Question 1 Documentation

Exec Family System Call and Basic IPC using Signals API

Program Logic & Flow:

I have added a system call in the kernel that does the following.
I used the macro SYSCALL_DEFINE4 to create a system call which accepts 4 parameters, the destination pointer, the source pointer, number of rows and number of columns. In the body, I run a nested for loop, that is responsible to iterate though the 2D matrix. After the copying action for an element is performed, we increment the 'src' and 'dest' pointers, so that now they point to the next element to be copied.

For each element of the matrix, the following happens:
First, we create a temporary float variable named 'val' in kernel space. This variable is responsible to hold each element of the user 2D matrix one by one. After that, we call the __copy_from_user() function, with the address of 'val' as the first parameter, the current address of element in source matrix1 (which is the address of current element while iterating in the source 2D matrix) as second parameter, and the size of float as the third parameter. Hence, this function copies sizeof(float) amount of bytes from user space to the variable in kernel space. Now, the variable in kernel space holds the same element.

After this has been done, we call the __copy_to_user() function, with the address of element in the destination matrix2 as the first parameter, the address of 'val' as the second parameter, and the size of float as the third parameter. This copies sizeof(float) amount of bytes from the address of the kernel space variable 'val' to the user space address. Now, the current element of user space matrix2 contains the same data as the kernel space variable 'val'.

This process happens for all the elements of the matrix one by one, and in the end, the destination matrix2 contains all the elements of source matrix1.

With this, we have achieved copying, using the kernel space that acts as a temporary storage area. On success, 0 is returned otherwise, 1 is returned.

Changes made in kernel

I have modified sys.c file present in kernel/sys.c with the above-described code. I have also modified the system call tables file present in arch/x86/entry/syscalls/syscall_64.tbl and added the above system call (i.e 'kernel_2d_memcpy') with the syscall number '448'. These changes are visible in the diff file generated.