

Build rootfs of Ubuntu

Editor: Zhen Bian

Get the basic rootfs of Ubuntu

Way 1: Using debootstrap

Reference website: <https://wiki.ubuntu.com/ARM/RootfsFromScratch/QemuDebootstrap>

```
sudo apt-get install debootstrap
sudo apt-get install qemu-user-static
sudo mkdir ~/ubuntu-rootfs
sudo qemu-debootstrap --arch armhf bionic ~/ubuntu-rootfs/ --variant=minbase --verbose
```

Way 2: Download from Ubuntu's website

```
mkdir ~/ubuntu-rootfs
cd ubuntu-rootfs
wget http://cdimage.ubuntu.com/ubuntu-base/releases/18.04.5/release/ubuntu-base-18.04.5-base-armhf.tar.gz
sudo tar -xvzf ubuntu-base-18.04.5-base-armhf.tar.gz
rm -f ubuntu-base-18.04.5-base-armhf.tar.gz
cd ..
sudo apt-get install qemu-user-static
sudo cp /usr/bin/qemu-arm-static ~/ubuntu-rootfs/usr/bin/
```

Mount related files and chroot

```
sudo cp -b /etc/resolv.conf ~/ubuntu-rootfs/etc/resolv.conf
sudo vim ~/ubuntu-rootfs/etc/apt/sources.list % Maybe you need to change the source if you
are in China and don't have VPN
```

```
#Source from USTC, replace the ubuntu's source if you need
deb http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic main universe restricted
deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic main universe restricted
deb http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-updates main restricted universe
multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-updates main restricted universe
multiverse
deb http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-security main restricted universe
multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-security main restricted universe
multiverse
deb http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-backports main restricted universe
multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ bionic-backports main restricted universe
multiverse
```

Wirte a bash script `ch-mount.sh` to mount related files:

```
#!/bin/bash

function help() {
    echo ""
    echo "usage: ch-mount.sh [-m <path>] [-u <path>] <command> [<args>]"
    echo ""
    echo "For example: bash ch-mount.sh -m /media/sdcard/"
    echo ""
}

while getopts "m:u:" arg
do
    case $arg in
        m)
            echo "I:MOUNTING"
            sudo mount -t proc /proc ${2}proc
            sudo mount -t sysfs /sys ${2}sys
            sudo mount -o bind /dev ${2}dev
            sudo mount -o bind /dev/pts ${2}dev/pts
            sudo chroot ${2}
            ;;
        u)
            echo "I:UNMOUNTING"
            sudo umount ${2}proc
            sudo umount ${2}sys
            sudo umount ${2}dev/pts
            sudo umount ${2}dev
            ;;
        ?)
            echo "E:Unknow parameter"
            help
            exit 1
    esac
done
```

```
sudo bash ch-mount.sh -m ~/ubuntu-rootfs/ % Run this command to mount
```

Install wanted packages

```
apt-get update
apt install language-pack-en-base sudo ssh net-tools ethtool wireless-tools ifupdown
network-manager iputils-ping bash-completion htop vim resolvconf rsyslog
```

User settings

```
passwd root
adduser XXX
sudo vim /etc/sudoers % Add `XXX ALL=(ALL:ALL) ALL` under the `root` line
echo "ubuntu-arm-zynq">/etc/hostname % Set the hostname
echo "127.0.0.1 localhost">>/etc/hosts
echo "127.0.1.1 ubuntu-arm-zynq">>/etc/hosts
dpkg-reconfigure resolvconf
```

Configure serial port

You need to add the file `/etc/init/ttyPS0.conf` to successfully connect serial port, otherwise you can't log in the serial port.

```
sudo vim /etc/init/ttyPS0.conf % Add the below content to this file
```

```
start on stoppedrc or RUNLEVEL=[12345]
stop on runlevel[!12345]
respawn
exec /sbin/getty -L 115200 ttyPS0 vt102
```

Configure SD card mount

```
sudo vim /etc/fstab % Add the below content to this file
```

```
# <file system> <dir> <type> <options> <dump> <pass>
/dev/mmcblk0p2 / ext4 defaults,noatime,errors=remount-ro 0 1
/dev/mmcblk0p1 /boot/uboot vfat defaults,noatime 0 0
```

Network settings

```
vim /etc/network/interfaces % Delete the line of `source-directory  
/etc/network/interfaces.d`, and add the below content to this file
```

```
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet dhcp
```

```
sudo vim /etc/systemd/system/network-online.target.wants/networking.service % Change  
`TimeoutStartSec=5min` to `TimeoutStartSec=15sec` to avoid waiting too much time for the  
start
```

Exit the chroot and unmount

```
exit % Exit the chroot  
sudo bash ch-mount.sh -u ubuntu-rootfs/ % Unmount files
```