**Mini project report format**

1. Fill out the role of the team members in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Participants’ name-Part A** | **Code Section** | **Report Section** | **Documentation Sec** | **Presentation Sec** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Participants name-Part B** | **Code Section** | **Report Section** | **Documentation Sec** | **Presentation Sec** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Participants name-Part C** | **Code Section** | **Report Section** | **Documentation Sec** | **Presentation Sec** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Create a table like the one provided in the project. But, include the line where do you make the modification, the reason for that modification, an a screenshot. Example:

|  |  |
| --- | --- |
| **File** | **Description** |
| *include/syscall.h* | Example: Create a new system call number, in lines X, and Y |
| *kernel/syscall.c* | Add a pointer to the new system call and then a prototype to the system call function  SCREENSHOT HERE |
| *kernel/sysproc.c* | Example:   1. We declare a counter and set it to zero, the counter is increased in system call sys\_getpid(). 2. We create a new system Call, partA() to return the value fo the counter |
| *user/usys.S* | Populate the call array |
| *user/user.h* | Declare the function for the new system call |
| *Makefile* | Add definition to emulate the OS for the QEMU |
| *user/makefile.mk* | Add the custom program to list of user programs so we can run the test |
| *user/tester*  kernel/sysfunc.h | Add the program to generate the output for the test of the new system call and get the PID  Register your system call there. |
| *Lessons Learned* | The project was useful, or it was a waste of time, etc. We learned blablabla. |
| *Elements to Improve for this project* | I would like to see …… |