Brock Sauvage
EECS 678
Lab 2

1. Hardware and software breakpoints, while being similar in a variety of ways, also carry fundamental differences between the two. Software breakpoints have little limitation in regards to how many are placed, and they may only be targeted at code. When being used, the debugger halts the processor whenever the code containing the breakpoint is being executed. Hardware breakpoints, on the other hand, have hardware support and are allowed to be used when reading, writing, or execution instructions are being used. However, hardware breakpoints are limited in usage, with only a few being able to be used at a time.
2. While a breakpoint is meant to halt the execution process of a program at a specific line of code, a watchpoint is a special type of breakpoint that finds the point in time in which a certain variable is changed.
3. It is possible using the command “set”, along with a “var” flag. For example, changing the value of variable “size” would look like “(gdb) set var size = 10”
4. In order to discover the length of “my_string” starting from a breakpoint that is assumed to be located at the beginning of main, a few steps must be taken. Operating under the assumption that we are working with gdb and that the “r” command has already been used to start the program, we would then run into our breakpoint at the beginning of main. From there, we should be able to see the declaration and initialization of the my_string variable. Since we cannot use a regular “print” command to find out it’s length (this just spits our garbage), we’ll have to try something else. By using the command “ptype my_string” on it, we can figure out that my_string is defined as a character array of length 256. Gdb will display something like “type = char[256]”.
5. One method would be to, before running our program, set a breakpoint on the line that our for-loop begins. This would look something like “b filename.c:#”, with “filename.c” being the actual name of our file being debugged, and “#” referring to the line number of the for-loop. Assuming the breakpoint is the first breakpoint that we have placed during this debugging session, we can use “cond 1 i = 10000” in order to set a conditional breakpoint for when the variable is equal to 10000. This will get rid of the need for manual loop traversal in gdb, allowing us to run the program until our desired iteration comes around. From there, we can step into the loop and inspect the code and function call.
6. In order to store the priority value, we should first set a breakpoint to stop at the scheduler function. Once we have run the program and reached this breakpoint, we can use a watchpoint with the command “watch priority” in order to see when the priority variable will be changed. At the point in time when the variable is changed, we can use “set $priority_variable = priority” in order to create a convenience variable, and then immediately store the actual priority variable from the program inside of it.