## Кросс-компиляция

Заходим в папку с исходником ядра и сначала вводим make distclean, для зачистки всех наших экспериментов.

С помощью ARCH=arm make defconfig, ARCH=arm make menuconfig собираем файлы конфигурации

Вводим ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- make -j16 zImage для сборки ядра Linux для архитектуры ARM Запускаем ARCH=arm make dtbs для сборки всех dtb файлов В папке /linux/arch/boot/dts ищем файл vexpress-v2p-ca9.dtb и помещаем его в ту же директорию что и zImage

В конце запускаем QEMU\_AUDIO\_DRV=none qemu-system-arm -M vexpress-a9 -kernel zImage -dtb vexpress-v2p-ca9.dtb append"console=ttyAMA0" -nographic эта команда запускает QEMU, эмулятор аппаратного обеспечения, для архитектуры ARM И получаем в результате вывод загрузки ядра, которая заканчивается Kernel panic

```
[ 2.81251] (driver?)
[ 2.81278] 0190 65536 ram0
[ 2.81378] 0190 65536 ram10
[ 2.81386] (driver?)
[ 2.81386] 0190 65536 ram10
[ 2.81386] 0190 65536 ram11
[ 2.81384] (driver?)
[ 2.81384] (driver?)
[ 2.81378] 0190 65536 ram12
[ 2.81378] 0190 65536 ram13
[ 2.81378] 0190 e 65536 ram13
[ 2.81378] 0190 e 65536 ram13
[ 2.81378] 0190 e 65536 ram14
[ 2.81387] 0190 e 65536 ram15
[ 2.81378] 0190 e 65536 ram15
[ 2.81378] 0190 e 65536 ram15
[ 2.81378] 0191 e 65536 ram15
[ 2.81378] 0191 e 131072 mtdblock0
[ 2.81387] 0191 e 131072 mtdblock0
[ 2.81387] 0191 e 131072 mtdblock0
[ 2.81387] 0191 e 131072 mtdblock0
[ 2.81378] 0191 e 131072 mtdblock0
[ 2.81388] 0191 e 131072 mtdblock0
[ 2.81388]
```