

三星 S20 ebpf 之 bcc 环境安装

环境安装

debian 环境准备

eadb 项目链接如下

- <https://github.com/tiann/eadb>

下载 `debianfs-arm64-full.tar.gz`，当前最新版是 `v0.1.6`

- <https://github.com/tiann/eadb/releases/download/v0.1.6/debianfs-arm64-full.tar.gz>

创建 `/data/eadb` 文件夹，推送 rootfs 到手机上

```
adb shell su -c "mkdir -p /data/eadb"
adb push debianfs-arm64-full.tar.gz /data/local/tmp/deb.tar.gz
adb shell su -c "mv /data/local/tmp/deb.tar.gz /data/eadb/deb.tar.gz"
```

Bash

下载/克隆 `eadb` 项目，进入 `assets` 目录，推送脚本到手机并给脚本加上权限

- <https://github.com/tiann/eadb/archive/refs/heads/main.zip>

将 eadb 的 assets 文件夹推送到 `/data/eadb` 目录下，并给其中的脚本添加执行权限

```
adb push assets /data/local/tmp
adb shell su -c "mv /data/local/tmp/assets/* /data/eadb"
adb shell su -c "rm -r /data/local/tmp/assets"
adb shell su -c "chmod +x /data/eadb/*"
```

Bash

进入 `adb shell`，切换到 root 用户执行 rootfs 的解压脚本

```
adb shell su
/data/eadb/device-unpack
```

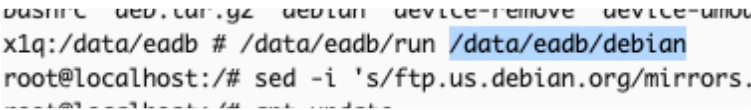
Bash

现在可以进入 debian 环境了（adb shell 后切换 root 用户）

```
/data/eadb/run /data/eadb/debian
```

Bash

成功之后如下图所示：



bcc 环境准备

修改下源，安装一些软件

```
sed -i 's/ftp.us.debian.org/mirrors.ustc.edu.cn/g' /etc/apt/sources.list
```

Bash

现在安装一些软件，这里的环境默认是 root，命令操作都不需要加 sudo

Bash

```
apt update
apt install git proxychains4 p7zip-full nano openssl-client openssl-server apt-utils
```

可以编辑.bashrc，在文件末尾加上 export LC_ALL=C

```
nano ~/.bashrc
source ~/.bashrc
```

Bash

如下图所示：

```
root@localhost:/# nano ~/.bashrc
root@localhost:/# ls
apex bin boot d dev etc get_kvers.sh home lib media mnt opt proc root run sbin srv sys
root@localhost:/# cat ~/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls` to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
export LC_ALL=C
root@localhost:/# source ~/.bashrc
root@localhost:/# █
```

git 代码拉取，执行命令

```
git clone https://github.com/tiann/bcc.git
```

Bash

成功之后如下图所示：

```
fatal: unable to access https://github.com/tiann/bcc.git/: couldn't connect to server
root@localhost:/# git clone https://github.com/tiann/bcc.git
Cloning into 'bcc'...
remote: Enumerating objects: 25871, done.
remote: Counting objects: 100% (2900/2900), done.
remote: Compressing objects: 100% (210/210), done.
remote: Total 25871 (delta 2736), reused 2690 (delta 2690), pack-reused 22971 (from 1)
Receiving objects: 100% (25871/25871), 16.40 MiB | 6.42 MiB/s, done.
Resolving deltas: 100% (17150/17150), done.
root@localhost:/# █
```

初始化子模块部分，命令如下：

```
git submodule init
git submodule update
```

Bash

执行成功如下图所示：

```
root@localhost:/bcc# ls .gitmodules
.gitmodules
root@localhost:/bcc# git submodule init
Submodule 'libbpf-tools/bpftool' (https://github.com/libbpf/bpftool) registered for path 'libbpf-tools/bpftool'
Submodule 'src/cc/libbpf' (https://github.com/libbpf/libbpf.git) registered for path 'src/cc/libbpf'
root@localhost:/bcc# git submodule update
Cloning into '/bcc/libbpf-tools/bpftool'...
Cloning into '/bcc/src/cc/libbpf'...
Submodule path 'libbpf-tools/bpftool': checked out '04c465fd1f561f67796dc68bbfe1aa7cfa956c3c'
Submodule path 'src/cc/libbpf': checked out '4eb6485c08867edaa5a0a81c64ddb23580420340'
root@localhost:/bcc# █
```

工程编译，执行下面的命令：

Bash

```
mkdir build && cd build
cmake ../
```

执行成功之后，如下图所示：

```
root@localhost:/bcc/build# cmake ../
-- The C compiler identification is GNU 10.2.1
-- The CXX compiler identification is Clang 11.0.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
Submodule 'libbpf' (https://github.com/libbpf/libbpf.git) registered for path 'libbpf-tools/bpftool/libbpf'
Cloning into '/bcc/libbpf-tools/bpftool/libbpf'...
Submodule path 'libbpf-tools/bpftool/libbpf': checked out '3591deb9bc6b0848721f831790b78ae0a593b4c5'
-- Latest recognized Git tag is v0.24.0
-- Git HEAD is 714219462192f5e342bc7ce159d156a11e883057
-- Revision is 0.24.0-71421946
-- Performing Test HAVE_NO_PIE_FLAG
-- Performing Test HAVE_NO_PIE_FLAG - Success
-- Performing Test HAVE_REALLOCARRAY_SUPPORT
-- Performing Test HAVE_REALLOCARRAY_SUPPORT - Success
-- Found LLVM: /usr/lib/llvm-11/include 11.0.1 (Use LLVM_ROOT environment variable for another version of LLVM)
-- Found BISON: /usr/bin/bison (found version "3.7.5")
-- Found FLEX: /usr/bin/flex (found version "2.6.4")
-- Found LibElf: /usr/lib/aarch64-linux-gnu/libelf.so
-- Performing Test ELF_GETSHDRSTRNDX
-- Performing Test ELF_GETSHDRSTRNDX - Success
-- Could NOT find LibDebuginfod (missing: LIBDEBUGINFOD_LIBRARIES LIBDEBUGINFOD_INCLUDE_DIRS)
-- Using static-libstdc++
-- Could NOT find LuaJIT (missing: LUAJIT_LIBRARIES LUAJIT_INCLUDE_DIR)
CMake Warning at tests/python/CMakeLists.txt:10 (message):
  Recommended test program 'netperf' not found

-- Configuring done
-- Generating done
-- Build files have been written to: /bcc/build
root@localhost:/bcc/build#
```

LuaJIT 和 netperf 这个可以暂时不管

然后开始编译 bcc，成功之后安装

Bash

```
make -j8
make install
```

执行 make -j8 如下图所示：

```
#define REQUIRE( expr ) INTERNAL_CATCH_TEST( expr, Catch::ResultDisposition::Normal, "REQUIRE" )

/bcc/tests/cc/catch.hpp:2062:32: note: expanded from macro 'INTERNAL_CATCH_TEST'
( __catchResult <= expr ).endExpression(); \

In file included from /bcc/tests/cc/test_sock_table.cc:23:
In file included from /bcc/src/cc/api/BPF.h:25:
In file included from /bcc/src/cc/api/BPFTable.h:30:
In file included from /bcc/src/cc/bcc_syms.h:24:
/bcc/src/cc/libbpf/include/uapi/linux/bpf.h:6362:1: warning: struct has size 0 in C, size 1 in C++ [-Wextern-c-compat]
struct bpf_raw_tracepoint_args {
^
1 warning generated.
[100%] Building CXX object tests/cc/CMakeFiles/test_libbcc.dir/utils.cc.o
[100%] Building CXX object tests/cc/CMakeFiles/test_libbcc.dir/test_parse_tracepoint.cc.o
In file included from /bcc/tests/cc/test_usdt_args.cc:20:
In file included from /bcc/src/cc/usdt.h:23:
In file included from /bcc/src/cc/bcc_proc.h:19:
In file included from /bcc/src/cc/bcc_syms.h:24:
/bcc/src/cc/libbpf/include/uapi/linux/bpf.h:6362:1: warning: struct has size 0 in C, size 1 in C++ [-Wextern-c-compat]
struct bpf_raw_tracepoint_args {
^
In file included from /bcc/tests/cc/test_usdt_probes.cc:23:
In file included from /bcc/src/cc/usdt.h:23:
In file included from /bcc/src/cc/bcc_proc.h:19:
In file included from /bcc/src/cc/bcc_syms.h:24:
/bcc/src/cc/libbpf/include/uapi/linux/bpf.h:6362:1: warning: struct has size 0 in C, size 1 in C++ [-Wextern-c-compat]
struct bpf_raw_tracepoint_args {
^
3 warnings generated.
1 warning generated.
In file included from /bcc/tests/cc/test_parse_tracepoint.cc:1:
In file included from /bcc/src/cc/api/BPF.h:25:
In file included from /bcc/src/cc/api/BPFTable.h:30:
In file included from /bcc/src/cc/bcc_syms.h:24:
/bcc/src/cc/libbpf/include/uapi/linux/bpf.h:6362:1: warning: struct has size 0 in C, size 1 in C++ [-Wextern-c-compat]
struct bpf_raw_tracepoint_args {
^
1 warning generated.
1 warning generated.
[100%] Built target PyPerf
1 warning generated.
1 warning generated.
[100%] Linking CXX executable test_libbcc
[100%] Built target test_libbcc
root@localhost:/bcc/build#
```

执行 make install 如下图所示：

```
-- Installing: /usr/share/bcc/tools/doc/threadsnoop_example.txt
-- Installing: /usr/share/bcc/tools/doc/tplist_example.txt
-- Installing: /usr/share/bcc/tools/doc/trace_example.txt
-- Installing: /usr/share/bcc/tools/doc/ttysnoop_example.txt
-- Installing: /usr/share/bcc/tools/doc/vfscount_example.txt
-- Installing: /usr/share/bcc/tools/doc/vfsstat_example.txt
-- Installing: /usr/share/bcc/tools/doc/virtiostat_example.txt
-- Installing: /usr/share/bcc/tools/doc/wakeuptime_example.txt
-- Installing: /usr/share/bcc/tools/doc/xfsdist_example.txt
-- Installing: /usr/share/bcc/tools/doc/xfs slower_example.txt
-- Installing: /usr/share/bcc/tools/doc/zfsdist_example.txt
-- Installing: /usr/share/bcc/tools/doc/zfs slower_example.txt
-- Installing: /usr/share/bcc/tools/lib/ucalls
-- Installing: /usr/share/bcc/tools/lib/uflow
-- Installing: /usr/share/bcc/tools/lib/ugc
-- Installing: /usr/share/bcc/tools/lib/uobjnew
-- Installing: /usr/share/bcc/tools/lib/ustat
-- Installing: /usr/share/bcc/tools/lib/uthreads
-- Installing: /usr/share/bcc/tools/doc/lib/ucalls_example.txt
-- Installing: /usr/share/bcc/tools/doc/lib/uflow_example.txt
-- Installing: /usr/share/bcc/tools/doc/lib/ugc_example.txt
-- Installing: /usr/share/bcc/tools/doc/lib/uobjnew_example.txt
-- Installing: /usr/share/bcc/tools/doc/lib/ustat_example.txt
-- Installing: /usr/share/bcc/tools/doc/lib/uthreads_example.txt
-- Installing: /usr/share/bcc/tools/old/bashreadline
-- Installing: /usr/share/bcc/tools/old/biosnoop
-- Installing: /usr/share/bcc/tools/old/compactsnop
-- Installing: /usr/share/bcc/tools/old/filelife
-- Installing: /usr/share/bcc/tools/old/gethostlatency
-- Installing: /usr/share/bcc/tools/old/hardirqs
-- Installing: /usr/share/bcc/tools/old/killsnoop
-- Installing: /usr/share/bcc/tools/old/memleak
-- Installing: /usr/share/bcc/tools/old/offcputime
-- Installing: /usr/share/bcc/tools/old/offwaketime
-- Installing: /usr/share/bcc/tools/old/oomkill
-- Installing: /usr/share/bcc/tools/old/opensnoop
-- Installing: /usr/share/bcc/tools/old/profile
-- Installing: /usr/share/bcc/tools/old/softirqs
-- Installing: /usr/share/bcc/tools/old/stackcount
-- Installing: /usr/share/bcc/tools/old/stacksnop
-- Installing: /usr/share/bcc/tools/old/statsnoop
-- Installing: /usr/share/bcc/tools/old/syncsnop
-- Installing: /usr/share/bcc/tools/old/tcpaccept
-- Installing: /usr/share/bcc/tools/old/tcpconnect
-- Installing: /usr/share/bcc/tools/old/wakeuptime
root@localhost:/bcc/build#
```

安装 python 环境，执行下面的命令

Bash

```
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
```

执行完成之后，如下图所示：

```
root@localhost:/bcc# wget https://bootstrap.pypa.io/get-pip.py
--2025-07-08 06:59:36-- https://bootstrap.pypa.io/get-pip.py
Resolving bootstrap.pypa.io (bootstrap.pypa.io)... 146.75.48.175, 2a04:4e42:7c::175
Connecting to bootstrap.pypa.io (bootstrap.pypa.io)|146.75.48.175|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2279307 (2.2M) [text/x-python]
Saving to: 'get-pip.py'

get-pip.py                               100%[=====>] 2.17M  4.37MB/s   in 0.5s

2025-07-08 06:59:37 (4.37 MB/s) - 'get-pip.py' saved [2279307/2279307]

root@localhost:/bcc# ls
CMakeLists.txt  CONTRIBUTING-SCRIPTS.md  INSTALL.md  LINKS.md  README.md  build  debian  docs  get-pip.py  introspection  man  snap  tests
CODEOWNERS     FAQ.txt               LICENSE.txt  QUICKSTART.md  SPECS     cmake  docker  examples  images  libbpf-tools  scripts  src  tools
root@localhost:/bcc# python get-pip.py
Collecting pip
  Downloading pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
Collecting setuptools
  Downloading setuptools-80.9.0-py3-none-any.whl.metadata (6.6 kB)
Collecting wheel
  Downloading wheel-0.45.1-py3-none-any.whl.metadata (2.3 kB)
Downloading pip-25.1.1-py3-none-any.whl (1.8 MB)
  1.8/1.8 MB 3.9 MB/s eta 0:00:00
Downloading setuptools-80.9.0-py3-none-any.whl (1.2 MB)
  1.2/1.2 MB 5.8 MB/s eta 0:00:00
Downloading wheel-0.45.1-py3-none-any.whl (72 kB)
Installing collected packages: wheel, setuptools, pip
Successfully installed pip-25.1.1 setuptools-80.9.0 wheel-0.45.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.
```

android 手机 安装应用，并且启动应用，如 shopee。启动之后，可以执行命令 ps -ef | grep shopee，如下图所示：

```
root@localhost:/bcc# ps -ef | grep shopee
10310   23896   955   1 06:52 ?           00:00:12 com.shopee.tw
root      26457 24738   0 07:04 ?           00:00:00 grep shopee
root@localhost:/bcc#
```

至此，开发环境已经搭建完成