**高级操作系统实验报告**

**姓名：施宇 学号：2017211384**

# Hyperkernel

阅读了Hyperkernel的state-machine specification以及system call的实现代码，确定了为hyperkernel增加新系统调用的流程。成功添加并验证了一个简单的系统调用。通过修改hyperkernel system call实现的代码，使其与state-machine specification产生不一致，并输出反例。

## 添加并验证新的系统调用

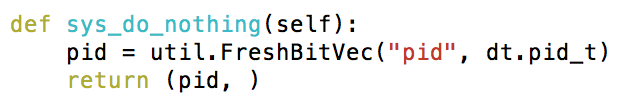
添加系统调用的步骤如下：

我们以添加一个什么都不做的系统调用int sys\_do\_nothing(pid\_t pid)为例子。

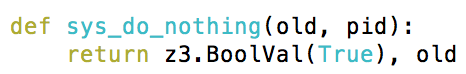
1. 在hv6/spec/kernel/main.py里的\_syscalls列表里加入系统调用的名字，以sys为开头，例如sys\_do\_nothing。



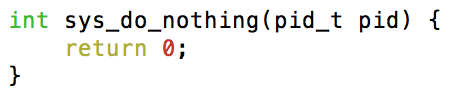
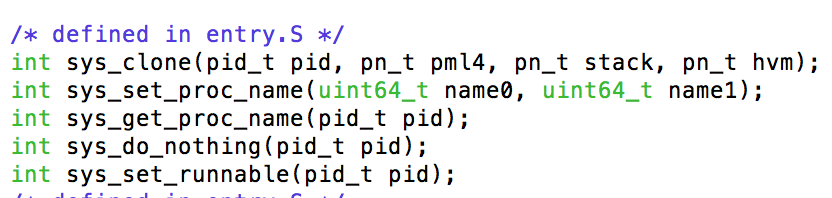
1. 在hv6/spec/kernel/syscall\_spec.py里定义一个同名的python函数。该函数用来指定系统调用的参数类型。例如，我们的sys\_do\_nothing只有一个pid\_t类型的参数pid，那么我们需要创建一个z3的FreshBitVec变量，名字是“pid”，类型是dt.pid\_t。并返回该变量。



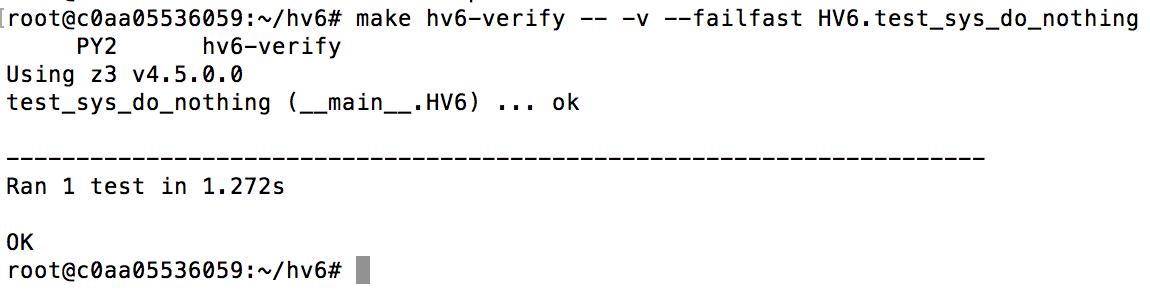
1. 在hv6/spec/kernel/spec/specs.p中写入state-machine specification。注意statemachine-specification应有两个返回值，第一表示转移到新的内核状态的条件，第二个表示新的内核状态。由于我们什么都没有改变，所以直接返回true以及旧的内核状态



1. 在内核的c代码中实现系统调用。我们把sys\_do\_nothing的声明和定义分别写在hv6/proc.h以及hv6/proc.c当中。

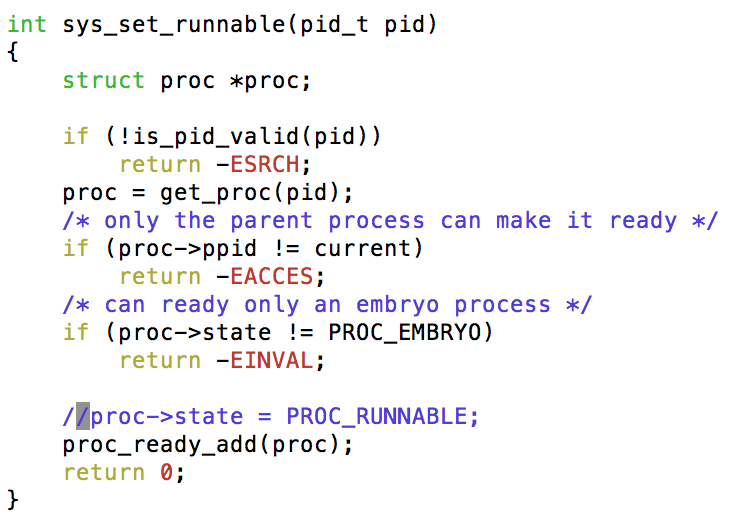


1. 验证系统调用

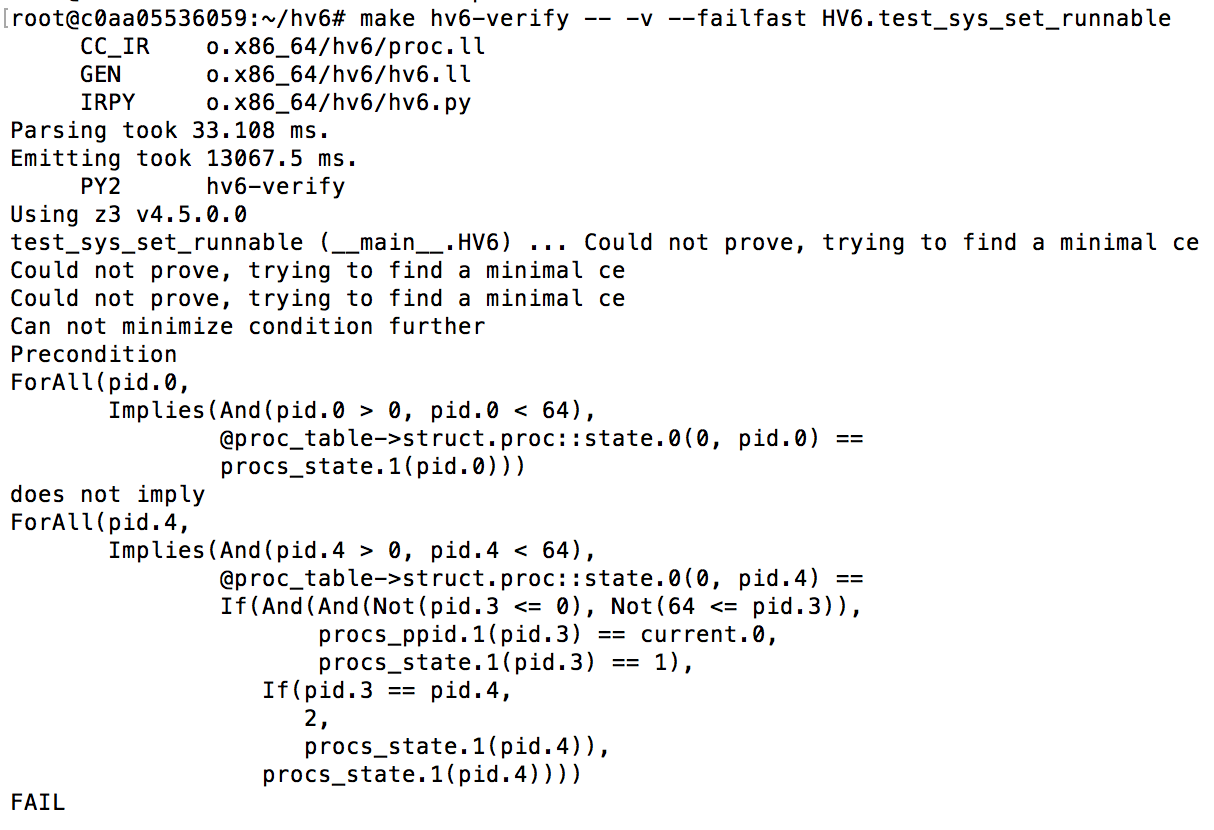


## 修改系统调用实现

1. 注释掉sys\_set\_runnable实现中的把进程状态设置成PROC\_RUNNABLE的语句



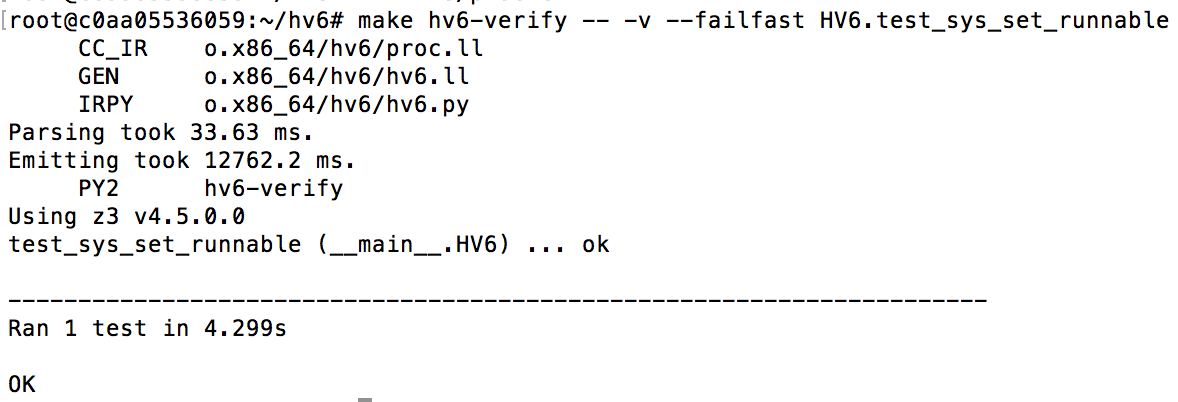
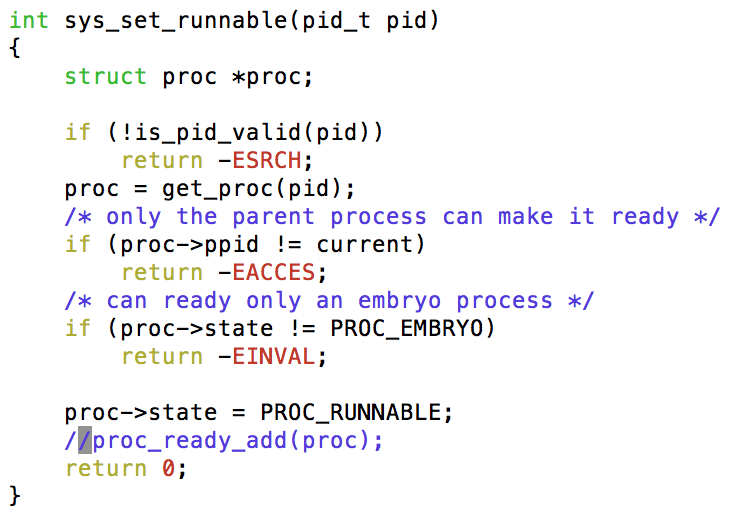
1. 重新验证sys\_set\_runnable



Z3检测到与state-machine specification不一致，并给出反例。

## 没有被state-machine specification跟踪的链表结构

在Hyperkernel中，为了实现的方便，还定义了两个链表结构，其中一个是用来存放准备运行的进程的ready queue。sys\_set\_runnable最后一行的proc\_ready\_add(proc)就是把proc加到ready queue里面去。由于链表的状态维护并不是一个finte的操作，因此要用Z3验证代价是很高的。Hyperkernel仅仅只是为了实现其他函数的方便而定义了这个链表。把proc\_ready\_add(proc)注释掉仍然可以通过验证



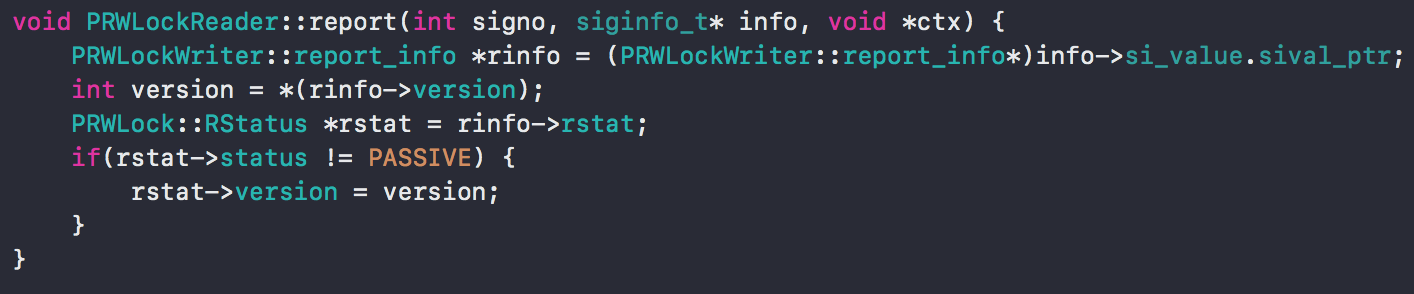
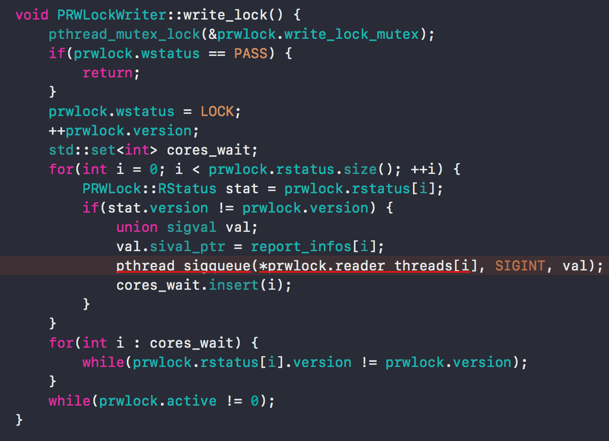
# Prwlock算法的实现与模拟

每个写者和读者都是一个线程，每个读者都有一个私有的状态，存储在一个全局共享的列表中。

使用pthread库实现了prwlock算法。其中prwlock类如下：

每个prwlock有一个用来写者锁，write\_lock\_mutex。Prwlock中还维护了一个rstatus列表，用来存储每个读者的version信息和状态信息。

在write\_lock的时候，当写者发现rstatus中有读者的version信息落后时，会使用pthread\_sigqueue告知相应的读者线程，然后读者线程会调用信号处理函数report，更新版本信息。



增加读者、写者的个数，测试平均每秒的读次数。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 读写者个数 | 100 readers  200 writers | 500 readers  500 writers | 1000 readers  1000 writers | 1000 readers  2000 writers |
| Read Throughput | 226 | 197.5 | 156 | 142.85 |
| Write Throughput | 5 | 3.3 | 2.6 | 2.4 |

当写者数目增多的时候prwlock的throughput还是会受影响。