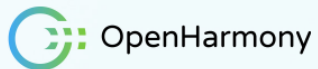




开放原子开源基金会  
OPENATOM FOUNDATION



OpenHarmony

# 技术筑生态 智联赢未来

第二届开放原子开源基金会OpenHarmony技术大会

2023.11.04 | 中国·北京

主办单位：OpenHarmony项目群技术指导委员会（TSC）

合作单位：华为、润开鸿、九联开鸿、软通动力、深开鸿

合作媒体：电子发烧友、51CTO、SegmentFault 思否、黄大年茶思屋科技网站、稀土掘金

# 基于飞腾CPU和AMDGPU的多媒体娱乐主机实践



许源申

高级开源工程师

湖南开鸿智谷数字产业发展有限公司

## 目录

- 1、国产信创硬件体系介绍
- 2、娱乐主机硬件平台分析
- 3、飞腾D2000适配经验分享
- 4、AMD显卡适配经验分享
- 5、基于OpenHarmony上的cocos引擎游戏开发过程

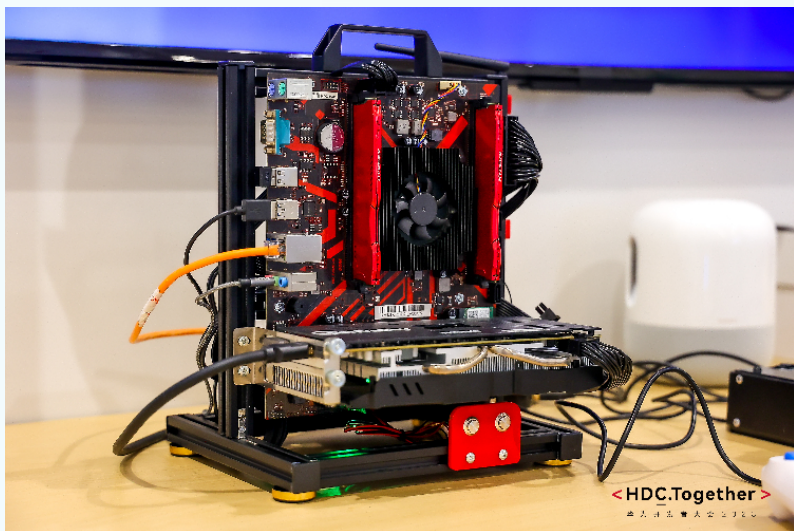
# 一、国产信创硬件体系介绍

# 信创硬件体系介绍

	龙芯	鲲鹏	飞腾	海光	兆芯	申威
研发单位	中科院计算所	华为	天津飞腾	天津海光	上海兆芯	江南计算所
指令集体系	MIPS/LoongArch	ARM	ARM	X86(AMD)	X86(VIA)	ALPHA
架构来源	指令集授权+自研	指令集授权	指令集授权	IP授权	威盛合资	指令集授权+自研
代表产品	龙芯1/龙芯2/龙芯3	鲲鹏920	D2000/S2500	Hygon C86-7285	ZXC KX-6000/KH-40000	申威 SW1621/SW26010
优势	MIPS架构功耗低，终端芯片不错。LoongArch自主性高。	ARM服务器芯片中性能最佳	终端芯片和服务器芯片整体性能较好	基于AMD最新的Zen架构，性能高	兼容性强，终端领域应用可以无缝对接，得到上海市资金扶持	不依赖商业机构授权，自主性较高
劣势	MIPS指令集已经停止发展。LoongArch生态能力弱。	兼容性和生态需要进一步打造	兼容性和生态需要进一步打造；商用性能需要进一步提升	因被列入美国实体名单，技术持续性较差	存在知识产权瑕疵，是否自主可控存疑；没有服务器芯片	Alpha指令集停止更新；主要用于超算和军队，市场和生态能力弱

## 二、娱乐主机硬件平台分析

## 飞腾D2000硬件配置



### 硬件规格:

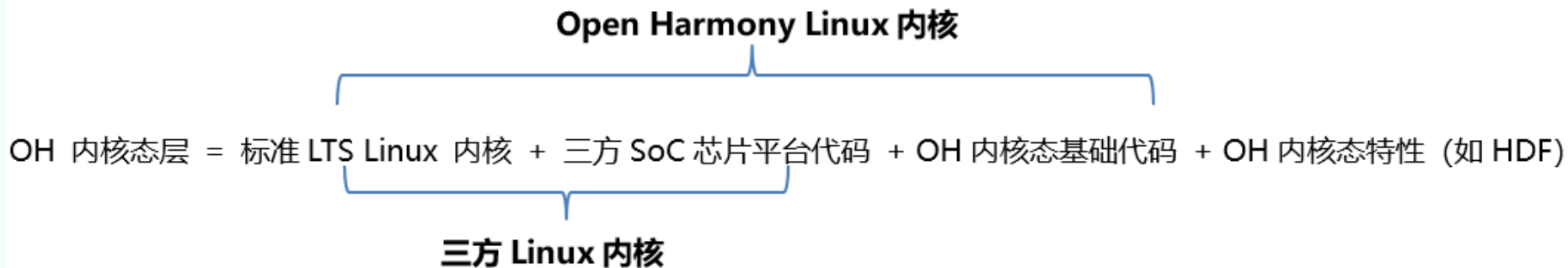
- CPU: 飞腾D2000
- GPU: AMD RX580 8G
- RAM: 光威16G DDR4 2666
- 摇杆+推杆

## 三、飞腾D2000适配经验分享

- 1.内核移植启动适配
- 2.显示能力适配
- 3.HDC连接能力适配



## 内核移植方案选型



方法一：OH 内核态层 = 三方 Linux 内核 + OH 内核态基础代码 + OH 内核态特性 (如 HDF, HMDFS 等)

方法二：OH 内核态层 = OH Linux 内核 + OH 内核态特性 (如 HDF, HMDFS 等)

## 内核移植启动适配（三方内核快速适配）

### 移植前准备

- 串口接通确认
- ko内核模块加载列表确认 (lsmod)
- 内核编译配置文件确认 (zcat /proc/config.gz)

### 驱动移植

- accesstokenid
- hilog
- hievent
- hisysevent
- zerohung
- hungtask
- blackbox
- android\_binder
- hdf

### 文件系统移植

- hmdfs
- sharefs
- epfs

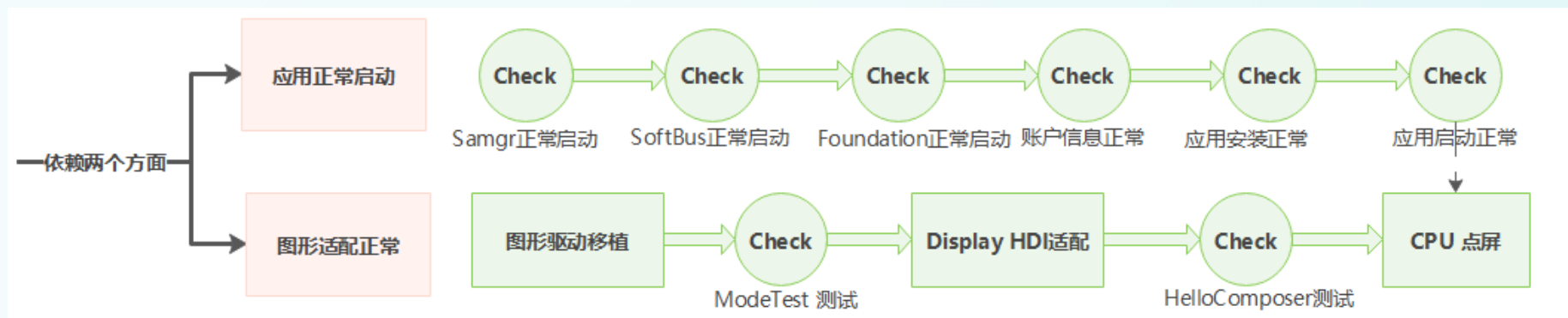
### 磁盘分区

- boot
- system
- vendor
- userdata

### 内核启动配置

- grub2
- 设备树
- boot参数

# CPU渲染显示适配



适配要点：

驱动选型

radeon

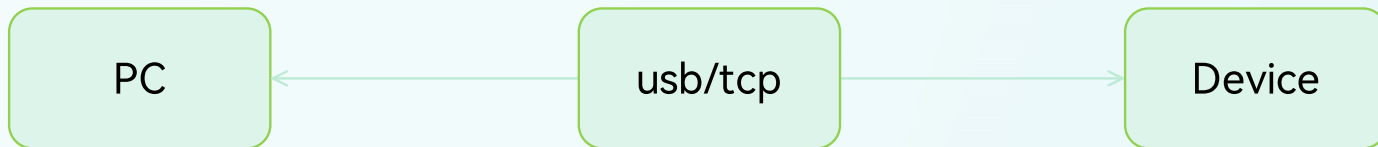
amdgpu

显卡固件获取

模块参数传递

radeon.si\_support=0 amdgpu.si\_support=1 amdgpu.dc=1

## HDC适配



### HDC Tcp网络连接

方法一：在init配置文件中使用setparam进行参数设置"`setparam persist.hdc.mode tcp`", "`setparam persist.hdc.port 5555`"

方法二：在para配置文件中加入`persist.hdc.mode=tcp`,  
`persist.hdc.port=5555`

## 四、AMD显卡适配经验分享

1. Mesa 3D开源显卡驱动编译
2. GPU合成适配

# 显卡选型

Name of GPU series	Vendor	March	3D Rage	Rage Pro	Rage 120i	R100	R200	R300	R400	R500	R600	RV670	R700	Evergreen	Northern Islands	Southern Islands	Sea Islands	Volcanic Islands	Arctic Islands/Polaris	Vega	Navit 1x	Navit 2x	Navit 3x					
Released	1985	1991	Apr 1996	Mar 1997	Aug 1998	Apr 2000	Aug 2000	Sep 2002	May 2004	Oct 2005	May 2007	Nov 2007	Jan 2008	Sep 2009	Oct 2010	Jun 2012	Sep 2013	Jun 2015	Jan 2016, Apr 2017, Aug 2019	Jan 2017, Feb 2019	Jul 2019	Nov 2020	Dec 2022					
Marketing Name	Wonder	March	3D Rage	Rage Pro	Rage 128	Radson 7000	Radson 8000	Radson 9000	Radson X700/X800	Radson X1000	Radson HD 2000	Radson HD 3000	Radson HD 4000	Radson HD 5000	Radson HD 6000	Radson HD 7000	Radson 200	Radson 300	Radson 400/500/600	Radson 100 Vega, Radson VII	Radson RX 5000	Radson RX 6000	Radson RX 7000					
AMD support Kind	X															✓												
Instruction set architecture	Not publicly known										TeraScale instruction set					GCN instruction set					RDNA instruction set							
Microarchitecture	Not publicly known										TeraScale 1 (RV600)		TeraScale 2 (RV630)	TeraScale 2 (RV630) up to 8KB	TeraScale 3 (RV670) in 8KB		GCN 1st gen	GCN 2nd gen	GCN 3rd gen	GCN 4th gen		GCN 5th gen	RDNA	RDNA 2	RDNA 3			
Type	Fixed pipeline <sup>[1]</sup>					Programmable pixel & vertex pipelines					Unified shader model																	
Direct3D	—	5.0	5.0	7.0	8.1	9.0	9.0	9.0c	10.0	10.0	10.0	10.0	10.1	11 (11_0)	11 (11_0)	11 (11_0)	11 (12_0)	11 (12_0)	11 (12_0)	11 (12_0)	11 (12_0)	11 (12_0)	11 (12_0)					
Shader model	—	—	—	—	—	1.4	2.0+	2.0b	3.0	4.0	4.0	—	—	5.0	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1					
OpenGL	—	—	1.1	1.2	1.3	—	—	—	—	—	—	—	—	4.5 (on Linux 4.5 (Mesa 3D 21.0)) <sup>[1][2]</sup>	4.6 (on Linux 4.6 (Mesa 3D 21.0))													
Vulkan																1.0 (Win 7+ or Mesa 11+)	1.2 (Adrenalin 20.1.2, Linux Mesa 3D 20.0) 1.1 (Linux 4 and above (with Adrenalin 22.1.2, Mesa 22.0))										1.3	
OpenCL	—										Close to Metal	1.1 (on Mesa 10)		1.2 (on Linux 1.1) (no Image support on driver, with by nvidia) with Mesa 30.1.2+ on GCN 1/Gcn)		2.0+ (Adrenalin driver on Win7+) (on Linux ROCm, Linux Mesa 3D 1.21 (no Image support in driver, but in nvidia) with Mesa 31.2.0+ and 3.0 with AMD drivers or AMD ROCm), 20+ open 2.2 win 10+ and Linux Metal 5.0+					2.2+ and 3.0 windows 8.1+ and Linux ROCm 3.0.1 (Mesa 31 nvidia 1.2+ and 3.0 (2.1+ and 2.2 win 10+ and Linux Metal 5.0+)) <sup>[1][2][3]</sup>							

## Mesa 3D开源显卡驱动编译适配要点

- 1) 修改OH默认编译脚本，修改GPU名为amdgpu
- 2) 参考makefile，添加libdrm\_radeon，libdrm\_amdgpu编译配置
- 3) 移植缺失三方库libelf等
- 4) 重编OH定制的LLVM工具链，添加生成目标AMDGPU
- 5) 编译LLVM动态运行库libLLVM-12.so

```

build.py M × M CMakeLists.txt M
toolchain > llvm-project > llvm-build > build.py > BuildConfig > _init_
42 self.xunit_xml_output = args.xunit_xml_output
43 self.enable_assertions = args.enable_assertions
44 self.need_libs = self.do_build and 'libs' not in args.no_build
45 self.need_lldb_mi = self.do_build and 'lldb-mi' not in args.no_build
46 self.need_lldb_server = self.do_build and 'lldb-server' not in args.no_build
47
48 self.no_build_arm = args.skip_build or args.no_build_arm
49 self.no_build_aarch64 = args.skip_build or args.no_build_aarch64
50 self.no_build_riscv64 = args.skip_build or args.no_build_riscv64
51 self.no_build_mipsel = args.skip_build or args.no_build_mipsel
52 self.no_build_x86_64 = args.skip_build or args.no_build_x86_64
53
54 self.discover_paths()
55
56 self.TARGETS = 'AArch64;ARM;BPF;Mips;RISCV;X86;AMDGPU'
57 self.ORIG_ENV = dict(os.environ)
58 self.VERSION = None # autodetected

```

```

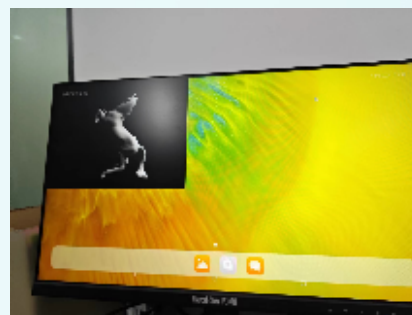
build.py M M CMakeLists.txt M ×
toolchain > llvm-project > llvm > tools > llvm-shlib > M CMakeLists.txt
27 add_custom_target(lldLLVMExports DEPENDS ${LLVM_EXPORTED_SYMBOL_FILE})
28 endif()
29
30 if (LLVM_LINK_LLVM_DYLIB)
31 set(INSTALL_WITH_TOOLCHAIN INSTALL_WITH_TOOLCHAIN)
32 endif()
33 add_llvm_library(LLVM SHARED DISABLE_LLVM_LINK_LLVM_DYLIB SONAME ${INSTALL_WITH_TOOLCHAIN} ${SOURCES})
34
35 list(REMOVE_DUPLICATES LIB_NAMES)
36 if((${CMAKE_SYSTEM_NAME} STREQUAL "Linux") OR (MINGW) OR (HAIKU)
37 OR (${CMAKE_SYSTEM_NAME} STREQUAL "FreeBSD")
38 OR (${CMAKE_SYSTEM_NAME} STREQUAL "GNU")
39 OR (${CMAKE_SYSTEM_NAME} STREQUAL "OpenBSD")
40 OR (${CMAKE_SYSTEM_NAME} STREQUAL "Fuchsia")
41 OR (${CMAKE_SYSTEM_NAME} STREQUAL "DragonFly")
42 OR (${CMAKE_SYSTEM_NAME} STREQUAL "OHOS"))
43 OR (${CMAKE_SYSTEM_NAME} STREQUAL "SunOS")) # FIXME: It should be "GNU ld for elf"
44 configure_file(
45 ${CMAKE_CURRENT_SOURCE_DIR}/simple_version_script.map.in

```

# GPU渲染合成适配要点



1. 在cpu点屏的基础上配置系统编译导入Mesa3D驱动
2. 通过glmark2测试GPU调用是否正常
3. 修改编译配置开启OH的GPU支持





## 五、基于OpenHarmony上的Cocos引擎游戏开发

1. 环境搭建与配置
2. 构建导出OpenHarmony工程
3. 编译运行

# Cocos引擎游戏开发环境搭建与配置

- 1) 下载安装DevEco Studio
- 2) 下载安装CocosCreator 3.8
- 3) 下载每日构建的SDK替换OpenHarmony SDK

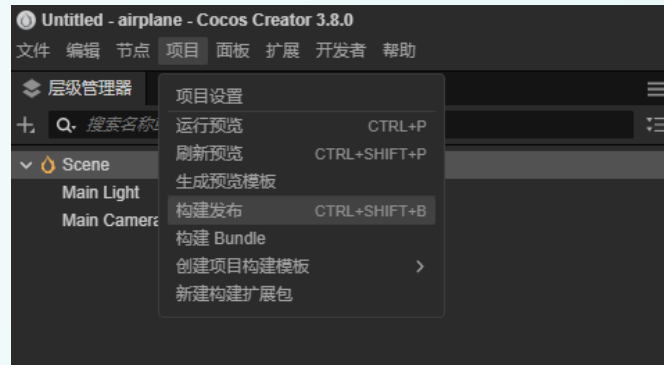
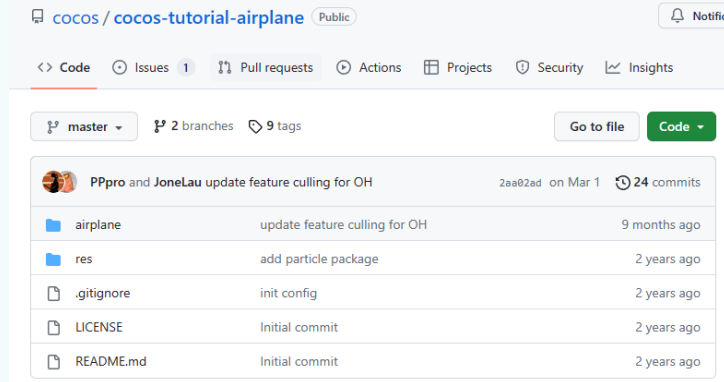
## 支持情况

引擎版本	OpenHarmony 版本	说明
v3.6.1 ~ v3.7.3	OpenHarmony 3.2 beta	该版本在社区公测，因此无法从 Dashboard 直接下载，开发者请从 <a href="#">社区下载</a>
v3.8	OpenHarmony 3.2, OpenHarmony 4.0	正式版，开发者可以选择从 Dashboard 中直接下载该编辑器版本。

流水线名称	设备型号	构建失败次数	构建总数	构建成功率	部署成功率	TD0	XTS	FUZZ	耗时	运行周期	版本下载地址
hispark_pegasus	Mini	0	1	100%	100%	--	--	--	73min	1次/天	<a href="#">下载链接</a>
hispark_taurus_LiteOS	Small	0	1	100%	--	--	--	--	8min	1次/天	<a href="#">下载链接</a>
ohos-sdk-full	Standard	0	1	100%	--	--	--	--	16min	1次/天	<a href="#">下载链接</a>
hispark_taurus_Linux	Small	0	1	100%	--	--	--	--	10min	1次/天	<a href="#">下载链接</a>
ohos-sdk-public	Standard	0	1	100%	--	--	--	--	17min	1次/天	<a href="#">下载链接</a>
dayu200	Standard	0	1	100%	100%	--	--	--	197min	1次/天	<a href="#">下载链接</a>

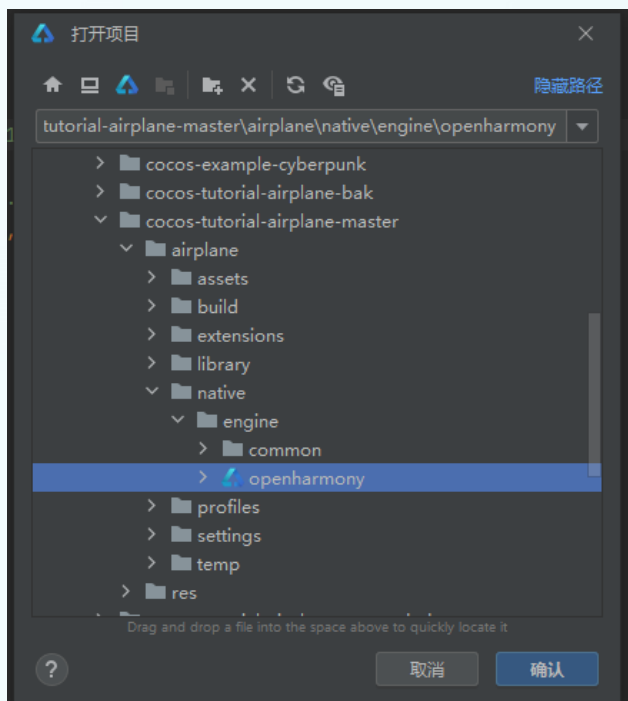
# CocosCreator导出OpenHarmony工程

下载cocos项目，导入CocosCreator，在CocosCreator中选择项目-构建发布导出OH工程



## DevEco Studio编译运行

在DevEco Studio中打开Cocos导出的OH项目  
配置项目签名，连接oh设备，编译运行



Thank you.

