

# AOSCC 2024 | 欢迎

衰落的王国

---

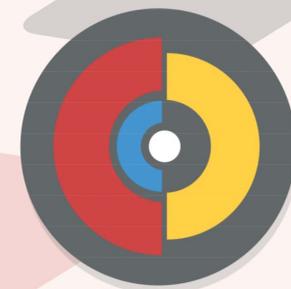
## MIPS 发行版及 基础设施的现状与无奈

Henry Chen

chenx97@aosc.io

杨欣辉

cyan@cyano.uk



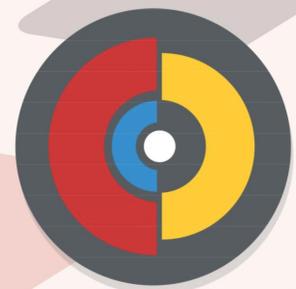
# 曾经辉煌的 MIPS

90 年代的 MIPS 几乎占领了工作站领域，且在游戏主机领域也有不少份额：

- SGI 图形工作站（Indigo、Indy、O2、Octane 等均使用 MIPS 架构的处理器）
  - SGI 机器参与渲染了著名的《玩具总动员》电影
- 任天堂 64 游戏主机
- 索尼 PlayStation、PlayStation 2 和 PSP



由上到下：  
Octane, Indigo 和 Indy  
图源：Wikipedia



# 曾经辉煌的 MIPS

- 最初被指定为 Windows NT 默认架构之一
- 即便 x86 逐渐成为主要平台，当时的开发团队始终维护着 MIPS 构建，且要求发版时能通过大部分测试
- Dave Cutler 自称“Mr. Mips”且一直对 MIPS 持有热情

G. P. Zachary, *Show-Stopper!: The Breakneck Race to Create Windows NT and the next Generation at Microsoft*. New York, Toronto, New York: Free Press ; Maxwell Macmillan Canada ; Maxwell Macmillan International, 1994.



# 衰落的王国：历史

Debian: 自 Debian 3.0 起加入正式 MIPS32 移植，  
自 Debian 9 起正式加入小尾端 MIPS64 移植

- MIPS32 大小端移植已经退出维护
- 尽管小尾端 MIPS64 移植还在维护，但维护成本愈来愈高
  - 由于设施资源不足，该架构已经与主线脱节<sup>[1]</sup>

[1]: <https://lists.debian.org/debian-devel-announce/2023/11/msg00005.html>



# 衰落的王国：历史

Gentoo: MIPS 移植自 2003 年就有活动

- 该移植的 Stage 3 tarball 涵盖大小端及 MIPS32/64 和 MIPS II
- 该项目仍在持续维护中，但基本处于无人看管的状态
  - 社区向广大 MIPS 爱好者呼救请求继续维护<sup>[1]</sup>
- 该移植依旧可以在 SGI Indy 等机器上运行！

[1]: <https://marc.info/?l=gentoo-dev&m=171965705309709>



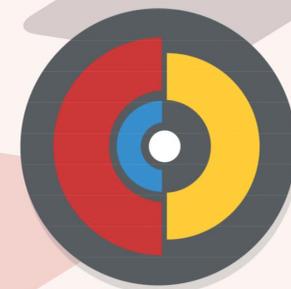
# 衰落的王国：历史

- OpenWrt: 自古以来，仍在继续
- NetBSD: 拥有各类 MIPS 移植
  - 支持了大量我们想都不敢想的平台，比如 SGI 工作站
- 除此之外，几乎没有别的发行版做 MIPS 移植



# 衰落的王国：现状

- 几乎只能在嵌入式领域见到该架构
- 来自其他架构（如 ARM）的竞争
  - 在路由器等嵌入式领域尤为明显
- 市场份额持续走低
- 开发兴趣低迷



# 衰落的王国：改革

MIPS Technologies 在 2014 年发布了 MIPS32/64 Release 6，该版本删除了 Hi/Lo 寄存器、删除了一些不常用的指令、新增及优化了一些跳转指令

- 一定程度上在硬件方面改善了历史包袱
- 因此 R6 与之前的版本不兼容



# 衰落的王国：改革

- 架构名称及定义上没有显示不兼容性
  - mips(64r2)el 和 mips(64)r6el 在内核、编译器主架构中都是 mips(64)el
  - 编译器内的角色是 `-march (-mcpu)`
- 并没有 MIPS R6 硬件流入市场
  - 开发工作只能借助 QEMU 进行



# 短暂的繁荣：龙芯 3 号

- 龙芯 3 号问世，MIPS 又一次有了桌面应用场景
- MIPS64 R2 兼容硬件大量流入市场
- 部分软件出现了私有移植
- 然后……龙芯就转向 LoongArch 了
- MIPS 发行版又几乎回到了缥缈的状态——



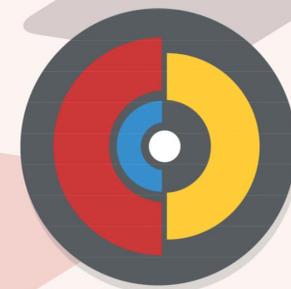
# 短暂的繁荣过后：无奈

- 历史包袱众多
- 用户越来越少
- 一直缺少关键软件移植
- 部分关键软件质量不高



# 在我们手里的是...

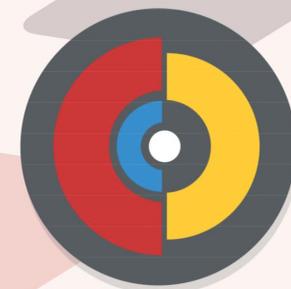
- 安同 OS 有龙芯 3 号和 R6 移植
- 龙芯 3 号较为完整，桌面环境可以满足日常使用
- 但是随着时间的推移，两者的维护愈发困难
- MIPS 发行版的接力棒有一部分已经到我们手中
- 同时还要维护基础设施



# 先了解现状

在我们正式开始修复各大组件的 MIPS 支持前：

- LLVM 原 MIPS 维护者 Simon Atanasyan 因个人原因于 2022 年离开该职位
- LLVM 的 MIPS 后端长期存在 Bug
- 无人理会 MIPS 工单
- Rust 将 MIPS 降级至 Tier 3
  - R6 的 Rust 基本编不出来



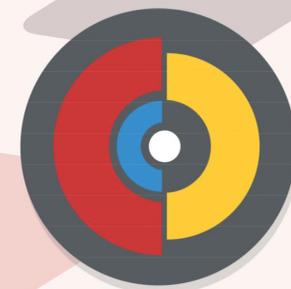
# 改善现状

AOSC 有许多 Rust 周边项目，怎么办？

- 修！
- 于是 R6 有 Rust 用了

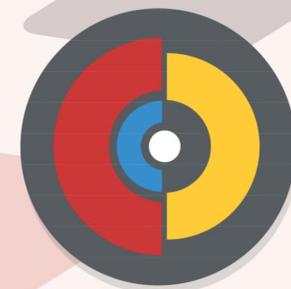
```
Sorting... Done
Full Text Search... Done
rustc/stable,stable,now 1:1.79.0 loongson3 [installed]
  General purpose, multi-paradigm programming language (compiler and runtime)
```

```
Sorting... Done
Full Text Search... Done
rustc/stable 1:1.79.0 mips64r6el
  General purpose, multi-paradigm programming language (compiler and runtime)
```



# 心有余而力不足

- 我们能做的也只是一些小事
- 还有一些残酷的现实摆在我们面前



# 历史包袱：前浪死在沙滩上

- 部分信号码（如 SIGBUS）与其他架构不同
    - 90 年代积极参与 UNIX 生态建设的 RISC 架构的历史背景所致
    - Alpha、SPARC 和 PA-RISC 也是如此<sup>1</sup>
    - 在 MIPS R5 之前，NaN 的处理与其他架构不同
  - 跳转范围和限制与其他 RISC 架构不同
    - 例如到 R6 才有原生的 PCREL 系列指令
  - ABI 定义也保留了那个年代的风格
- 1: 详见手册 `signal(7)` § Signal numbering for standard signals  
当时的 Unix ABI 基本都是 RISC 架构的天下

Signal	x86/ARM most others	Alpha/ SPARC	MIPS
SIGHUP	1	1	1
SIGINT	2	2	2
SIGQUIT	3	3	3
SIGILL	4	4	4
SIGTRAP	5	5	5
SIGABRT	6	6	6
SIGIOT	6	6	6
SIGBUS	7	10	10
SIGEMT	-	7	7
SIGFPE	8	8	8
SIGKILL	9	9	9
SIGUSR1	10	30	16
SIGSEGV	11	11	11
SIGUSR2	12	31	17
SIGPIPE	13	13	13
SIGALRM	14	14	14
SIGTERM	15	15	15
SIGSTKFLT	16	-	-
SIGCHLD	17	20	18
SIGCLD	-	-	18
SIGCONT	18	19	25
SIGSTOP	19	17	23
SIGTSTP	20	18	24
SIGTTIN	21	21	26
SIGTTOU	22	22	27
SIGURG	23	16	21
SIGXCPU	24	24	30
SIGXFSZ	25	25	31
SIGVTALRM	26	26	28
SIGPROF	27	27	29
SIGWINCH	28	28	20
SIGIO	29	23	22
SIGPOLL			
SIGPWR	30	29/-	19
SIGINFO	-	29/-	-
SIGLOST	-	-/29	-

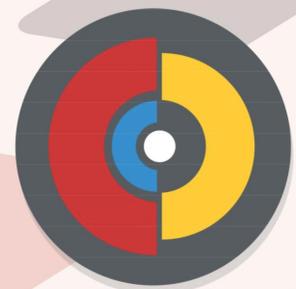
# 历史包袱：过时的 ABI 规范

MIPS 的 ABI 问题主要在于二进制链接方面：自安同 OS 开始维护 MIPS 移植起，就时不时和链接器斗智斗勇

- MIPS SVR4 psABI 规范要求：开了 `-fPIC` 后必须通过 GOT 寻找函数入口
- 编译器默认生成只有 16 位偏移量的寻址，明显无法满足现代二进制的需求
- 不是所有链接器都能得到同等及时的修复



relocation truncated  
to fit:  
R\_MIPS\_CALL16

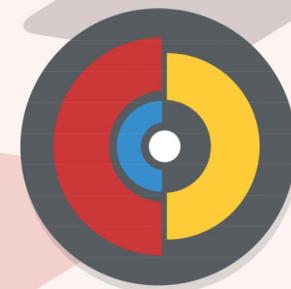
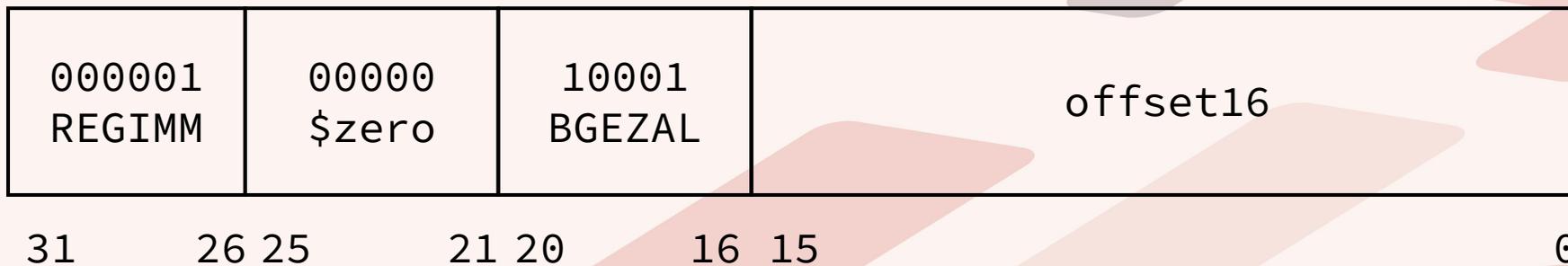


# 历史包袱：过时的 ABI 规范

## JAL

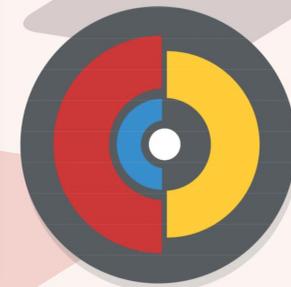
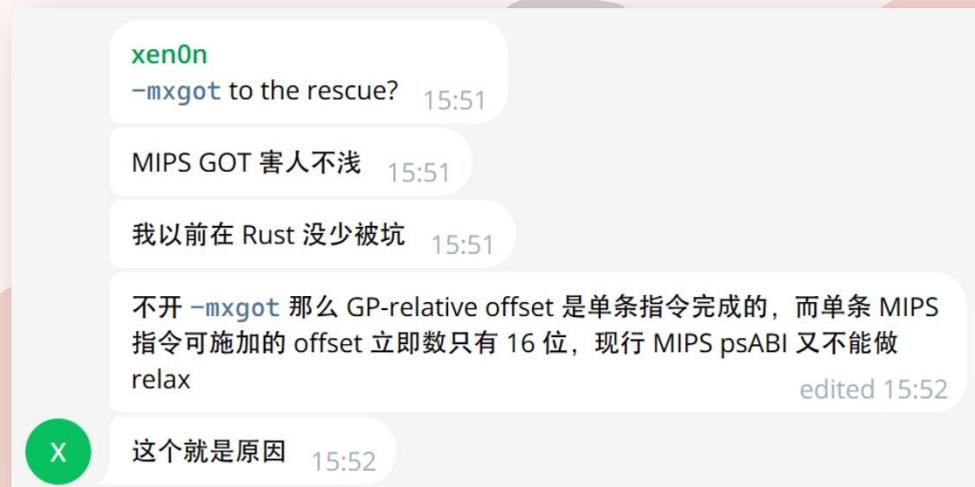


## BAL



# 历史包袱：过时的 ABI 规范

- 可以使用 `-mxgot` 选项，让编译器生成 `R_MIPS_[GOT|CALL]_[HI|LO]16`，扩展 GOT 到 32 位
- 或许可以考虑让工具链默认打开这个选项



# 历史包袱：时常出错的工具链

除之前介绍的链接器问题之外，还有一些工具链相关的问题：

- LLVM 会生成错误的指令
  - Rust 因此无法通过 CI
- LLVM 在编译场景会莫名其妙崩溃
  - 可以在安同 OS 的源码树上看见许多  
`USECLANG__LOONGSON3=0`



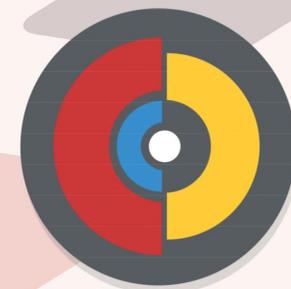
# 历史包袱：破而不立的 MIPS R6

- 没有硬件，只能用 QEMU 起容器造空气
  - 也无法进入实际场景调试
- 不做区分，会无意中链接到针对 R2 的汇编代码
  - Rust 成天因为这个问题苦恼
- LLVM 尝试生成调试信息时会 SIGSEGV
- GDB 也时不时不工作
  - 地址到处飞，但是重跑一下似乎又可以



# 尴尬的结局

- mold 链接器作者表示老子不干了
- 除非有商业支持，不然链接器的坑填起来过于依赖时间的沉淀
- 还是先保证 GCC 和 LLVM 的维护罢（悲）



“

I made a fair amount of effort to try to support MIPS, but it turned out that it is much harder than expected. I punted it instead of making further efforts.

The problem is the MIPS ABI is hostile to the linker in the modern environment.

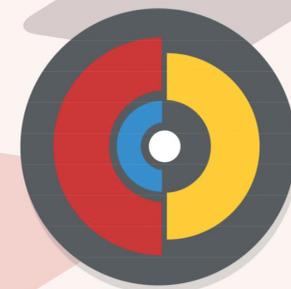
[...] MIPS requires the linker to implement tons of workarounds for its legacy ABI assumptions.

[...] At this point, I don't think it is productive to implement workarounds for the old ABI that is stuck in the 90s.

— Mold 作者,

<https://github.com/rui314/mold/commit/db5fa8a8cd9884fc8b493fd5642d72a11b9a5aff>

”



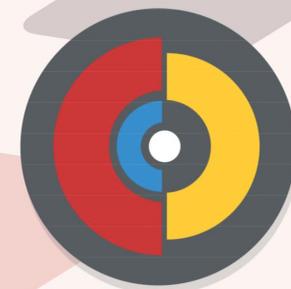
# 衰落的王国：虚无的未来

- 尽管 R6 硬件上改善了弊端，但软件方面依旧过时
  - 没有端上改善了跳转限制、能合理利用 PCREL 系列指令的新 ABI
  - 没有使 Linux 信号码与常见架构同步，依然是独一份定义
- 破坏了兼容性却没有甩掉包袱
- **R6 没有硬件！**
- MIPS Technologies 在 2021 年宣布停止 MIPS 架构研发，转投 RISC-V



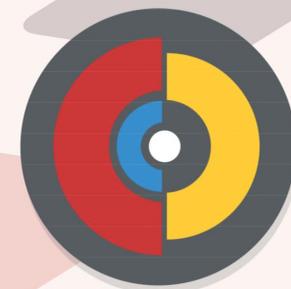
# 持续维护

- 芯联芯同事：维护 MIPS 工具链
  - 参与修复了数个 LLVM Bug
- 依旧热衷于 MIPS 的开发者、贡献者们：不会放弃
  - Gentoo 的呼救也得以响应
- 苏运强：Debian 还不能死
  - 同时一直在维护 GCC 和 Binutils

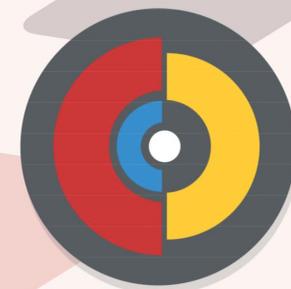


# 活在当下

- 同事和社区贡献者将继续努力
- 尽量把小事做好，让 MIPS 发挥余热
- 安同 OS 的 MIPS 移植会持续进行



**Q & A**



本 PPT 和演讲得到了一些社区好友和同事的帮助，在此鸣谢：

- 白铭骢
- Jiaxun Yang (FlyGoat)
- WANG Xuerui (xen0n)
- YunQiang Su (wzssyqa)

# Thanks

