



GOTC 2023

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

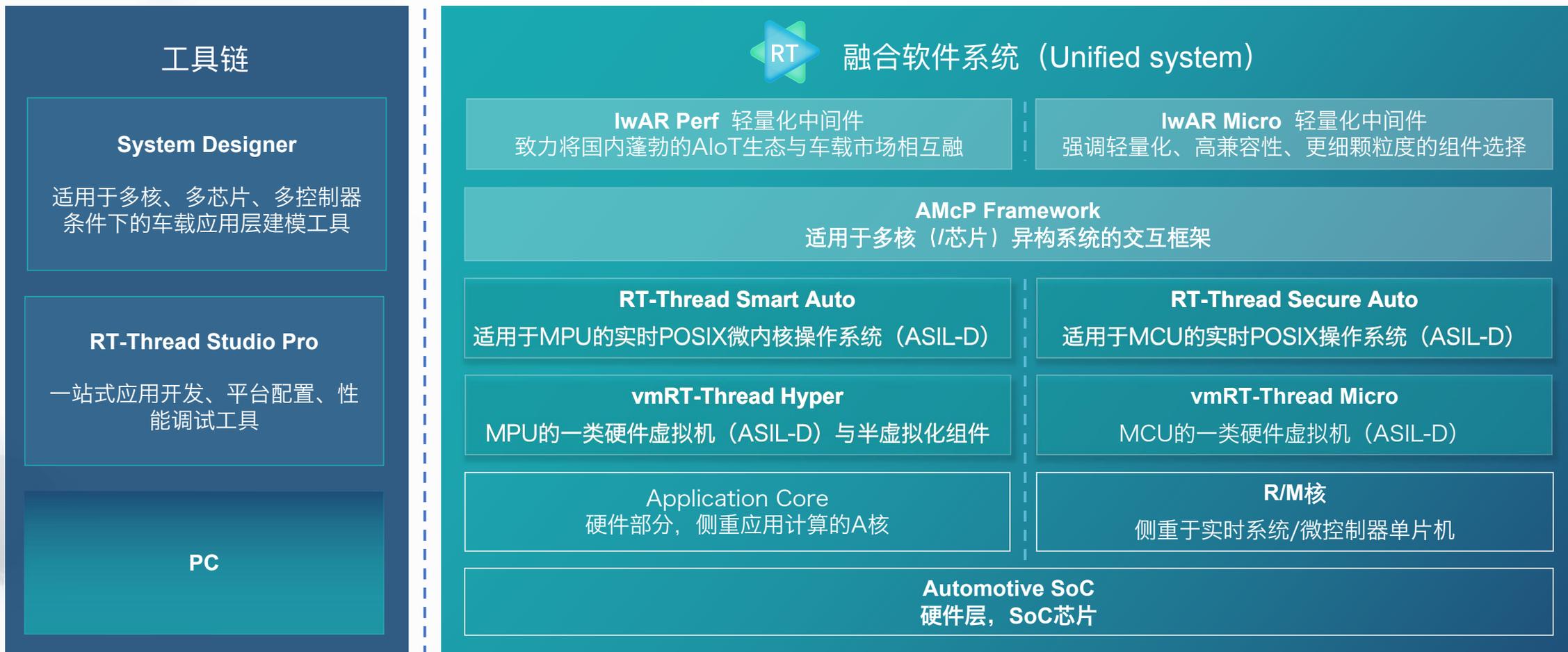
OPEN SOURCE, INTO THE FUTURE

「OpenSDV软件定义汽车」专场

RT-Thread Smart 操作系统的实践

熊谱翔 2023年05月28日

全栈式车规级OS产品组合，满足在各系统层面的需求

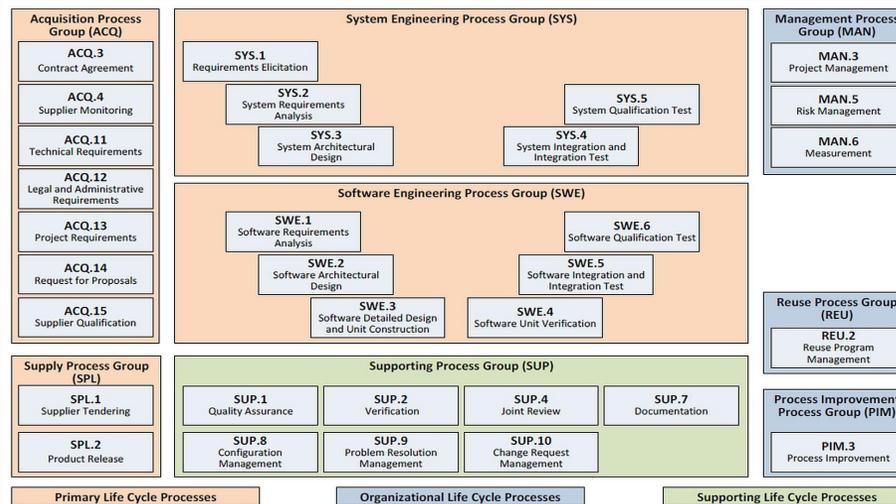


产品安全

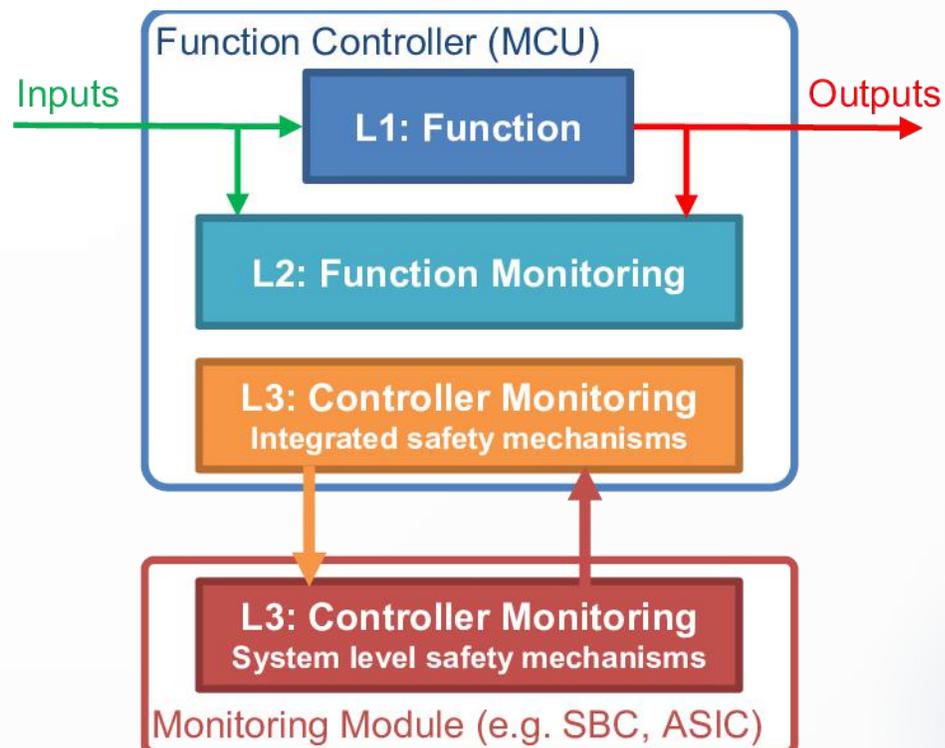
- 安全分析, 安全机制
- 软件代码模型检查
- 测试覆盖度

流程安全

- 遵循一定的流程开发机制
- 开发过程管理
- 变更管理

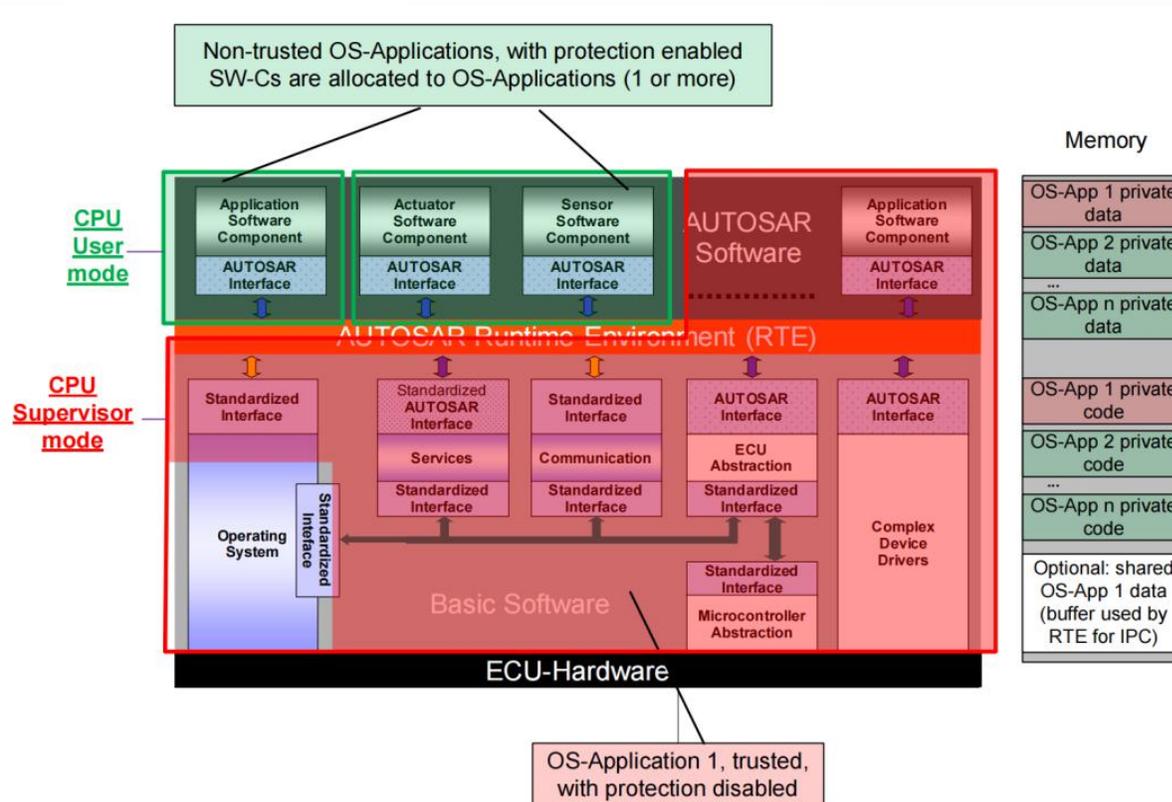


功能安全, 不仅仅是产品的功能安全, 也包括功能安全的流程。



四种功能安全机制

- 内存分区 (Memory Partitioning)
- 时间监控 (Timing Monitoring)
- 逻辑监督 (Logical Supervision)
- 端到端保护 (End-2-End Protection)





上层应用

- 传统的各类应用程序

中间件

- 图形栈，多媒体栈
- 网络应用协议等

系统服务

- 把经典的一些操作系统设施放到用户态
- 文件系统，网络协议栈，外设驱动等

微内核

- 一般包括基础调度，IPC等；
- 代码比较少，资源占用也相对少；

内核单独运行在内核态，特权模式下

系统服务、应用都运行在用户态，受限模式下

应用与系统服务之间，通过消息通信方式交互。

丰富的应用环境

- 承载自己的生态

各种应用：图形栈，多媒体栈，AI

运行环境

libc/libstdc++

POSIX

设备驱动

文件系统

网络协议栈

微内核

虚拟机方式

- 其他的操作系统、应用

虚拟机管理器



设备驱动

客户机

微内核

源于经典RT-Thread系统

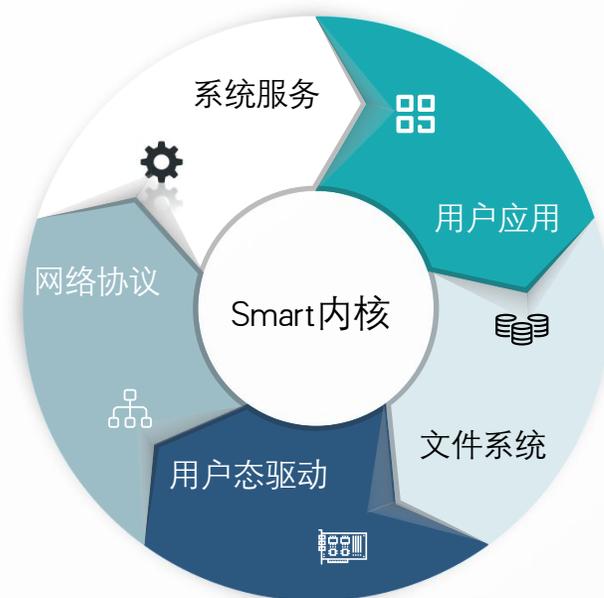
- 基于15+年的RT-Thread实时操作系统;
- 可抢占式硬实时内核, 延续**高实时特性**;
- 实时系统经典IPC设施, 信号量, 互斥锁, 邮箱, 消息队列;
- 支持32位/64位, 单核/多核处理器;

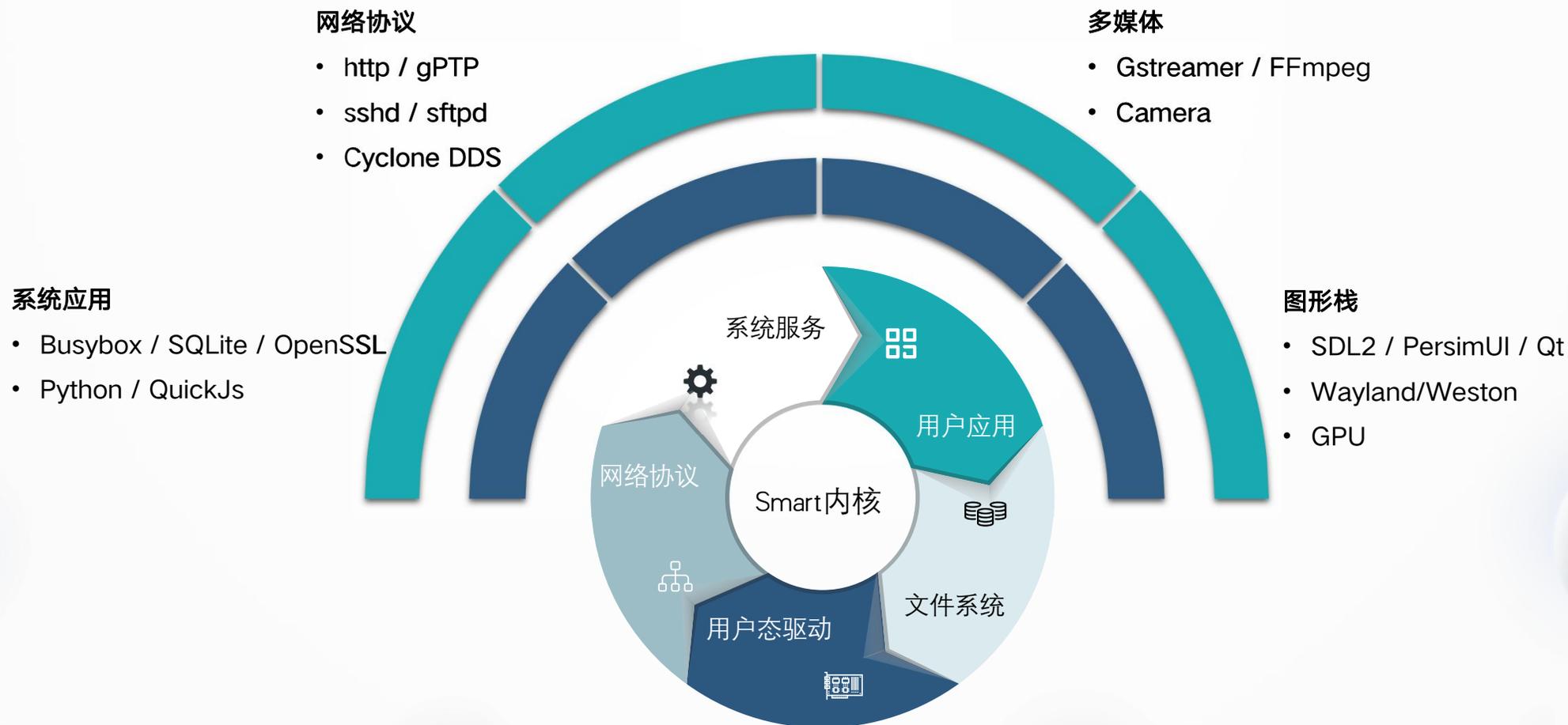
采用混合内核架构

- 按需加载功能组件, 极速启动;
- 内核态与用户态分离, 安全的内存隔离环境;

POSIX兼容用户态环境

- POSIX用户态运行环境, 兼容Linux/Unix
开源生态;





RT-Thread Smart的特点：实时、安全、POSIX生态



在用户态完整兼容 PSE53 API标准：

- 支持多进程，包括经典的fork, vfork, pipe系统调用；
- 完整的网络支持，具备标准的Socket Abstraction Layer (SAL)，并支持UNIX套接字，netlink等；
- 支持POSIX signal, threads等；



01. 完整支持stdc++库

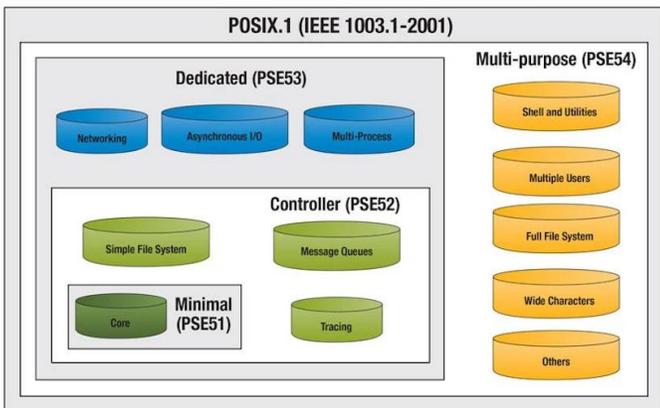
- C++11/14中的标准库，算法等
- std::vector, std::map, 容器等

02. 并发编程及智能指针

- std::thead、std::future
- std::mutex、std::condition_variable、std::atomic等
- shared_ptr、unique_ptr等

03. 系列语法糖

- auto自动类型；
- lambda表达式及函数等



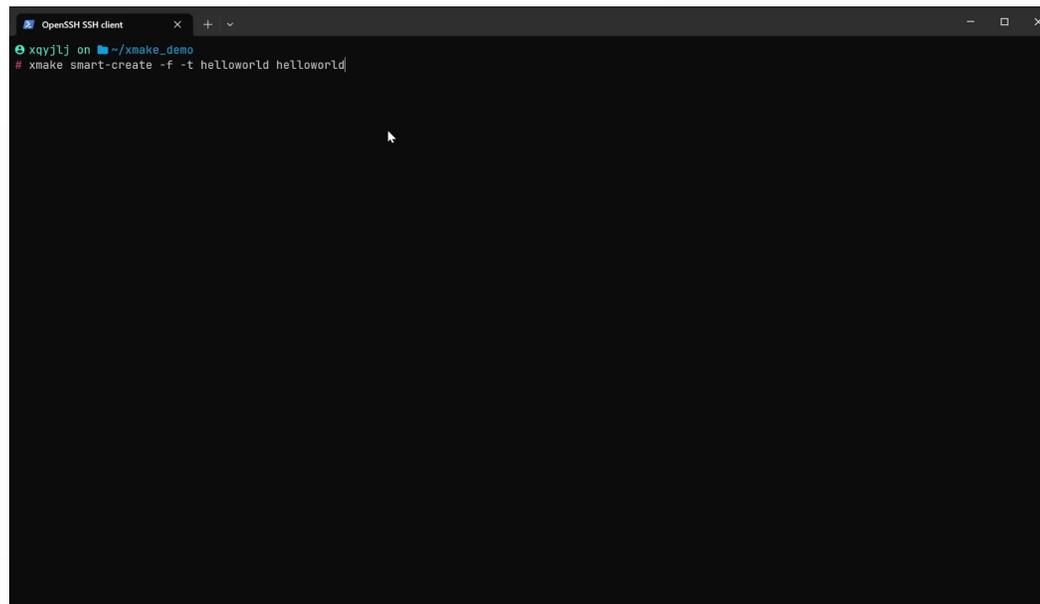


xmake

- A cross-platform build utility based on Lua
- Modern C/C++ build tool: Simple, Fast, Powerful dependency package integration

基于xmake的构建系统

- 国产的现代化构建系统
- 快速构建、分布式构建
- 自带包管理器 xrepo
- 重用部分包
- 可以非常好的对接到Makefile, configure & make, CMake

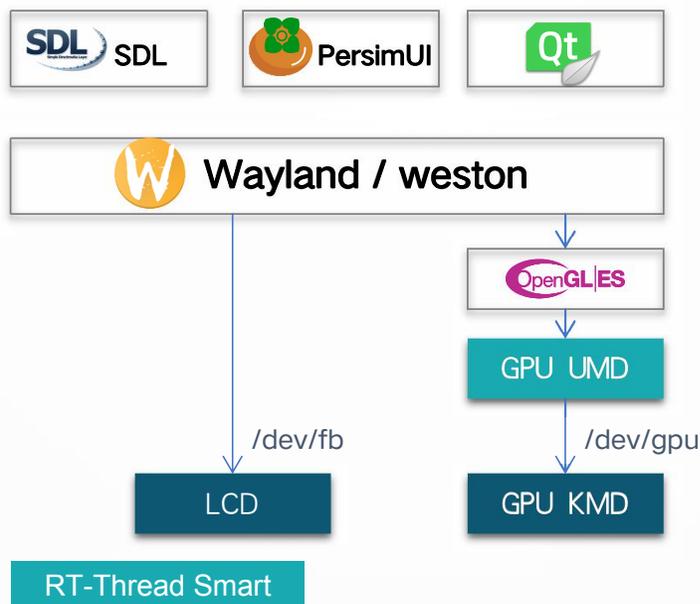


```
OpenSSH SSH client
xqylj on ~/xmake_demo
# xmake smart-create -f -t helloworld helloworld
```



PersimUI Kit

- 前端化的现代UI系统，支持JavaScript小程序开发
- 支持多种常用控件、动画，并可进行自定义，实现各类炫酷效果；支持半透明、Alpha混合、抗锯齿等效果；



RT-Thread Smart

UI Toolkit

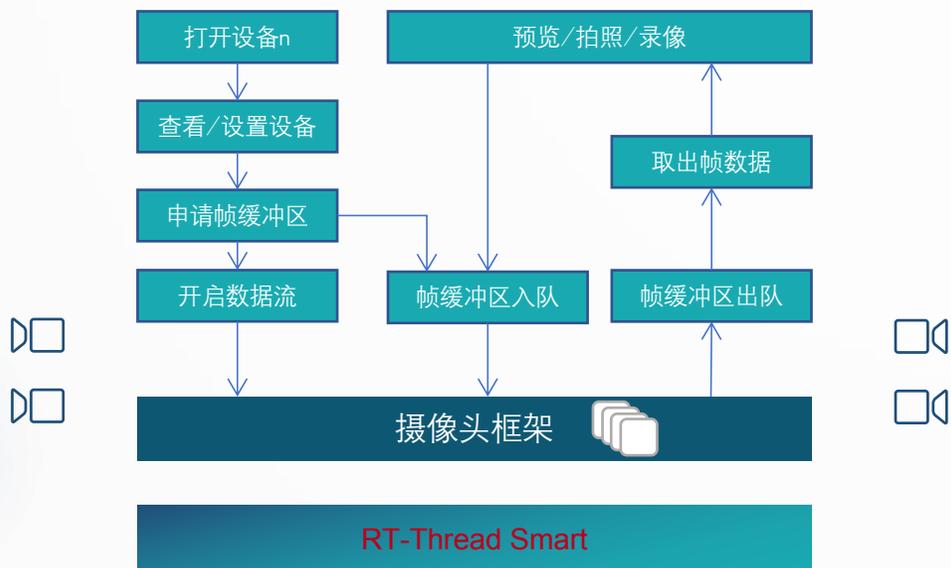
- SDL2 轻量级高效图形渲染
- 柿饼UI (PersimUI Kit)
- Qt [soon]

窗口合成器

- 支持Wayland/Weston
- 自有C++实现的窗口合成器 [soon]

GPU支持

- 底层提供对3D加速支持，支持OpenGL ES、Vulkan等
- ARM Mali GPU驱动
- Imagination GPU驱动 [soon]



GNU的多媒体应用

- FFmpeg 能够编解码、转码、复用/解复用、过滤、播放等的音视频多媒体框架
- Gstreamer 插件式的多媒体框架

摄像头框架

- 类似V4L2的摄像头框架
- 可同时支持多个摄像头

时间同步

- 支持应用层通过socket获取时间戳
- 网络高精度时间同步 (PTP)

DMA buffer

- 用于各外设之间的数据buffer分享
- 跨进程分享fd

vscode + smart扩展方式开发:

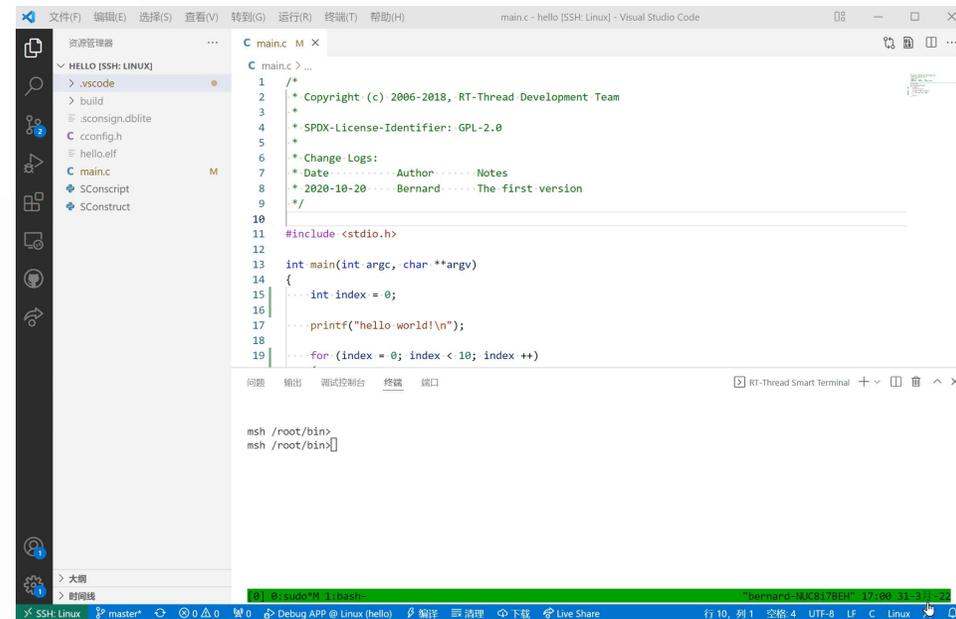
创建应用程序;

连接到远程ssh server进行开发;

支持直接gdb调试;

支持Windows & Linux平台

Linux平台: 方便进行移植、调试GNU Apps;



Networking/USB



内存的问题

- 内存越界，野指针
- 内存泄露



KASAN (Kernel Address Sanitizer)

- 动态监控代码运行情况
- 速度会降低

ASAN (Address Sanitizer)

- 用户态应用程序的动态内存错误检测器
- 野指针，堆溢出，栈溢出
- 返回后继续使用
- 超出范围后继续使用
- 初始化序列问题



动态追踪框架

- 动态追踪框架是一个事件采集框架，具有可配置化、实时性、可拓展等优势与特性。
- 它通过采集多种类型的软件和硬件事件满足不同的调试需求

```
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058578, op=0x200058579, arg=0x20005857a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058578, op=0x200058579, arg=0x20005857a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058898, op=0x200058899, arg=0x20005889a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058898, op=0x200058899, arg=0x20005889a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] ioctl(fd=0x0, cmd=0x1, data=0x2)
[I/tracing.ftrace] [python:802e2d78] ioctl() => 0x0
[I/tracing.ftrace] [python:802e2d78] write(fd=0x1, buf=0x2, nbytes=0x3)
MicroPython v1.19-22-gf236ab234-dirty on 2023-05-15[I/tracing.ftrace] [python:802e2d78] write() => 0x33
[I/tracing.ftrace] [python:802e2d78] write(fd=0x1, buf=0x2, nbytes=0x3)
; rt-smart [GCC 10.1.0] version[I/tracing.ftrace] [python:802e2d78] write() => 0x1f
[I/tracing.ftrace] [python:802e2d78] write(fd=0x1, buf=0x2, nbytes=0x3)

Use Ctrl-D to exit, Ctrl-E for paste mode
[I/tracing.ftrace] [python:802e2d78] write() => 0x2b
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058898, op=0x200058899, arg=0x20005889a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] pmutex(umutex=0x200058898, op=0x200058899, arg=0x20005889a)
[I/tracing.ftrace] [python:802e2d78] pmutex() => 0x0
[I/tracing.ftrace] [python:802e2d78] ioctl(fd=0x0, cmd=0x1, data=0x2)
[I/tracing.ftrace] [python:802e2d78] ioctl() => 0x0
[I/tracing.ftrace] [python:802e2d78] ioctl(fd=0x0, cmd=0x1, data=0x2)
[I/tracing.ftrace] [python:802e2d78] ioctl() => 0x0
[I/tracing.ftrace] [python:802e2d78] write(fd=0x1, buf=0x2, nbytes=0x3)
>>> [I/tracing.ftrace] [python:802e2d78] write() => 0x4
[I/tracing.ftrace] [python:802e2d78] read(fd=0x0, buf=0x1, nbytes=0x2)
```

RT-Thread Smart 中使用动态插桩技术采集系统调用事件的例子



RT-Thread Smart for Safety v1.0
是一份获得功能安全产品认证的版本：

- IEC61508 SIL3 标准认证
- EN50128 标准认证
- ISO 26262 ASIL-D 标准认证

硬件芯片支持情况

ARM®

ARMv8-A



A1000



征程5



RK3568/3588



S32G



Zynq MPSoC UltraScale

ARMv7-A



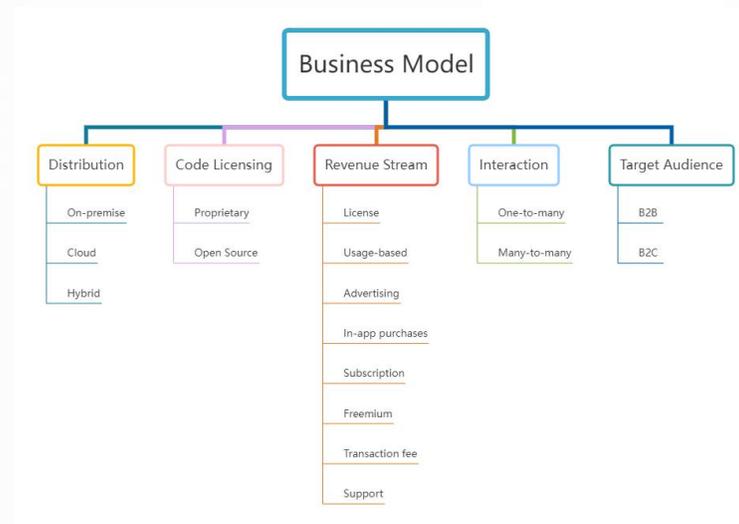
Zynq7000



STM32MP1



i.MX6ULL



持续开源

- 源于RT-Thread，开源于RT-Thread社区
- 内核开源，系列的 GNU Apps 开源

商业结合

- 服务
- 安全保证
- 工具等……



谢谢

关注RTThread公众号获得更多资讯