Hyperkernel: Push-Button Verification of an OS Kernel

`OS` `Kernel` `Verification`

Nelson L, Sigurbjarnarson H, Zhang K, et al. Hyperkernel: Push-Button Verification of an OS Kernel[C]//Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017: 252-269.

背景

设计证明一个系统的正确性一直是一个很重要的问题，此前的工作的工作通过正确性证明来消除一类系统bug，但是，有很大的实现上开销，对于写出证明的人也有很高的要求。本文希望提供一个方法，借助Z3（一种由微软开发的SMT求解器），来构造一个系统同时通过很低的成本（Push-button）完成正确性证明。
操作系统有着复杂的函数、数据结构和层次关系。路径数目更大，不便于Z3求解。论文中提出的方法，一方面，把证明的内容从C代码变为llvm中间代码，使得证明的语义更加规范。另一方面，限制了系统的特性，进行证明的是一个单处理器关中断且简化虚拟内存实现的操作系统。最后，还修改了一般接口实现，要求接口实现添加约束，事实上使得证明的代码实现变得简单。

简单理解

> 论文提供了：一个低开销的方法来构造一个验证过的操作系统内核;一种便于SMT求解的接口设计方法;一个有合适的性能的操作系统（Hyperkernel）的实现

首先，系统证明同样需要一个specification，同时具有完备的数据结构来描述系统本身，可以依靠它来实现一个实际的系统，又不能太过复杂使得证明路径爆炸。限制内核接口，避免无限的循环和递归。

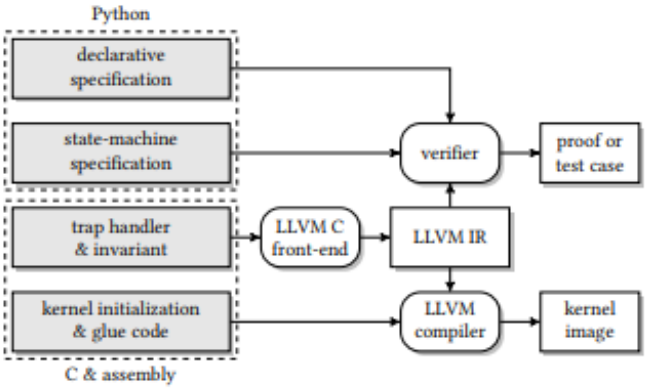

Figure 3: The Hyperkernel development flow. Rectangular boxes denote source, intermediate, and output files; rounded boxes denote compilers and verifiers. Shaded boxes denote files written by programmers.

其次，系统是通过c语言实现的。然而c语言由于语义问题和底层操作（指针运算和访存），以及c标准的不清晰，不便于形式化表述。论文证明使用llvm中间表示（IR）来进行证明。IR具有更加简单而且清晰的语义的同时，保留了高级的信息。

最后，Hyperkernel将内核与用户地址空间分开，简化了虚拟内存的实现。

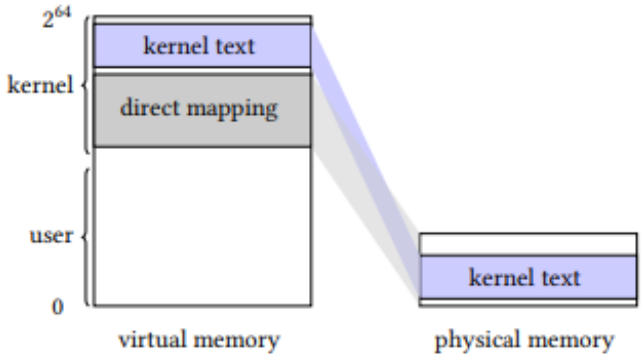

Figure 1: A simplified memory layout of Linux on x86-64: the kernel and user space are mapped to the upper half and lower half of the virtual address space, respectively. The ABI recommends the kernel text to be mapped to the top 2 GiB, as required by the kernel code model [52]; the kernel also has a direct mapping of all physical memory.

State-machine Style Specification：提出了一个内核调用和中断处理函数的状态机，以及这个specification的“正确的”C实现。

Declarative Style Specification: 用具体的语言描述specification，提炼这两种specification。

结论

本文实现了一个基于xv6的单CPU系统内核Hyperkernel，通过Z3求解器检查了45个系统调用和中断处理程序。实验证明Hyperkernel可以避免xv6中发现的相似的bug，同时可以以较小的代价得到内核正确性的证明，也就是所谓的push-button。证明部分由python实现，兼顾了实现的简易性和可以简便地调用Z3接口。