

Jemalloc 记录文档

编辑历史

| 更新日期 | 作者 | 更新内容 |
|------------|-----|----------------|
| 2024.02.04 | 胡益华 | Initialization |
| | | |
| | | |

目录

| | |
|----------------------------|----|
| <i>Jemalloc</i> 记录文档 | I |
| 编辑历史 | I |
| 目录 | II |
| 1. 引言 | 1 |
| 1.1 概述 | 1 |
| 1.2 jemalloc | 1 |
| 2. 安装 | 2 |

1. 引言

1.1 概述

众所周知，C/C++编写的程序拥有极致的运行速度，并且编译后的程序所占的存储空间非常小。因此，C/C++在嵌入式中被广泛应用。

不过其安全问题也是一直被诟病。一般来说，安全问题主要出在内存上。了解 C 语言的工程师对内存泄露、内存碎片等一定不会陌生。尤其是某些嵌入式设备，内存就这么点大，这要还泄露了，那还得了。

如果一块 2KB 的堆区内存只释放了 1.5KB，那么没有被释放 0.5KB 就是内存碎片。这一块内存碎片的影响不仅体现在进程运行时，占用的内存变大了一些；甚至可能因此受到内存溢出攻击。例如有一名黑客将未被释放的内存块中的数据或代码覆盖甚至做替换，后果是不堪设想的。

虽然业内都在调侃：没有不安全的语言，只有不安全的工程师。但是再好的工程师也难免出错，为了对抗 C 的内存问题，工程师们也是绞尽脑汁。开发者们耳熟能详的 `asan`、`tcmalloc`、`jemalloc` 等本质上都是为了解决 C 语言内存不安全的问题。这些都是我使用过，体验很好的工具，本文档主要对 `jemalloc` 的使用做一些记录。

1.2 jemalloc

`jemalloc` 是一款开源的软件。gitbub 地址如下：

<https://github.com/jemalloc/jemalloc/>

官方 wiki 地址如下：

<https://github.com/jemalloc/jemalloc/wiki>

官方环境变量说明地址如下：

<http://jemalloc.net/jemalloc.3.html#opt>

就像 `malloc` 一样，它也是一款内存分配器，可以高效地分配内存、管理内存。在多线程的情况下，也可以一定程度上减少内存碎片。当然因为其开源的特性，许多公司的项目中就不可以使用它来管理内存。不过 `jemalloc` 同时也提供了记录内存开销，定位内存问题的功能，所以用它来查找内存上的错误也是很好的选择。

2. 安装

以 Linux, centos7 环境为例, 下载源码后解压。

(1) `./autogen.sh`

运行 `autogen.sh` 文件, 用于生成 `configure`。

(2) `./configure --prefix=/home/$USER/tool/jemalloc-dev/artifacts/hi3519dv500 \`
`--enable-prof \`
`--build=x86_64-unknown-linux-gnu \`
`--host=aarch64-linux-gnu \`
`CC=aarch64-linux-gnu-hi3519dv500-v2-gcc \`
`CXX=aarch64-linux-gnu-hi3519dv500-v2-g++`

运行 `configure` 文件, 用于生成 `Makefile`。假设需要在 hi3519dv500 芯片上使用 `jemalloc` 检查内存问题。

1) `--enable-prof` 表示启用堆分析和泄漏检测功能。

2) `--build` 指定当前系统环境, 不指定也可以。

3) `--host` 表示交叉编译的工具链, 不使用 `--host` 只指定编译工具链, 会提示:
If you meant to cross compile, use '--host'.

所以 x86_64 平台下不需要指定 `--host` 参数。

(3) `make -j && make install`

(4) 指定可执行文件环境变量等。