Adding a Character Device:

Define the necessary data structures:

struct cdev: Represents the character device.

struct file\_operations: Contains the operations provided by the device driver.

Initialize the cdev structure and assign the file operations:

struct cdev myDevice;

cdev\_init(&myDevice, &fops);

Allocate and assign a device number to the character device:

dev\_t deviceNumber;

alloc\_chrdev\_region(&deviceNumber, minorNumber, count, "device\_name");

Register the character device with the kernel:

cdev\_add(&myDevice, deviceNumber, count);

Create a device file for user-space access:

struct class \*deviceClass;

struct device \*device;

deviceClass = class\_create(THIS\_MODULE, "class\_name");

device = device\_create(deviceClass, NULL, deviceNumber, NULL, "device\_name");

Implement the necessary device driver functionalities in the file operations structure.

Removing a Character Device:

Remove the character device registration:

cdev\_del(&myDevice);

Destroy the device file and device class:

device\_destroy(deviceClass, deviceNumber);

class\_destroy(deviceClass);

Release the allocated device number:

unregister\_chrdev\_region(deviceNumber, count);

Clean up any other resources used by the character device driver.

By following these steps, you can add a character device to the Linux system, allowing user-space applications to interact with the device driver. Similarly, removing the character device involves unregistering the device, destroying the device file and class, and releasing the allocated device number.

Remember to include the necessary headers (linux/fs.h and linux/device.h) and define the file operations (fops) for your character device driver.

Note: Proper error handling and cleanup routines should be implemented to handle potential failures during the device addition and removal process, ensuring resource deallocation and preventing resource leaks.