CMPE283 : Virtualization
Assignment 3: Instrumenting KVM

Due: Nov 27, 2017 before midnight

In this assignment you will learn how to place instrumentation in the KVM hypervisor. This will be used to determine the frequency of each type of exit, and how long is spent processing the exits. This lab assignment is worth up to 20 points and may be done in groups of up to **two people max.** Each team member can receive up to 20 points. It is expected that groups of more than one student will find an equitable way to distribute the work outlined in this assignment. You can form groups from both Monday and Tuesday sections, as both sections will receive the same assignment.

Prerequisites

- You will ideally implement this assignment using the same setup you used for assignment 2.

The Assignment

Your assignment is to add counters into the KVM that track the following information:

- Total number of exits (for each type of exit KVM enables)
- Max/Min/Average number of CPU cycles for each exit type
- Total amount of cycles spent processing all exits

An ideal place to implement this functionality is inside the top-level exit handler function in KVM. We outlined where this could be found in class a few weeks ago.

You do not need to use atomic variables or locking for this assignment (eg, the values outlined above may be slightly inaccurate based on SMP effects). The values outlined above are to be calculated hypervisor-wide, meaning they are to be reported for the entire hypervisor, across all PCPUs/VCPUs/VMs.

Periodically, output the statistics above to the system dmesg log (via pr_info/printk/etc – look to assignment 1 and use the same output routines if unsure). At a minimum, output the statistics information every 500 total exits (if this generates too much output, you may opt to output the statistics information every 1000 or 2000 exits, but please mention this in your answers to the questions supplied below). The format of the output is left to you but please ensure it is easily understood (specifically, indicate on each line of output what is being reported).

This assignment, like assignment 2 before it, is a standalone assignment, meaning you should start with a fresh (or reverted) git repository before starting. Do not build this assignment on top of your existing assignment 1 or 2 commits.

On or before the due date, turn a code listing and answers to the questions below via E-Mail. **Make sure to follow the instructions on diff/code format below.**

Grading

This assignment will be graded and points awarded based on the following:

- 15 points for the implementation and code producing the output above
- 5 points for the answers to the questions below

Submissions shall be made via email to the email addresses listed in the class syllabus/green sheet. DO NOT WAIT UNTIL LATE ON THE DUE DATE, as email server lags or delays may result in a late

submission. Since you have three weeks to complete this assignment, I will not accept "email server outage or delay" as an excuse for late submissions. If you are concerned about this, submit your assignment early. This is one area that I am extremely picky with – even 1 second late will result in a zero score.

I will be comparing all submissions to ensure no collaboration has taken place. Make sure you do not copy another group's work. If you copy another group's work, members of both groups will receive an F in the class and be reported to the department chair for disciplinary action. If you are working in a group, make sure your partners do not copy another group's work without your knowledge, as all group members will be penalized if cheating is found.

Special Notes

I am implementing an automated framework to test these submissions. Therefore, you **must** follow the subsequent submission rules precisely. I will be using a script that will automatically process my mailspool to extract your submissions, and the script will expect the submission email to formatted a specific way, as described below:

- Use a kernel built from the master Linux git repository
  - https://github.com/torvalds/linux.git
  - Record the head commit ID of your tree:
    - For example, if the output of "git log" shows the following:
      commit 89970a04d70c6c9e5e4492fd4096c0b5630a478c
      Merge: 806276b7f07a 3ea3217cf918
      Author: Linus Torvalds <torvalds@linux-foundation.org>
      Date: Wed Mar 29 19:59:49 2017 -0700
    - .. you would record "commit 89970a04d70c6c9e5e4492fd4096c0b5630a478c"
- Use 'git add' and 'git commit' to add your tree changes. **Include all changes required to build your assignment.**
- Submit a plain text file containing a unified diff ("git diff" format) diff file containing your entire submission. Name the diff file "cmpe283-3.diff" in your submission when attaching to the email.
- When submitting, submit **only**:
  - A plain text email (no HTML)
  - A single PDF file attachment containing the answers to the questions (the PDF can have whatever name you want)
  - The diff file in plain text format as described above.
  - Your commit ID as calculated above, by including a line starting with "commit" followed by a space, followed by the SHA value of the commit ID, with no other text on that line.
  - A list of student IDs and names for members of the group, in the following format, one per line:
    - ID <id> <name>
      - (example) ID 0123456789 Larkin, Michael
- The subject line of the email must be "CMPE283 Assignment 3 Submission" (no quotes)
- Do not submit .zip, .tar, .rar, etc. Send me an email with two attachments, a PDF and a .diff file, as described above.
- Do not mangle your diff
- Do not give me commit IDs of your local commits
- Do test your diff before submitting
- Make sure to follow all other instructions in this assignment precisely.
- **Failure to follow the instructions above may result in a zero score for the assignment.**

Questions

1. For each member in your team, provide 1 paragraph detailing what parts of the lab that member implemented / researched. (You may skip this question if you are doing the lab by yourself).
2. Describe in detail the steps you used to complete the assignment. Consider your reader to be someone skilled in software development but otherwise unfamiliar with the assignment. Good answers to this question will be recipes that someone can follow to reproduce your development steps.

Note: I may decide to follow these instructions for random assignments, so you should make sure

they are accurate.

3. Note whether or not you used a larger count of exits between outputs (1000 or 2000 exits vs the suggested 500).