

Kernel memory copy

There is a new system call added to kernel which allows to copy a 2d floating point array from source to destination.

The code for this new system call is part of kernel_2d_memcpy.c along with a Makefile to compile it as part of kernel compilation. There is an entry done in syscall_64.tbl file for this new system call with id – 451 (KERNEL_2D_MEMCPY_SYSCALL)

There are following diff in this kernel vs. the original custom kernel without support for this system call -

```
diff -r linux-6.0/Makefile original/linux-6.0/Makefile
1104c1104
< core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ kernel_2d_memcpy/
--

diff -r linux-6.0/arch/x86/entry/syscalls/syscall_64.tbl original/linux-6.0/arch/x86/entry/syscalls/syscall_64.tbl
375d374
< 451    common    kernel_2d_memcpy    sys_kernel_2d_memcpy
--

Only in linux-6.0: kernel_2d_memcpy
ls linux-6.0/kernel_2d_memcpy/ -l
-rw-r--r-- 1 root root    28 Nov 24 22:34 Makefile
-rw-r--r-- 1 root root   148 Nov 24 22:50 built-in.a
-rw-r--r-- 1 root root    608 Nov 24 22:36 kernel_2d_memcpy.c
-rw-r--r-- 1 root root 148960 Nov 24 22:50 kernel_2d_memcpy.o
-rw-r--r-- 1 root root     0 Nov 24 22:50 modules.order
~
~
```

There are 2 calls used here –

1. `__copy_from_user()` which copies data from user address space to kernel address space, and
2. `__copy_to_user()` which copies data from kernel address space to user address space.

Further, the system call assumes that 2d floating point array have total 25 floating point numbers, so a 5X5 matrix is copied from source to destination by this sample system call.

The test code added has `syscall()` with `KERNEL_2D_MEMCPY_SYSCALL` and passes a 2d floating point array (5X5) with some values which are copied to destination. 2 arrays are checked to be equivalent after this system call.