

# Process Tables

**Process management in Minix.** Process table is where Minix stores information about processes, and Minix has three different process tables. Part of the reason why Minix has three process tables is that Minix is a micro-kernel system, in which memory management and file system are designed as two server processes (MM and FS) that are separated from the kernel. As a result, there are three process tables in Minix: `proc` for the kernel (`/usr/src/kernel/proc.h`), `mproc` for MM (`/usr/src/mm/mproc.h`), and `fproc` for FS (`/usr/src/fs/fproc.h`). Since MM and FS share some common information in their process tables, it is important to maintain the consistency of their process tables. The consistency maintenance is achieved by communication between the MM and FS processes.

When a data member of `mproc` is changed to a new value, the MM process will call function `tell_fs()` (in `usr/src/mm/utility.c`) to send this new value to the FS process. In the FS process, based on the parameters given by the caller, a corresponding function will be called within FS (see `usr/src/fs/table.c`); this function will set the new value in `fproc`.

**Message Data Structure.** Data exchanges between two processes are achieved via message sending. A process can pass values and pointers to another process. The message data structure is defined in `/usr/include/minix/type.h`.

```
typedef struct {
    int m_source;           /* who sent the message */
    int m_type;            /* what kind of message is it */
    union {
        mess_1 m_m1;
        mess_2 m_m2;
        mess_3 m_m3;
        mess_4 m_m4;
        mess_5 m_m5;
        mess_6 m_m6;
    } m_u;
} message;

/* The following defines provide names for useful members. */
#define m1_i1  m_u.m_m1.m1i1
#define m1_i2  m_u.m_m1.m1i2
#define m1_i3  m_u.m_m1.m1i3
#define m1_p1  m_u.m_m1.m1p1
#define m1_p2  m_u.m_m1.m1p2
#define m1_p3  m_u.m_m1.m1p3
```

The first three are used to pass values and the last three are used to pass pointers. If one wants to pass a structure to or get a structure from a server process (e.g., i-node information), the entries for pointers should be used.