

用 usbip 轉發 raspberry pi 的 USB 鍵盤鼠標給 Arch Linux 的 PC



惠狐 megumifox 寫了篇 用PulseAudio將電腦的聲音用手機放出來，文末提到想知道我怎麼用樹莓派轉發 USB 的，於是寫篇文章記錄一下。

起因

家裏有個裝了 Arch Linux ARM 的樹莓派3B 閒置着，裝了 Arch Linux ARM 偶爾上電更新一下，不過因為性能實在不適合做別的事情於是一直在吃灰。某日給老婆安利幻想萬華鏡和老婆看片的時候，老婆不吃安利於是遷怒鍵盤鼠標鍵盤鼠標被長長的 USB 線扯着感覺很難受，於是偶發奇想，能不能利用一下樹莓派的多達 4 個 USB 2.0 端口接鼠標鍵盤呢，這樣鼠標鍵盤就可以跟着樹莓派來回走，不用拖着長長的 USB 線了。

上網搜了一下，Linux 環境有個 usbip 工具正好能做到這個。原理也很直觀，usbip 能把 USB 端口上的數據封裝成 IP 協議通過網絡轉發出去，從而兩個網絡間相互聯通的電腦就可以遠程轉發 USB 了。設置好的話，就像是一臺 PC 多了幾個位於樹莓派上的 USB 端口，插上樹莓派的 USB 設備統統作為 PC 的設備。

這篇文章假設有一個裝了 Arch Linux 的 PC，和一個裝了 Arch Linux ARM 的樹莓派，並且兩者間能通過網絡互相訪問到。別的發行版上大概也可以這麼做，只是我沒有試過。usbip 工具似乎普遍被發行版打包了，除此之外需要的也只是 Linux 內核提供好的功能而已。

設置 Arch Linux ARM 的樹莓派端

假設樹莓派上面網絡已經設置妥當，開機插電就能自動聯網。接下來安裝 usbip 工具：

```
1 $ sudo pacman -Syu usbip
```

然後需要記錄一下樹莓派的 IP 地址：

```
1 $ ip addr
2 3: wlan0: .....
3 inet 192.168.0.117/24 brd 192.168.0.
255 scope global noprefixroute wlan0
4 .....
```

接下來給 udev 添加一個規則，當插入 usb 設備的時候，執行我的腳本 usbipall.sh 把 usb 設備通過 usbip 共享出去：

```
1 $ cat /etc/udev/rules.d/usbipall.rules
2 ACTION=="add", SUBSYSTEM=="usb", RUN
+="/usr/bin/bash /usr/local/bin/usbipal
l.sh"
```

這個 rules 文件 可以在我的 dotfiles 裏面找到。

然後規則調用的 `usbipall.sh` 我這麼寫的，文件同樣在我的 `dotfiles` 裏面：

```
1  #!/bin/sh
2  (
3  allusb=$(usbip list -p -l)
4  for usb in $allusb
5  do
6      busid=$(echo "$usb" | sed "s|#.*
||g;s|busid=||g")
7      if [ "$busid" = "1-1.1" ]
8      then
9          # ignoring usb ethernet
10         continue
11     fi
12     echo "$(date -Iseconds): Exporti
ng $busid"
13     usbip bind --busid="$busid"
14 done
15 ) >>/var/log/usbipall.log 2>&1
```

這個腳本做了這樣幾件事。

1. 調用 `usbip list --local` 列出本地所有 usb 設備。
2. 針對每個設備
 1. 取出它的 `busid`
 2. 判斷是不是樹莓派的 USB 以太網卡，不是的話繼續
 3. 通過 `usbip bind --busid=` 命令把這個

usb 設備導出到網上

3. 最後把所有輸出記錄到 /var/log/usbipall.log 日誌裏面

樹莓派這邊設置就完成了。從此之後插入的 usb 設備就會統統導出出去。

這裏需要注意一下，啓用了 udev 規則之後，就無法插鍵盤鼠標到樹莓派上控制它了……我都是從另一端 ssh 上樹莓派操作的。如果有什麼地方設置錯誤，可能需要把樹莓派的 SD 卡拔下來插到電腦上，刪除掉 rules 文件……

仔細檢查設置正確了之後，重新載入 udev 規則，或者重啓樹莓派：

```
1 # systemctl restart systemd-udev
```

這樣樹莓派這邊就設置好了。

設置 Arch Linux 的 PC 端

同樣假設 PC 這邊也已經聯網。接下來同樣安裝 usbip 工具：

```
1 $ sudo pacman -Syu usbip
```

然後我寫了個小腳本去鏈接樹莓派端，這個文件
usbiprpi3.sh 也在我的 dotfiles：

```
1  #!/bin/sh
2  rpi3="192.168.0.117"
3
4  modprobe vhci-hcd
5
6  allusb=$(usbip list -p -r $rpi3 | cu
t -d":" -f1 -s | sed 's|^[ \t]*||;/^$/d'
)
7  for busid in $allusb
8  do
9      if [ "$busid" = "1-1.1" ]
10      then
11          # ignoring usb ethernet
12          continue
13      fi
14      echo "Attaching $busid"
15      usbip attach --remote=$rpi3 --bu
sid="$busid"
16  done
```

其中腳本第一行填入上面記錄下來的樹莓派的 IP 地址，接下來腳本做了這麼幾件事：

1. 用 modprobe 確認加載 vhci-hcd 通用虛擬鍵鼠驅動
2. 用 usbip list --remote= 列出遠程設備上已經導出了的 USB 設備，取出他們的 busid

3. 對每個設備用 `usbip attach` 接上該設備

然後就已經準備妥當，接下來是見證奇蹟的時刻：

```
1 $ sleep 10; sudo ./usbiprpi3.sh
2 Attaching 1-1.4.3
3 Attaching 1-1.4.1
```

因爲只有一套鍵盤鼠標，所以先 `sleep` 個 10 秒，在此期間快速把鍵鼠拔下來插到樹莓派的 USB 口上去。如果對自己手速沒自信也可以把時間設長一點。然後用 `root` 權限執行 `usbiprpi3.sh`。

一切正常的話，先能觀測插上樹莓派的鍵盤鼠標被樹莓派初始化了一下，比如鍵盤燈會亮，然後這些設備會被導出出去，從而鍵盤燈滅掉，然後 10 秒等待結束後他們被遠程接到了 PC 端，又會被初始化一下，同時 PC 端這邊會有上述 `Attaching` 的輸出。然後鍵盤鼠標就能像平常一樣用啦。

使用體驗

因爲就是通過 IP 轉發 USB 嘛，所以就和普通地接 USB 的體驗差不多，當然前提是網絡環境足夠穩定。在我家間隔 5 米到無線路由器的環境下，基本感覺不到網

絡延遲的影響。通過這種方式聊天上網應該和直接接 USB 設備完全一樣。本文就是在通過樹莓派轉發的前提下用鍵盤打字寫的。

不過如果網絡負載本身就很大的話，可能會一些延遲，比如我開着 OBS 直播打東方的時候，原本就手殘的我感覺更加手殘了……

試過拿着樹莓派在房間到處走，走到無線信號覆蓋不到的地方，usbip 會斷掉，PC 上的現象就像是 USB 設備被拔下來了……所以如果無線網絡不穩的話，可能需要對上面腳本做個循環？不過那樣可能會用起來很彆扭吧。

以及，上述操作 usbip 是走 TCP 3240 端口，數據包大概完全沒有加密，所以考慮安全性的話，最好還是在內網環境使用。不過轉念一想，萬一有別人接上了我導出出去的 USB，也就是截獲我的鍵盤，PC 這邊沒法 attach 設備了，應該馬上會發現吧。我敲打 sudo 之類命令的時候 shell 裏面沒有回顯，就不會再繼續敲密碼了。而且似乎對攻擊者也沒有什麼好處？要是他 usb attach 到了我的設備上，我就能控制他的鍵盤了耶~