## Корневая файловая система

Создали файл init.c и собрали его под arm архитектуру arm-linux-gnueabihf-gcc -static init.c -o init

Заархивировали его с помощью команды echo init | cpio -o -H newc | gzip > initramfs.cpio.gz

Запускаем наше временную корневую файловую систему QEMU\_AUDIO\_DRV=none qemu-system-arm -M vexpress-a9 -kernel zlmage -initrd initramfs.cpio.gz -dtb vexpress-v2p-ca9.dtb -append "console=ttyAMA0" -nographic

```
Q 2.112761] usbcore: registered new Interface driver usbild

2.112781] usbcore: registered new Interface driver usbild

2.112782] usbchid: USB HID core driver

2.112783] usbchid: USB HID core driver

2.112783 usbblid: USB HID core driver

2.112783 usb
```

Далее мы должны скачать busybox и собирать его

ARCH=arm make qemu\_arm\_defconfig
ARCH=arm make menuconfig | ставим статическую сборку и заполняем префикс
ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- make -j 16
QEMU\_AUDIO\_DRV=none qemu-system-arm -M virt -bios u-boot.bin -nographic

получаем busybox файл, который нам потребуется для сборки корневой файловой системы

Собираем её QEMU\_AUDIO\_DRV=none qemu-system-arm -M vexpress-a9 -kernel zlmage -initrd initramfs.cpio.gz -dtb vexpress-v2p-ca9.dtb -append "console=ttyAMA0 rdinit=/bin/ash" -nographic

## И в конечном итоге получили простую корневую файловую систему