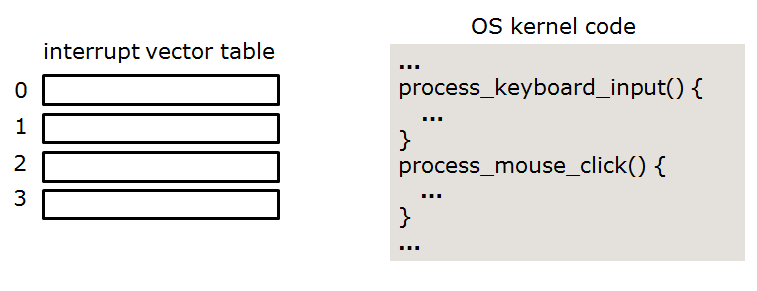
CST 334 (Operating Systems)

Dr. Glenn Bruns

# Lab: Limited Direct Execution

The purpose of this lab is to help you understand the mechanics of interrupts. Please work with a partner. This is a paper-and-pencil lab -- you don't need to log into mlc104.

1. The images below show a blank interrupt vector table, and some operating system code related to the handling of a mouse click.   
   Without looking at the lecture slides or your notes, fill in the sequence of steps that occur when a mouse click happens. DIscuss with your partner to try to remember and understand how this works. Fill in some entries of the interrupt vector table.
   * user process 123 is running
   * user clicks mouse
   * ?
   * ?
   * etc.



Please take your time. When you're finished, please compare your answer to the lecture slides.

1. Suppose a text editor user process is running (as in the previous problem) and a mouse click occurs that is related to a browser process. Is the mouse click interrupt still processed immediately by the OS?
2. When your code does a divide by zero, does either an interrupt or trap occur? If so, is it an interrupt or a trap?
3. What kind of value is found in an entry in the interrupt vector table?
4. Could you write code on mlc104 that would call the process\_mouse\_click() function of the Linux kernel? If not, why not?

If you still have time, please do the following:

1. Figure out the process ID of your current bash shell. Hint: try to remember the command we use for listing processes.
2. Use command ‘ps’ to show all the processes currently running. Read the man page for ps and try a few of the options.
3. Explain what this bash command does:

$ ps -A | wc -l

1. In class we learned that a process can have child processes. Try command ‘pstree’. Read the man page and understand what it does. Try a few of the command line options.
2. Using pstree, write down the path in the tree from the first ‘init’ process to your bash shell.