
Subject: [PATCH 12/12] proc: remove ->get_info infrastructure
Posted by [Alexey Dobriyan](#) on Tue, 08 Apr 2008 15:04:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Now that last dozen or so users of ->get_info were removed, ditch it too.
Everyone sane should have switched to seq_file interface long ago.

P.S.: Co-existing 3 interfaces (->get_info/->read_proc/->proc_fops) for proc
is long-standing crap, BTW, thus
a) put ->read_proc/->write_proc/read_proc_entry() users on death row,
b) new such users should be rejected,
c) everyone is encouraged to convert his favourite ->read_proc user or
I'll do it, lazy bastards.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>

```
fs/proc/generic.c      | 7 +-----
include/linux/proc_fs.h | 15 +-----
2 files changed, 2 insertions(+), 20 deletions(-)
```

```
--- a/fs/proc/generic.c
+++ b/fs/proc/generic.c
@@ -69,12 +69,7 @@ proc_file_read(struct file *file, char __user *buf, size_t nbytes,
     count = min_t(size_t, PROC_BLOCK_SIZE, nbytes);
```

```
    start = NULL;
-   if (dp->get_info) {
-       /* Handle old net routines */
-       n = dp->get_info(page, &start, *ppos, count);
-       if (n < count)
-           eof = 1;
-   } else if (dp->read_proc) {
+   if (dp->read_proc) {
        /*
         * How to be a proc read function
         * -----
```

```
--- a/include/linux/proc_fs.h
+++ b/include/linux/proc_fs.h
@@ -40,7 +40,7 @@ enum {
    * /proc file has a parent, but "subdir" is NULL for all
    * non-directory entries).
    *
-   * "get_info" is called at "read", while "owner" is used to protect module
+   * "owner" is used to protect module
    * from unloading while proc_dir_entry is in use
    */
```

```

@@ -48,7 +48,6 @@ typedef int (read_proc_t)(char *page, char **start, off_t off,
    int count, int *eof, void *data);
typedef int (write_proc_t)(struct file *file, const char __user *buffer,
    unsigned long count, void *data);
-typedef int (get_info_t)(char *, char **, off_t, int);

struct proc_dir_entry {
    unsigned int low_ino;
@@ -69,7 +68,6 @@ struct proc_dir_entry {
    * somewhere.
    */
    const struct file_operations *proc_fops;
- get_info_t *get_info;
    struct module *owner;
    struct proc_dir_entry *next, *parent, *subdir;
    void *data;
@@ -187,14 +185,6 @@ static inline struct proc_dir_entry *create_proc_read_entry(const char
*name,
    return res;
}

-static inline struct proc_dir_entry *create_proc_info_entry(const char *name,
- mode_t mode, struct proc_dir_entry *base, get_info_t *get_info)
-{
- struct proc_dir_entry *res=create_proc_entry(name,mode,base);
- if (res) res->get_info=get_info;
- return res;
-}
-
extern struct proc_dir_entry *proc_net_fops_create(struct net *net,
    const char *name, mode_t mode, const struct file_operations *fops);
extern void proc_net_remove(struct net *net, const char *name);
@@ -234,9 +224,6 @@ static inline struct proc_dir_entry *proc_mkdir(const char *name,
static inline struct proc_dir_entry *create_proc_read_entry(const char *name,
    mode_t mode, struct proc_dir_entry *base,
    read_proc_t *read_proc, void * data) { return NULL; }
-static inline struct proc_dir_entry *create_proc_info_entry(const char *name,
- mode_t mode, struct proc_dir_entry *base, get_info_t *get_info)
- { return NULL; }

struct tty_driver;
static inline void proc_tty_register_driver(struct tty_driver *driver) {};
--

```

1.5.4.3