

## Lab 2:

1. A hardware breakpoint will halt the CPU to stop the program at a specified instruction in a program. Mainly this is used to debug memory corruption and set breakpoints at certain memory addresses. The program breaks when a register assigned to a breakpoint is accessed. The number of hardware breakpoints that can be assigned is limited.

A software breakpoint will halt the CPU to stop the program at a specified line of code in a program. Mainly this is used to follow the return addresses of code and step into and over functions where breakpoints have been assigned. There is no explicit limit on how many software breakpoints can be assigned. Software breakpoints offer a large quantity of breaks at a high level whereas the hardware breakpoints give in-depth memory diagnostics.

2. Breakpoint is where you specify a line of code and when the program arrives at that line it will stop there. Watchpoint is where you specify a condition that must occur and when that condition is satisfied the program will stop at whatever line its at (e.g. stop the moment a value changes for a variable that's being watched).

3. It is possible to change the value of a variable during a debug session. The variable must be in scope at the point where the program has stopped and then the variable can be changed to a new value and the program can resume. The gdb command for doing this is "set [variable] = [value]". Type "c" to resume.

4. Type "p sizeof(my\_string)". This will print the size of the array my\_string.

5. Put a breakpoint in the loop body then type "c 9999". This means continue 9999 times. It will pass over the breakpoint 9999 times and break normally after the 9999<sup>th</sup> iteration.

6. Type "set \$var = priority" then type "b run if priority < \$var". This will store priority variable into a convenience variable called 'var' and then use a conditional break to break when the task is called to run if a new priority variable is less than the one in 'var'.