

Procfs Kernel Module

Procfs Kernel Module

- Hello World for /proc
- Steps
 - Create entry in module_init
 - Register reading function with procfs_read
 - Delete entry in module_cleanup

Headers and Globals

```
#include <linux/module.h>
```

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

```
#include <linux/proc_fs.h>
```

```
MODULE_LICENSE("GPL");
```

```
#define ENTRY_NAME "helloworld"
```

```
#define PERMS 0644
```

```
#define PARENT NULL
```

```
struct proc_dir_entry *proc_entry;
```

```
int procfile_read(char *buf, char **buf_location, off_t offset, int  
buffer_length, int *eof, void *data);
```

Creation

```
int hello_proc_init(void) {
    proc_entry = create_proc_entry(ENTRY_NAME, PERMS, PARENT);

    if (proc_entry == NULL)
        return -ENOMEM;
    proc_entry->read_proc = procfile_read;
    proc_entry->mode = S_IFREG | S_IRUGO;
    proc_entry->uid = 0;
    proc_entry->gid = 0;
    proc_entry->size = 11;

    printk("/proc/%s create\n", ENTRY_NAME);
    Return 0;
}
```

Reading

```
int procfile_read(char *buf, char **buf_location,
off_t offset, int buffer_length, int *eof, void *data) {
    int ret;
    printk("/proc/%s read called.\n", ENTRY_NAME);

    /* Setting eof. We exhaust all data in one shot */
    *eof = 1;
    ret = sprintf(buf, "Hello World!\n");

    return ret;
}
```

Reading In General

- `int procfile_read(char *buf, char **buf_location, off_t offset, int buffer_length, int *eof, void *data)`
 - Return bytes read
 - `buf` is the buffer of data read
 - `buf_location` is simply a pointer to a string (`buf`)
 - Allows you to point to a different buffer than the one provided
 - `offset` is how far into the file you are
 - `buffer_length` is how much memory is allocated for `buf`
 - Mark `eof` as 1 when you reach the end of the file
- Goal is to fill a buffer with data from your module

Deletion

```
void hello_proc_exit (void) {  
    remove_proc_entry(ENTRY_NAME, NULL);  
    printk("Removing /proc/%s.\n", ENTRY_NAME);  
}
```

Registration

```
module_init(hello_proc_init);  
module_exit(hello_proc_exit);
```

Testing

```
$ sudo make
```

```
$ sudo insmod hello_proc.ko
```

```
$ sudo tail /var/log/syslog
```

```
$ cat /proc/helloworld
```

```
$ sudo rmmmod hello_proc
```