Sysfs tagging

(Taken almost verbatim from Eric Biederman's netns tagging patch commit msg)

The problem. Network devices show up in sysfs and with the network namespace active multiple devices with the same name can show up in the same directory, ouch!

To avoid that problem and allow existing applications in network namespaces to see the same interface that is currently presented in sysfs, sysfs now has tagging directory support.

By using the network namespace pointers as tags to separate out the the sysfs directory entries we ensure that we don't have conflicts in the directories and applications only see a limited set of the network devices.

Each sysfs directory entry may be tagged with zero or one namespaces. A sysfs_dirent is augmented with a void *s_ns. If a directory entry is tagged, then sysfs_dirent->s_flags will have a flag between KOBJ_NS_TYPE_NONE and KOBJ_NS_TYPES, and s_ns will point to the namespace to which it belongs.

Each sysfs superblock's sysfs_super_info contains an array void *ns[KOBJ_NS_TYPES]. When a a task in a tagging namespace kobj_nstype first mounts sysfs, a new superblock is created. It will be differentiated from other sysfs mounts by having its s_fs_info->ns[kobj_nstype] set to the new namespace. Note that through bind mounting and mounts propagation, a task can easily view the contents of other namespaces' sysfs mounts. Therefore, when a namespace exits, it will call kobj_ns_exit() to invalidate any sysfs_dirent->s_ns pointers pointing to it.

Users of this interface:
- define a type in the kobj_ns_type enumeration.
- call kobj_ns_type_register() with its kobj_ns_type_operations which has
  - current_ns() which returns current's namespace
  - netlink_ns() which returns a socket's namespace
  - initial_ns() which returns the initial namesapce
- call kobj_ns_exit() when an individual tag is no longer valid