

Lecture 18: May 11

*Lecturer: Alessandro Pellegrini**Scribe: Anzhelo Xhebraj*

18.1 Module Parameters

Parameters might define how the module should behave when executing. In linux, to have variables passed to modules you define global variables and "register" them.

To define global variables of a certain type as module parameters we can use `module_param`.

When declaring these parameters some objects/pseudofiles are placed in `Sysfs`. We have to define the permission mask.

Internal modules are the ones belonging to the main tree of the linux kernel and are included by default in `/lib/modules`. While external kernel modules are external piece of codes. `insmod` allows to insert the external module while `modprobe` checks `/lib/modules`. `rmmod` unloads the module.

18.2 Kernel Module Loading

The kernel allocates some memory to load both code and data structures. This piece of code is subject to relocation. The memory is allocated through `vmalloc` therefore it might not be contiguous in RAM but just virtually.

In kernel 2.4 the relocation was much more limited. So most of the job of the relocation was done in userspace. When applications wanted to load kernel modules they first resolved symbols and then loaded it in ram.

From kernel 2.6 the resolution of symbols is done by the kernel. The set of tools shown previously trigger the kernel actions for memory allocation, module loading and address resolution.

Kbuild for building the kernel modules. A `.ko` file starts with a normal object file. `modpost` creates an additional file with extension `.mod` that has information about the symbols of the `.o` file.

`EXPORT_SYMBOL` tells to Kbuild that it should take into account to make "visible" some symbols of the kernel for module developers to interact with the kernel.

`kprobes` allow to inject code into the kernel for monitoring performance of kernel code. With `kprobe` everything will be probed except what is marked as not to be. Through `INT 3` breakpoints are introduced. It requires only one byte differently from other interrupts. Its code is `0xcc`.

Setting `TF` flag the processor will stop execution at each instruction. `pushf/popf` allows to set the `TF` flag.

Prob handlers are run with preemption disabled.

`flush_tlb_all` is not exported. To find its virtual address we can use `kprob` to get it.

18.3 Kernel Messaging

Specific modules that expose facilities to print on screen or files.

`printk` similar in spirit to `printf`. Accepts a format string and the variables needed to be printed. `printk("%pk", ..)` allows to print the real virtual address of some kernel memory. When issuing a call to `printk` we have to specify a priority.

The different loglevels tell what the kernel has to do with messages that were generated or messages with no priority level specified.

`/proc/sys/kernel/printk` keeps 4 integers about the various log levels. `current`, `default`, `min` etc.

The prints are managed through a ring buffer.

`automake`

References