WIP: Rough notes - Devmfc Debian 13 Trixie on tvbox s905x4 or s905x3

#Devmfc Debian 13 Trixie Linux on a tv box

#see older similar note about Devmfc Debian 12 Bookworm https://justpaste.it/jrdwq

#image file from Github, "Devmfc_Debian-Trixie_6.12.41-meson64_Minimal-25.08.09.img.xz"

#Linux headers file (needed for zfs): "linux-headers-6.12.41-meson64_20250808_arm64.deb"

this is NOT Armbian or Ophub, nor affiliated with them. ☺

Decompress xz file, and then burn the img file to a fresh USB 2.0 pen drive

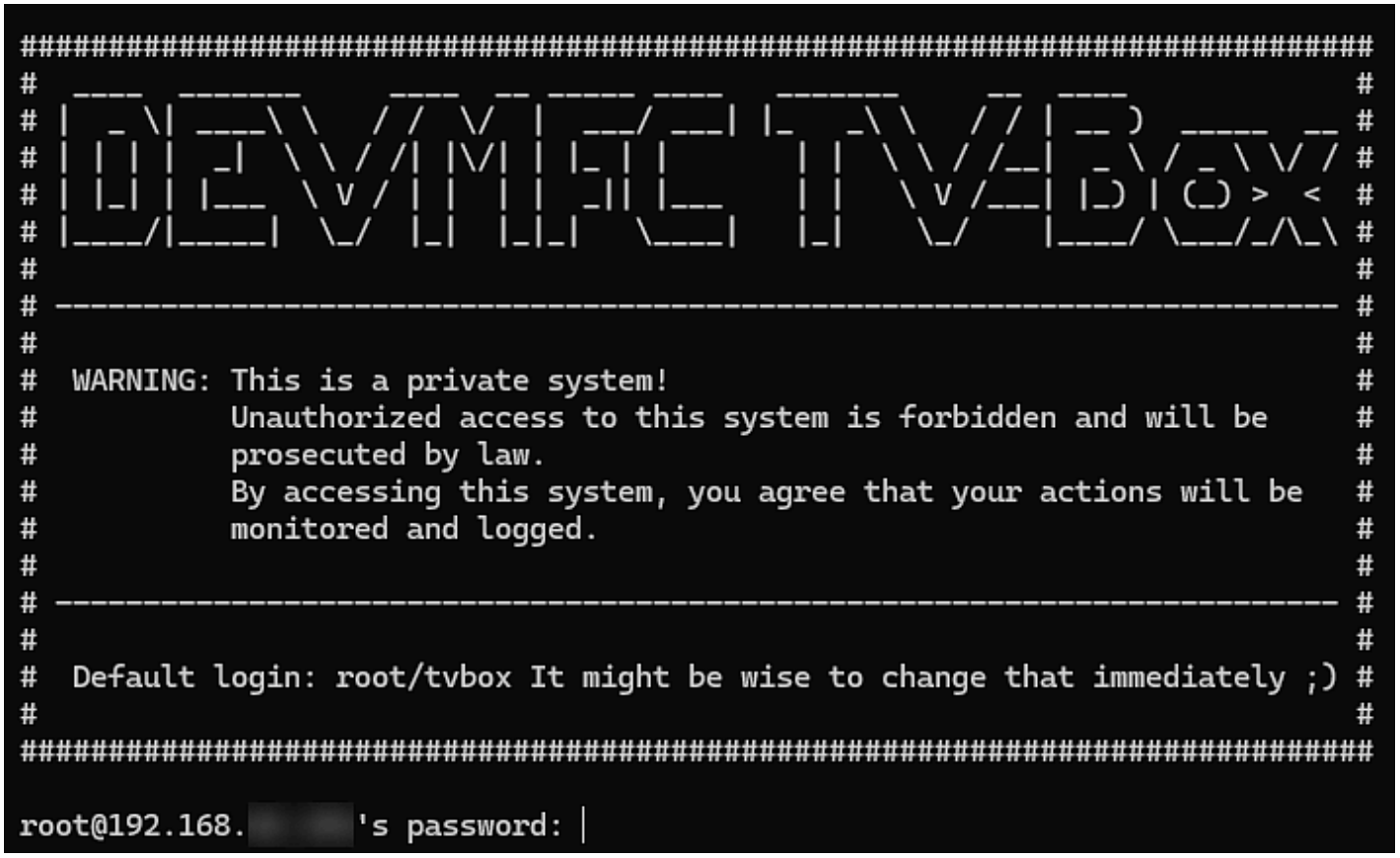
This distribution is very minimal. Not even "sudo" or "man" or "wget" are installed.

These notes can apply to other Amlogic tv boxes such as H96 Max X3

#* NOTE: zfs-dkms on AMLOGIC-ARM64 supports up to kernel 6.14

```
configure: error:
        *** Cannot build against kernel version 6.18.10-meson64.
        *** The maximum supported kernel version is 6.14.
```

#_link, https://github.com/devmfc/debian-on-amlogic/tags



☺

Install devmfc image file - use a USB 2.0 pen/flash drive

#decompress (unzip) file "Devmfc_Debian-Trixie_6.12.41-meson64_Minimal-25.08.09.img.xz"
#and burn the .img file to a new USB 2.0 pen/flash drive or to a memory card that is in a
#USB 2.0 card adapter. put it in the USB 2.0 black port on the tv box.
#Edit the BOOT/boot.config text file. For a "hk1 rbox x4" tv box

```
## Uncomment one of these box configs for s905x4:
 box=s905x4_generic_gigabit

## or uncomment another box config for s905x3, for example a h96 max
 box=h96maxx3
```

#disconnect power from the tv box, put the USB 2.0 pen drive or card adapter
#into the USB 2.0 black port, press and hold the hidden button in the A/V port, and
#connect power to the tv box. a micro sd card in the TF slot may also work.

Disable some things that alleviate future boot problems

#disable waiting for a network connection, disable unattended upgrades
systemctl disable --now systemd-networkd-wait-online.service
systemctl disable --now unattended-upgrades

#hold the current kernel, so that it won't be updated/upgraded
apt-mark hold linux-image-$(uname -r)

Make a backup of the internal EMMC (Android)

#get zstd compression tool for ARM64 linux from github, and make a new link "zstdmt" to "zstd"
#"zstdmt" is multi-threaded for zst compression
#multi-threaded tools use all CPU cores. it's faster than regular gzip/bzip2
apt update -y
apt install -y zstd

#can instead download the arm64 linux file zstd onto a PC and copy it to the tv box
#- https://github.com/aspect-build/zstd-prebuilt/releases/tag/v1.5.6-bcr2
#or use the tv box itself to download the file
wget -O zstd https://github.com/aspect-build/zstd-prebuilt/releases/download/v1.5.6-bcr2/zstd_linux_arm64

chmod +x zstd
ln -s zstd zstdmt

#do the backup, compressing data on the fly. list block storage with "lsblk"
#use multi threaded "zstdmt". be patient and wait ☺
lsblk

dd if=/dev/mmcblk1 bs=1M conv=sparse status=progress | zstdmt > /root/android-emmc-backup-dd.img.zst

ls -l android-emmc-backup-dd.img.zst

Install devmfc system onto internal emmc

#be sure you made a backup of internal emmc before continuing
#first install multiboot so that devmfc automaticaly boots itself
#copy the backup file "uEnv.bak" to a safe place on a PC hard drive
cd /root
./aml-multiboot-setup.sh

#install the system onto emmc
cd /root
./aml-install-to-emmc.sh

#run the command to shutdown the tv box and remove the USB pen/flash
#drive or card adapter. after full shutdown, disconnect and reconnect power cable
poweroff

Booting from the internal EMMC for the first time and install/setup basics

#connect power supply cable to the tv box. be sure USB pen drives are removed.
set a static IP (v4) address on the ethernet connection. edit the file, put in the text,
save and close, and restart systemd-networkd. use public DNS 9.9.9.9 or another.
DNS 76.76.2.1 is from the Control-D website. "/24" is netmask 255.255.255.0
nano /etc/systemd/network/10-eth0.network

    [Match]
    Name=eth0

```
[Network]
DHCP=no
Address=192.168.1.3/24
Gateway=192.168.1.1
DNS=76.76.2.1
DNS=9.9.9.9

[DHCPv4]
RouteMetric=100
[IPv6AcceptRA]
RouteMetric=100
```

systemctl restart systemd-networkd
(or command: *networkctl reload*)

```
#check the IP address information, and check that it works
ip a s eth0
hostname -I       # capital eye
resolvectl | grep -i "dns server"
ip route
ping4 bing.com
ping6 yahoo.com

#set the hostname. edit the file and put in the hostname you want, then save, and close
nano /etc/hostname
                tinkerbox

#apply the hostname. must logout/login to see the new hostname
hostname -F /etc/hostname

# Install basic tools
apt update -y
apt install -y sudo man-db python3 python3-pip zip unzip bzip2 p7zip-full less systemd-cron
apt install -y file unrar htop perl xz-utils smartmontools gdisk rsync ftp-ssl wget curl fdisk coreutils

#Install multi-threaded gzip and xz and zstd compression tools, faster than the regular tools
#these tools use all CPU cores simultaneously for better performance
#"zstdmt" is multi-threaded for zst compression - facebook uses it ☺
#use them like this: "pigz bigfile.img"  or  "tar -Ipigz -cvf backup.tar.gz infile1 infile2 infile3"
```

#or "zstdmt bigfile.img" or like this: "tar -Izstdmt -cvf backup.tar.zst infile1 infile2 infile3"
apt install -y pixz pigz
apt instal -y zstd # if not installed before

#set time zone. Amurika! F^ck yeah! Coming to save the mutha f^cken dayay!
timedatectl set-timezone America/Chicago

#check time zone and local time & date. and check if time date services are running
timedatectl status
systemctl status systemd-timedated.service
systemctl status systemd-timesyncd.service

#* Optional - can delete the backup file .img.zst from /root on EMMC
rm -vi /root/android-emmc-backup-dd.img.zst

Optional: Install setup rsyslog to save logs in directory /var/log

#external small devices such as routers, Raspberry Pi, self-contained NAS products, (etc) can send
#their logs via a TCP/IP network and a single device running Linux with rsyslog can capture those logs.
#can install "logrotate" tool to manage, rotate and compress current and old log files
https://justpaste.it/jbca7
#create user syslog and group syslog for rsyslog application, and add user syslog to groups adm and tty
groupadd -r syslog
useradd -M -r -g syslog -G adm,tty -s /usr/sbin/nologin syslog

#install rsyslog application and stop the application
apt install -y rsyslog
mkdir -p /var/spool/syslog
systemctl stop rsyslog.service syslog.socket

#create and edit the config file /etc/rsyslog.conf - put in the text, save and close
nano /etc/rsyslog.conf

    module(load="imuxsock") # provides support for local system logging
    module(load="imklog" permitnonkernelfacility="on") # provides kernel and non-kernel messages
    $ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat # enable a simpler timestamp format
    $RepeatedMsgReduction on # Filter duplicated messages

```
 ## Set the default permissions for all log files.
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog          # PrivDropToUser/Group settings are optional
$PrivDropToGroup syslog         # - maybe omit them
$WorkDirectory /var/spool/rsyslog

 ## standard log files. do the log by facility (category)
*.*;authpriv.none               -/var/log/syslog
 auth.*,authpriv.*              -/var/log/auth.log

#make sure log files exist, and owner and permissions are right
#repeat touch, chown, chmod for each log file
touch /var/log/syslog
chown syslog:adm /var/log/syslog
chmod 640 /var/log/syslog

#restart (or start) rsyslogd and check if it is working
systemctl restart rsyslog
systemctl status rsyslog
journalctl -u rsyslog     # press space bar key

#Experiment with setting "ForwardToSyslog=no" and "ForwardToSyslog=yes"
#first configure journald to not send messages to (r)syslog. as "ForwardToSyslog=no"
#and configure journald to retain all log data, including boot logs. as "Storage=persistent"
#Decrease how often the logs are written to the persistant storage (also known as 'flushed' or
'synced')
nano /etc/systemd/journald.conf

    Storage=persistent
    ForwardToSyslog=no
    Compress=no
    SyncIntervalSec=5m    # can increase or decrease by a few minutes if there are problems


#restart journald, and check it is working
systemctl restart systemd-journald
```

systemctl status systemd-journald

Read log files in /var/log and read logs in stored in journal

#read boot log in journal
journalctl -b 0
journalctl -b 1

#read other logs in journal. search for a keyword
journalctl | grep [keyword]
journalctl | grep -i zfs
journalctl | grep -i docker | more

#read log files in /var/log
more /var/log/syslog
more /var/log/auth.log

Install ZFS support

☺ be patient, and wait. zfs software module version 2.3.2 or higher will be installed
#and ZFS pool version will be 5000, and ZFS filesystem version will be 5
#reminder: this is Devmfc and not Armbian or Ophub
#need to download and install file "linux-headers-6.12.41-meson64_20250808_arm64.deb"
#google drive link https://drive.google.com/file/d/1YSBy3H1kNymdU8cHMbNCBhiziRn0oNXU/view
#github link https://github.com/devmfc/debian-on-amlogic/releases/tag/v6.12.41
#install file linux headers

wget https://github.com/devmfc/debian-on-amlogic/releases/download/v6.12.41/linux-headers-6.12.41-meson64_20250808_arm64.deb
apt install -y ./linux-headers-6.12.41-meson64_20250808_arm64.deb

install packages for ZFS. ☺ be patient and wait. do NOT force quit anything.
apt update -y # if not already done after linux-headers
apt -y install systemd-cron # for cron jobs in /etc/cron.d and for command "crontab"
apt install -y dkms
apt install -y --no-install-recommends zfs-dkms # **go take a break, cook some food, make a drink**
apt install -y zfsutils-linux
modprobe zfs

check that zfs is loaded and is generally working
ls -l /lib/modules/$(uname -r)/updates/dkms/
zfs version

```
zpool version
lsmod | grep zfs
dmesg | grep -i zfs | head          # use option -i with grep

# make sure to enable zfs services. for whatever odd reason, some are maybe disabled in Devmfc (?)
systemctl enable zfs-load-module.service
systemctl enable zfs-volume-wait.service
systemctl enable zfs-mount.service
systemctl enable zfs-import-cache.service
systemctl enable zfs-import-scan.service
systemctl enable zfs-share.service

#import existing zpool (if one or more exists)
zpool import     # list available pools
zpool import -f POOLNAME    # replace POOLNAME with actual pool

#create a zfs zpool filesystem
#see my other notes - https://justpaste.it/ij0my
#possibly use "wipefs" tool to clear all data from hard drives

#disable automatic setup of samba sharing
zfs set sharesmb=off POOL
zfs set sharesmb=off POOL/DATASET

# on my s905x4 hk1 rbox, the system, boots very quickly and the pool is not automatically
#importing or mounting at boot time, so i tweaked a script /etc/rc.local to make sure all
#pools are imported and mounted.
# edit the file, put in the text, save and close. and make the script executable,
#change sleep time to 3 or more if zpool is not found or not mounted,
# put "#!/usr/bin/bash" at the top, and put an extra line under "exit 0"
nano /etc/rc.local

        #!/usr/bin/bash
        /usr/sbin/blkid > /dev/null
        /usr/bin/sleep 2
        /usr/bin/lsblk  > /dev/null
        /usr/bin/sleep 2
        /usr/bin/systemctl start zfs-import-cache.service
        /usr/bin/sleep 2
        /usr/bin/systemctl start zfs-mount.service
```

```
        /usr/bin/sleep 2
        exit 0

#make rc.local executable
chmod +x /etc/rc.local

#check that your zfs zpool works by rebooting and doing other commands
#change POOL to your pool's name
reboot
mount | grep zfs
zfs get mounted POOL
zpool list -v POOL
zpool status -v POOL
zfs list POOL
ls -l /POOL
df -h /POOL

#... to be continued ...
```