System Call Sequence (we use `chmod` program as an example)

Mingdong Shang and Wenliang Du (Department of EECS, Syracuse University)

/usr/src/commands/simple/chmod.c
line 198, in main()
if (do_change(*argv++)) ex_code = 1;

/usr/src/commands/simple/chmod.c
line 221, in do_change(name)
if (chmod(name, m))

/usr/src/lib/posix/_chmod.c
line 13, in chmod(name, mode)
return (_syscall(FS, CHMOD, &m));

/usr/src/lib/other/system.c
line 11, in _syscall(who, syscallnr, msgptr)
status = _sendrec(who, msgptr);

/usr/src/lib/i386/rts/_sendrec.s
line 50
int   SYSVEC // trap to the kernel

/usr/src/kernel/protect.c
gate_table[]:
{s_call, SYS386_VECTOR, USER_PRIVILEGE },
        /* corresponding 386 system call is called */

/usr/src/kernel/mpx386.s
_s_call: /*save registers*/
call _sys_call   ! sys_call(function,
        src_dest, m_ptr)   /* call sys_call*/

/usr/src/kernel/proc.c
sys_call(function, src_dest, m_ptr) /*verify
the param, then call mini_send and mini_rec
to process the message*/

/usr/src/servers/fs/main.c
get_work():
/*FS receives the message, calls the
do_chmod() to do the work*/
This is done via using
(*call_vec[fs_call])();
The value of fs_call is CHMOD=15.

/usr/src/servers/fs/protect.c
line 51, in PUBLIC int do_chmod(),
rip->i_mode   =   (rip->i_mode   &
~ALL_MODES) | (mode & ALL_MODES);
//perform the real change mode