Note

1. Cross compile Linux kernel of different architecture

1.1 Preparation

1.1.1 Cross compile tool

The architecture we need to compile is `arm`.

```
#The following installation method is not recommended.
apt install gcc-arm-linux-gnueabi #armel
apt install gcc-arm-linux-gnueabihf #armh
```

gcc-9 or less .Based on your kernel version.Check it by `linux-x.x.x/include/linux/compiler-gccX.h` and downloaded matched version.The versoin here we choose is `4.9.4` and architecture is *armhf*

1.1.2 Linux source code

Download source code from Linux kernel mirrors.The version here we choose is 3.2.1

1.2 Compile

1.2.1 Prepare config file

Generate `.config` with template config file `vexpress_deconfig` and specify compiler and architecture.

```
make CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm vexpress_defconfig
```

Edit config file

- To avoid .Set `CONFIG_LBDAF=y`

- Set `CONFIG_IPV6=y`

Edit CONFIG_CMDLINE( )

If you are compiling the linux kernel version `3.x.x` or lower. You must specify `CONFIG_CMDLINE` directly instead of `-append` while booting by *qemu-system* if your current kernel version does not support `CONFIG_ATAGS` or else the following error may occur.

```
No filesystem could mount root, tried:  ext3 ext2 ext4 cramfs vfat
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(179,0)
```

Here is mine

```
CONFIG_CMDLINE="root=/dev/mmcblk0p2 rw console=ttyAMA0 mem=128M"
```

- `root=/dev/mmcblk0p2` specify a device to mount root file system
- `rw` mount file system as read-write
- `console=ttyAMA0` specify console echo
- `mem=128M` Without this field you may meet

1.2.2 Compile

After complting your compile configuration. Type the following command to compile kernel

```
make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm
```

Get error while compiling please check 1.4.1 About Errors

1.3. Boot

1.3.1 Prepare File System

Create a disk image with

```
qemu-img create -f raw disk.img 512M
```

And format it to a specific format(`ext3 | ext4`, based on your kernel)

```
mkfs -t ext3 ./disk.img
```

Mount it on a temporary directory and copy your rootfs into it

```
> mount -o loop ./disk.img tmpfs/
> cp -r rootfs/* tmpfs/
> umount tmpfs/ #Don't forget it
```

1.3.2 Booting

Source code of kernel version `3.2.1` without *Device tree blob* file of *vexpress* so we have to download it specifically.

```
qemu-system-arm -M vexpress-a9 -m 512M -kernel /kernel/linux/3.1.2/arch/armhf/zImage
-dtb  /kernel/dtb/vexpress-v2p-ca9.dtb -nographic -sd disk.img \-net nic -net
tap,ifname=tap0,script=no,downscript=no
```

1.4 Errors

1.4.1 About Errors

Normally there is won't cause any error while compiling. If it happened. You must get something wrong(like gcc version does not match the kernel) but the source code. compile kernel in force while your compile version not match the kernel version

The following error just meant to record my compile progress but without any purpose

Even though you can compile the kernal successful after using some forced way to fix the errorYou can't run it successfully either. You may stuck while *Uncompressing Linux... done, booting the kernel* as same as I used to.

libz.so.1: cannot open shared object file

```
apt-get install zlib1g
apt-get install zlib1g:i386
apt-get install libc6-i386 lib32stdc++6 lib32gcc1 lib32ncurses5
```

arch/arm/kernel/return_address.c:66:7: error: redefinition of

'return_address'

```
@arch/arm/include/asm/ftrace.
- extern inline void *return_address(unsigned int level)
+ static inline void *return_address(unsigned int level)
```

```
@arch/arm/kernel/return_address.c
- void *return_address(unsigned int level)
- {
-       return NULL;
- }
```

Can’ t use ‘defined(@array)’ (Maybe you should just omit the defined()?)

```
@kernel/timeconst.pl
- if(!defined(@val))
+ if(!@val)
```

1.5. Reference

- Difference between armhf & armel
- Simulate arm with qemu
- selected processor does not support cpsid i in ARM mode
- Kernel compile config:CONFIG_LBDAF
- linux: running self compiled kernel in qemu: VFS: Unable to mount root fs on unknown wn-block(0,0)
- Enable & Disable IPV6 on Linux
- Compile and simulte arm with qemu
- Vexpress device tree blob

1.6. Reflection

1.6.1 About `Unable to mount root fs on unknown-block`

Mostly it stems from incorrect kernel booting parameters. I tested on kernel version `4.12` first so when I switched to `3.2.1`. I encounter this problem and I spent a whole day to solved it :(.Finally I realized that it cause by compile configuration `CONFIG_CMDLINE` because this version doesn’t support dynamically specifying kernel booting parameters dynamically. So what ever how I change *qemu-system*’s parameter `-append` it still went kernel panic.

1.6.2 About Kernel error `Out of Memory`

It won't occur on kernel version `4.12` but `3.2.1` will. I attempted to specify *qemu-system*'s parameter `-m` to increase memory but it doesn't worked. After many tests I found that is affacted by again. `mem 128M` must be specified in `CONFIG_CMDLINE`.

1.6.3 SSH host from qemu `exited: No matching algo kex`

```
echo "KexAlgorithms diffie-hellman-group1-sha1" >> /etc/ssh/sshd_config
```

1.6.4 SSH host from qemu `exited: No matching algo hostkey`

```
echo "HostKeyAlgorithms +ssh-rsa" >> /etc/ssh/sshd_config
```