Allwinner A64 kernel build procedure

* apt-get install libncurses-dev
* apt-get install gawk
* sudo su
* Kernel Configuration
  1. $cd lichee/linux-3.10
  2. $make menuconfig ARCH=arm64 CROSS\_CPMPILE=../brandy/armv8\_toolchain/gcc-linaro-linux-gnueabihf-4.9-2014.09\_linux/bin/arm-linux-gnueabihf-

//----------------------------------------------------------------------------

No need :

* 3G
* NFC
* LED driver
* DS18B20
* Gsensor
* CMOS camara

//----------------------------------------------------------------------------

* 設定Display
* $export XY=800x480
* $export XY=1024x600
* $export XY=hdmi
* $export XY=vga
* SDK compile
* $ ./build.sh
* SDK 打包固件
  1. $./build.sh pack
  2. Check 🡪 ./tools/pack/sun50iw1p1\_ubuntu\_mate\_xxxxx.img
* Backup emmc to SD card image file -🡪??????
  + - Backup🡪 dd if=/dev/mmcblk0 of=/medial/root/32G/xxx.img bs=64K conv=noerror,sync
    - Restore 🡪 dd if=/media/root/32G/xxx.img of=/dev/mmcblk0
  1. Full backup from emmc:
     + dd if=/dev/mmcblk0 of=/media/debian/8G/mmcblk0
  2. Restore from SD to emmc
     + gunzip /media/debian/8G/mmcblk0.img.gz | pv | dd of=/dev/mmcblk0 conv=sync,noerror bs=64K
* **Remove libreOffice**

sudo apt-get remove --purge libreoffice\*

sudo apt-get clean

sudo apt-get autoremove

* **Linux VisualBox 分享資料夾方法**
  1. Sudo mount –t vboxsf sourceDir DesDir

( example: $sudo mount –t vboxsf tool /home/hant/share )

* Check UBUNTU version & linux kernel version

$ cat /etc/\*-release

Or

$ cat /etc/issue

Or

$ uname –a check linux kernal version

* **tar 指令的常用語法常用參數**
* -c 打包一個 tar 檔案
* -x 解開一個 tar 檔案
* -t 檢視 tar 檔案的內容
* -z 使用 gzip 壓縮
* -v 顯示建立 tar 檔案的過程
* -P 使用絕對路徑
* -f 指定 tar 檔案的檔案名稱。此參數的後面要接檔案名稱，因此要注意參數的順序 (通常是把 f 參數寫在最後一個，或者是與其它參數拆開使用)
* 建立壓縮檔

tar -czvf 檔案名稱.tgz 來源檔案

將 來源檔案 壓縮後，打包成一個名稱為 檔案名稱.tgz 的壓縮檔，來源檔案 可以是檔案或目錄。

來源檔案 也可以有很多個，以空白字元分隔:

tar -czvf 檔案名稱.tgz 來源檔案1 來源檔案2 ... 來源檔案n

* 檢視壓縮檔的內容

tar -tzvf 檔案名稱.tgz

* 解開壓縮檔的檔案到目前的目錄

tar -xzvf 檔案名稱.tgz

* 複製目錄

tar -cvf - 來源目錄 | tar -xvf -

將 來源目錄 下的所有檔案及子目錄複製到目前的目錄，這樣做的好處是可以保留原來的檔案屬性。

**How to make Bootable SD card**

<http://linux-sunxi.org/Bootable_SD_card>

<http://linux-sunxi.org/Manual_build_howto#Setting_up_the_rootfs>

**Backup UBUNTU filesystem**

1. cd /
2. tar -cvpzf ubuntu.tar.gz --exclude=/ubuntu.tar.gz --exclude=/lost+found --one-file-system /

**Restore UBUNTU filesystem**

1. sudo tar -xvpzf /path/to/ubuntu.tar.gz -C /media/whatever --numeric-owner

**Backup Over a Network**

1. on receive ( port 1024): 🡪

nc -l 1024 > backup.tar.gz

1. on send (port 1024): 🡪

tar -cvpz <all those other options like above> / | nc -q 0 <receiving host> 1024

**View Linux boot service**

<https://www.linux.com/learn/cleaning-your-linux-startup-process>

* **list enabled boot services**

$systemctl list-unit-files --type=service | grep enabled

* **generate a list of all services:**

$systemctl list-unit-files --type=service

**Useful Tools to install**

* + tightvncserver
* **使用NetCat 指令:**
* 測試特定的 TCP 連接埠（port :5000）是否有開啟
  + - nc -v 192.168.0.175 5000
* 傳送測試用的 UDP 封包到遠端伺服器
  + - echo -n "foo" | nc -u -w1 192.168.1.8 5000
* 開啟 UDP 連接埠接收資料
  + - nc -lu localhost 5000
* 在兩台主機之間複製檔案
  + - Host A : nc -l 5000 > my.jpg
    - Host B : nc hostB.com 5000 < my.jpg
* 在兩台主機之間複製整個目錄
  + - Host B: nc -l 5000 | tar xvf –
    - Host A: tar cvf - /path/to/dir | nc hostB.com 5000
* 固定頁面的網頁伺服器
  + - while true; do nc -l 8000 < test.html; done
    - 使用瀏覽器開啟 http://HOST\_IP\_ADDRESS:8000/test.html
* **Modify CUPS rules**
  + Modify : /lib/udev/rules.d
  + Add a new rules : 10-cups-usb.rules

ATTR{idVendor}=="04b8", ATTR{idProduct}=="000a", MODE:="0664", GROUP:="lp", ENV{libsane\_matched}:="yes"

$udevadm control --reload-rules

* Install nodejs
  + **Node.js v5.x**

# Using Ubuntu

$curl -sL https://deb.nodesource.com/setup\_5.x | sudo -E bash -

$sudo apt-get install -y nodejs

# Using Debian, as root

$curl -sL https://deb.nodesource.com/setup\_5.x | bash -

$apt-get install -y nodejs

* + Update npm

$ npm i -g npm-check-updates

$ npm-check-updates -u

* Build Nodejs Project

$ npm i -g npm-check-updates

$ npm-check-updates –u

$ npm update

$ npm install

$ npm install java

==Build code ===================

$ ./compile-java.sh ( build java source)

--- Run nodejs code ----------------

$ nodejs index-Serial.js

* + Open web from : 192.168.38.103:8081
* Auto run nodejs on startup
  1. Add new file **start-nodeprinter.sh** in /etc/init.d
  2. In start-nodeprinter.sh : add

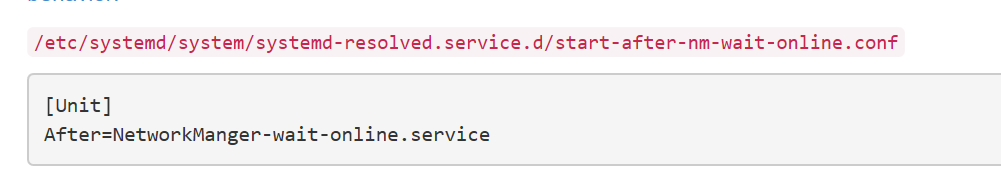
#!/bin/bash

Echo “run nodeprinter….”

Cd /root/nodeprinter

./run.sh

* 1. $ chmod +x /etc/init.d/**start-nodeprinter.sh**
  2. $update-rc.d **start-nodeprinter.sh** defaults 100
  3. 取消開機執行 : $update-rc.d -f **start-nodeprinter.sh** remove
* How to fix “ Failed to start Network Name Resolution”



* Show all system service

$ systemd-analyze blame

* Disable unnecessary service

$systemctl disable ????.service