EMBEDDED DEVICE DRIVERS

Linux Device Drivers on Beaglebone Black

Kernel module: Parameters

- Kernel modules can take parameters
  - Via cmd-line ***(insmod / modprobe)***
  - Or through the **sysfs** entry

    /sys/module/<MY_MODULE>/parameters/<MY_PARAM>

  - Support is through kernel-level macros
    - *module_param()*
      - Used to initialize argument
    - *module_param_array()*
      - Used to send an array as an argument
    - *module_param_cb()*
      - Used to get notification on *argument change*

  - Header file:

    #include <linux/moduleparam.h>

LKM: module_param()

- Macro definition

  module_param(name, type, perms);

- name: The variable name

- type: The variable type
  - *(inv)bool, charp, byte, short, ushort, int, uint, long, ulong*

- perms: Permissions for the **sysfs** entry
  - S_I: Prefix
  - R/W/X: Read/Write/Execute
  - USR/GRP/UGO: User / Group / User-Group-and-others
    - Examples: **S_IRUSR, S_IXGRP, S_IRUGO**
  - Can be ORed together

LKM: module_param_array()

- Macro definition

  module_param_array(name, type, &count, perms);

- name: The variable name
- type: The variable type
  - Same as in *module_param()*
- count: No. of array elements received *(this is an output)*
- perms: Permissions for the sysfs entry
  - Same as in *module_param()*

LKM: module_param_cb()

- Macro definition

  module_param_cb(name, ops, args, perms);

- name: The variable name
- ops: A ***kernel_param_ops*** structure that handles setting, getting and freeing the parameter

  ```
  struct kernel_param_ops {
    int (*set)(const char *val, const struct kernel_param *kp);
    int (*get)(char *buffer, const struct kernel_param *kp);
    void (*free)(void *arg);
  }
  ```

  - **These functions are called (via callback) when the parameter variable is set / read**
  - **The kernel defines set and get functions – which are overridden by these ones**

- args: The arguments to functions in **ops**

- perms: Permissions for the sysfs entry
  - Same as in *module_param()*

LKM: MODULE_PARM_DESC

- Human-readable text strings
  - Describing parameters
  - Visible in ***modinfo***
  - Help user pass proper parameters when loading

- Format

  MODULE_PARM_DESC(name, description);
  - *Example:*

    MODULE_PARM_DESC(myint, “This is an integer variable”);

LKM: Mod-params exercise (1/2)

- Refer the ***mod2*** directory – we deal with mod-params here
  - ***mod21.c*** contains the module src code
    - Study the usage of the *module_param_*()* macros
  - Compile the module and transfer to BBB

```
$ modinfo mo21.ko
```

  - Default load

```
# insmod mod21.ko
# dmesg
# cat /sys/module/mod21/parameters/myint etc.
# rmmod mod21
```

  - Load with cmd line parameters

```
# insmod mod21.ko myint=1000 mycharp="World!" myarr=1,2,3,4,5
# dmesg
# cat /sys/module/mod21/parameters/myint etc.
# rmmod mod21
```

LKM: Mod-params exercise (2/2)

- Refer the ***mod2*** directory – we deal with callbacks here
  - ***mod22.c*** contains the module src code
    - Study the usage of the *module_param_*()* macros
  - Compile the module and transfer to BBB

```
$ modinfo mo22.ko
```

  - Default load

```
# insmod mod22.ko
# dmesg
# cat /sys/module/mod22/parameters/myshort
# rmmod mod22
```

  - Load with cmd line parameters

```
# insmod mod22.ko myshort=900
# dmesg
# cat /sys/module/mod22/parameters/myshort
# echo 345 > /sys/module/mod22/parameters/myshort
# dmesg
# rmmod mod22
```

THANK YOU!