# Bind CAPS+hjkl to arrow keys; CAPS to ESC

Asked 3 years, 1 month ago Active 1 month ago Viewed 6k times

11

4

Currently running Antergos Linux. The set-up I would like to have is the following.

1. Pressing and releasing CAPS sends ESC.
2. The combinations of CAPS and any of CAPS+h, CAPS+j, CAPS+k, CAPS+l send Left, Down, Up, and Right respectively. Upon release of CAPS, ESC is not sent.

Of course, the goal here is to get some VIM-style bindings in programs which do not have them.

It seems that xcape would be useful here:

https://github.com/alols/xcape

But the examples do not get me quite as far as I would like.

Any help is appreciated.

EDIT: I came across a very useful answer here:

https://unix.stackexchange.com/a/163675/267068

Can anybody help me figure how to modify the procedure so that I get CAPS+hjkl as needed. Could I use Hyper_L instead of the Super_L in that answer, and then map Hyper_L + hjkl to left

down, up, right?

keyboard-shortcuts

Share Improve this question Follow edited Feb 3 '18 at 1:57 asked Jan 5 '18 at 6:42

Nonnus
175 1 7

## 3 Answers

Active | Oldest | Votes

11

I wanted to do the exact same thing, and after some search and experiment, finally got it working.

**Solution 1**

(See solution 2 below, which is potentially better)

**Step 1 (mapping caps_lock + hjkl)**: Follow this answer and add the config. You should add to the "us" file if you are using the US keyboard layout and skip the other keybindings that you're not interested in. Then run `setxkbmap -layout us`.

**Step 2 (caps_lock as escape)**: Run `xcape -e 'ISO_Level3_Shift=Escape'`. You can add this line to your `/etc/profile` so you don't have to run it manually after reboot.

==========

**Solution 2 (probably better)**

I was happy with solution 1, until I realized I couldn't use the key bindings in IntelliJ, which is a big bummer. Eventually I figured out that I could just use xmodmap and xcape to do the job, while still being able to use them in IntelliJ!

**Step 1 (mapping caps_lock + hjkl)**: Create a file (say "~/.xmodmap") with the following content:

```
keycode 66 = Mode_switch
keysym h = h H Left
keysym l = l L Right
keysym k = k K Up
keysym j = j J Down
keysym u = u U Prior
keysym i = i I Home
keysym o = o O End
keysym p = p P Next
```

Feel free to skip the last 4 lines. I pasted them because they might be useful to you as well. In

fact I'm really hoping to get the caps_lock enhancement working in Linux.

Then, run `xmodmap ~/.xmodmap` .

**Step 2 (caps_lock as escape)**: Run `xcape -e 'Mode_switch=Escape'` .

**Step 3 (optional)**: To avoid manually applying the keybindings, put the above 2 commands into your `/etc/profile` .

Share Improve this answer Follow edited Mar 28 '18 at 23:30 answered Mar 28 '18 at 18:38

matrinica
**126** 2 4

Many thanks! Sorry for not responding sooner. I was off-grid for some time. However, with Solution 2, I get the following error when I run xcape: "No keycode found for keysym Escape in mapping Mode_switch." – Nonnus Apr 20 '18 at 15:43

My mistake. Forgot to remove some old settings. This is working great. Thank you. – Nonnus Apr 20 '18 at 15:52

4 Looks like solution 2 (step 1) breaks other layouts. It works in English, but when I switch (Win + Space) e.g. to Russian I cannot type letters on these keys and caps + hjkl also doesn't work until I switch back to English. – AlexP11223 Jul 2 '19 at 10:26

Also **shift** + caps + hjkl (text selection) doesn't seem to work in IntelliJ, in other apps it works. – AlexP11223 Jul 9 '19 at 9:08

0

I was having trouble getting this to work properly with /etc/profile, so expanding on the answer from @matrinica I was able to achieve success (on ubuntu/gnome) with the following method:

## Step 0 : install xclip and xcape

```
sudo apt install xclip
sudo apt install xcape
```

## Step 1 : create ~/.xmodmap

Copy the following text:

```
keycode 66 = Mode_switch
keysym h = h H Left
keysym l = l L Right
keysym k = k K Up
keysym j = j J Down
```

Create file with contents:

```
touch ~/.xmodmap
```

```
touch ~/.xmodmap
xclip -o > ~/.xmodmap
```

## Step 2 : create script

Copy the following text:

```
xmodmap ~/.xmodmap
xcape -e 'Mode_switch=Escape'
```

Create file with contents:

```
touch ~/xmodmap.sh
xclip -o > ~/xmodmap.sh
```

Make file executable:

```
chmod +x ~/xmodmap.sh
```

## Step 3 : create autostart entry

Copy the following text:

```
[Desktop Entry]
Type=Application
Exec=sh -c "$HOME/xmodmap.sh"
Hidden=false
X-GNOME-Autostart-enabled=true
Name=xmodmap
Comment=xmodmap script
```

Create file with contents:

```
touch ~/.config/autostart/xmodmap.deskop
xclip -o > ~/.config/autostart/xmodmap.deskop
```

### Note:

This method does not work with Wayland and so is not futureproof. For Wayland compatible methods of achieving similar results please consult:

https://askubuntu.com/a/898462

https://wiki.archlinux.org/index.php/Keyboard_input

https://realh.co.uk/wp/linux-keymap-hacking/

If you are able to come up with a clean method which is Wayland compatible please post it here.

Share Improve this answer Follow

edited Jun 11 '20 at 14:16
Community ♦
1

answered Sep 24 '19 at 20:07
Ace.C
121 3

0

I use 3rd layer symbols and assign arrows to letters `hjkl`

Install `gnome-tweaks` tool

Inside gnome-tweaks go to Additional Layout Options and choose Caps Look as a key to choose the 3rd level.

Then open your terminal and go to `/usr/share/X11/xkb/symbols/` There are all keyboard layouts store on your computer

Make a backup of your layout

```
cp us us_old
```

Modify file as root

```
sudo nano us
```

Third values in arrays are your keys:

```
key <AC01> {[a,A,Home]};
key <AC02> {[s,S,BackSpace]};
key <AC03> {[d,D,Delete]};
key <AC04> {[f,F,End]};
key <AC05> {[g,G]};
key <AC06> {[h,H,Left]};
key <AC07> {[j,J,Down]};
key <AC08> {[k,K,Up]};
key <AC09> {[l,L,Right]};
```

Share Improve this answer Follow

answered Jan 4 at 19:34
Danila Popov
1

You'll need to press ALT+F2, enter "r" to reload the gnome session or just log out and in again after altering the `symbols/us` file. – Jp_ Feb 14 at 20:01