OS Labs 使用指南

一、搭建开发环境

建议直接使用 UniProton 提供的 docker 镜像环境，可以避免后续使用 CMake 编译代码和对镜像做烧写前的签名校验时不必要的麻烦。

有关 docker 的安装请自行搜索相关资料。

使用如下命令拉去 docker 镜像:

```
docker pull swr.cn-north-4.myhuaweicloud.com/openeuler-embedded/uniproton:v004
```

执行完成之后，创建容器并进入（默认挂载当前执行命令的目录为容器内的/home/uniproton目录）：

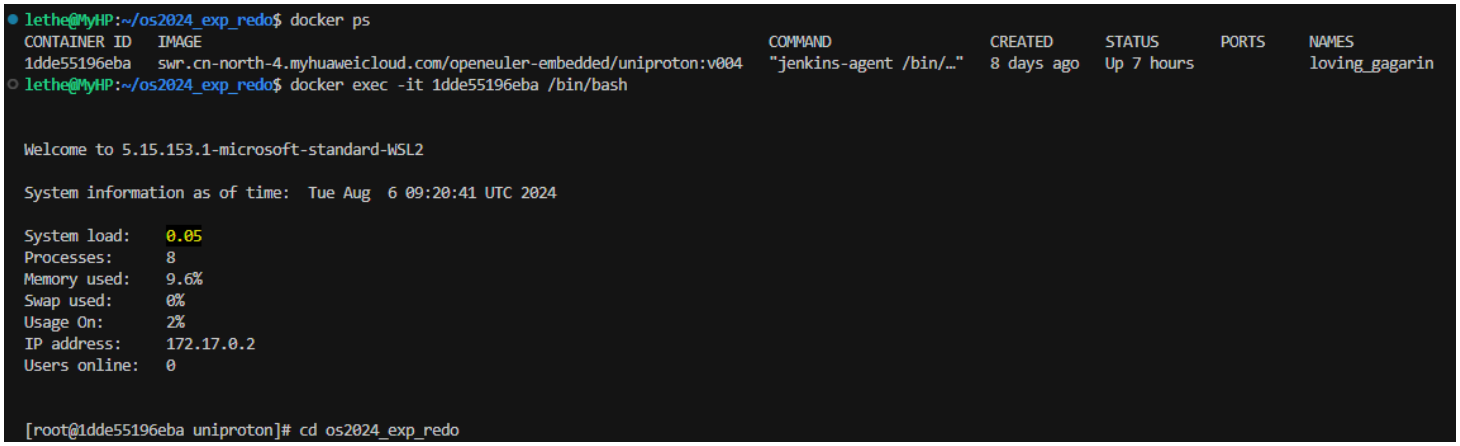
```
docker run -it -v $(pwd):/home/uniproton swr.cn-north-4.myhuaweicloud.com/openeuler-embedded/uni
```

至此你已经搭建好所需要的开发环境，并且与 CMakeLists.txt 中配置的工具链相适配。

之后每次进入 docker 环境无需重复创建容器的命令(否则会如字面意思一般创建很多相同的容器)，而是运行对应的容器，并使用如下命令:

```
> docker ps
CONTAINER ID    IMAGE    OMMAND    CREATED    STATUS    PORTS    NAMES
Your_ID    .........

> docker exec -it Your_ID /bin/bash
```



二、编译、签名及烧录镜像

目前只支持在 X86 架构的 Windows 系统上操作。

1. 编译

以在 `lab1/` 目录下为例，只需执行如下命令：

```
sh ./makeMiniEuler.sh
```

所生成的镜像为 `lab1/build/miniEuler.bin` 和 `lab1/build/miniEuler.elf` ，后续签名和烧写我们使用 `.bin` 文件

2. 签名

将 `miniEuler.bin` 文件复制到 `KunPengDevBoard-SignTool/output/Image` 文件（*NB: `miniEuler.bin` 和 `Image` 是相同的文件，只是文件名不一样*）

进入 `KunPengDevBoard-SignTool/` 目录下执行以下命令完成签名(在 `docker` 镜像中执行):

```
sh ./sign_image.sh
```

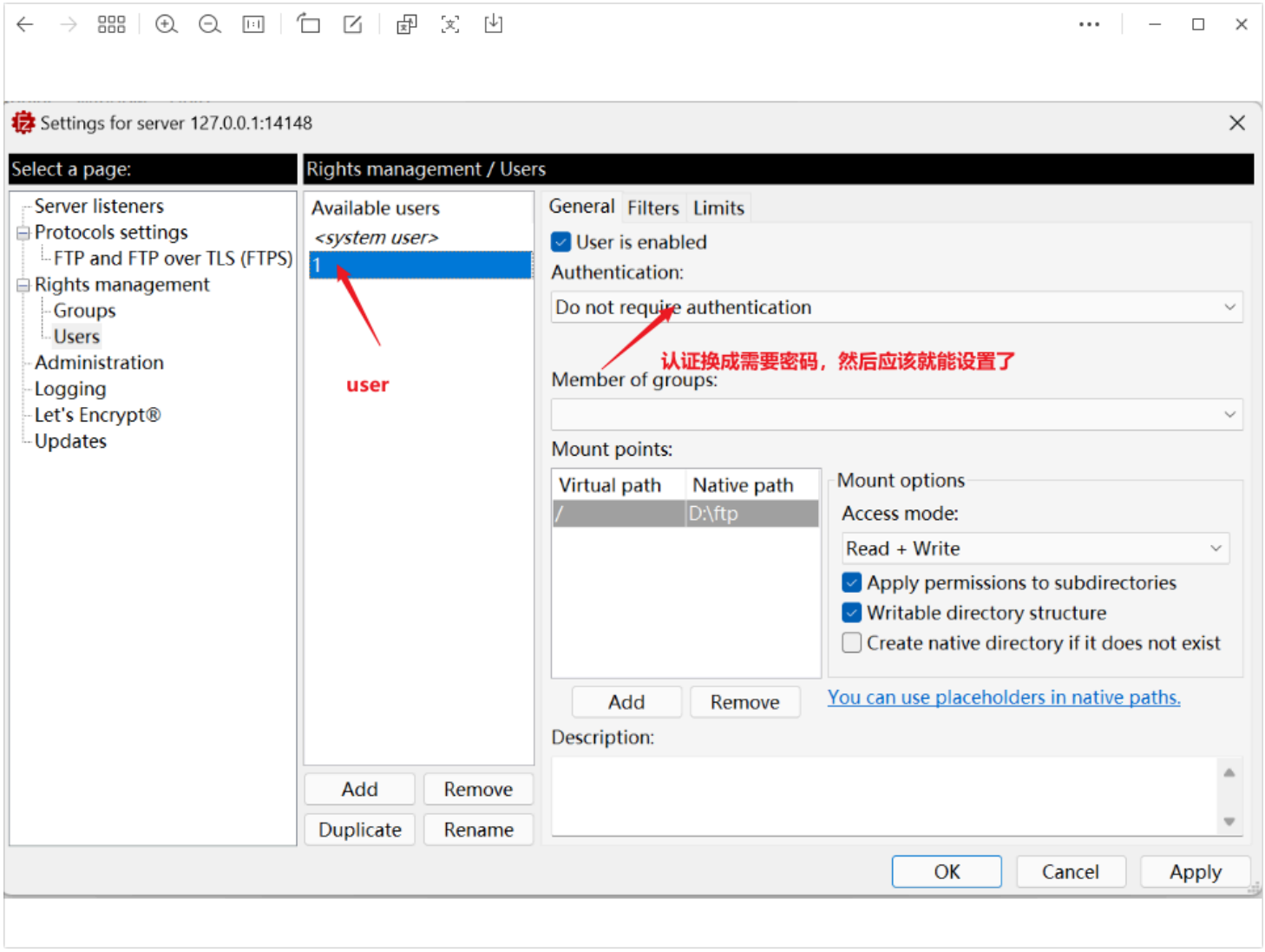
签名后的文件依旧是 `KunPengDevBoard-SignTool/output/Image` 文件

3. 烧录

a. 烧录环境准备

需要准备 `IPOP4.1.exe` 和 `FTP Server` 。关于 `FTP Server` 推荐使用 FileZilla Server 搭建。
需要注意以下几点：

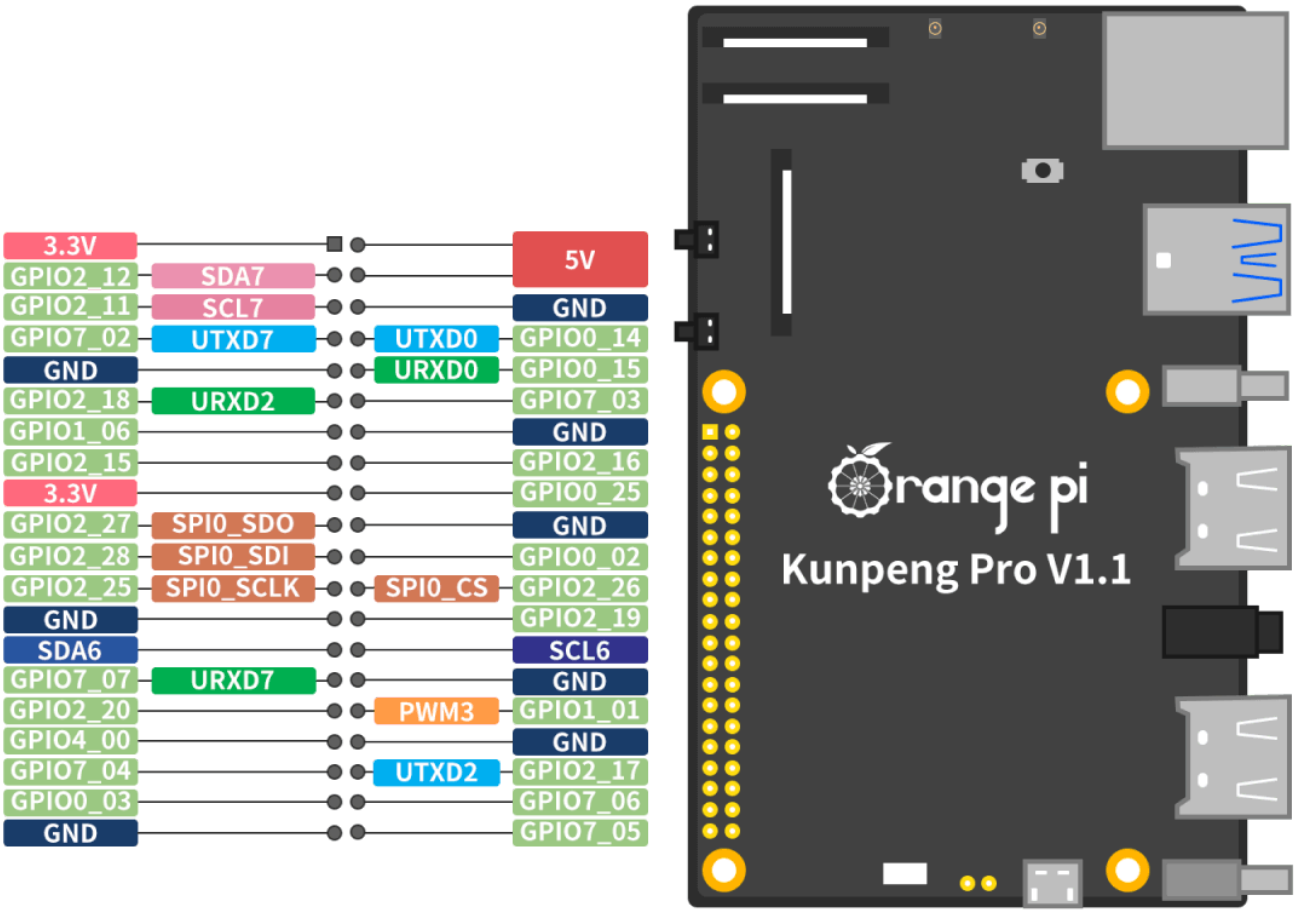
- 确保 `FTP Server` 开启了用户 `1`
- 确保 `FTP` 通信没有被 Windows 防火墙拦截（简单有效但不太安全的解决方案：关闭所有防火墙）
- 关于 `Authentication` 可以直接使用默认设置 `Do not require authentication`



至此烧录环境搭建成功。

b. 硬件连线

需要连接网线、MicroUSB 调试线和电源，lab10 需要单独的 USB 转 TTL 模块来与 UART2 进行通信，引脚图如图所示：

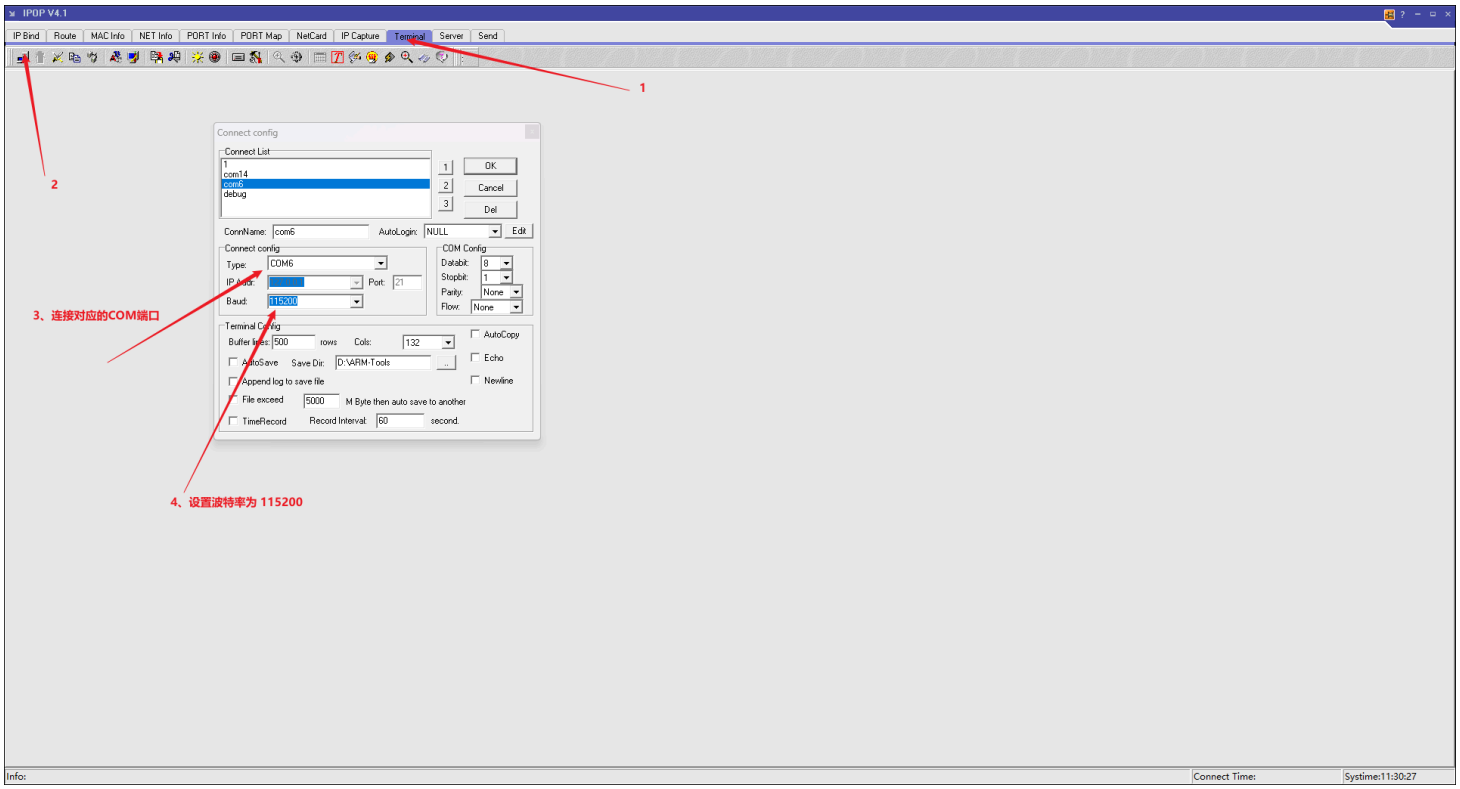


连接完网线之后将以太网配置成如下静态IP：

IP 分配:	手动
IPv4 地址:	192.168.2.11
IPv4 掩码:	255.255.255.0
IPv4 网关:	192.168.2.1

c. 烧录

打开 `IPOP4.1.exe` 连接 `COM` 端口：



给开发板上电，并在 IPOP 中不断按 delete 建，直至进入BIOS，如图为成功界面：

```
current keypoint[71], last keypoint[0].
macmode:0x1, ulPort:0x0, ulPhyID:0x1CC910, ulMacSpeed:0x2!
The default boot selection will start in    1 secondsDisable Wdg
Embedded Boot Loader (EBLTHE PROGRAM IS DISTRIBUTED UNDER THE BSD LICENSE ON AN 'AS IS' BASIS,
WITHOUT WARRANTIES OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.
Please send feedback to edk2-devel@lists.sourceforge.net
Ebl >
```

依次输入如下命令:

```
ifconfig -s eth0 192.168.2.50 255.255.255.0 192.168.2.1 # 设置开发板的IP地址
emmc 2
provision 192.168.2.11 -u 1 -p 1 -f lab10 -a 0x28FFDF00 # 通过 FTP 下载镜像文件，lab10记得修改为对
emmc 11 0x28FFDF00 0x1000000 0x2700000 0x0
```

结果如图所示：

```
current keypoint[71], last keypoint[0].
macmode:0x1, ulPort:0x0, ulPhyID:0x1CC910, ulMacSpeed:0x2!
The default boot selection will start in   1 secondsDisable Wdg
Embedded Boot Loader (EBLTHE PROGRAM IS DISTRIBUTED UNDER THE BSD LICENSE ON AN 'AS IS' BASIS,
WITHOUT WARRANTIES OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.
Please send feedback to edk2-devel@lists.sourceforge.net
Ebl >ifconfig -s eth0 192.168.2.50 255.255.255.0 192.168.2.1
Ebl >emmc 2
emmc init
Ebl >provision 192.168.2.11 -u 1 -p 1 -f lab10 -a 0x28FFDF00
Connected to the ftp server.
Download file lab10 from server.
[FtpGet]:[931L] Success
[FtpGet]:[934L] Success
[FtpGet]:[939L] Success
[FtpGet]:[941L] Success
[FtpGet]:[944L] Success
[FtpGet]:[951L] Success
[FtpGet]:[957L] Success
[FtpGet]:[967L] Success
[FtpGet]:[969L] Success
lab10: 173060 bytes.
[FtpGet]:[978L] Success
[FtpGet]:[980L] Success
..

[FtpGet]:[1023L] CurOffset=0, FileBuffer=28FFDF00
mdfile \lab10 0x28FFDF00 0x2A404
[FtpGet]:[1034L] Success
[FtpGet]:[1043L] Success
[FtpGet]:[1045L] Success
Download file lab10 succ!
Ebl >emmc 11 0x28FFDF00 0x1000000 0x2700000 0x0
Ebl >
```

重新上电即可运行烧录的镜像。

```
current keypoint[99], last keypoint[0].
 _ __ ___ (_)_ __ (_) ____|   _| | ___ _ __  | |__  _   _  | | | | \ | | | | | ___ _ __
| '_ ` _ \| | '_ \| |  _|| | | | |/ _ \ '__| | '_ \| | | | | |_| |  \| | | | |/ _ \ '__|
| | | | | | | | | | | |__| |_| | |  __/ |    | |_) | |_| | |  _  | |\  | |_| |  __/ |
|_| |_| |_|_|_| |_|_|_____\__,_|_|\___|_|    |_.__/ \__, | |_| |_|_| \_|\___/ \___|_|
                                                      |___/
IDLE
```

三、各个实验结果

各个实验的镜像在 `release/` 目录下。

lab 1

```
current keypoint[99], last keypoint[71].
AArch64 Bare Metal
```

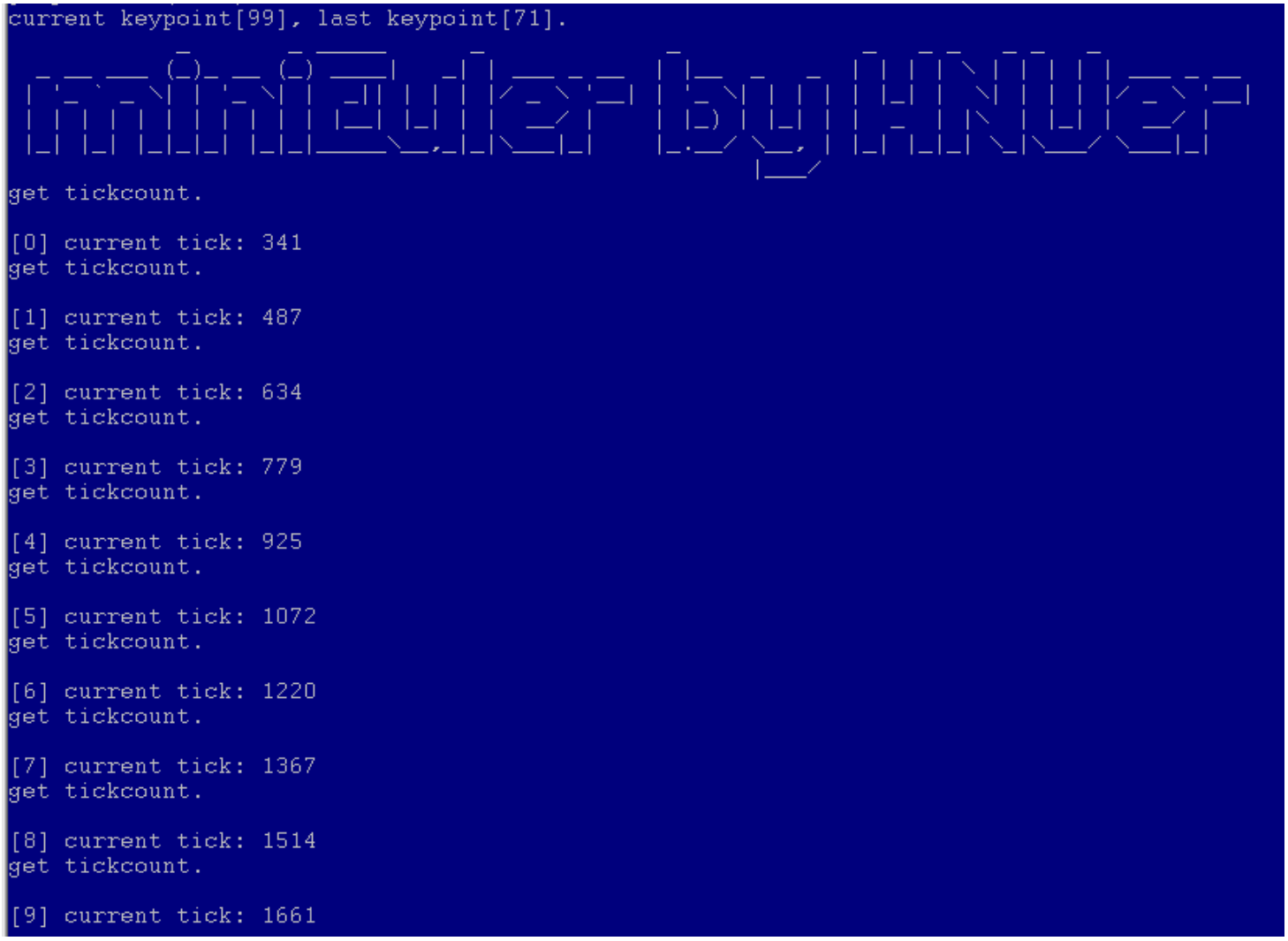
lab 2


current keypoint[99], last keypoint[71].
[uniproton]!
Test PRT_Printf int format 10

lab 4


current keypoint[99], last keypoint[71].
Catch a SVC call.
syscall number: 1, param 0: 0x290049ea
Hello, my first system call!

lab 5


current keypoint[99], last keypoint[71].
get tickcount.
[0] current tick: 341
get tickcount.
[1] current tick: 487
get tickcount.
[2] current tick: 634
get tickcount.
[3] current tick: 779
get tickcount.
[4] current tick: 925
get tickcount.
[5] current tick: 1072
get tickcount.
[6] current tick: 1220
get tickcount.
[7] current tick: 1367
get tickcount.
[8] current tick: 1514
get tickcount.
[9] current tick: 1661

lab 6


current keypoint[99], last keypoint[71].
Test1Task
task1run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
exit
task2runrun
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
exit
IDLE

lab 7


current keypoint[99], last keypoint[71].
Test1Task
task2runrun
task1run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
exit
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
exit
IDLE

lab 8

```
current keypoint[99], last keypoint[71].
 _ __ ___ (_)_ __ (_) ____|   _| | ___ _ __  | |__  _   _  | | | | \ | | | | | ___ _ __
| '_ ` _ \| | '_ \| |  _|| | | | |/ _ \ '__| | '_ \| | | | | |_| |  \| | | | |/ _ \ '__|
| | | | | | | | | | | |__| |_| | |  __/ |    | |_) | |_| | |  _  | |\  | |_| |  __/ |
|_| |_| |_|_|_| |_|_|_____\__,_|_|\___|_|    |_.__/ \__, | |_| |_|_| \_|\___/ \___|_|
                                                       |___/
Test1Task
task2runrun
task1run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
task 2 run ...
exit
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
task 1 run ...
exit
IDLE
```

lab 10

```
current keypoint[99], last keypoint[0].
 _ __ ___ (_)_ __ (_) ____|   _| | ___ _ __  | |__  _   _  | | | | \ | | | | | ___ _ __
| '_ ` _ \| | '_ \| |  _|| | | | |/ _ \ '__| | '_ \| | | | | |_| |  \| | | | |/ _ \ '__|
| | | | | | | | | | | |__| |_| | |  __/ |    | |_) | |_| | |  _  | |\  | |_| |  __/ |
|_| |_| |_|_|_| |_|_|_____\__,_|_|\___|_|    |_.__/ \__, | |_| |_|_| \_|\___/ \___|_|
                                                       |___/
IDLE
```

Shell 界面：

```
miniEuler # top
PID             Priority        Stack Size
1               63              4096
0               9               4096
Total 2 tasks
miniEuler # tick
Current Tick: 33270
miniEuler #
```