



OSDT2021 AOSP for RISC-V 社区开源进展报告

中科院软件所智能软件中心 PLCT 实验室 汪辰

- AOSP RISC-V 社区发展简史
- AOSP RISC-V 移植现状
- PLCT Lab AOSP 12 移植工作计划

- **AOSP RISC-V 社区发展简史**
- AOSP RISC-V 移植现状
- PLCT Lab AOSP 12 移植工作计划

AOSP RISC-V 社区发展简史

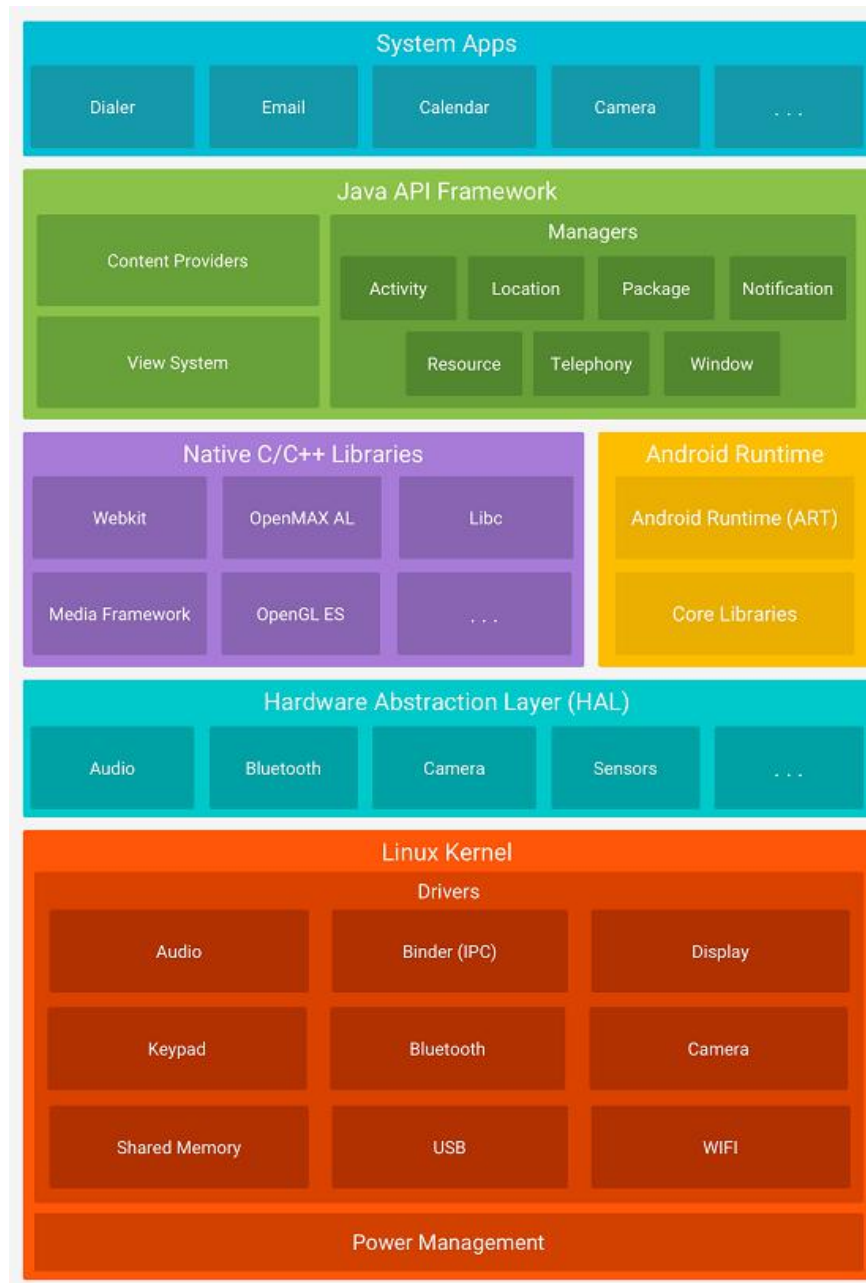


AOSP RISC-V 社区发展简史

- **2020 年 8 月**，中科院软件所 PLCT 实验室开始尝试基于 AOSP 10 实现 RISC-V 的移植工作并于 2020 年 12 月完成 bionic 的移植并实现了一个 第一个 RISC-V 上的 “Android 最小系统” (<https://zhuanlan.zhihu.com/p/302870095>) 。
- **2021 年 1 月**，阿里巴巴旗下的平头哥半导体 (T-Head) 成功将 AOSP 10 移植到自己的 RISC-V 芯片上，并开源部分代码，开源仓库地址在：<https://github.com/T-head-Semi/aosp-riscv>。同时 PLCT 实验室停止了相关 AOSP 移植工作，原有工作都备份到 <https://gitee.com/aosp-riscv-bionic-porting>。
- **2021 5 月 20 日**，RVI Android SIG 成立 - Han Mao (Alibaba), Zheng Zhang (Imagination) 担任 acting chairs . <https://lists.riscv.org/g/sig-android/message/1> 。
- **2021 7 月 20 日**，第一次 RVI Android SIG 会议，建立 RVI Android SIG 的官方主页 <https://lists.riscv.org/g/sig-android>，并制定了初步的开发任务列表: <https://lists.riscv.org/g/sig-android/message/5> 。
- **2021 10 月 20 日**，RVI Android SIG 的官方源码仓库建立（基于 AOSP 10）：<https://github.com/riscv-android-src>。中科院软件所 PLCT 实验室同时宣布加入，并贡献了第一个 PR: https://github.com/riscv-android-src/toolchain-llvm_android/pull/1 。
- **2021 10 月 29 日 ~ 现在**，PLCT lab 重启 AOSP 移植工作，最新的目标是将 AOSP 12 移植到 RV64 上。目前的所有工作都同步开源在：<https://github.com/aosp-riscv> 。

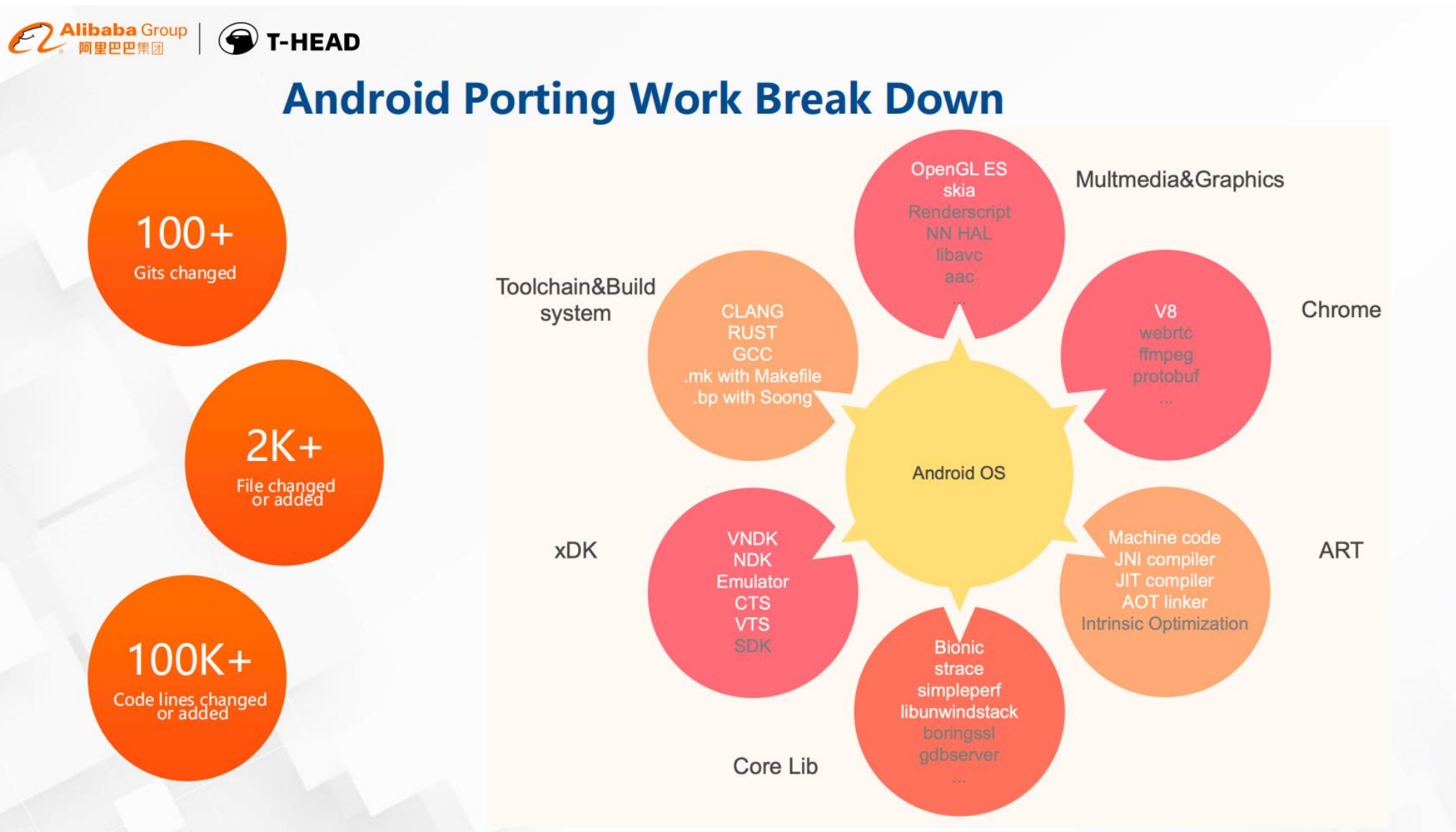
- AOSP RISC-V 社区发展简史
- **AOSP RISC-V 移植现状**
- PLCT Lab AOSP 12 移植工作计划

AOSP RISC-V 移植现状



图片来源:
https://developer.android.google.cn/guide/platform/images/android-stack_2x.png

AOSP RISC-V 移植现状



来源: https://chipsalliance.org/wp-content/uploads/sites/83/2021/10/porting-android-chips_alliance-slides-v1.2-Han-Mao.pdf

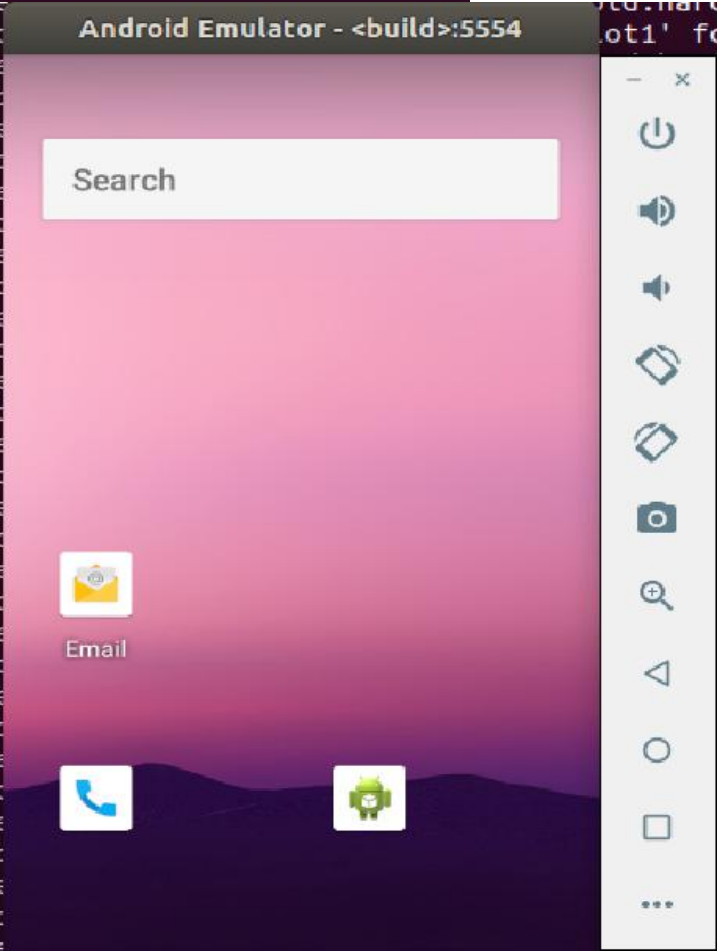
AOSP RISC-V 移植现状

```
=====
u@u18:~/ws/aosp-thead-10/aosp-riscv/aosp$ export ANDROID_PRODUCT_OUT="/home/u/ws/aosp-thead-10/aosp-riscv/aosp/out/target/product/generic_riscv64"
u@u18:~/ws/aosp-thead-10/aosp-riscv/aosp$ emulator -shell -selinux permissive -qemu -smp 2 -m 3800M -bios /home/u/ws/aosp-thead-10/aosp-riscv/aosp/p
rebuilts/qemu-kernel/riscv64/ranchu/fw_jump.bin
emulator: WARNING: encryption is off
kernel@80200000, initrd@88200000
qemu-system-riscv64: plic: invalid register write: 000001fc

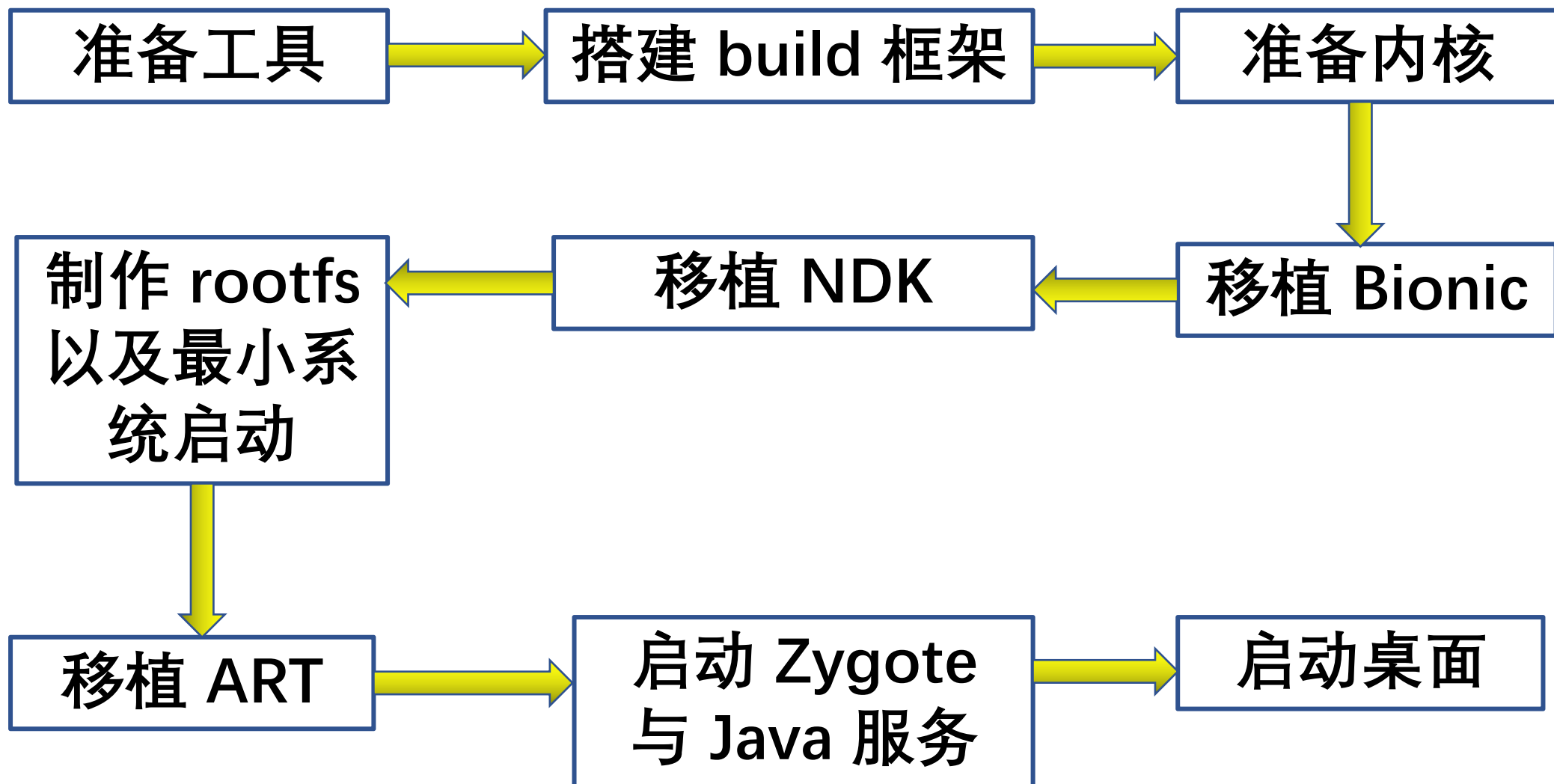
OpenSBI v0.7-3-g6585fab

Platform Name      : QEMU Virt Machine
Platform HART Features : RV64ACDFMSU
Current Hart      : 0
Firmware Base     : 0x80000000
Firmware Size     : 128 KB
Runtime SBI Version : 0.2

MIDELEG : 0x0000000000000222
MEDELEG : 0x0000000000000b109
PMP0    : 0x0000000000000000-0x0000000000000000
PMP1    : 0x0000000000000000-0xffffffffffffffff
emulator: ERROR: AdbHostServer.cpp:102: Unable to
qemu-system-riscv64: plic: invalid register read:
qemu-system-riscv64: plic: invalid register write:
qemu-system-riscv64: plic: invalid register read:
qemu-system-riscv64: plic: invalid register write:
[ 0.000000] OF: fdt: Ignoring memory range 0x[
[ 0.000000] Linux version 5.4.48-00598-g1457c
11.0.3 (https://android.googlesource.com/toolcha
lvm-project/llvd cc96790d85f5928e236eb3158491a2bc8
[ 0.000000] Initial ramdisk at: 0x(____ptrval)
[ 0.000000] Zone ranges:
[ 0.000000] DMA32    [mem 0x0000000000000000-0x0000000000000000]
[ 0.000000] Normal  [mem 0x0000000000000000-0x0000000000000000]
[ 0.000000] Movable zone start for each node
[ 0.000000] Early memory node ranges
[ 0.000000] node 0: [mem 0x0000000000000000-0x0000000000000000]
[ 0.000000] Initmem setup node 0 [mem 0x0000000000000000-0x0000000000000000]
[ 0.000000] software IO TLB: mapped [mem 0xfbf11111-0xfbf11111]
[ 0.000000] elf_hwcap is 0x112d
[ 0.000000] percpu: Embedded 26 pages/cpu s672000000
console:/ $ ls -l
ls -l
total 132
dr-xr-xr-x 24 root root 0 1970-01-01 00:00 acct
drwxr-xr-x 14 root root 280 1970-01-01 00:00 apex
lrw-r--r-- 1 root root 11 2021-10-08 06:53 bin -> /system/bin
lrw-r--r-- 1 root root 50 2021-10-08 06:53 bugreports -> /data/user
drwxrwx--- 2 system cache 4096 2021-10-08 06:12 cache
lrw-r--r-- 1 root root 19 2021-10-08 06:53 charger -> /system/bin/cha
drwxr-xr-x 3 root root 0 1970-01-01 00:00 config
lrw-r--r-- 1 root root 17 2021-10-08 06:53 d -> /sys/kernel/debug
drwxrwx--x 42 system system 4096 1970-01-01 00:00 data
drwxr-xr-x 2 root root 4096 2021-10-08 06:12 debug_ramdisk
lrw----- 1 root root 23 2021-10-08 06:53 default.prop -> system/e
drwxr-xr-x 17 root root 2860 1970-01-01 00:00 dev
lrw-r--r-- 1 root root 11 2021-10-08 06:53 etc -> /system/etc
lrwxr-x--- 1 root shell 16 2021-10-08 06:53 init -> /system/bin/init
-rwxr-x--- 1 root shell 1653 2021-10-08 06:12 init.environ.rc
-rwxr-x--- 1 root shell 33632 2021-10-08 06:36 init.rc
-rwxr-x--- 1 root shell 7690 2021-10-08 06:35 init.usb.configfs.rc
-rwxr-x--- 1 root shell 5649 2021-10-08 06:35 init.usb.rc
-rwxr-x--- 1 root shell 563 2021-10-08 06:35 init.zygote32.rc
-rwxr-x--- 1 root shell 959 2021-10-08 06:35 init.zygote32_64.rc
-rwxr-x--- 1 root shell 981 2021-10-08 06:35 init.zygote64_32.rc
drwx----- 2 root root 16384 2021-10-08 06:53 lost+found
drwxr-xr-x 2 root root 4096 2021-10-08 06:12 metadata
drwxr-xr-x 12 root system 260 1970-01-01 00:00 mnt
drwxr-xr-x 2 root root 4096 2021-10-08 06:12 odm
drwxr-xr-x 2 root root 4096 2021-10-08 06:12 oem
dr-xr-xr-x 121 root root 0 1970-01-01 00:00 proc
lrw-r--r-- 1 root root 15 2021-10-08 06:53 product -> /system/product
lrw-r--r-- 1 root root 24 2021-10-08 06:53 product_services -> /sys
drwxr-xr-x 3 root root 4096 2021-10-08 06:36 res
drwxr-x--- 2 root shell 4096 2021-10-08 06:12 sbin
lrw-r--r-- 1 root root 21 2021-10-08 06:53 sdcard -> /storage/self/
drwxr-xr-x 3 root root 60 1970-01-01 00:00 storage
dr-xr-xr-x 12 root root 0 1970-01-01 00:00 sys
drwxr-xr-x 13 root root 4096 2021-10-08 06:53 system
-rw-r--r-- 1 root root 2608 2021-10-08 06:12 ueventd.rc
drwxr-xr-x 8 root root 4096 2021-10-08 06:36 vendor
console:/ $
```



- AOSP RISC-V 社区发展简史
- AOSP RISC-V 移植现状
- **PLCT Lab AOSP 12 移植工作计划**



准备工具

- **LLVM/Clang(12.0.7)**
 - 添加 riscv64-linux-android target
 - 构建 android-riscv 版本的运行时库(compiler-rt)、libunwind、cxxlib, 支持静态/动态链接
- **GNU toolchain (gcc 11.1.0)**
- **Go**
- **Rust**
 - 添加 riscv64gc-linux-android target
 - 添加 rust libc 对 riscv64gc-linux-android 的符号绑定信息
- **Emulator**

搭建 build 框架 (1)

```
u@u-OptiPlex-7080:~/ws/dev-aosp12$ lunch aosp_riscv64-eng
```

build/ma

```
=====
```

```
PLATFORM_VERSION_CODENAME=REL
```

```
PLATFORM_VERSION=12
```

```
TARGET_PRODUCT=aosp_riscv64
```

```
TARGET_BUILD_VARIANT=eng
```

```
TARGET_BUILD_TYPE=release
```

```
TARGET_ARCH=riscv64
```

build/ma

```
TARGET_ARCH_VARIANT=riscv64
```

```
TARGET_CPU_VARIANT=generic
```

```
HOST_ARCH=x86_64
```

```
HOST_2ND_ARCH=x86
```

```
HOST_OS=linux
```

```
HOST_OS_EXTRA=Linux-5.4.0-91-generic-x86_64-Ubuntu-18.04.6-LTS
```

```
HOST_CROSS_OS=windows
```

build/ma

```
HOST_CROSS_ARCH=x86
```

```
HOST_CROSS_2ND_ARCH=x86_64
```

```
HOST_BUILD_TYPE=release
```

build/ma

```
BUILD_ID=SP1A.210812.016.A1
```

```
OUT_DIR=out
```

```
PRODUCT_SOONG_NAMESPACES=device/generic/goldfish device/generic/goldfish-opengl
```

```
=====
```

```
u@u-OptiPlex-7080:~/ws/dev-aosp12$
```

中增加一项

一项

v64

SCV64

搭建 build 框架 (2)

Android.bp

Android.mk

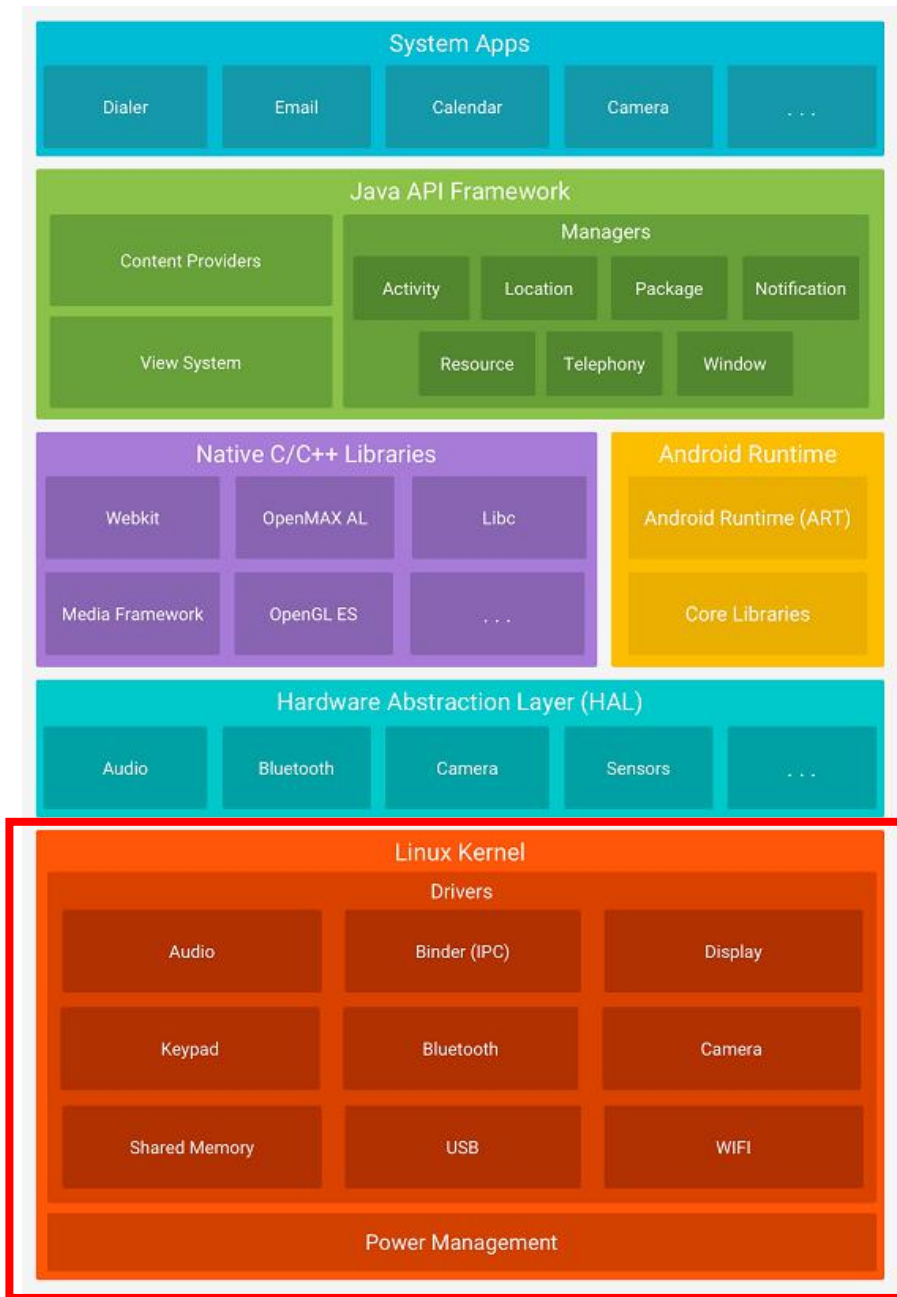
```
u@u-OptiPlex-7080:~/ws/dev-aosp12$ m --skip-ninja
build/make/core/soong_config.mk:195: warning: BOARD_PLAT_PUBLIC_SEPOLICY_DIR has been deprecated. Use S
build/make/core/soong_config.mk:196: warning: BOARD_PLAT_PRIVATE_SEPOLICY_DIR has been deprecated. Use
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=12
TARGET_PRODUCT=aosp_riscv64
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_ARCH=riscv64
TARGET_ARCH_VARIANT=riscv64
TARGET_CPU_VARIANT=generic
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-5.4.0-91-generic-x86_64-Ubuntu-18.04.6-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=SP1A.210812.016.A1
OUT_DIR=out
PRODUCT_SOONG_NAMESPACES=device/generic/goldfish device/generic/goldfish-opengl hardware/google/camera
=====
[ 96% 433/451] including system/sepolicy/Android.mk ...
system/sepolicy/Android.mk:57: warning: BOARD_PLAT_PUBLIC_SEPOLICY_DIR has been deprecated. Use SYSTEM
system/sepolicy/Android.mk:62: warning: BOARD_PLAT_PRIVATE_SEPOLICY_DIR has been deprecated. Use SYSTEM
[100% 454/454] writing packaging rules ...

#### build completed successfully (01:37 (mm:ss)) ####

u@u-OptiPlex-7080:~/ws/dev-aosp12$
```

android
images

PLCT Lab AOSP 12 移植工作计划



图片来源:
https://developer.android.google.cn/guide/platform/images/android-stack_2x.png

准备内核

Android platform release	Launch kernels	Feature kernels
Android 9 (2018)	android-4.4-p android-4.9-p android-4.14-p	android-4.4-p android-4.9-p android-4.14-p
Android 10 (2019)	android-4.9-q android-4.14-q android-4.19-q	android-4.9-q android-4.14-q android-4.19-q
Android 11 (2020)	android-4.14-stable android-4.19-stable android11-5.4	android-4.14-stable android-4.19-stable android11-5.4
Android 12 (2021)	android-4.19-stable android11-5.4 android12-5.4 android12-5.10	android12-5.4 android12-5.10

图片来源:

<https://source.android.com/devices/architecture/kernel/android-common#feature-and-launch-kernels>

```
$ git clone https://android.googlesource.com/kernel/common && cd common && git checkout 5.10-android12-9
```

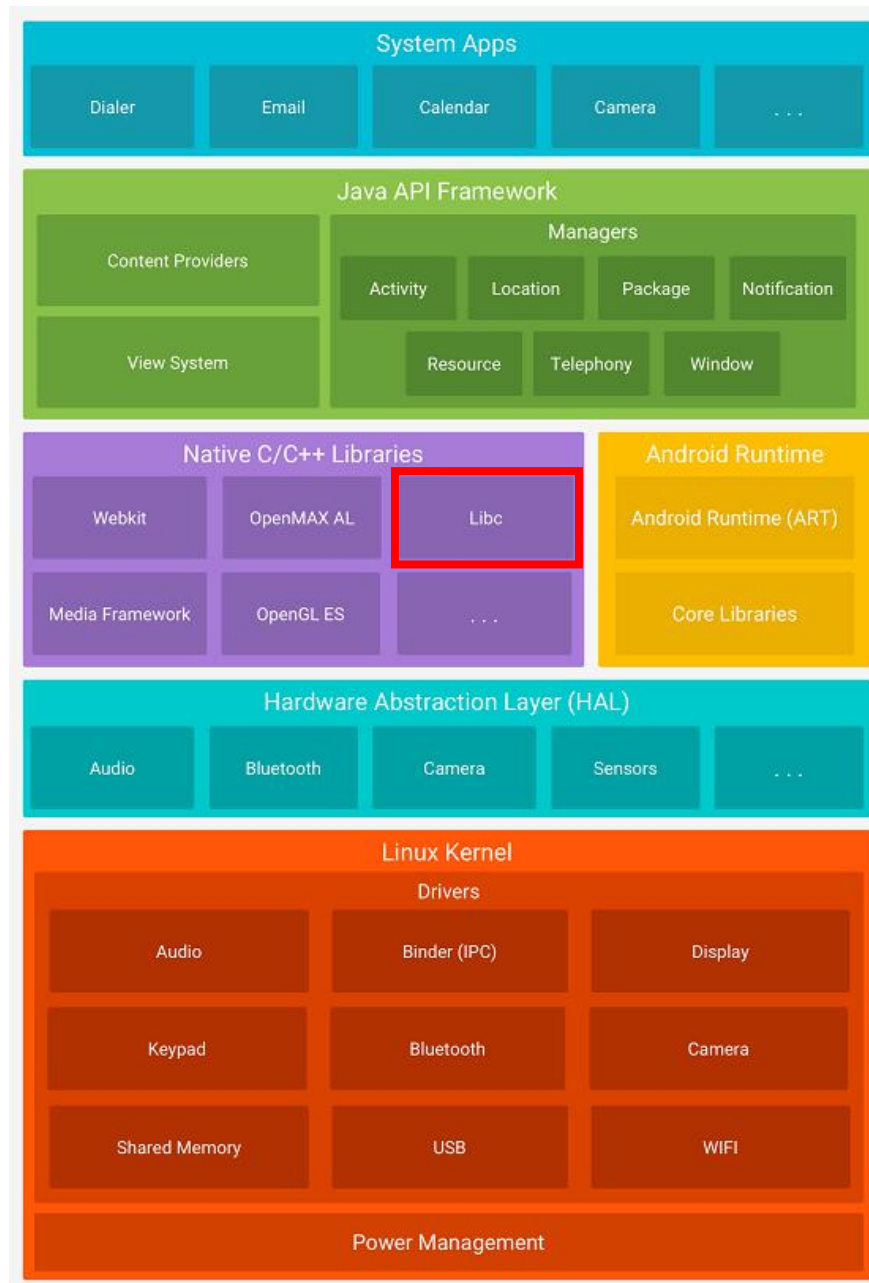
```
$ git clone https://android.googlesource.com/kernel/configs && cd configs && git checkout android-12.0.0_r3
```

```
$ cd common
```

```
$ ARCH=riscv ./scripts/kconfig/merge_config.sh arch/riscv/configs/defconfig ../configs/android-5.10/android-base.config
```

```
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- -j $(nproc)
```

PLCT Lab AOSP 12 移植工作计划

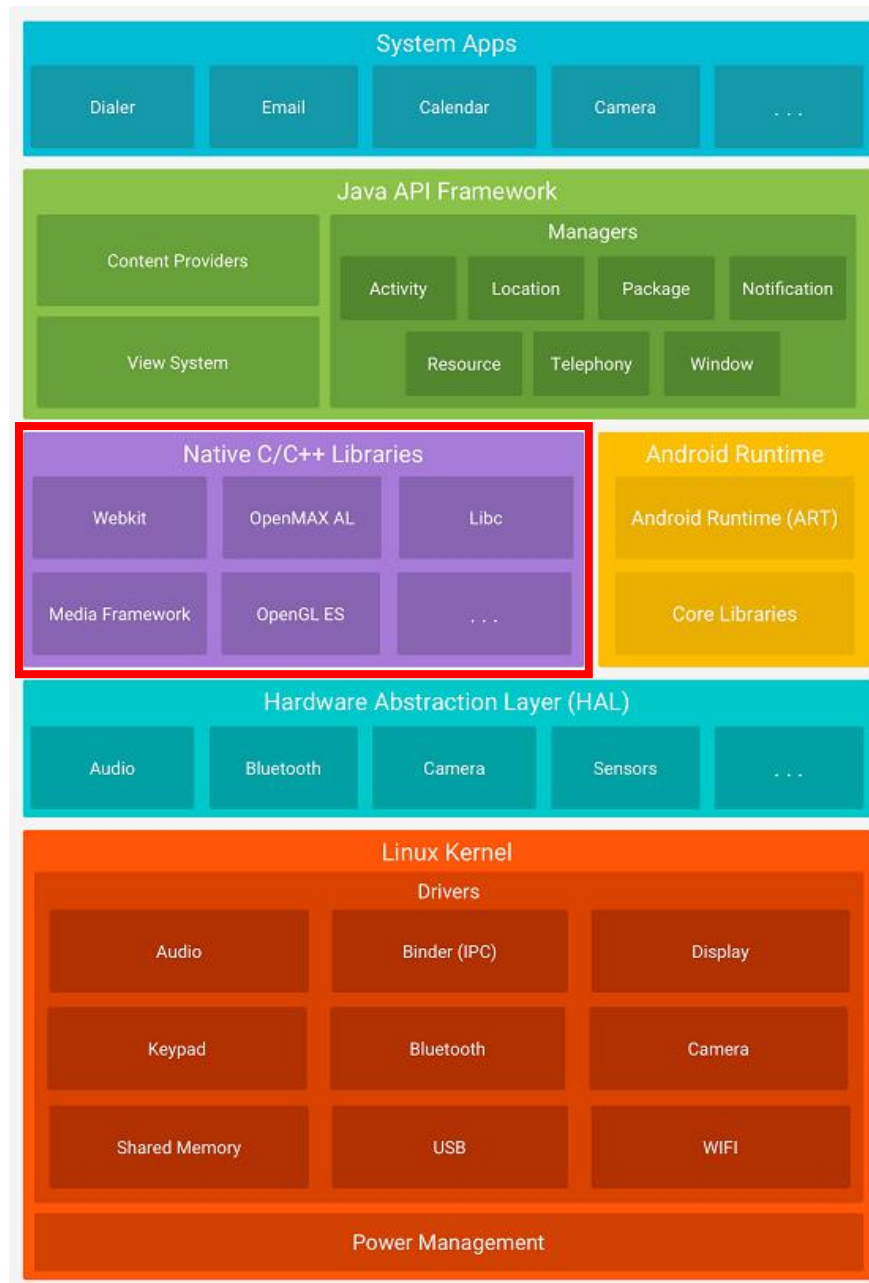


图片来源:
https://developer.android.google.cn/guide/platform/images/android-stack_2x.png

移植 Bionic

libc/	Android C 库, 包含以下主要功能: <ul style="list-style-type: none">● CRT (crtbegin, crtend)● 系统调用封装● stdio/stdlib/unistd/string 等 POSIX 标准 C 库函数● POSIX 线程库	libc.so, libc.a
libm/	Math 库	libm.so, libm.a
libdl/	dynamic linker interface 库 (dlopen/dlclose/.....)	libdl.so
libstdc++	提供 C++ ABI support functions 供 clang++ 调用。	libstdc++.so
linker	动态链接器	linker/linker64
tests	基于 gtest 构建的自动化单元测试程序 (包括单元测试集)	
benchmarks	性能测试程序	

PLCT Lab AOSP 12 移植工作计划



图片来源:

https://developer.android.google.cn/guide/platform/images/android-stack_2x.png

移植 NDK

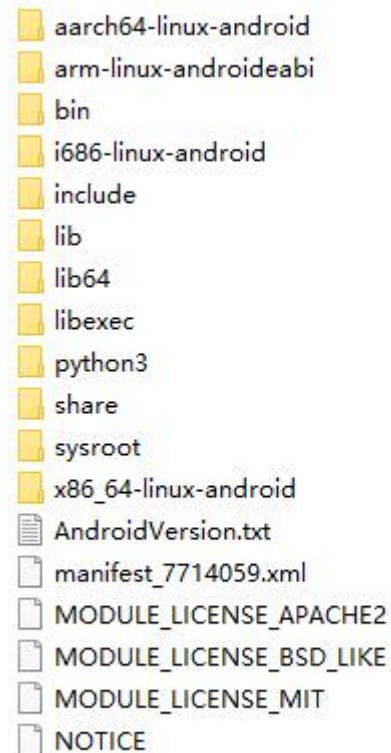
NDK 是一套包含了众多平台库用于 C/C++ 程序开发工具套件，包括了

- 编译工具链
- platform libraries
 - Core C/C++ libraries (libc/libc++/liblog/libz/...)
 - Graphics 图像处理 (libGLES/libEGL/libvulkan/...)
 - Camera 摄像相关 (libcamera2ndk)
 - Media 多媒体相关 (libmediandk/libOpenMAXAL)
 - Android native application APIs(libandroid/libnativewindow)
 - Audio 音频处理 (libaaudio/libOpenSLES)
 - Neural Networks API (libneuralnetworks)
- 头文件

移植 NDK

NDK 是一套包含了众多平台库用于 C/C++ 程序开发工具套件，包括了

- **编译工具链**
- platform libraries
 - Core C/C++ libraries (libc/libc++/liblog/libz/...)
 - Graphics 图像处理 (libGLES/libEGL/libvulkan/...)
 - Camera 摄像相关 (libcamera2ndk)
 - Media 多媒体相关 (libmediandk/libOpenMAXAL)
 - Android native application APIs(libandroid/libnativewindow)
 - Audio 音频处理 (libaaudio/libOpenSLES)
 - Neural Networks API (libneuralnetworks)
- 头文件



移植 NDK

NDK 是一套包含了众多平台库用于 C/C++ 程序开发工具套件，包括了

- 编译工具链
- **platform libraries**
 - Core C/C++ libraries (libc/libc++/liblog/libz/...)
 - Graphics 图像处理 (libGLES/libEGL/libvulkan/...)
 - Camera 摄像相关 (libcamera2ndk)
 - Media 多媒体相关 (libmediandk/libOpenMAXAL)
 - Android native application APIs(libandroid/libnativewindow)
 - Audio 音频处理 (libaaudio/libOpenSLES)
 - Neural Networks API (libneuralnetworks)
- 头文件

▢ crtbegin_dynamic.o	▢ crtbegin_so.o	▢ crtbegin_static.o
▢ crtend_android.o	▢ crtend_so.o	▢ libaaudio.so
▢ libamidi.so	▢ libandroid.so	▢ libbinder_ndk.so
▢ libc.a	▢ libc.so	▢ libc++.a
▢ libc++.so	▢ libcamera2ndk.so	▢ libcompiler_rt-extras.a
▢ libdl.a	▢ libdl.so	▢ libEGL.so
▢ libGLESv1_CM.so	▢ libGLESv2.so	▢ libGLESv3.so
▢ libicu.so	▢ libjnigraphics.so	▢ liblog.so
▢ libm.a	▢ libm.so	▢ libmediandk.so
▢ libnativehelper.so	▢ libnativewindow.so	▢ libneuralnetworks.so
▢ libOpenMAXAL.so	▢ libOpenSLES.so	▢ libstdc++.a
▢ libstdc++.so	▢ libsync.so	▢ libvulkan.so
▢ libz.a	▢ libz.so	

移植 NDK

NDK 是一套包含了众多平台库用于 C/C++ 程序开发工具套件, 包括了

- 编译工具链
- platform libraries
 - Core C/C++ libraries (libc/libc++/liblog/libz/...)
 - Graphics 图像处理 (libGLES/libEGL/libvulkan/...)
 - Camera 摄像相关 (libcamera2ndk)
 - Media 多媒体相关 (libmediandk/libOpenMAXAL)
 - Android native application APIs(libandroid/libnativewindow)
 - Audio 音频处理 (libaaudio/libOpenSLES)
 - Neural Networks API (libneuralnetworks)
- 头文件

aarch64-linux-android
android
asm-generic
camera
GLES
i686-linux-android
media
net
OMXAL
SLES
unicode
x86_64-linux-android
alloca.h
byteswap.h
ctype.h
elf.h
errno.h
features.h
fts.h
glob.h
ifaddrs.h
langinfo.h
limits.h
malloc.h
mntent.h
paths.h
pty.h
resolv.h
semaphore.h
spawn.h
stdio.h
string.h
sysexit.h
termio.h
time.h
uconfig_local.h

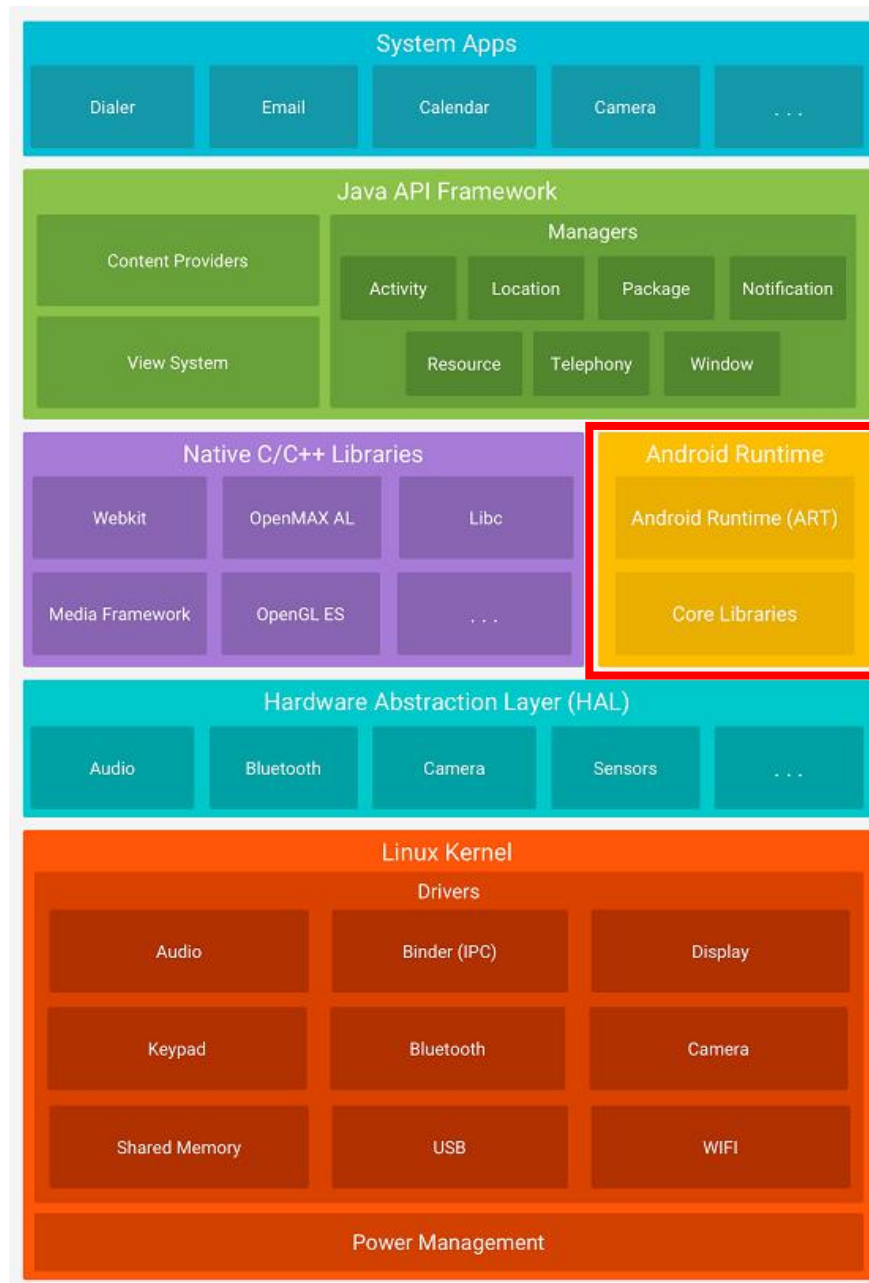
aaudio
arm-linux-androideabi
bits
drm
GLES2
KHR
misc
netinet
rdma
sound
video
xen
ar.h
complex.h
dirent.h
endian.h
error.h
fenv.h
ftw.h
grp.h
inttypes.h
lastlog.h
link.h
math.h
netdb.h
poll.h
pwd.h
sched.h
setjmp.h
stdatomic.h
stdio_ext.h
strings.h
syslog.h
termios.h
time64.h
ucontext.h

amidi
arpa
c++
EGL
GLES3
linux
mtd
netpacket
scsi
sys
vulkan
.clang-format
assert.h
cpio.h
dlfcn.h
err.h
fcntl.h
fnmatch.h
getopt.h
iconv.h
jni.h
libgen.h
locale.h
memory.h
nl_types.h
pthread.h
regex.h
search.h
signal.h
stdint.h
stdlib.h
syscall.h
tar.h
threads.h
uchar.h
unistd.h

制作 rootfs 以及最小系统启动

- 内核的启动
- 文件系统的加载
- init 运行以及各类初始化 rc 脚本
- 启用 selinux相关环境(具体调试时可能需要先关闭), 启动 rc 脚本注册的各种服务 (此时不包括 Zygote与 Java服务)
- 初始化命令行

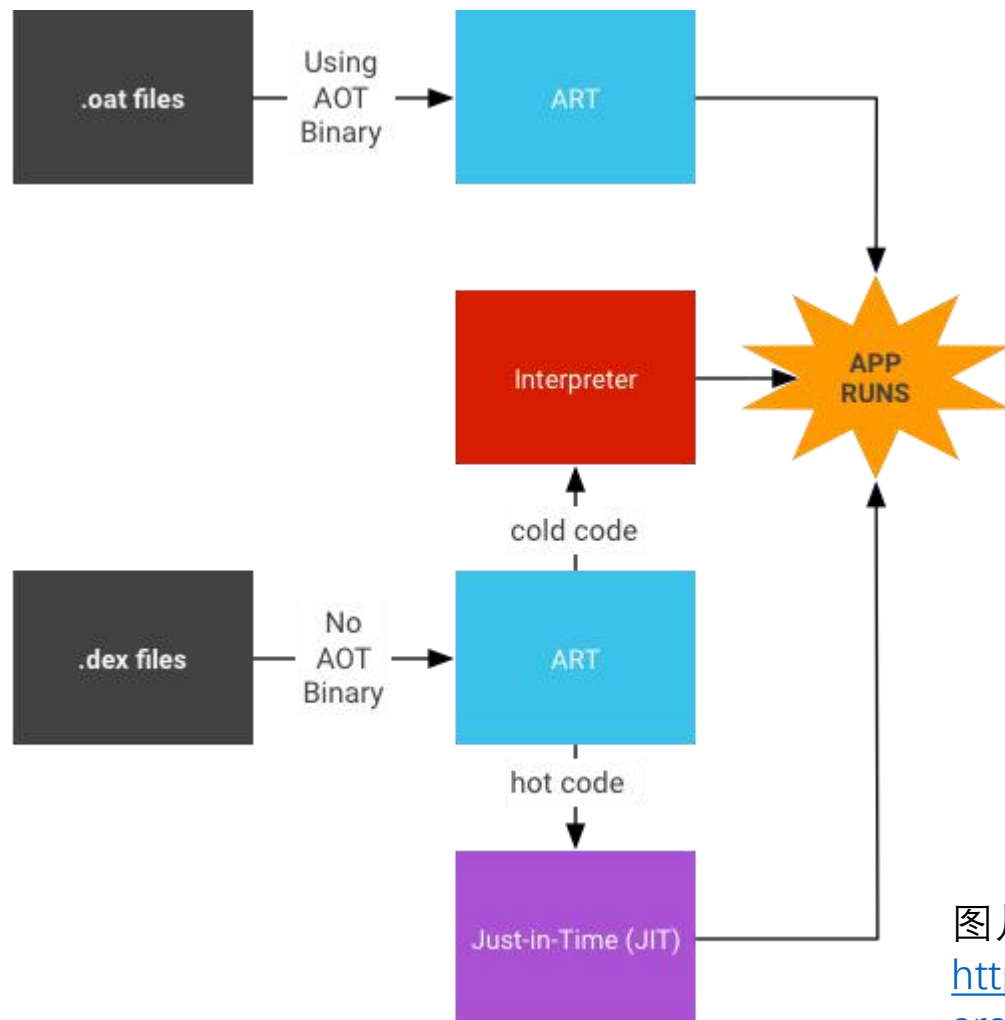
PLCT Lab AOSP 12 移植工作计划



图片来源:

https://developer.android.google.cn/guide/platform/images/android-stack_2x.png

移植 ART



主要有三部分内容是移植 RISC-V 的关键：

- Interpreter: dex 字节码解释器，对未编译的 Android 的 dex 字节码采用解释方式执行。
- JIT 编译器: dex 字节码运行过程中，ART 会将热点方法记录下来，并生成 profiling 信息。在设备 idle 或者充电时 JIT 编译器会根据 profiling 信息对热点方法进行编译。
- AOT (Ahead-Of-Time) 编译器: 即 .oat 文件的生成工具 dex2oat。AOT 编译器作用是将 dex 字节码编译成二进制格式的 oat 文件。缺省配置下，在编译时或者安装时，会调用 dex2oat 来完成编译，加速程序的执行。

图片来源：

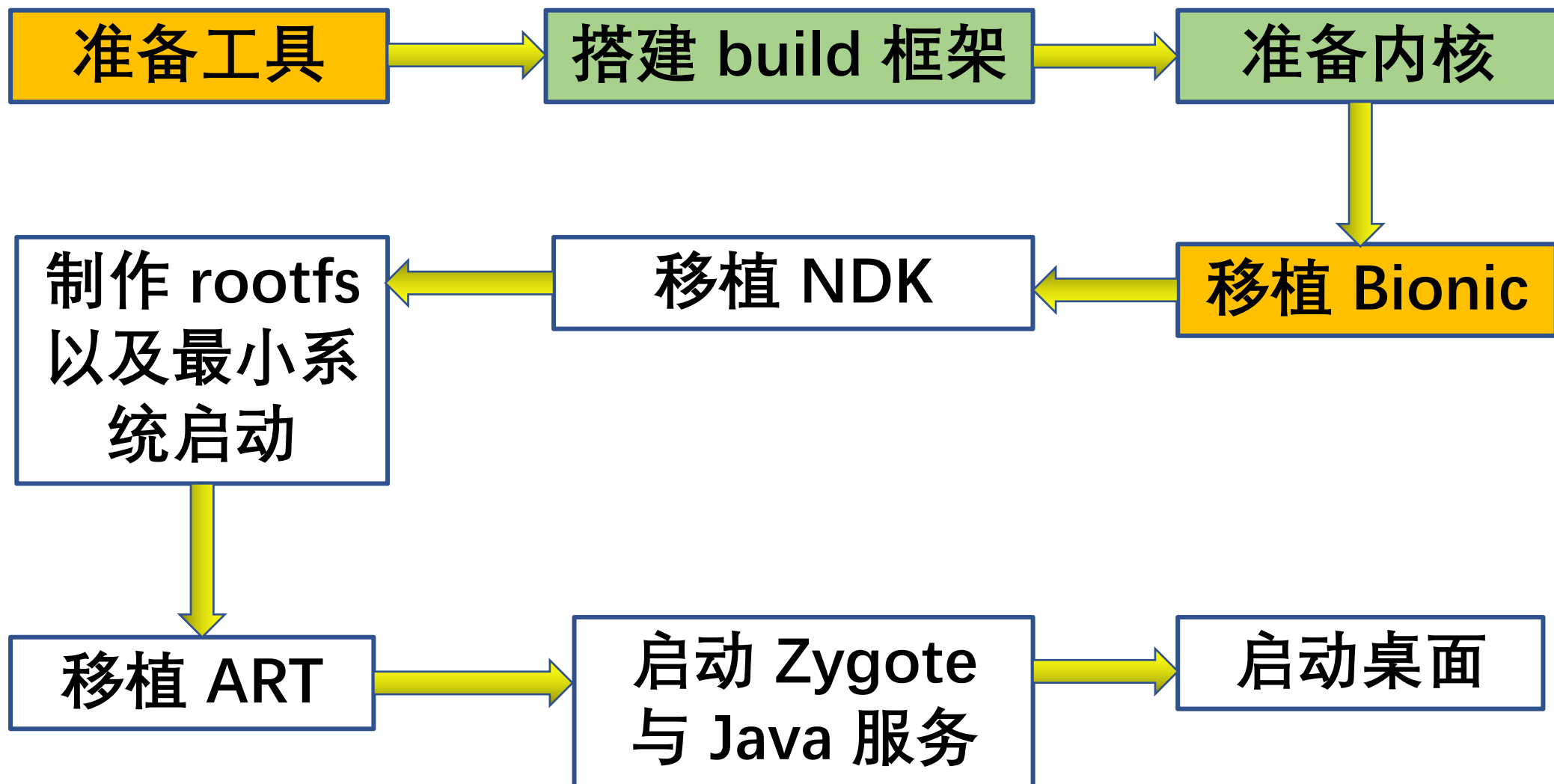
<https://source.android.google.cn/devices/tech/dalvik/images/jit-arch.png>

启动 Zygote 与 Java 服务

zygote 是整个系统创建新进程的核心进程。zygote 进程在内部会先启动 ART 虚拟机，继而加载一些必要的系统资源和系统类，最后进入一种监听状态。在之后的运作中，当其他系统模块（比如 AMS）希望创建新进程时，只需向 zygote 进程发出请求，zygote 进程监听到该请求后，会相应地 fork 出新的进程，于是这个新进程在初生之时，就先天具有了自己的虚拟机以及系统资源。

启动桌面

在原生程序和 Java 服务都调试稳定的理想状况下，将启动到桌面环境。




PLCT Lab AOSP 12 移植工作计划

The screenshot shows the GitHub profile page for the organization 'aosp-riscv'. The page has a dark theme. At the top, there's a navigation bar with links for Pulls, Issues, Marketplace, and Explore. Below the navigation bar, the organization's name 'aosp-riscv' is displayed next to its logo, which features a green Android head and a stylized 'R' and 'V'. A large orange 'Welcome~' text is prominently displayed. Below the welcome message, there's a row of tabs: Overview (selected), Repositories (39), Packages, People (9), Teams, and Projects (1). The main content area is divided into three columns. The left column has a 'Pinned' section featuring a repository named 'working-group' with a description: 'meta repo for AOSP for RISC-V Project. General tasks and issues here.' It shows 55 stars and 6 forks. Below this is a 'Repositories' section with a search bar and filters for Type, Language, and Sort, along with a 'New' button. The right column has a 'People' section showing avatars of team members and an 'Invite someone' button. At the bottom right, there's a 'Top languages' section with colored circles representing C++, Python, C, Java, and Go.


https://github.com/aosp-riscv

Search or jump to... / Pulls Issues Marketplace Explore

 aosp-riscv **Welcome~**

Overview Repositories 39 Packages People 9 Teams Projects 1

Pinned Customize your pins


 **working-group** Public

meta repo for AOSP for RISC-V Project.
General tasks and issues here.

☆ 55 🍴 6

Repositories

Find a repository...

Type Language Sort 

People

Invite someone

Top languages

● C++ ● Python ● C
● Java ● Go

谢谢

欢迎交流合作