Loading and unloading kernel modules in Linux can be accomplished using several commands, each with specific use cases.

Loading Kernel Modules:

- `insmod`: This command is used to load a kernel module directly from its `.ko` file path. It does not resolve dependencies, meaning if the module requires other modules to be loaded, those must be loaded manually beforehand.
  Code
  ```
  sudo insmod /path/to/module.ko
  ```

- `modprobe`: This is the preferred command for loading modules as it intelligently handles dependencies. It searches for modules in the standard module directories (e.g., `/lib/modules/$(uname -r)`) and loads any required dependent modules automatically.
  Code
  ```
  sudo modprobe module_name
  ```

Unloading Kernel Modules:

- `rmmod`: This command is used to unload a specific kernel module. It requires the module to not be in use and any dependent modules that rely on it must be unloaded first.
  Code
  ```
  sudo rmmod module_name
  ```

- `modprobe -r`: This command is used to unload a kernel module and automatically handles the removal of any dependent modules that are no longer required.
  Code
  ```
  sudo modprobe -r module_name
  ```

Important Considerations:

- **Dependencies**:

  `modprobe` is generally recommended over `insmod` and `rmmod` due to its ability to manage module dependencies automatically, simplifying the process and preventing potential issues.

- **Module in Use**:

  A module cannot be unloaded if it is currently in use by a process or another loaded module. Identifying and terminating processes using the module or unloading dependent modules first may be necessary.

- **Root Privileges**:

  Loading and unloading kernel modules typically require root privileges.

- **Module Locations**:

Kernel modules are commonly located in subdirectories under `/lib/modules/$(uname -r)/kernel/`.

- **Listing Loaded Modules**:
  The `lsmod` command can be used to list all currently loaded kernel modules and their dependencies.