*Operating Systems with C/C++*
Semester 1, 2012
Eike Ritter

The University of Birmingham
19 November, 2012

# Assessed Exercise 4

**Deadline: Fri 7 December, 12noon**

## The Task

The purpose of this exercise is to re-implement the task of the previous exercise in kernel space. The firewall should implement some rate limitations. More precisely, for a given port, the firewall should drop all incoming connections on this port if more than a given number of connections arrive in a second. After a specified interval the firewall should accept again connections on this port, subject again to the rate limit.

Your program should work as follows:

- You should write a kernel module which when loaded sets up the additions to the firewall. When the module is unloaded, these additions are removed.
- An example user space program which demonstrates the packet inspection and the timer are found on the webpage.
- For implementing the specified interval during which the firewall should not accept any connections you should use a timer.
- Make sure you handle concurrency correctly. In particular, both the timer code and the code handling the packets is called from an interrupt and can run at any time. Your code should work even if several packets arrive at the same time. You should minimise the length of any critical sections needed.

## Marking Scheme

Please use the School submission system for submitting your code. Please submit only the source files you have written yourself. We will compile and run your code on the Linux machines and mark it accordingly. Please in particular note that we will use the compiler option introduced in the lecture and will deduct 6 marks immediately if there is any compiler error or warning.

We will award marks as follows:

- 5 marks for correct handling of the packets
- 5 marks for correct handling of the rate limitations.
- 5 marks for correctly accepting connections again after timeout.
- 5 marks for the correct handling of the concurrency.