Documentation Q2

Steps to implement System call:

- Adding code into -> kernel/sys.c
- Adding new syscall entry into-> entry/syscalls/syscall_64.tbl
- Generating patch using diff diff -ruN and then applying patch into the stock kernel and compiling it later on.

Logic Of System Call:

- For copying matrix from user __copy_from_user() is used and data is stored in a temporary 2d array/matrix in kernel space.
- Then this temporary array is copied into the destination 2d array/matrix using ___copy_to_user().
- If __copy_to_user() and __copy_to_user() both work successfully 0 is returned otherwise -1 is returned by the kernel_2d_memcpy() function.

Sample Program:

```
#define SYS_kernel_2d_memcpy 451
```

- The above header is defined for our new system call and using Syscall() system call is called. Using the rand() function random floating point numbers are generated. If our system call returns 0 copied matrix is printed otherwise an error message is printed.

Compilation:

A bash script is written for the compilation of stock kernel and patching.

A makefile is written for the running of bash script and sample program