Assignment 2

Question 2

Part 1:

Write a system call to copy a 2d floating point matrix to another matrix.
Name of system call: copytwodarray.
Name of file containing syscall: a.c
Name of directory: 2darray
Syscall is written using SYSCALL_DEFINEN().
Copy to user and copy from user are used to copy the matrix using a matrix called buffer[40]. Thus this syscall will only work for a 2d matrix containing 40 values at maximum.

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>

SYSCALL_DEFINE4(copytwodarray,float *,source,float *,destination,int,rows,int,columns){
    float buffer[40];
    if(__copy_from_user(buffer,source,rows*columns*sizeof(float)))
        return -EFAULT;
    if(__copy_to_user(destination,buffer,rows*columns*sizeof(float)))
        return -EFAULT;
    return 0;
}
```

Part 2:

Make changes to Makefile in linux source code.This is done by adding ...../crypto/ 2darray.
Add syscall in syscall_64.tbl at the number 451.

Part 3:

Write a C program to test the syscall. This syscall is invoked to copy a matrix from src a 5X5 matrix to a destination matrix. EFAULT is raised if the syscall is not invoked. If it is invoked the program checks if the values were copied correctly from src to dest. Error is raised if some value does not match.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/syscall.h>
#include <errno.h>
#include <linux/kernel.h>

#define KERNEL_TWOD_COPY 451
int check2(float *a,float *b,int len){
    for(int i=0;i<len;i++){
        printf("%f %f\n",a[i],b[i]);
        if(a[i]!=b[i]){
            return -1;
        }
    }
    return 0;
}
void check(int size1,int size2,float src[size1][size2],float dest[size1][size2]){
    for(int i=0;i<size1;i++){
        if(check2(src[i],dest[i],size2)==-1){
            printf("Error system call does not work\n");
            return;
        }
    }
    printf("syscall workd!! Congrats\n");
}
int main(){
    float src[5][5]={{1,1,1,1,1},{2,2,2,2,2},{3,3,3,3,3},{4,4,4,4,4},{5,5,5,5,5}};
    float dest[5][5];
    long sys_call_status;
    sys_call_status=syscall(KERNEL_TWOD_COPY,src,dest,5,5);
    if(sys_call_status!=EFAULT){
        printf("Message: System call invoked congratulations\n");
        check(5,5,src,dest);
    }
    return 0;
}
```

Part 4:

Run the diff command to generate the patchfile between the stock kernel source code and the new kernel. This file can be patched into linux source code to get required functionality.

```
diff --git a/Makefile b/Makefile
index 1f27c4bd0..c66b399ea 100644
--- a/Makefile
+++ b/Makefile
@@ -1100,7 +1100,7 @@ export MODORDER := $(extmod_prefix)modules.order
 export MODULES_NSDEPS := $(extmod_prefix)modules.nsdeps

 ifeq ($(KBUILD_EXTMOD),)
-core-y         += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/
+core-y         += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ 2darray/
 core-$(CONFIG_BLOCK)   += block/
 core-$(CONFIG_IO_URING)    += io_uring/

diff --git a/arch/x86/entry/syscalls/syscall_64.tbl b/arch/x86/entry/syscalls/syscall_64.tbl
index c84d12608..4f5c2f675 100644
--- a/arch/x86/entry/syscalls/syscall_64.tbl
+++ b/arch/x86/entry/syscalls/syscall_64.tbl
@@ -372,7 +372,7 @@
 448    common  process_mrelease    sys_process_mrelease
 449    common  futex_waitv     sys_futex_waitv
 450    common  set_mempolicy_home_node sys_set_mempolicy_home_node
-
+451    common  kernel_2d_memcpy    sys_copytwodarray
 #
 # Due to a historical design error, certain syscalls are numbered differently
 # in x32 as compared to native x86_64.  These syscalls have numbers 512-547.
```