UnixBench测试集

UnixBench是一个类Unix系统(Unix，BSD，Linux)下的开源性能测试工具，被广泛用与测试Linux系统主机的性能。UnixBench的主要测试项目有：系统调用、读写、进程、图形化测试、2D、3D、管道、运算、C库等系统基准性能提供测试数据。

UnixBench v5.1.3下载地址：<https://github.com/kdlucas/byte-unixbench>

测试项

Dhrystone 2 using register variables

此项用于测试 string handling，因为没有浮点操作，所以深受软件和硬件设计（hardware and software design）、编译和链接(compiler and linker options)、代码优化（code optimazaton）、对内存的cache（cache memory）、等待状态（wait states）、整数数据类型（integer data types）的影响。

Double-Precision Whetstone

这一项测试浮点数操作的速度和效率。这一测试包括几个模块，每个模块都包括一组用于科学计算的操作。覆盖面很广的一系列 c 函数：sin,cos,sqrt,exp,log 被用于整数和浮点数的数学运算、数组访问、条件分支（conditional branch）和程序调用。此测试同时测试了整数和浮点数算术运算。

Execl Throughput

此测试考察每秒钟可以执行的execl系统调用的次数。execl 系统调用是 exec 函数族的一员。它和其他一些与之相似的命令一样是 execve()函数的前端。

File copy

测试从一个文件向另外一个文件传输数据的速率。每次测试使用不同大小的缓冲区。这一针对文件 read、write、copy 操作的测试统计规定时间（默认是10s）内的文件 read、write、copy 操作次数。

Pipe Throughput

管道（pipe）是进程间交流的最简单方式，这里的 Pipe throughtput 指的是一秒钟内一个进程可以向一个管道写512字节数据然后再读回的次数。需要注意的是，pipe throughtput 在实际编程中没有对应的真实存在。

Pipe-based Context Switching

这个测试两个进程（每秒钟）通过一个管道交换一个不断增长的整数的次数。这一点很向现实编程中的一些应用，这个测试程序首先创建一个子进程，再和这个子进程进行双向的管道传输。

Process Creation

测试每秒钟一个进程可以创建子进程然后收回子进程的次数（子进程一定立即退出）。process creation 的关注点是新进程进程控制块（process control block）的创建和内存分配，即一针见血地关注内存带宽。一般说来，这个测试被用于对操作系统进程创建这一系统调用的不同实现的比较。

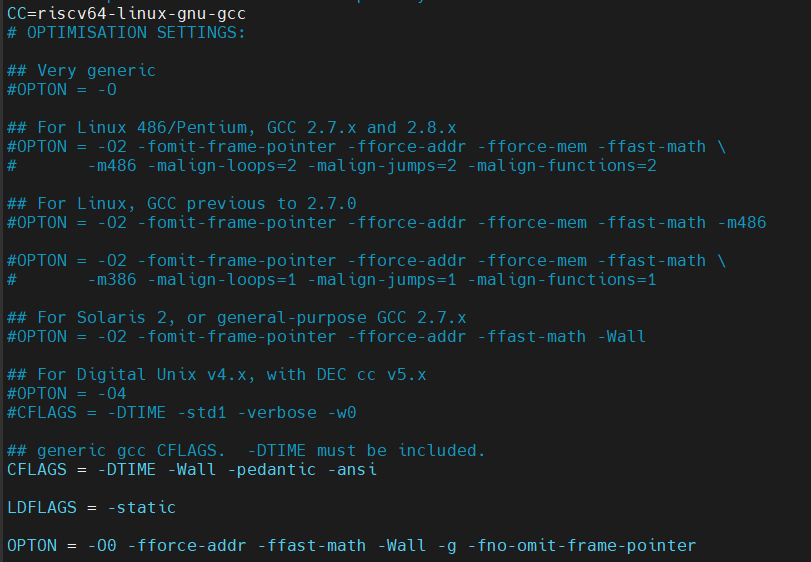
System Call Overhead

测试进入和离开操作系统内核的代价，即一次系统调用的代价。它利用一个反复地调用 getpid 函数的小程序达到此目的。

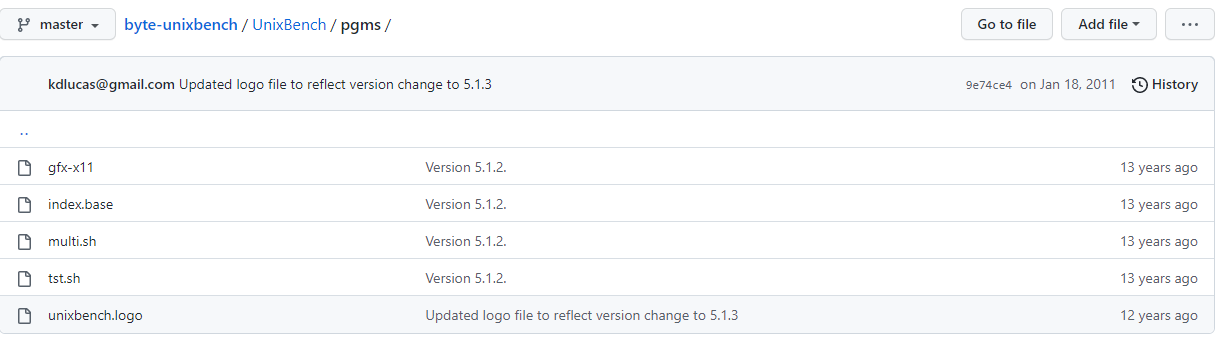
Shell Scripts

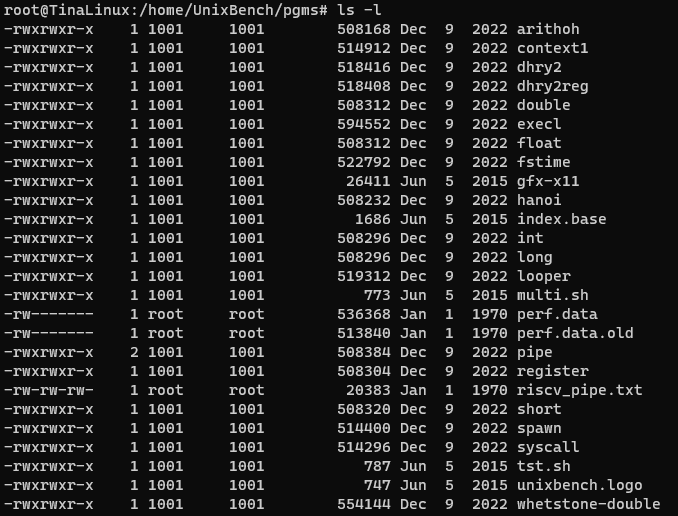
测试一秒钟内一个进程可以并发地开始一个 shell 脚本的 n 个拷贝的次数，n 一般取值1,2,4,8。（我在测试时取 1,8）。这个脚本对一个数据文件进行一系列的变形操作（transformation）。

修改Makefile(UnixBench目录下的Makefile)



交叉编译生成可执行文件



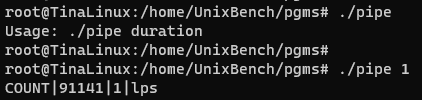


测试脚本UnixBench\_process.sh使用说明

sh UnixBench\_process.sh 待测试程序的目录 生成结果的指定目录 CPU架构类型 执行perf record的次数

示例: sh UnixBench\_process.sh ./pgms riscv\_perf riscv 5

测试



perf record -e cpu-cycles -g ./pipe 1

perf report --no-children -G > riscv\_pipe.txt

adb pull /home/UnixBench/pgms/riscv\_pipe.txt D:\TinaDocument

perf stat -e cache-misses -e cache-references -e branches -e branch-misses ./pipe 1

