**VirtIO-Serial**

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改者 | 日期 | 描述 |
| V1.0 | 汪博文 | 2023-09-18 | 初版 |
| V1.1 | 汪博文 | 2023-10-07 | 添加emulator连接host串口（物理/虚拟）的方法 |

**1 文档目的**

Vela中基于现有的Virtio框架实现了Virtio-Serial驱动，本文档说明如何在QEMU中使用该设备让Vela能和Linux进行通信。

**2 编译和运行**

**配置**

|  |
| --- |
| Makefile CONFIG\_OPENAMP=y CONFIG\_DRIVERS\_VIRTIO=y CONFIG\_DRIVERS\_VIRTIO\_MMIO=y CONFIG\_DRIVERS\_VIRTIO\_SERIAL=y |

**编译**

下面的两个配置默认使能了VIRTIO\_SERIAL，不需要进行额外的配置即可使用

|  |
| --- |
| Bash /\* 社区支持的qemu配置 \*/ ./build.sh ./nuttx/boards/arm64/qemu/qemu-armv8a/configs/netnsh -j14 /\* 内部支持的goldfish配置 \*/ ./build.sh vendor/qemu/boards/vela/configs/goldfish-armeabi-v7a-ap -j14 或者 ./build.sh vendor/qemu/boards/vela/configs/goldfish-arm64-v8a-ap -j14 |

**运行**

使用如下的命令运行，创建两个virtio-serial设备（/dev/ttyV0和/dev/ttyV1），并使用qemu启动编译好的vela，在host中可以通过telnet与创建的两个串口设备进行通讯

|  |
| --- |
| Bash /\* 社区支持的配置直接使用qemu运行即可 \*/ qemu-system-aarch64 -cpu cortex-a53 -nographic \ -machine virt,virtualization=on,gic-version=3 \ -chardev stdio,id=con,mux=on -serial chardev:con \ -global virtio-mmio.force-legacy=false \ **-device virtio-serial-device,bus=virtio-mmio-bus.0 \** **-chardev socket,telnet=on,host=127.0.0.1,port=3450,server=on,wait=off,id=foo0 \** **-device virtconsole,chardev=foo0 \** **-device virtio-serial-device,bus=virtio-mmio-bus.1 \** **-chardev socket,telnet=on,host=127.0.0.1,port=3451,server=on,wait=off,id=foo1 \** **-device virtconsole,chardev=foo1 \** -mon chardev=con,mode=readline -kernel ./nuttx/nuttx  /\* 内部支持的goldfish版本，使用emulator.sh脚本启动 \*/ ./emulator.sh -no-window -qemu \ **-device virtio-serial-device,bus=virtio-mmio-bus.0 \** **-chardev socket,telnet=on,host=127.0.0.1,port=3450,server=on,wait=off,id=foo0 \** **-device virtconsole,chardev=foo0 \** **-device virtio-serial-device,bus=virtio-mmio-bus.1 \** **-chardev socket,telnet=on,host=127.0.0.1,port=3451,server=on,wait=off,id=foo1 \** **-device virtconsole,chardev=foo1** |

|  |
| --- |
| Bash /\* 启动后可以看到如下的/dev/ttyV0和/dev/ttyV1设备 \*/  NuttShell (NSH) NuttX-12.0.0-vela nsh> ls /:  dev/  proc/ nsh> ls /dev /dev:  console  null  telnet  ttyS0  ttyV0  ttyV1  zero nsh> |

Telnet连接命令

|  |
| --- |
| Bash /\* 连接 virtio-serial 0 \*/ telnet 127.0.0.1 3450 /\* 连接 virtio-serial 1 \*/ telnet 127.0.0.1 3451 |

演示如下

**[2023-03-22-14-19-37.mp4]**

**ex3 测试**

virtio-serial可以实现让host和Vela通过串口来进行通讯，上面演示了通过telnet和Vela进行通讯，在编写测试用例时，host可以通过socket（telnet）来和Vela进行通讯，启动host端socket使用提前定义好的端口即可，如下：

Vela:

|  |
| --- |
| C #include <nuttx/config.h>  #include <sys/ioctl.h> #include <stdio.h> #include <unistd.h> #include <fcntl.h> #include <errno.h> #include <debug.h>  int main(int argc, FAR char \*argv[]) {  ssize\_t ret = 0;  ssize\_t ret1;  char buf[256];  int fd;   fd = open("/dev/ttyV0", O\_RDWR);  if (fd < 0)  {  printf("Open failed, err=%d\n", errno);  return ERROR;  }   printf("Echo test start\n");  while (1)  {  ret = read(fd, buf, 256);  if (ret <= 0)  {  printf("Read failed, err=%d\n", errno);  break;  }   lib\_dumpbuffer("Read data:", (const uint8\_t \*)buf, ret);  ret1 = write(fd, buf, ret);  if (ret1 != ret)  {  ret = ret1;  printf("Write failed, ret1=%zd, err=%d\n", ret1, errno);  break;  }  }   close(fd);  return ret; } |

Host（Linux）：

|  |
| --- |
| C /\* 连接 virtio-serial 0 \*/ telnet 127.0.0.1 3450 /\* 连接 virtio-serial 1 \*/ telnet 127.0.0.1 3451 |

**4 连接Host串口**

QEMU支持了将不同的Linux设备作为Virtio Serial设备后端的能力，上面演示了将socket作为Virtio Serial后端的能力，实际上，使用如下的启动命令，可以在vela中直接访问Host（Linux）上的串口（物理或者虚拟）

**4.1 内部支持的goldfish配置**

内部支持的goldfish版本，使用emulator.sh脚本启动，但是访问/dev/ttyACM0需要root权限，可以使用如下的两种方式：

可以临时使用如下的命令为user添加访问权限，但缺点是每次重新插入设备都需要执行一次

|  |
| --- |
| C sudo chmod 0666 /dev/ttyACM0 |

另外一种方法是永久修改用户的访问权限，执行如下命令后，重新插入USB设备即可：

|  |
| --- |
| C sudo echo KERNEL=="ttyACM\*", OWNER="root", GROUP="users", MODE="0666" > my\_usb\_dev.rules sudo cp my\_usb\_dev.rules /etc/udev/rules.d/ sudo service udev restart |

启动命令如下：

|  |
| --- |
| C /\* 内部支持的goldfish版本，使用emulator.sh脚本启动 \*/ ./prebuilts/emulator/tools/emulator.sh -no-window -qemu \ -device virtio-serial-device,bus=virtio-mmio-bus.0 \ -chardev serial,path=/dev/ttyACM0,id=foo0 \ -device virtconsole,chardev=foo0 |

**4.2 社区支持配置**

社区支持的配置直接使用qemu运行即可:

|  |
| --- |
| C /\* 社区支持的配置直接使用qemu运行即可 \*/ sudo qemu-system-aarch64 -cpu cortex-a53 -nographic \ -machine virt,virtualization=on,gic-version=3 \ -chardev stdio,id=con,mux=on -serial chardev:con \ -global virtio-mmio.force-legacy=false \ -device virtio-serial-device,bus=virtio-mmio-bus.0 \ -chardev serial,path=/dev/ttyACM0,id=foo0 \ -device virtconsole,chardev=foo0 \ -mon chardev=con,mode=readline -kernel ./nuttx/nuttx \ -gdb tcp::7777 |

实现细节参考[VirtIO Serial](https://xiaomi.f.mioffice.cn/wiki/wikk4ideCKKtWZI6p1qnaSSvenf)。