**Questions 1 & 2.**

*Screenshots at the end of the document.*

**Question 3.**

1. What files did you need to change and why did you need to change them?
   1. syscall.c
      1. Here we add the implementations of our custom-made system calls. Here is where we add out counter, getters and setters etc. We could say this is the main point where we want to create a system call.
   2. syscall.h
      1. We need to create a header for our new system calls in the array of system calls and assign the proper value to identify it afterwards.
   3. sysproc.c
      1. Here we add the functionality or the main code of the functions that we are working with. Here is where we write the code that operates our system calls. In other words, this is where we write what does or system calls do.
   4. usys.S
      1. I think this file is invokes the assembly interruptions to each system call. When we interrupt, we trigger the system calls, and this file is responsible for this.
   5. user.h
      1. Here we just declare the system call functions that our user will be using.
   6. defs.h
      1. Here we store all the definitions. We are talking about variables, constants, data structures, functions, and function prototypes.
   7. myprog.c
      1. This is our program and we wanted to change it because we no longer wanted to print a “hello world”, we wanted to show the number of times our system calls happened.
2. When your program calls the two system calls, what it occurring?
   1. It first run one of the two system calls. It goes to the kernel mode and will be running it until, it finishes or when the interrupt comes along by some error. Then it will go for the next system call and will get the information when the interrupt hits. Between the two system calls the control does no come back to the user, it goes to the System Call handler to define what would we run next. I think it is trying to complete the user request handling the resources as it was taught, the best way possible.
3. Why did we need to edit or not edit the Makefile?
   1. I did no edit the Makefile and I think we did not need to edit it because we already did that for our program. Now, we are calling the system calls from within the program, so the Makefile is still calling the program and the content of the program will be mostly irrelevant to the system call if it compile with the rest of the xv6. We would need to edit it we the files mentioned above in (a) were correctly linked but since we could run assignment 1, the work normally.

Texto

Descripción generada automáticamente

A screenshot of a computer

Description automatically generated