REPORT

2. KERNEL MEMORY COPY (Implementing system call)

- Syscall_64.tbl is the registry that contains all the information about the system calls available on that operating system.
- All changes in the system call system will result in change of Syscall_64.tbl

  To implement a system call:

  1. Open the Syscall_64.tbl file and add the entry of a new system call
     a. 451 x86 2dmemcpy sys_2dmemcpy
        1. 451 is the system call number
        2. x86 is the operating system
        3. 2dmemcpy is the name of system call
        4. sys_*function name*

  2. Creating a directory in the root of the kernel with two files one containing the main code of the system call and other one being the makefile.

- The code explanation in the 2dmemcpy.c file:
  1. SYSCALL_DEFINE2 macro defined already included system call without going into much details of that system call being used in the code.
  2. It takes inputs or arguments along with the name of the system call.

     ```
     long copy = __copy_from_user(&output[i][j],&input[i][j],4);
     ```

  3. The above function '__copy_from_user copies the matrix from user space and copies it to the output matrix.
  4. It throws an error if it didn't copy and gives EFAULT error

- The code explanation in the Testing.c file:
  1. Defining the system call TWOMEMCPY which is defined in 2dmemcpy.c file which contains source and destination 2D matrices.
  2. It tests and checks whether the matrix made through the system call is equal to the matrix given initially as the input to the user.
  3. It prints yes or no accordingly.

- The makefile includes the rule which is used to recompile the Testing.c file.

- The patch file includes all the differences in the custom kernel and stalk kernel in the makefile, Syscall_64.tbl and the files and folders present in the custom kernel.