

实验环境

实验平台

- Window 10 Pro 21H2 19044.1766
- Arch Linux on Windows 10 x86_64 (Kernel: 5.10.102.1-microsoft-standard-WSL2)

工具依赖

- TigerVNC (Windows)
- NASM 2.16.01 (WSL)
- QEMU emulator 8.0.3 (WSL)

实验步骤

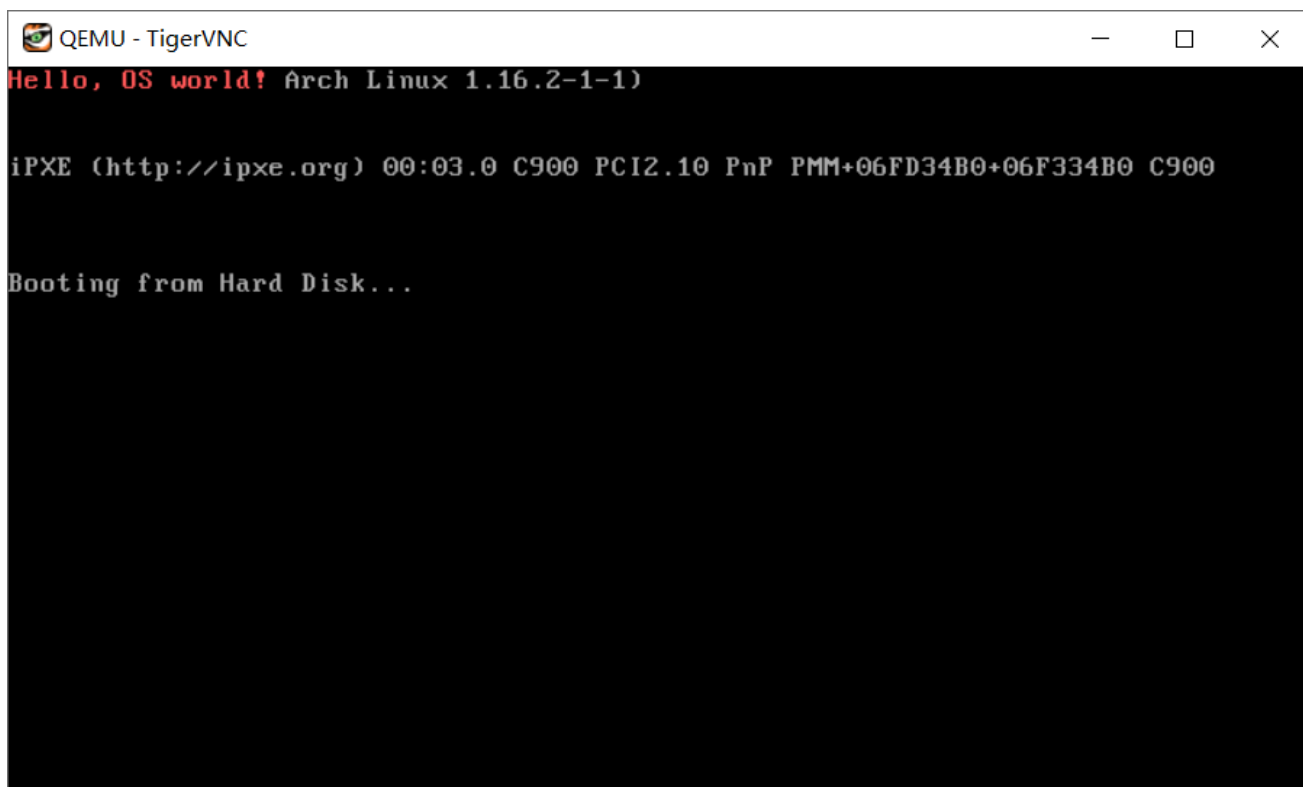
目 录 (一)

```
nasm -o boot.bin boot.asm  
qemu-system-i386 -boot order=c -drive  
file=boot_1.bin,format=raw
```

此时 qemu 将完成虚拟机的创建及 boot 的挂载并启动 VNC 服务端。

```
VNC server running on 127.0.0.1:5900
```

使用 TigerVNC 观察 qemu 模拟的输出情况：



```
QEMU - TigerVNC
Hello, OS world! Arch Linux 1.16.2-1-1)

iPXE (http://ipxe.org) 00:03.0 C900 PCI2.10 PnP PMM+06FD34B0+06F334B0 C900

Booting from Hard Disk...
```

红色块 "Hello, OS world!" 为 boot 的预期行为。

内容 (二)

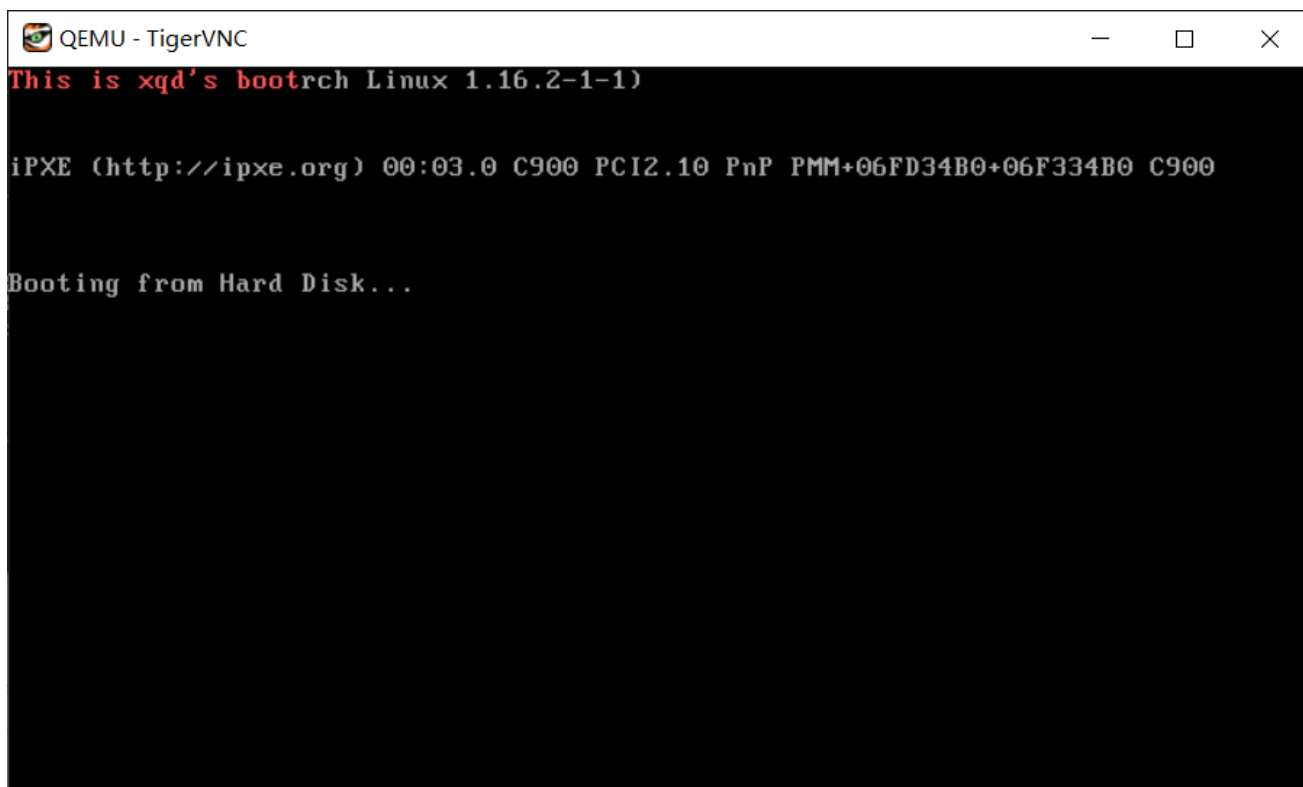
原 boot.asm 调用 10h 号中断的 13h 号服务，该服务要求的参数如下：

```
BH: page number
BL: attribute
CX: total chars to print
DH: row
DL: column
```

待替换字符串为 "This is xqd's boot"，字符串长度为 18（不含 '\0'）。

故修改目标字符串的内容并将送入 CX 寄存器的值修改为 18。

得到结果如下：



内容 (三)

清空屏幕调用 10h 中断的 00h 视频模式重置服务的 03h 显示模式实现。

重置视频模式将清空屏幕并将光标定位至左上角，且 03h 80x25 chars, Colour 模式是显卡加电自检设置的默认模式，故选用该模式可以保证屏幕清楚的同时不改变原有屏幕显示模式。

```
ClearScreen:
    mov ax, 0003h
    int 10h
    ret
```

定位输出 "NWPU" 即在实验 (一) 的基础上利用 13h 服务的 dx 寄存器设定输出坐标。

重写 DispStr 使其匹配函数签名 `void DispStr(char *str, int len, int style, int row, int col)` 并约定 ax 传递 str, cx 传递 n, bx 传递 style, dh 传递行数, dl 传递列数。

DisplayString:

```
mov bp, ax
mov ax, 1301h
mov bh, 00h
int 10h
ret
```

添加 MoveCursorTo 用于单独设定光标位置，并约定 ah 传递行数，al 传递列数：

MoveCursorTo:

```
mov dx, ax
mov ah, 02h
int 10h
ret
```

主要逻辑及实验结果：

```
call ClearScreen      ; ClearScreen()
mov  ax, .LC0
mov  cx, 4h
mov  bx, 1fh
mov  dx, 1326h
call DisplayString    ; DisplayString("NWPU", 4, 0x1f,
19, 38)
mov  ax, 0a0ah
call MoveCursorTo     ; MoveCursorTo(10, 10)
```



内容 (四)

err_boot.asm 对应的 boot 信息:

```
QEMU - TigerVNC
Booting from Hard Disk...
Boot failed: not a bootable disk

Booting from Floppy...
Boot failed: could not read the boot disk

Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.20.1+ (g4bd0) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP iSCSI TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:12:34:56 using 82540em on 0000:00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:12:34:56)..... ok
net0: 10.0.2.15/255.255.255.0 gw 10.0.2.2
Nothing to boot: No such file or directory (http://ipxe.org/2d03e13b)
No more network devices

Press Ctrl-B for the iPXE command line..._
```

x86 架构使用小端序, 而 bootloader 的尾部魔术数为 0x55 0xaa。

在 `err_boot.asm` 中 `dw 0x55aa` 实际指定的是数据段 `0xaa 0x55`，故调整字节序修正为 `dw 0xaa55` 后即可被正常识别为 bootloader。

目录内容 (五)

Makefile 实现目标：

- [默认] `%.asm` 编译到 `%.bin`
- `run [TARGET=<target>]`
- `clean`

为了使 `%.asm` 编译到 `%.bin` 为 `make` 默认行为，需要将目标 `%.bin` 追加到 `all` 目标依赖。

两次 `make` 执行实验结果如下（确保第一次执行时 `.bin` 文件皆删除，且第二次执行时未改动 `.asm`）：

```
@zymelaii → reprod make
Compiling boot_1.asm
Compiling boot_2.asm
Compiling boot_3.asm
Compiling boot.asm
Compiling err_boot.asm
@zymelaii → reprod make
make: Nothing to be done for 'all'.
@zymelaii → reprod
```

`clean` 目标执行 `rm -rf *.bin` 清除生成文件。

`run` 目标通过增加外部参数指定需要 `qemu` 运行的 bootloader（默认值为 `boot`）。

其中 `clean`、`run` 不生成输出项，故标记为 `.PHONY` 隐规则。