

# AccessibleJoy

Leandro Santana







# Introdução e Problema

- Deficiência física
- Distância do teclado
- Facilidade
- Customizável (Script ou atalho)

## General

Ctrl+Alt+Down	Show the window selection screen
Ctrl+Alt+Up	Show the workspace selection screen
Super+D	Show the desktop
Super+S	Show desklets
Alt+Tab	Cycle through open windows
Alt+F2	Run dialog
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+Z	Undo
Ctrl+Y	Redo

To modify or add keyboard shortcuts, open System Settings -> Keyboard: Shortcuts  
The super key is the same as the windows key.

## System / Screen Shots

Ctrl+Alt+- Delete	Log out
Ctrl+Alt+End	Shut down
Super+L	Lock screen
Ctrl+Alt+Esc	Restart Cinnamon
Super+P	Re-detect display devices
Super+O	Orientation lock

## Screenshots

Shift+Print	Take a screenshot of an area
Shift+Ctrl+Print	Copy a screenshot of an area to clipboard
Print	Take a screenshot
Ctrl+Print	Copy a screenshot to clipboard
Alt+Print	Take a screenshot of a window

## Launchers

Ctrl+Alt+T	Launch terminal
Super+E	Launch home folder

## Universal Access

Alt+Super+=	Zoom in
Alt+Super+-	Zoom out

Must be enabled in System Settings -> Accessibility

## Xed - Text Editor

### Documents

Ctrl+N	Create new document
Ctrl+O	Open a document
Ctrl+S	Save a document
Shift+Ctrl+S	Save the document with new filename
Shift+Ctrl+L	Save all documents
Ctrl+W	Close the document
Shift+Ctrl+W	Close all documents
Ctrl+Alt+PgDn	Switch to the next document
Ctrl+Alt+PgUp	Switch to the previous document
Alt+1 ... 9	Switch to the nth document

### Windows and Panes

F9	Show side pane
Ctrl+F9	Show the bottom pane
F11	Fullscreen on / off
Ctrl+Q	Quit xed

### Find and Replace

Ctrl+F	Find
Ctrl+G	Find the next match
Shift+Ctrl+G	Find the previous match

## Workspaces

Super+F1	Switch to workspace 1
Super+F2	Switch to workspace 2
Super+F3	Switch to workspace 3

And so on... up to Super+F12

## Nemo - File Manager

### General

Ctrl+N	New window
Ctrl+W	Close window or tab
Ctrl+Q	Close all windows
F3	Toggle extra pane
Ctrl+F	Search (find)
Ctrl+D	Bookmark current location
Ctrl+B	Edit bookmarks
Ctrl+M	Make link
Shift+Ctrl+D	Pin/Unpin selection
F1	Show help
Ctrl+F1	Show shortcuts

### Opening

Enter / Ctrl+O	Open
Shift+Ctrl+T	Open in new tab
Shift+Ctrl+O	Open in new window
Ctrl+Alt+O	Open item location (recent files and search only)

### Tabs

Ctrl+T	New tab
Ctrl+PgUp	Go to previous tab
Ctrl+PgDn	Go to next tab
Alt+0 ... 8	Switch to tab #



# Roda qualquer script ou execução de aplicativo.

10 lines (9 sloc) | 162 Bytes

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <unistd.h>
4
5  int main()
6  {
7      char *arg[] = { "/usr/bin/drawing", NULL };
8      execvp(arg[0], arg);
9      return 0;
10 }
```



## Solução

- Modificação de um módulo já existente do Linux de Joystick bluetooth (hid-sony.c);
- Utilização do Joystick direito como cursor (L1 como BTN\_LEFT e R1 como BTN\_RIGHT) e o esquerdo junto com outros botões para realizar comandos;
- Após um botão ou sequência de comandos ser pressionado, um script será executado.



## **Passos/Planejamento:**

1. Prototipação e análise de viabilidade; Concluído
2. Testes com o driver que será modificado; Concluído
3. Modificações básicas como analógico->cursor; Concluído
4. Inserção dos Scripts/Macros em comandos e sequência de comandos; Concluído
5. Finalizações e melhorias. Concluído



## Como foi possível executar scripts:

- A execução dos scripts foi possível devido à função `call_usermodehelper` após o apertado do botão, que permite a execução de um programa direto do kernel.
- Para facilitar uma futura alteração dos scripts, o `call_usermodehelper` chama um script da pasta `Programas` que foi setado a um botão ou sequência de comando específica. Podendo ser alterado depois.



```
if((rd[offset+4] & 0x10) && (rd[offset]==128 && rd[offset+1]==128) && (limitadorq==0)){  
  
    char * envp[] = { "SHELL=/bin/bash",  
    "PWD=/",  
    "HOME=/home/leandro",  
    "USER=leandro",  
    "DISPLAY=:0.0",  
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin",  
    NULL };  
    char * argv[] = { "/home/leandro/AccessibleJoy/Programas/quadrado", NULL };  
    call_usermodehelper(argv[0], argv, envp, UMH_WAIT_EXEC);  
    limitadorq=1;  
}else if(!(rd[offset+4] & 0x10)&&(limitadorq==1)){  
limitadorq=0;
```




## Como foi possível mover o mouse e apertar os botões:

Ao adicionar o `BTN_LEFT` e `BTN_RIGHT` ao joystick, automaticamente ele foi identificado como input de movimento do mouse, fazendo o `ABS_Y` e `ABS_X` determinar a posição absoluta do cursor.

Para poder mover o mouse sem que ele sempre voltasse à posição original ( joystick e sua mola) foi criado duas variáveis acumuladoras ( `posx` e `posy` ), que vão sendo modificadas aos poucos após o movimento do analógico direito.

Para colocar o clique esquerdo(`BTN_LEFT`) e direito(`BTN_RIGHT`) no `L1` e `R1` respectivamente, foi adicionado às opções do joystick esses dois botões. Além disso, a tecla `ENTER` foi inserida ao invés do `START`.



```
if(rd[offset+2]<128&&posx>1){  
    posx=posx-1;  
}  
if(rd[offset+2]>128&&posx<255){  
    posx=posx+1;  
}  
if(rd[offset+3]<128&&posy>1){  
    posy=posy-2;  
}  
if(rd[offset+3]>128&&posy<255){  
    posy=posy+2;  
}  
input_report_abs(input_dev, ABS_X, posx);  
input_report_abs(input_dev, ABS_Y, posy);
```



## XDOTOOL foi usado para execução de atalhos do teclado.

10 lines (9 sloc) | 177 Bytes

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <unistd.h>
4
5  int main()
6  {
7      char *arg[] = { "/usr/bin/xdotool", "key", "alt+F4", NULL };
8      execvp(arg[0], arg);
9      return 0;
10 }
```



# Mercado e Diferenciais

- Existem softwares userspace que realizam mapeamentos parecidos, porém são limitadas a o que já está definido no kernel.
- Também existem hardwares que possuem pessoas com deficiência motora como público alvo, mas costumam ter um valor alto.
- O diferencial é o custo e maior possibilidade de ações ( relacionada a comandos específicos ).



# Xbox Adaptive Controller

Jogue do seu jeito.

R\$ 999\*

**ENCONTRE UM VAREJISTA >**



# Time e Fontes

## Realizador:

Leandro de Jesus Dias Santana

## Fontes:

<https://github.com/torvalds/linux/blob/master/drivers/hid/hid-sony.c>

<https://01.org/linuxgraphics/gfx-docs/drm/input/input-programming.html>

<https://www.xbox.com/pt-BR/accessories/controllers/xbox-adaptive-controller>