Operating System Lab Week 3

Question 1. Keep in mind the utility of fork() and exec() system calls in process creation. Exec is used to load a process and fork() creates a child process.

- In the xv6 directory, type “make qemu-gdb”, to start qemu along with GNU Debugger.
- Open another terminal window, type “gdb” while in the same xv6 directory. Enter “source .gdbinit”. Now gdb is connected to the xv6 operating system.
- In the same terminal (gdb), add a breakpoint for exec system call by typing “break exec”. Then type “continue” to reach the first instance where exec() is called. Typing continue again takes us to the next point where the exec() is invoked. Type continue the third time, and in the xv6 OS window run any shell command (ls, cat etc.)

Examine the contents of what gdb reports in all the three instances when breakpoint exec() is reached in the above example. Write a document file with as much details as possible an explanation of what for the exec() is being used by the operating system to do,
1. **initially**
2. **at the second breakpoint**
3. **and why is no breakpoint reached the third time until we execute any of the commands in xv6 operating system**

Question 2. On xv6 shell, type “ls” command. Your task is to make this command more user friendly, by finding out what the numbers in the second column of the result represent.
The xv6 source code files are well documented using comments.
Edit the C file for this command, find out what the numbers in the second column represent (hint : you might want to inspect the #included files) and modify the code appropriately so that it prints a descriptive string in the second column instead of a number, when “ls” is called.
Note: every time you modify xv6, make has to be called again to implement those changes.

Write a document file whatever changes you have made along with the output screenshot.