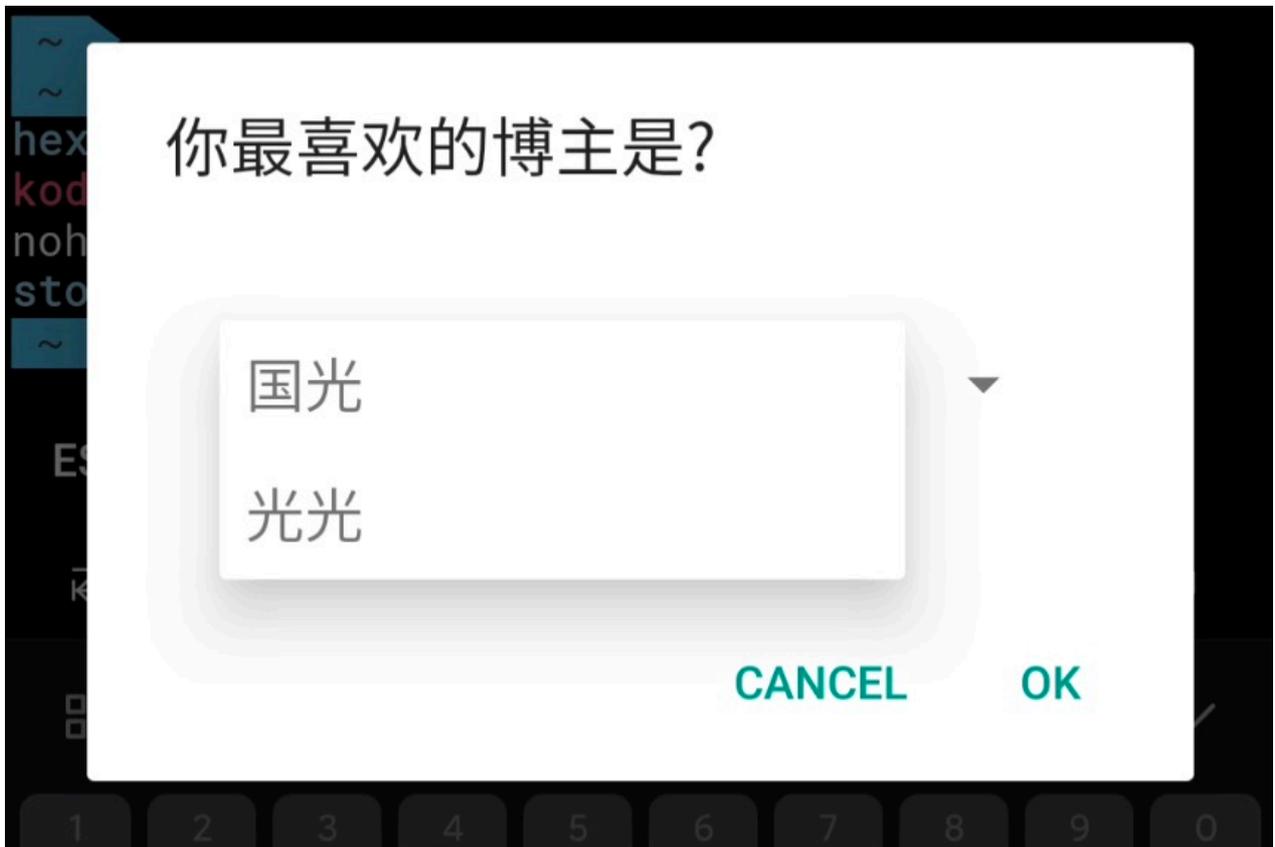
```
{  
  "code": 0,  
  "text": "国光",  
  "index": 1  
}
```

spinner

spinner - 从下拉微调器中选择一个值
[-v ",,,"] 多个值用逗号隔开 (必选)
[-t 标题] 设置对话框的标题 (可选)

国光的演示:

```
termux-dialog spinner -v '国光,光光' -t '你最喜欢的博主是?'
```



返回内容:

```
{
  "code": -1,
  "text": "国光",
  "index": 0
}
```

text

text - 输入文本 (如果未指定小部件, 则为默认值)

- [-i 提示]** 文本提示 (可选)
- [-m]** 多行输入 (可选) *
- [-n]** 输入数字 (可选) *
- [-p]** 输入密码 (可选)
- [-t 标题]** 设置对话框的标题 (可选)

*不能将[-m]与[-n]一起使用

国光的演示:

```
termux-dialog text -i '密码:' -t '请输入核弹爆炸密码'
```



返回内容:

```
{  
  "code": -1,  
  "text": "666666",  
  "index": 0  
}
```

time

```
time - 选择一个时间值  
[-t 标题] 设置对话框的标题 (可选)
```

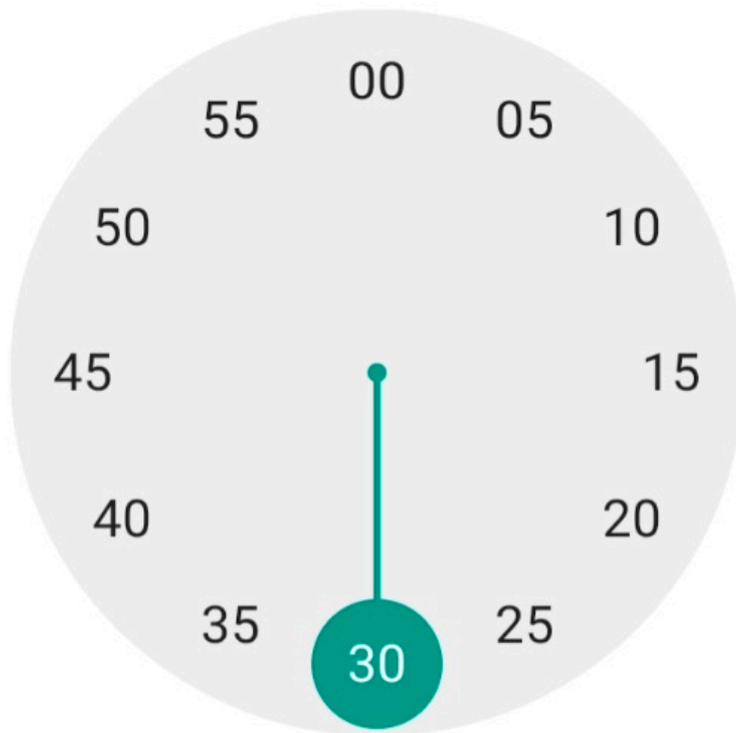
国光的演示:

```
termux-dialog time -t '你每天多少点睡觉?'
```

你每天多少点睡觉？

上
下

7:30



CANCEL

OK

返回内容:

```
{
  "code": -1,
  "text": "19:30",
  "index": 0
}
```

Termux:Boot

用于将自定义命令开机自启使用，不要每次重启完重复敲命令了。(当然你的手机性能很强，从来需要重启)

安装 Termux:Boot

- [Termux:Boot Google Play 下载地址](#)
- [Termux:Boot F-Droid 下载地址](#)

请勿在Google Play 和 F-Droid 之间混合安装Termux 和 插件。



这是一个收费的应用，一贫如洗的国光是买不起的，但是还得写一下这个，万一真的有壕买了这个APP呢

使用方法

安装完成启动这个应用程序后，创建 `~/.termux/boot/` 目录，将需要开机自启的脚本放在这个目录下，如果有多个文件的话，将他们按照排序顺序执行，如果要确保设备进入睡眠状态，建议脚本前面先运行 `termux-wake-lock` 命令。

下面是一个开机自启 `sshd` 服务的脚本，文件的完整路径为: `~/.termux/boot/start-sshd` 内容如下:

```
#!/data/data/com.termux/files/usr/bin/sh
termux-wake-lock
sshd
```

Termux:Float

可以将 Termux 悬浮窗形式显示，看上去比较酷炫。

- [Termux: Float Google Play 下载地址](#)
- [Termux: Float F-Droid 下载地址](#)

请勿在Google Play 和 F-Droid 之间混合安装Termux 和 插件。



这是依然是一个收费的应用，尽管它看上去很酷炫，但是家境贫寒的国光还是买不起，下面就晒一个效果图吧：

gives the same result.

```
bash4$ echo $$
11015

bash4$ echo $BASHPID
11015

bash4$ ps ax | grep bash4
11015 pts/2 R 0:00 bash4
```

But ...

```
#!/bin/bash4

echo "\$$ outside of subshell = $$"
echo "\$BASH_SHELL inside of subshell = $BASH_SHELL"
echo "\$BASHPID outside of subshell = $BASHPID"

echo

( echo "\$$ inside of subshell = $$"
  echo "\$BASH_SHELL inside of subshell = $BASH_SHELL"
  echo "\$BASHPID inside of subshell = $BASHPID" )
# Note that $$ returns PID of parent process.
```

```
$ echo $$
7151
$ echo $BASHPID
7151
$ ps ax | grep bash
5640 u0_a106 0:00 -bash
7151 u0_a106 0:00 bash
7183 u0_a106 0:00 grep bash
$
```

`$BASH_VERSINFO[n]`

A 6-element [array](#) containing version information about the installed release of

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p å

Termux:Styling

Termux 官方提供了终端的一些美化方案

可以将 Termux 悬浮窗形式显示，看上去比较酷炫。

- [Termux:Styling Google Play 下载地址](#)
- [Termux:Styling F-Droid 下载地址](#)

请勿在Google Play 和 F-Droid 之间混合安装Termux 和 插件。



Termux:Styling
Fredrik Fornwall

4.6 ★
380 条评价

1万+
下载次数

3
3 岁以上 ⓘ

₩2,500

这还是一个收费的应用，虽然我们前面的一键安装脚本已经可以达到这个美化效果了，但是如果你想尝试一下官方的可以试试看。

Termux:Widget

提供了 Andorid 的 Termux 小部件。

- [Termux:Widget Google Play 下载地址](#)
- [Termux:Widget F-Droid 下载地址](#)

请勿在Google Play 和 F-Droid 之间混合安装Termux 和 插件。



依然是收费插件，比较冷门，使用场景很少，不推荐大家入手这个，所以用法这里国光就不列出了。(这个偷懒的理由很好 2333)

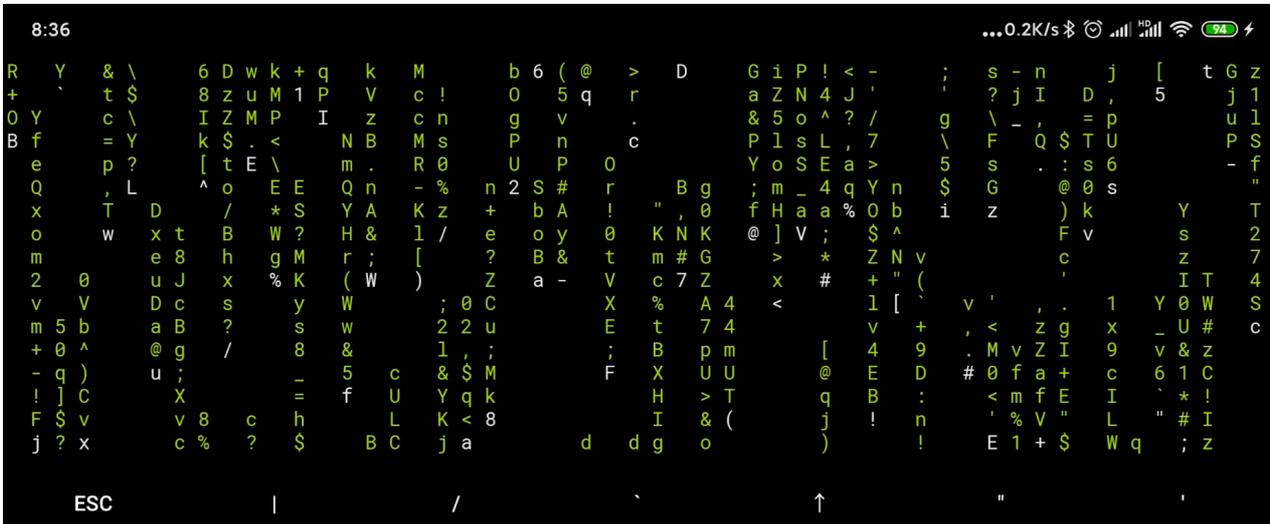
无聊专区

一些无聊有趣的知识，如果你是一个正经讲究人，可以跳过这个版块以节约你的阅读时间。

cmatrix

《黑客帝国》的代码雨视觉特效。

```
pkg install cmatrix
cmatrix
```




```
pkg install figlet
figlet -f 字体 '文本内容'
```

内置如下样式:

```
ls $PREFIX/share/figlet

646-ca.flc  646-jp.flc  8859-7.flc  circle.tlf  mini.flf
smbraille.tlf
646-ca2.flc 646-kr.flc  8859-8.flc  digital.flf  mnemonic.flf
smmono12.tlf
646-cn.flc  646-no.flc  8859-9.flc  emboss.tlf  mono12.tlf
smmono9.tlf
646-cu.flc  646-no2.flc  ascii12.tlf  emboss2.tlf  mono9.tlf
smscript.flf
646-de.flc  646-pt.flc  ascii9.tlf   frango.flc   moscow.flc
smshadow.flf
646-dk.flc  646-pt2.flc  banner.flf   future.tlf   pagga.tlf
smslant.flf
646-es.flc  646-se.flc  big.flf      hz.flc      script.flf
standard.flf
646-es2.flc 646-se2.flc  bigascii12.tlf  ilhebrew.flc  shadow.flf
term.flf
646-fr.flc  646-yu.flc  bigascii9.tlf  ivrit.flf     slant.flf
upper.flc
646-gb.flc  8859-2.flc  bigmono12.tlf  jis0201.flc  small.flf
ushebrew.flc
646-hu.flc  8859-3.flc  bigmono9.tlf  koi8r.flc    smascii12.tlf
uskata.flc
646-irv.flc 8859-4.flc  block.flf     lean.flf     smascii9.tlf
utf8.flc
646-it.flc  8859-5.flc  bubble.flf    letter.tlf   smblock.tlf
wideterm.tlf
```

国光的演示:

```
figlet -f future 'www.sqlsec.com'
```



hollywood

在 Linux 终端中伪造好莱坞黑客屏幕，假装自己是一名黑客。

```
pkg install hollywood
hollywood
```

```
8:50 ...0.2K/s 96%
[ , / ) PERLDBMFILTER(1) 00000fe0 00 00 00 00 00 0 Screen must be at least 4
# ?I ;? U 0 00 00 00 00 00 00 00 00 00 00 00
: 1? L X ?2 Perl Programmers Reference Guide
< u 2? M C : 00000ff0 e1 04 00 00 12 0
G ] m p? P t ? 0 00 00 00 00 00 00 00 00
s T t I Z?m y $ o 0 / 4 _ $ u 1 " A 00 00 |.....|
o?L y ; u T @ B?V y \ u \ , i Z 3 00 00 |.....|
i K D ;? . r # k P? W 3 \ z 0 g 00001000

|-----[SHA256]-----|
Change: 2020-04-21 08:48:02.252505559 +0800 Birth: -
Log ended: 2020-04-21 08:48:05
14 directories, 96 files /data/data/com.termux/files/home/www
index.php
0 directories, 1 file

II 0: bash- 1: hollywood* 2! 14h57m 0.22, 9x1.6GHz 5.4G72% 711058492% 2020-04-21 08:50:15

ESC | / \ ^ _ ` ~
CTRL ← →
```

neofetch

Neofetch 是一个简单但有用的命令行系统信息工具。它会收集有关系统软硬件的信息，并在终端中显示结果。

```
pkg install neofetch
neofetch
```

```
u0_a387@localhost ~ neofetch
-o o- u0_a387@localhost
+hydNNNNdyh+ OS: Android 10 aarch64
+mMMMMMMMMMMMMMM+ Host: Xiaomi Redmi K20
`dMMm: NMMMMMMN :mMMd` Kernel: 4.14.117-perf+
hMMMMMMMMMMMMMMMMMMh Uptime: 14 hours, 17 mins
.. yyyyyyyyyyyyyyyyyyy .. Packages: 125 (dpkg), 1 (pkg)
.mMMm`MMMMMMMMMMMMMMMMMMMM`mMMm. Shell: zsh 5.8
:MMMM-MMMMMMMMMMMMMMMMMMMMM-MMMM: Terminal: /dev/pts/1
:MMMM-MMMMMMMMMMMMMMMMMMMMM-MMMM: CPU: Qualcomm SDMMAGPIE (8) @ 1.804GHz
:MMMM-MMMMMMMMMMMMMMMMMMMMM-MMMM: Memory: 3961MiB / 5557MiB
-MMMM-MMMMMMMMMMMMMMMMMMMMM-MMMM-
+yy+ MMMMMMMMMMMMMMMMMMMMMMM +yy+
mMMMMMMMMMMMMMMMMMMMM
`/++MMMMh++hMMMM++/`
MMMMo oMMMM
MMMMo oMMMM
oNMm- -mMNs
```

nyancat

彩虹貓（英語：**Nyan Cat**）是在2011年4月上传在Youtube 的视频，并且迅速爆红于网络，並在2011年YouTube浏览量最高的视频中排名第五，B站这个小猫也很多，主要是BGM比较魔性，感兴趣的朋友可以自己去搜索看看。

```
pkg install nyan-cat  
nyan-cat
```

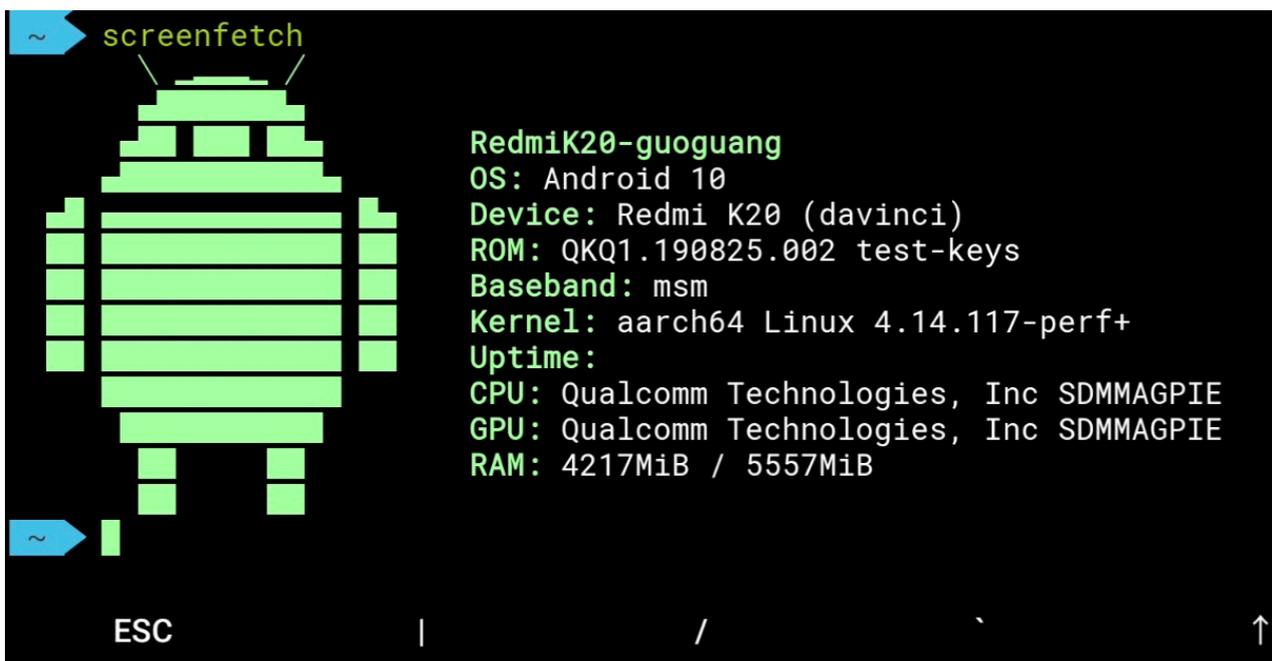


使用 `Ctrl + C` 快捷键退出魔性循环

screenfetch

Screenfetch 是一个适用于 Linux 的小工具，用于显示系统信息及ASCII化的Linux发行版图标。

```
pkg install screenfetch  
screenfetch
```



显示其他 Linux 发行版的 logo:

```
screenfetch -A 发行版
```

现在内置的发行版 logo 有:

```
ALDOS, Alpine Linux, Amazon Linux, Antergos, Arch Linux (Old and Current Logos), ArcoLinux, Artix Linux, blackPanther OS, BLAG, BunsenLabs, CentOS, Chakra, Chapeau, Chrome OS, Chromium OS, CrunchBang, CRUX, Debian, Deepin, DesaOS, Devuan, Dragora, elementary OS, EuroLinux, Evolve OS, Sulin, Exherbo, Fedora, Frugalware, Fuduntu, Funtoo, Fux, Gentoo, gNewSense, Guix System, Hyperbola GNU/Linux-libre, januslinux, Jiyuu Linux, Kali Linux, KaOS, KDE neon, Kogaion, Korora, LinuxDeepin, Linux Mint, LMDE, Logos, Mageia, Mandriva/Mandrake, Manjaro, Mer, Netrunner, NixOS, OBRevenge, openSUSE, OS Elbrus, Oracle Linux, Parabola GNU/Linux-libre, Pardus, Parrot Security, PCLinuxOS, PeppermintOS, Proxmox VE, PureOS, Qubes OS, Raspbian, Red Hat Enterprise Linux, ROSA, Sabayon, SailfishOS, Scientific Linux, Siduction, Slackware, Solus, Source Mage GNU/Linux, SparkyLinux, SteamOS, SUSE Linux Enterprise, SwagArch, TinyCore, Trisquel, Ubuntu, Viperr, Void and Zorin OS and EndeavourOS
```

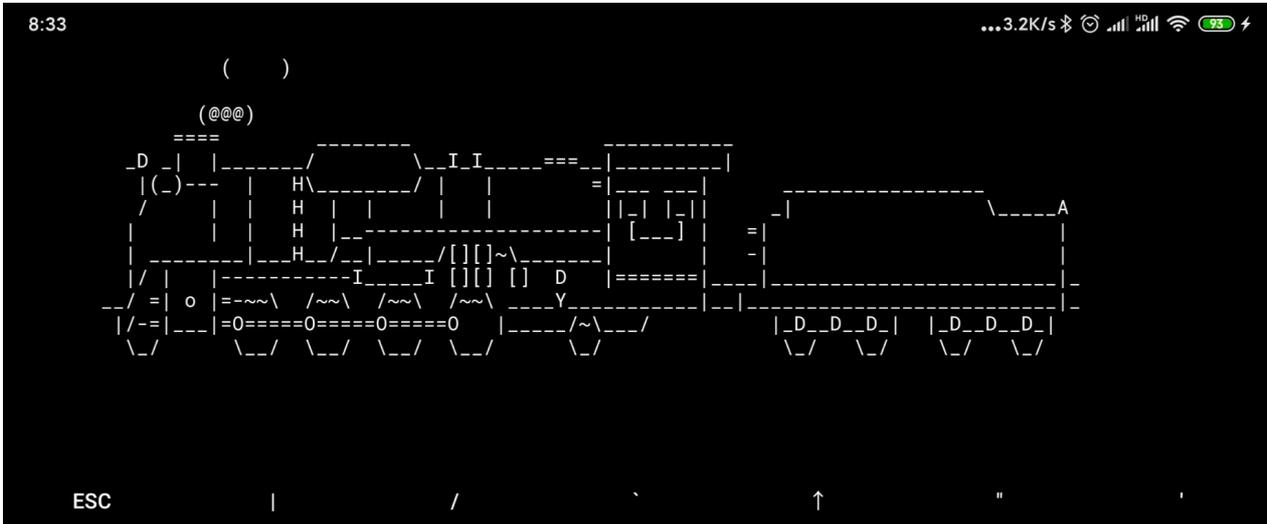
内置的操作系统 logo 有:

```
Dragonfly/Free/Open/Net BSD, Haiku, Mac OS X, Windows
```

sl

某编程牛人也经常犯把 ls 敲成 sl 的错误，所以他自己编了一个程序娱乐一下，这个程序的作用很简单，就是当你输入 sl 的时候终端会出现一个火车呼啸而过~~

```
pkg install sl  
sl
```



toilet

toilet 能用字母拼写出更大字母的工具，具体拼出什么字由命令后面的参数决定，不仅如此，它还能打印出各种风格的效果，比如彩色，金属光泽等。

```
pkg install toilet
toilet -f 字体 -F 颜色参数 '文本信息'
```

内置如下字体:

```
$ ls $PREFIX/share/figlet
```

```
ascii12.tlf      bigmono12.tlf  emboss2.tlf    mono9.tlf      smblock.tlf
wideterm.tlf
ascii9.tlf       bigmono9.tlf   future.tlf     pagga.tlf      smbaille.tlf
bigascii12.tlf  circle.tlf     letter.tlf     smascii12.tlf  smmono12.tlf
bigascii9.tlf   emboss.tlf     mono12.tlf    smascii9.tlf   smmono9.tlf
```

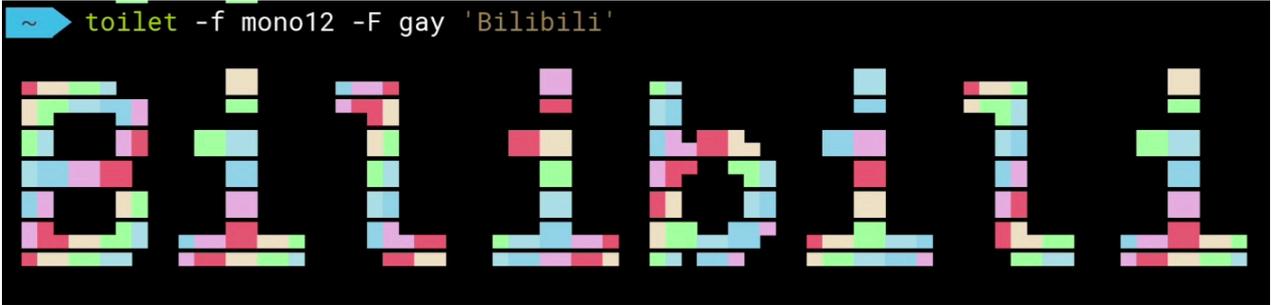
内置如下颜色效果:

```
$ toilet --filter list
```

```
Available filters:
"crop": crop unused blanks
"gay": add a rainbow colour effect
"metal": add a metallic colour effect
"flip": flip horizontally
"flop": flip vertically
"180": rotate 180 degrees
"left": rotate 90 degrees counterclockwise
"right": rotate 90 degrees clockwise
"border": surround text with a border
```

国光的简单演示:

```
toilet -f mono12 -F gay 'Bilibili'
```



终端二维码

Linux 命令行下的二维码,主要核心是这个网址:<http://qrenco.de/>

```
echo "https://www.sqlsec.com" |curl -F-=\<- qrenco.de
```





如果你不嫌无聊的话还可以扫描这个二维码，然后就又打开我的这篇文章了。

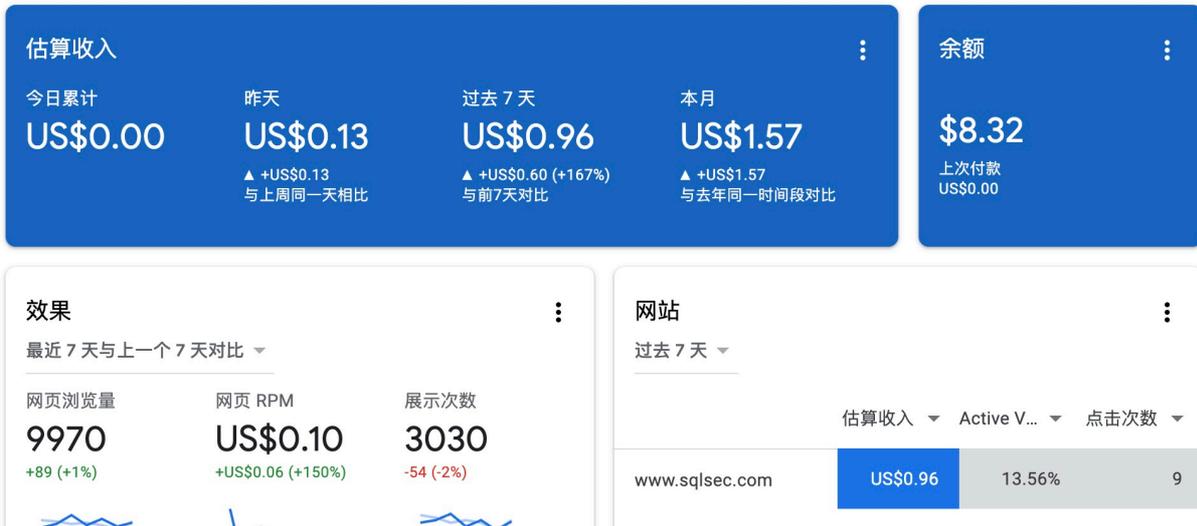
文章总结

相对来说 国外的Termux DIY的氛围比国内好很多，Youtube上的视频都有很高的播放量。当然国内也有这么一批执着于 Termux 的玩家，当初写完这篇文章的时候，相对来说还比较小众的，写这篇文章只想让更多的人认识到 Termux 的生产力，使用 Termux 来做一些很 cool 的事情。期待有更多 Termux 的优秀文章出现~

另外如果你喜欢这篇文章的话 不防点一下网站最下方不起眼的广告表示支持！Thanks♪(ω·)♪

发现一个尴尬的问题，看我这篇文章大人大多数是极客玩家，你们的浏览器自带去广告插件，尽管我的文章中的广告位置已经很不起眼了，但是你们可能连点的机会都没有...

博客我也写了快 4 年了，文章广告到现在的收益只有 8 美元，有图有真相：



所以如果你恰巧财力雄厚，感觉本文对你有帮助的话，可以考虑打赏一下本文，用以维持高昂的服务器运营费用（域名费用、服务器费用、CDN费用等），对博客用CDN了，否则访客不会有这么快速友好的体验。



“请随意打赏吧 (☒●>∪<●)”



国光(*怒嵩)

