

RISC-V 指令模拟器中增加对系统调用的支持

(Version 160321)

易江芳

概述

本实验要求学生在实验 2.1 的基础上，增加对标准库函数 glibc 和系统调用的支持。最终，在指令模拟器上：

- (1) 实现 printf 的标准输出
- (2) 参照 printf 的实现原理，实现其他系统调用，正确运行测试基准程序 Dhrystone

提交

最终提交内容：

- 实验报告
- 指令级模拟器代码

关于库函数和系统调用

应用程序中使用到的 printf 函数是 glibc 标准库函数中的一员。这些库函数是 GNU 发布的 libc 库，即 c 运行库。glibc 是 linux 系统与应用程序交互的最基本 api。在 glibc 中，包含了字符串处理、信号处理、共享库加载、socket 接口、动态内存分配、线程管理及标准库函数等众多必不可少的内容，是支撑整个操作系统的运行框架。

以 printf 为例，对 glibc 库函数和系统调用进行简介。

应用程序使用 printf 进行标准输出，在 glibc 中有函数 printf 与之对应，libc 里的 printf 函数会使用系统调用 write(printf() 会调用 fputc();而 fputc()会调用 _sys_write())。

对于系统调用 _sys_write(), 不同体系结构的实现不同。如在 X86 结构中，会被转换成 4 号系统调用，即设置 eax=4。同时，需显示的内容会放在之后的寄存器中，然后再通过 0x80h 进入软中断处理。在中断处理中，会根据 stdout 的指向，将需要显示的数据送到串口(uart)或者 console，并调用串口(uart)或者 console 相关驱动的 write 函数进行打印。

因此，如果需要处理系统调用，最简单的做法就是接管所有系统调用的处理。而系统调用的处理通常在系统的 ABI (Application Binary Interface) 中定义，包括系统调用的类型号以及系统调用的参数传递规则。

关于测试基准程序

Dhrystone 是一个计算机性能评测基准程序 (benchmark)，于 1984 年由 Reinhold P.Weicker 开发，用来测试通用 CPU 的定点性能，浮点性能由另一个测试程序 whetstone 来评测。

该测试程序收集了很多软件程序如 Fortran、PL/1、SAL、ALGOL68 和 pascal，包括了很多类型的结构如程序调用、指针、赋值等。最早的版本用 Ada 语言编写，后续有了 C 版本。

Dhrystone 的计量单位为每秒计算多少次 Dhrystone，后来把在 VAX-11/780 机器上的测试结果 1757 Dhrystones/s 定义为 1 Dhrystone MIPS(缩写为 DMIPS)。因此，表征结果的方式是每秒运行的 Dhrystone 次数/1757 (DMIPS)