

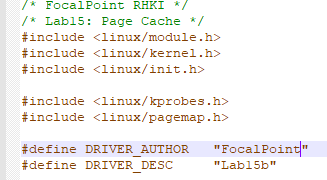
**Page Cache – B**

***Objective: In this lab you will create a LKM called*** cachetest***. It will intercept (via*** kprobes***) all calls to*** add\_to\_page\_cache ***and*** remove\_from\_page\_cache() ***and keep running counters to these calls.***

Kprobes provide us with a powerful interface to intercept any kernel function before and after entry points. This allows us to inspect kernel functions, which might not be otherwise accessible via syscalls. “find\_get\_page” is an example of this. We need to use a Kprobe to access this routine since it lives only in kernel memory.

**File(s) for this lab:**

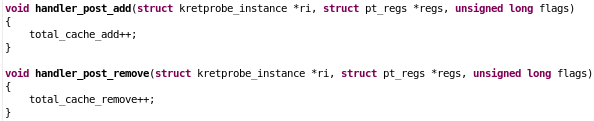
1. To use Kprobes include the header kprobes.h. pagecache.h gives us access to the “add\_to\_page\_cache\_lru” and “delete\_from\_page\_cache”.



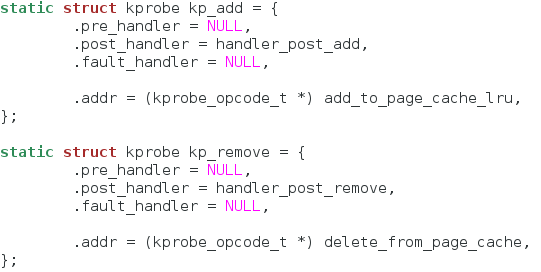
1. Declare some globals for holding the statistics for total accesses.



1. The routines to handle the Kprobe is as follows:



1. The kprobe structure needs the address or the name of the function to intercept.



1. Compile and run the module for a minute or so. Unload the module to see your results.

