



湖南大学
HUNAN UNIVERSITY

课程实验报告

课 程 名 称: 操作系统
实验项目名称: 鲲鹏云 ECS 的构建及内核编译
专 业 班 级: 软件 2203 班
姓 名: 丁海桐
学 号: 202226010304
指 导 教 师: 周四望
完 成 时 间: 2024 年 4 月 21 日

信息科学与工程学院

实验题目：鲲鹏云 ECS 的构建及内核编译

- 实验目的：
- 学习掌握如何安装构建 ECS
 - 学习掌握如何编译操作系统内核
 - 了解内核模块编程。

实验环境：
华为云、win11

实验内容及操作步骤：

1、创建 VPC

根据表格中参数创建虚拟私有云

参数	配置
区域	华北-北京四
名称	vpc-test
网段	192.168.1.0/24
企业项目	default
默认子网可用区	可用区1
默认子网名称	subnet-test
子网网段	如192.168.1.0/24



在“访问控制中”创建安全组



2、购买 ECS

根据表格中参数配置弹性云服务器

参数	配置
计费模式	按需计费
区域	华北-北京四
可用区	可用区1
CPU架构	鲲鹏计算
规格	鲲鹏通用计算增强型 kc1.xlarge.2 4vCPUs 8GB
镜像	公共镜像 openEuler openEuler 20.03 64bit with ARM(40GB)
系统盘	通用型SSD 40GB

参数	配置
网络	vpc-test subnet-test 自动分配IP地址
安全组	sg-test
弹性公网IP	现在购买
线路	全动态BGP
公网带宽	按流量计费
带宽大小	5Mbit/s

参数	配置
云服务器名称	openEuler (输入符合规则名称)
登录凭证	密码
密码	请输入8位以上包含大小写字母、数字和特殊字符的密码，如

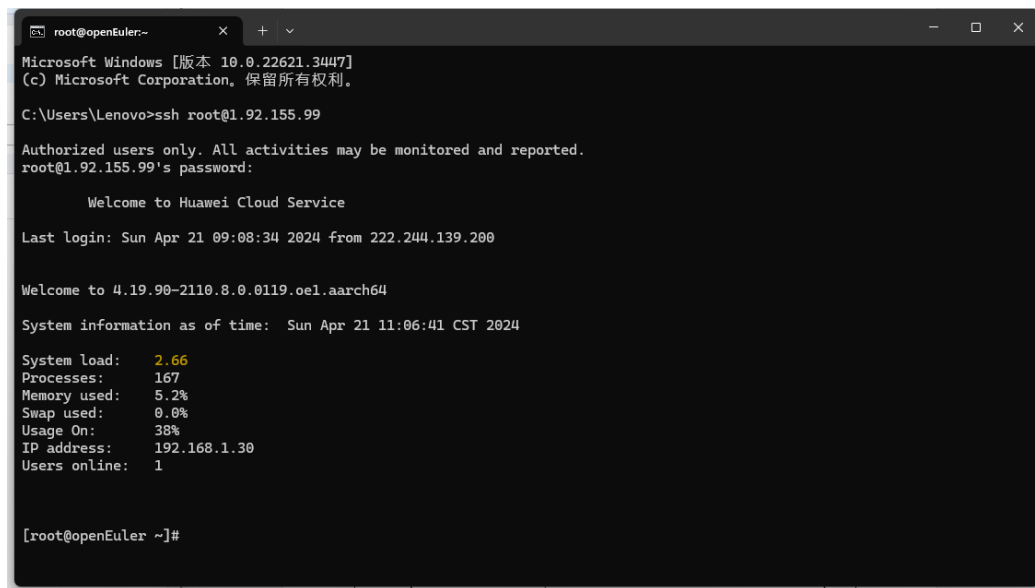


3、通过 SSH 登录系统

查看弹性公网的 IP 地址



在宿主机上用 SSH 登录



4、openEuler 内核编译与安装

1) 安装工具，构建开发环境：

```
[root@openEuler ~]# yum group install -y "Development Tools" --nogpgcheck
```

```
[root@openEuler ~]# yum install -y bc --nogpgcheck
```

```
[root@openEuler ~]# yum install -y openssl-devel --nogpgcheck
```

2) 备份 boot 目录以防后续步骤更新内核失败

```
[root@openEuler ~]# tar czvf boot.origin.tgz /boot/
```

保存当前内核版本信息

```
[root@openEuler ~]# uname -r > uname_r.log
```

注意上述命令中的“-”为英文字符。

3) 获得源码并解压

```
[root@openEuler~]# wget
```

```
https://gitee.com/openeuler/kernel/repository/archive/openEuler-20.03-LTS-SP3.zip
```

```
[root@openEuler ~]# unzip openEuler-20.03-LTS-SP3.zip
```

4) 编译内核

```
[root@openEuler ~]# mv kernel-openEuler-20.03-LTS-SP3/ kernel
```

```
[root@openEuler ~]# cd kernel/
```

```
[root@openEuler kernel]# make openeuler_defconfig
```

在这里，我们按源代码文件 kernel/arch/arm64/configs/openeuler_defconfig 的配置配

置内核。

```
[root@openEuler kernel]# make help | grep Image
```

* Image.gz - Compressed kernel image (arch/arm64/boot/Image.gz)

Image - Uncompressed kernel image (arch/arm64/boot/Image)

这一步查看了可编译的 Image。

```
[root@openEuler kernel]# make -j4 Image modules dtbs
```

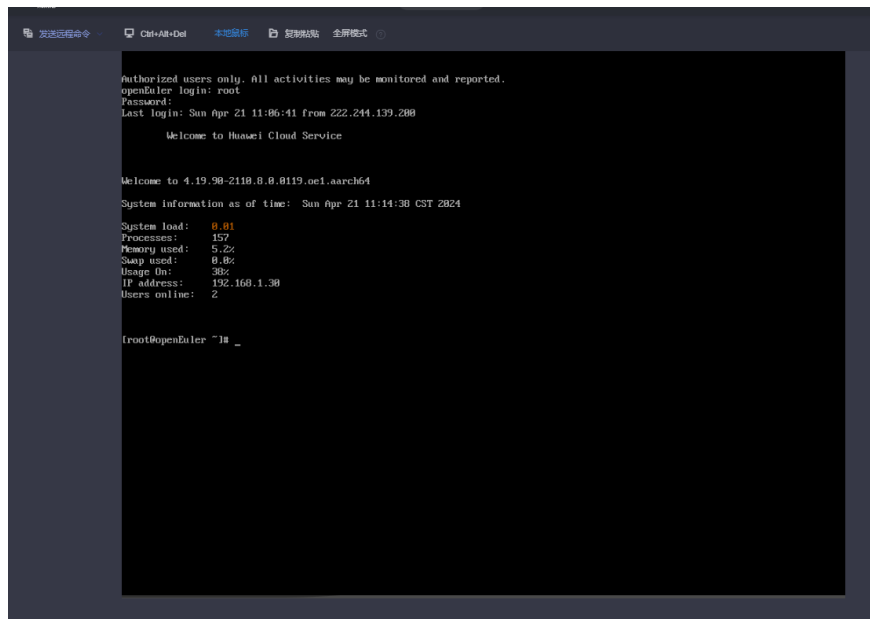
这一步是编译内核的 Image、modules 和 dtbs。

5) 安装内核

```
[root@openEuler kernel]# make modules_install
```

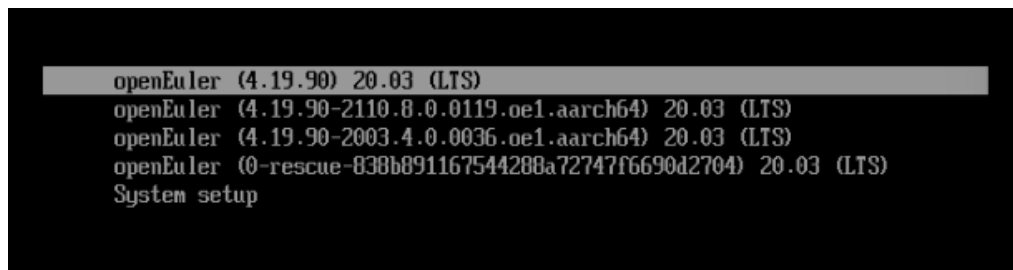
```
[root@openEuler kernel]# make install
```

6) 以 VNC 登录 ECS



7) 登录验证

在 VNC 窗口中选择以新编译出来的内核启动系统：



```
Authorized users only. All activities may be monitored and reported.  
openEuler login: root  
Password:  
Last login: Sun Apr 21 11:17:43 on tty1
```

```
      Welcome to Huawei Cloud Service
```

```
Welcome to 4.19.90
```

```
System information as of time:  Sun Apr 21 11:18:50 CST 2024
```

```
System load:    3.32  
Processes:      163  
Memory used:    5.2%  
Swap used:      0.0%  
Usage On:       38%  
IP address:     192.168.1.30  
Users online:   1
```

```
[root@openEuler ~]# uname -r  
4.19.90  
[root@openEuler ~]# _
```

5、Hello, world!

1) 编写 hello-world.c 文件和 Makefile 文件

```

[root@openEuler ~]# cd Lab-1/
[root@openEuler Lab-1]# cat hello_world.c
/* Hello from Kernel! */

#include <linux/module.h>

MODULE_LICENSE("GPL");

static char* guy = "Kernel";
module_param(guy, charp, 0644);
MODULE_PARM_DESC(guy, "char* param\n");

static int year = 2021;
module_param(year, int, 0644);
MODULE_PARM_DESC(year, "int param\n");

void hello_world(char* guy, int year)
{
    printk(KERN_ALERT "Hello, %s, %d!\n", guy, year);
}

int __init hello_init(void)
{
    printk(KERN_ALERT "Init module.\n");
    hello_world(guy, year);

    return 0;
}

void __exit hello_exit(void)
{
    printk(KERN_ALERT "Exit module.\n");
}

module_init(hello_init);
module_exit(hello_exit);

```

```

[root@openEuler Lab-1]# cat Makefile
# Build module hello_world

ifneq ($(KERNELRELEASE),)
    obj-m := hello_world.o
else
    KERNELDIR ?=/root/kernel
    PWD := $(shell pwd)
default:
    $(MAKE) -C $(KERNELDIR) M=$(PWD) modules
endif

.PHONY:clean
clean:
    -rm *.mod.c *.o *.order *.symvers *.ko

[root@openEuler Lab-1]# |

```

2) 编译源文件

```
[root@openEuler Lab-1]# make
make -C /root/kernel M=/root/Lab-1 modules
make[1]: Entering directory '/root/kernel'
Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/root/kernel'
```

- 3) 加载编译完成的内核模块，并查看加载结果。

```
[root@openEuler Lab-1]# insmod hello_world.ko guy="Dinu" year=2013
[root@openEuler Lab-1]# lsmod | grep hello_world
hello_world      262144  0
```

- 4) 卸载内核模块，并查看结果。

```
[root@openEuler Lab-1]# rmmod hello_world
[root@openEuler Lab-1]# dmesg | tail -n5
[ 213.234022] hello_world: loading out-of-tree module taints kernel.
[ 213.234500] hello_world: module verification failed: signature and/or required key missing - tainting kernel
[ 213.235595] Init module.
[ 213.235767] Hello, Dinu, 2013!
[ 225.152570] Exit module.
```

实验结果及分析：

1、实验结果分析

- 1) [root@openEuler hello-world]# make: 这个命令执行了 Makefile 中的默认目标，通常是用来编译源文件生成目标文件或可执行文件。在这个例子中，它编译了 hello_world.c 源文件生成 hello_world.ko 内核模块文件。
- 2) [root@openEuler hello-world]# insmod hello_world.ko guy="Dinu" year=2013: 这个命令使用 insmod 工具加载了编译完成的内核模块 hello_world.ko 到内核中，并传递了两个参数 guy 和 year。这些参数将会被模块的初始化函数使用。
- 3) [root@openEuler hello-world]# lsmod | grep hello_world: 这个命令使用 lsmod 工具查看已加载的内核模块，并使用 grep 进行过滤，以查找包含 hello_world 关键字的模块。结果显示了 hello_world 模块已成功加载到内核中。
- 4) [root@openEuler hello-world]# rmmod hello_world: 这个命令使用 rmmod 工具卸载了名为 hello_world 的内核模块。
- 5) [root@openEuler hello-world]# dmesg | tail -n5: 这个命令使用 dmesg 工具查看了内核消息日志的最后五行，即最近的五条内核消息。

2、在 Linux 内核模块中，代码的执行顺序如下：

- 1) 加载模块：当使用 insmod 或 modprobe 命令加载内核模块时，内核首先会调用模块的初始化函数 hello_init。
- 2) 初始化函数：初始化函数 hello_init 被调用，它会执行模块的初始化操作，包括打印初始化消息和调用 hello_world 函数打印欢迎消息。
- 3) 退出函数：当使用 rmmod 命令卸载内核模块时，内核会调用模块的退出

函数 `hello_exit`。

- 4) 退出函数执行：退出函数 `hello_exit` 被调用，它会执行模块的清理操作，包括打印退出消息。

收获与体会：

第一次尝试下载系统内核进行编译配置安装，接触了新的操作系统 `openEuler`，上手模块编程。对于操作系统的系统级函数调用理解更加深刻。

实
验
成
绩

