Tutorial: Key Mapping and Identifying Unused Keys in Arch Linux with `evtest`

Key Mapping: Key mapping allows you to assign specific functions or actions to keys on your keyboard.

1. **Identify Key Events:**
   - Use `evtest` command to monitor input events:

     ```
     sudo evtest
     ```

2. **Press Keys:**
   - Press the keys you want to map. `evtest` will display events for each key press.
3. **Note Key Codes:**
   - Note down the key codes for the keys you want to map. Key codes are displayed in the `code` field of the event output.
4. **Map Keys:**
   - Use a tool like `xmodmap` to map keys to desired actions or functions.

Finding Unused Keys: Identifying unused keys can help in customizing key mappings more effectively.

1. **Use `evtest` to Monitor Input Events:**

   ```
   sudo evtest
   ```

2. **Press Keys:**
   - Press all keys on your keyboard.
3. **Observe Output:**
   - Look for keys that generate events but are not in use or rarely used.
4. **Note Unused Key Codes:**
   - Note down the key codes of unused keys. These are potential candidates for remapping.

Example Output:

```
Right Keypad keys 1 to 9
Event: time 1717731615.983575, type 1 (EV_KEY), code 75 (KEY_KP1),
value 1

Other Right Keypad keys / * - +
Event: time 1717732042.614526, type 1 (EV_KEY), code 83 (KEY_KPDOT),
value 1
Event: time 1717732044.581674, type 1 (EV_KEY), code 98 (KEY_KPSLASH),
value 0
Event: time 1717732045.118088, type 1 (EV_KEY), code 55
```

```
(KEY_KPASTERISK), value 0
Event: time 1717732045.708025, type 1 (EV_KEY), code 74 (KEY_KPMINUS),
value 0
Event: time 1717732050.260868, type 1 (EV_KEY), code 78 (KEY_KPPLUS),
value 1
Event: time 1717732051.163450, type 1 (EV_KEY), code 96 (KEY_KPENTER),
value 1
```

This output shows key events and their corresponding codes. Use this information to map keys or identify unused keys for customization.

Now you’re ready to customize your keyboard mappings efficiently and effectively!