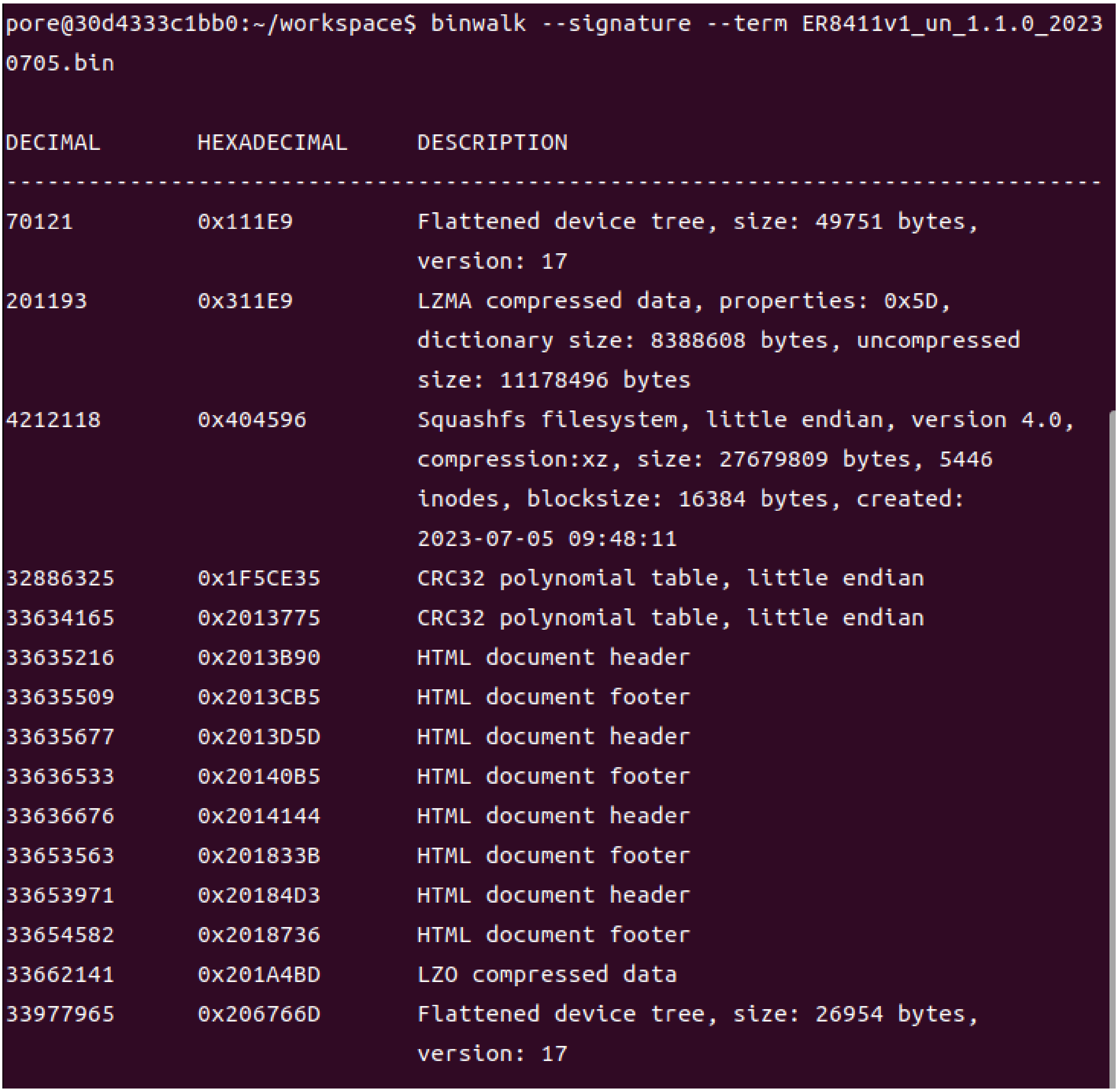
PoRE_lab10

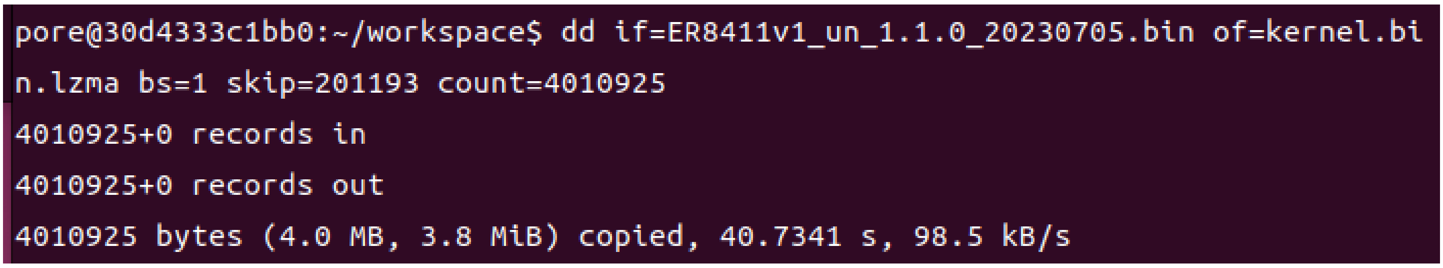
任务一

md5值=b9995e2de290344f28f9cfc5c16cdeb4



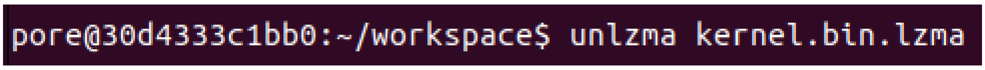
```
pore@30d4333c1bb0:~/workspace$ binwalk --signature --term ER8411v1_un_1.1.0_2023
0705.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
70121         0x111E9         Flattened device tree, size: 49751 bytes,
                              version: 17
201193        0x311E9         LZMA compressed data, properties: 0x5D,
                              dictionary size: 8388608 bytes, uncompressed
                              size: 11178496 bytes
4212118       0x404596        Squashfs filesystem, little endian, version 4.0,
                              compression:xz, size: 27679809 bytes, 5446
                              inodes, blocksize: 16384 bytes, created:
                              2023-07-05 09:48:11
32886325      0x1F5CE35       CRC32 polynomial table, little endian
33634165      0x2013775       CRC32 polynomial table, little endian
33635216      0x2013B90       HTML document header
33635509      0x2013CB5       HTML document footer
33635677      0x2013D5D       HTML document header
33636533      0x20140B5       HTML document footer
33636676      0x2014144       HTML document header
33653563      0x201833B       HTML document footer
33653971      0x20184D3       HTML document header
33654582      0x2018736       HTML document footer
33662141      0x201A4BD       LZO compressed data
33977965      0x206766D       Flattened device tree, size: 26954 bytes,
                              version: 17
```

没有明确标识出kernel的位置，推测kernel以压缩的形式存储。



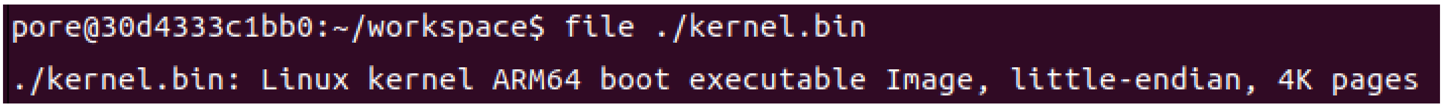
```
pore@30d4333c1bb0:~/workspace$ dd if=ER8411v1_un_1.1.0_20230705.bin of=kernel.bi
n.lzma bs=1 skip=201193 count=4010925
4010925+0 records in
4010925+0 records out
4010925 bytes (4.0 MB, 3.8 MiB) copied, 40.7341 s, 98.5 kB/s
```

开始位置在201193，结束位置在4212118，然后解压



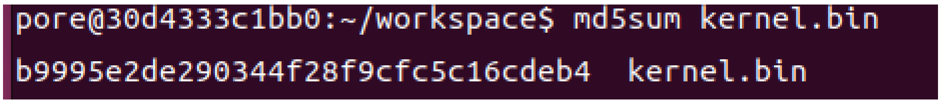
```
pore@30d4333c1bb0:~/workspace$ unlzma kernel.bin.lzma
```

使用file命令出现"Linux kernel ARM64 boot executable Image, little-endian, 4K pages"



```
pore@30d4333c1bb0:~/workspace$ file ./kernel.bin
./kernel.bin: Linux kernel ARM64 boot executable Image, little-endian, 4K pages
```

最后提取md5值

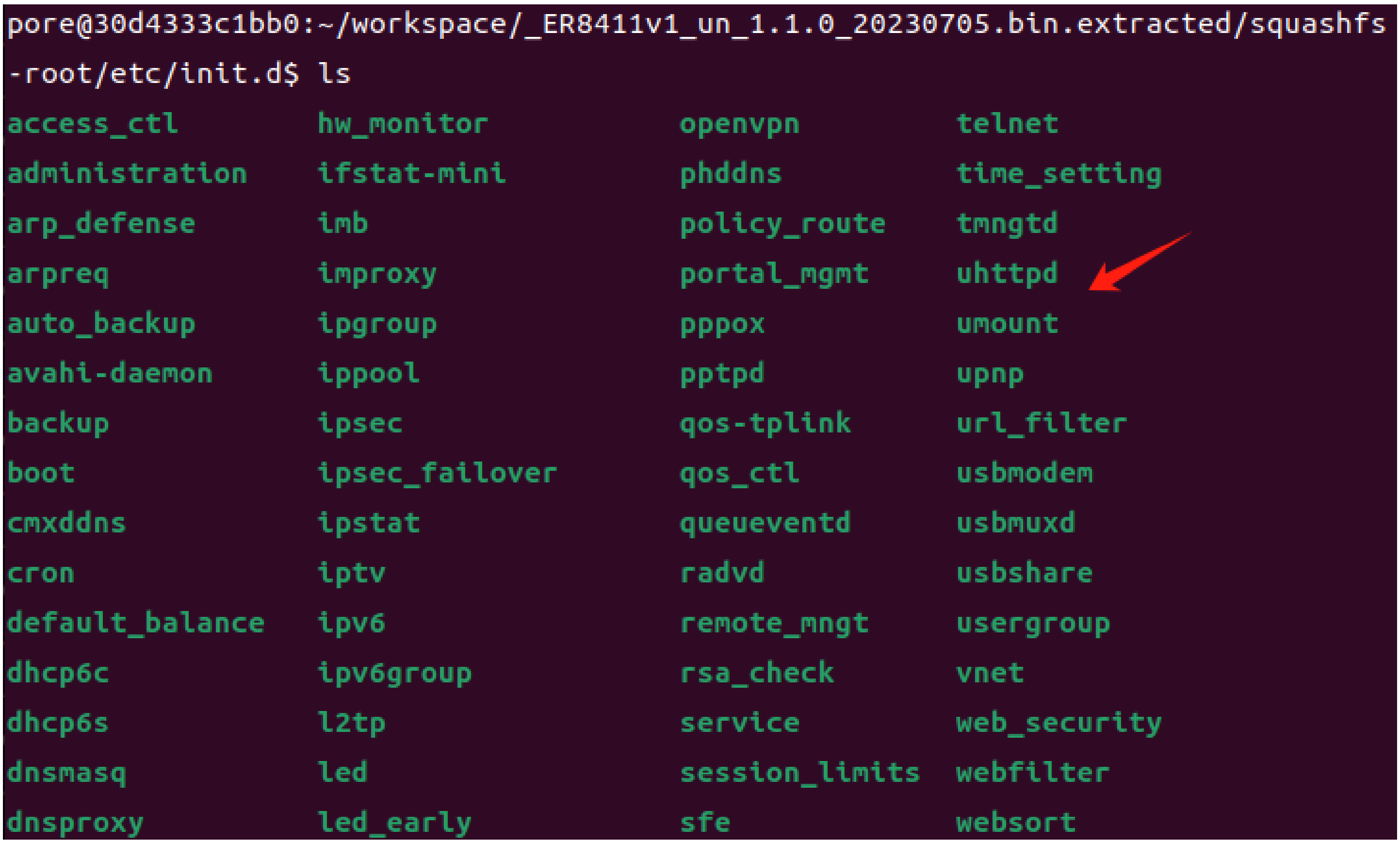


```
pore@30d4333c1bb0:~/workspace$ md5sum kernel.bin
b9995e2de290344f28f9cfc5c16cdeb4  kernel.bin
```

任务二

1.固件中的应用启动脚本在什么文件夹下，简要说明固件的应用启动流程

（1）固件中的应用启动脚本在/etc/init.d下，打开可以看到有uhttpd



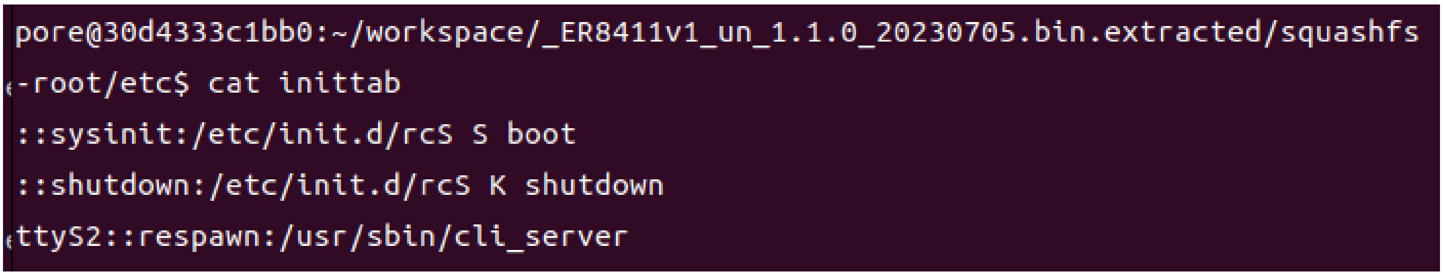
```
pore@30d4333c1bb0:~/workspace/_ER8411v1_un_1.1.0_20230705.bin.extracted/squashfs
-root/etc/init.d$ ls
access_ctl        hw_monitor       openvpn         telnet
administration    ifstat-mini      phddns          time_setting
arp_defense       imb              policy_route    tmngtd
arpreq            improxy          portal_mgmt     uhttpd
auto_backup       ipgroup          pppox           umount
avahi-daemon      ippool           pptpd           upnp
backup            ipsec            qos-tplink      url_filter
boot              ipsec_failover   qos_ctl         usbmodem
cmxddns           ipstat           queueventd      usbmuxd
cron              iptv             radvd           usbshare
default_balance   ipv6             remote_mngt     usergroup
dhcp6c            ipv6group        rsa_check       vnet
dhcp6s            l2tp             service         web_security
dnsmasq           led              session_limits  webfilter
dnsproxy          led_early        sfe             websort
```

（2）固件应用启动流程：

1. Bootloader阶段：
    - 硬件初始化和自检。
    - Bootloader加载到内存中，并启动。
    - Bootloader引导内核。
2. 内核加载阶段：
    - 内核初始化：包括初始化设备、文件系统，建立进程管理等。
    - 内核读取启动命令行：内核获取启动参数，确定根文件系统等配置，其中包含了内核启动脚本位置
3. 启动init进程：
    - nit进程读取并解析/etc/inittab文件，确定系统的运行级别（运行级别1到6）和需要启动的服务与守护进程。
    - 执行与运行级别相关的初始化脚本（例如/etc/rc.d/rc.sysinit）。
4. 目标应用启动阶段：
    - 目标应用启动脚本设置目标进程命令行参数、环境变量并启动应用：目标应用启动脚本配置应用程序的运行参数和环境变量，最终启动应用程序进程。

2.目标固件http服务对应的二进制文件路径是什么，你是怎样找到的

目标固件http服务对应的二进制文件路径是/usr/sbin/uhttpd，可以看到有uhttpd



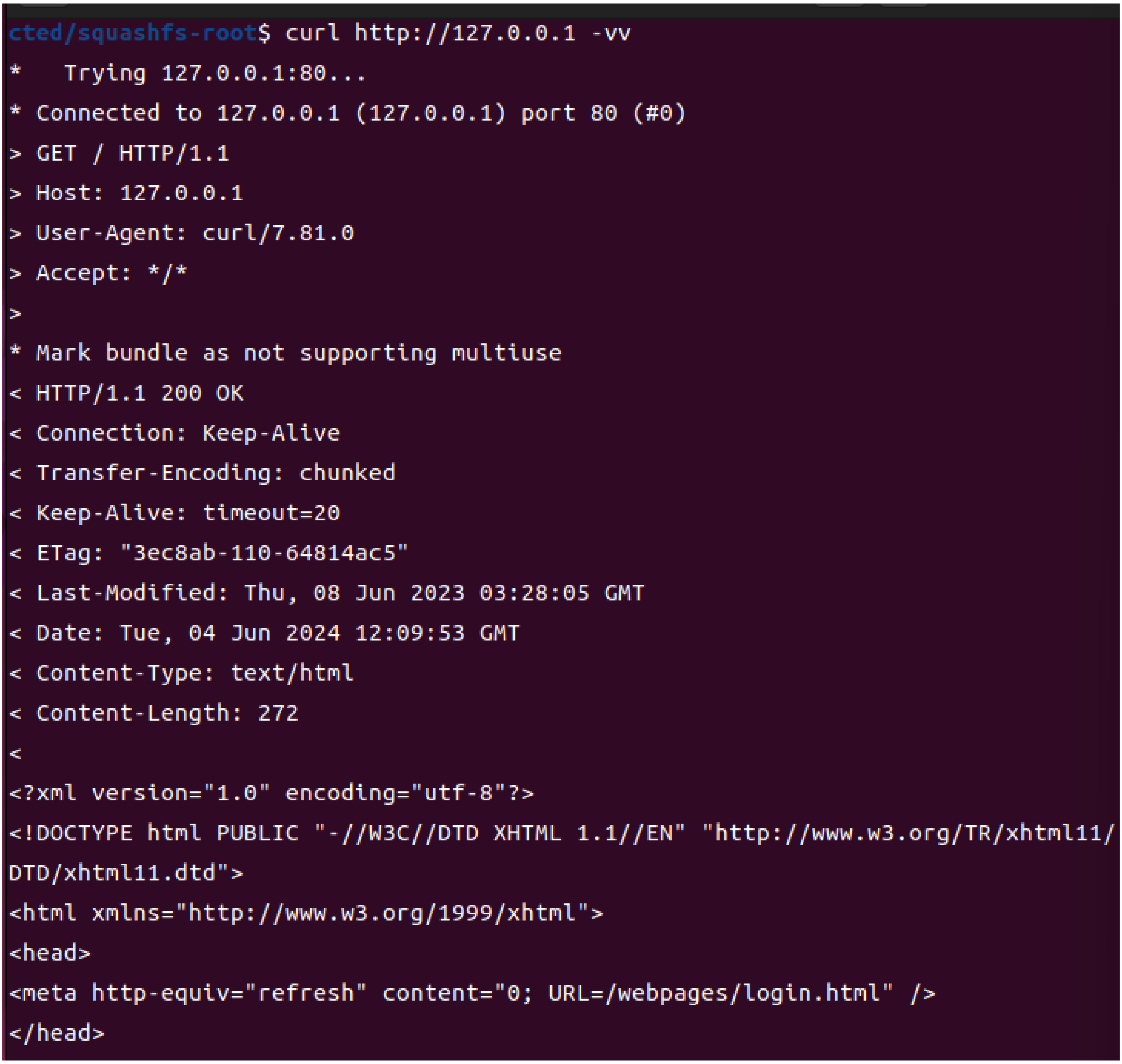
```
pore@30d4333c1bb0:~/workspace/_ER8411v1_un_1.1.0_20230705.bin.extracted/squashfs
-root/etc$ cat inittab
::sysinit:/etc/init.d/rcS S boot
::shutdown:/etc/init.d/rcS K shutdown
ttyS2::respawn:/usr/sbin/cli_server
```

可以看到启动后会执行 `/usr/sbin/cli_server` 程序，所以可以推测http服务在这个目录下

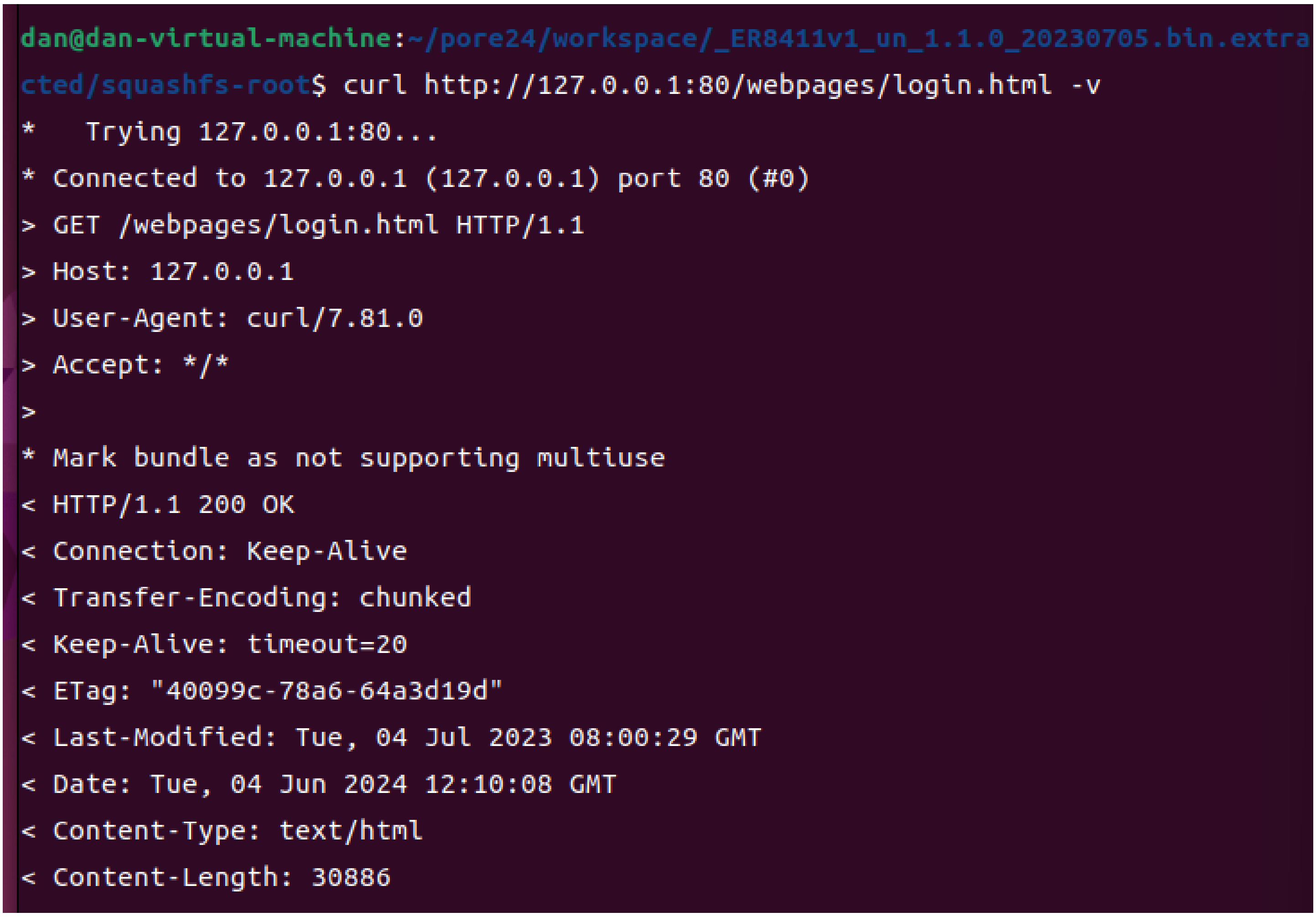
任务三

输入指令sudo chroot . ./qemu-aarch64-static -L . ./usr/sbin/uhttpd -p 80 -h /www

然后另开一个终端



```
cted/squashfs-root$ curl http://127.0.0.1 -vv
*   Trying 127.0.0.1:80...
* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> Host: 127.0.0.1
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Connection: Keep-Alive
< Transfer-Encoding: chunked
< Keep-Alive: timeout=20
< ETag: "3ec8ab-110-64814ac5"
< Last-Modified: Thu, 08 Jun 2023 03:28:05 GMT
< Date: Tue, 04 Jun 2024 12:09:53 GMT
< Content-Type: text/html
< Content-Length: 272
<
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/
DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="refresh" content="0; URL=/webpages/login.html" />
</head>
```



```
dan@dan-virtual-machine:~/pore24/workspace/_ER8411v1_un_1.1.0_20230705.bin.extra
cted/squashfs-root$ curl http://127.0.0.1:80/webpages/login.html -v
*   Trying 127.0.0.1:80...
* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)
> GET /webpages/login.html HTTP/1.1
> Host: 127.0.0.1
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Connection: Keep-Alive
< Transfer-Encoding: chunked
< Keep-Alive: timeout=20
< ETag: "40099c-78a6-64a3d19d"
< Last-Modified: Tue, 04 Jul 2023 08:00:29 GMT
< Date: Tue, 04 Jun 2024 12:10:08 GMT
< Content-Type: text/html
< Content-Length: 30886
```