

南京理工大学计算机科学与技术学院

软件课程设计Ⅲ实验报告

组员 1

组员 2

组员 3

目录

1 构建实验环境.....	3
1.1 实验介绍	3
1.2 构建实验环境.....	3
2 openEuler 系统环境实验.....	3
2.1 实验介绍	3
2.2 系统编程环境实验.....	4
3 内核编程实验.....	5
3.1 实验介绍	5
3.1.1 关于本实验	5
3.1.2 实验目的.....	5
3.2 内核编程实验.....	5
3.2.1 内核的编译与安装	5
3.2.2 Hello, world!	10
3.2.3 char 数据类型	11
4 实验遇到的问题	12
4.1 问题一	12
4.1.1 问题描述.....	12
4.1.2 解决方法.....	12
5 小组分工.....	12

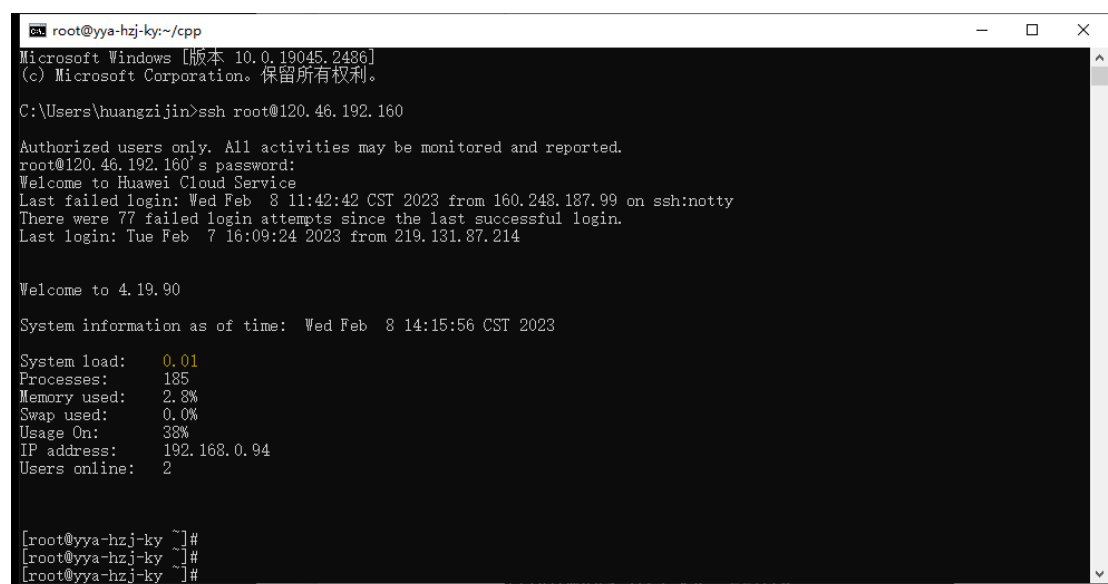
1 构建实验环境

1.1 实验介绍

openEuler 是一款通用服务器操作系统，支持 x86 和 ARM 等多种处理器架构，本实验旨在熟悉基于 Kunpeng 架构弹性云服务器 ECS 上 openEuler 操作系统基本系统环境、学习轻量级容器 iSulad 的基本用法以及了解智能调优引擎 A-Tune 的调优过程。

1.2 构建实验环境

我们购买了华为云服务器，搭载的是鲲鹏芯片，操作系统是 openEuler，ip 地址是 120.46.192.160，使用 ssh 命令登录服务器。

A terminal window titled 'root@yaya-hzj-ky:~/cpp' showing an SSH session. The user 'root' connects to IP '120.46.192.160'. The terminal displays a welcome message from Huawei Cloud Service, login statistics, and system information. The system information includes system load (0.01), processes (185), memory used (2.8%), swap used (0.0%), usage on disk (33%), IP address (192.168.0.94), and 2 users online. The prompt changes from root@120.46.192.160 to [root@yaya-hzj-ky ~]#.

```
root@yaya-hzj-ky:~/cpp
Microsoft Windows [版本 10.0.19045.2486]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\huangzijin>ssh root@120.46.192.160

Authorized users only. All activities may be monitored and reported.
root@120.46.192.160's password:
Welcome to Huawei Cloud Service
Last failed login: Wed Feb  8 11:42:42 CST 2023 from 160.248.187.99 on ssh:notty
There were 77 failed login attempts since the last successful login.
Last login: Tue Feb  7 16:09:24 2023 from 219.131.87.214

Welcome to 4.19.90

System information as of time: Wed Feb  8 14:15:56 CST 2023

System load:   0.01
Processes:    185
Memory used:  2.8%
Swap used:    0.0%
Usage On:     33%
IP address:   192.168.0.94
Users online: 2

[root@yaya-hzj-ky ~]#
[root@yaya-hzj-ky ~]#
[root@yaya-hzj-ky ~]#
```

图表 1ssh 命令登录服务器

2 openEuler 系统环境实验

2.1 实验介绍

本实验通过运行 shell 命令查看系统信息以达到了了解 openEuler 操作系统的目的。

2.2 系统编程环境实验

使用 `uname -m` 可以查看到，使用的是 aarch64 架构，如下图。

```
[root@yya-hzj-ky ~]# cd .ssh
[root@yya-hzj-ky .ssh]# dir
authorized_keys
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]# uname -m
aarch64
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
```

使用 `uname -r` 可以查看到版本号，如下图。

```
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]# uname -r
4.19.90-2110.8.0.0119.oel.aarch64
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
[root@yya-hzj-ky .ssh]#
```

使用 `lscpu` 查看 cpu 的参数：8 个 cpu，编号从 0 到 7，有一个插槽，一个插槽 8 个核心，产商是 HiSilicon 半导体，鲲鹏 920 的芯片。如下图。

```

< \
< /
[root@yya-hzj-ky /]#
[root@yya-hzj-ky /]# lscpu
Architecture:          aarch64
CPU op-mode(s):        64-bit
Byte Order:             Little Endian
CPU(s):                 8
On-line CPU(s) list:    0-7
Thread(s) per core:     1
Core(s) per socket:     8
Socket(s):              1
NUMA node(s):          1
Vendor ID:              HiSilicon
Model:                  0
Model name:             Kunpeng-920
Stepping:               0x1
CPU max MHz:            2400.0000
CPU min MHz:            2400.0000
BogoMIPS:               200.00
L1d cache:              512 KiB
L1i cache:              512 KiB
L2 cache:               4 MiB
L3 cache:               32 MiB
NUMA node0 CPU(s):      0-7
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf:      Not affected
Vulnerability Mds:       Not affected
Vulnerability Meltdown:  Not affected
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1: Mitigation; __user pointer sanitization
Vulnerability Spectre v2: Not affected
Vulnerability Srbds:     Not affected
Vulnerability Tsx async abort: Not affected
Flags:                   fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm jscvt fc
                        ma dcpop asimddp asimdfhm
[root@yya-hzj-ky /]# cd
< /

```

3 内核编程实验

3.1 实验介绍

3.1.1 关于本实验

本实验在鲲鹏云 ECS 上编译、安装 openEuler 操作系统内核并编写一个简单的内核模块以验证安装是否成功。

3.1.2 实验目的

- 熟悉 openEuler 内核的编译与安装；
- 了解内核模块编程的过程。

3.2 内核编程实验

3.2.1 内核的编译与安装

安装环境

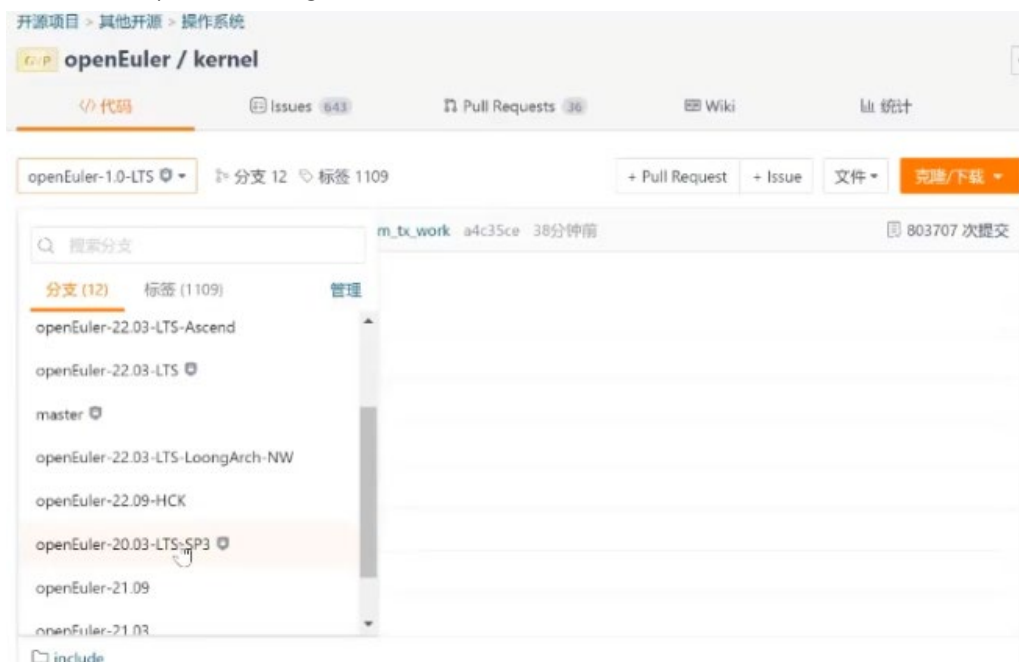
```
yum group install -y "Development Tools"
yum install -y bc
yum install -y openssl-devel --nogpgcheck
```

```
Total download size: 87 k
Installed size: 293 k
Downloading Packages:
bc-1.07.1-11.oe1.aarch64.rpm
-----
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : 
  Installing     : bc-1.07.1-11.oe1.aarch64
  Verifying      : bc-1.07.1-11.oe1.aarch64

Installed:
  bc-1.07.1-11.oe1.aarch64

Complete!
[root@yya-hzj-ky /]# yum install -y openssl-devel
```

获取内核（从 openEuler 的 gitee 中获取下载链接）



下载内核

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

```

[root@yya-hzj-ky ~]# wget https://gitee.com/openEuler/kernel/repository/archive/openEuler-20.03-LTS-SP3.zip
--2023-02-07 14:51:52-- https://gitee.com/openEuler/kernel/repository/archive/openEuler-20.03-LTS-SP3.zip
Resolving gitee.com (gitee.com)... 212.64.63.190, 212.64.63.215
Connecting to gitee.com (gitee.com)|212.64.63.190|:443... connected.
HTTP request sent, awaiting response... 302 Found
location: https://gitee.com/openEuler/kernel/repository/blazearchive/openEuler-20.03-LTS-SP3.zip?Expires=1675753913&Signature=jjj9thC5AAKzjcyi9
%2BKVMm%2Fto3b1OC7ZdcP81FGOL0s%3D [following]
--2023-02-07 14:51:53-- https://gitee.com/openEuler/kernel/repository/blazearchive/openEuler-20.03-LTS-SP3.zip?Expires=1675753913&Signature=jj
j9thC5AAKzjcyi9%2BKVMm%2Fto3b1OC7ZdcP81FGOL0s%3D
Reusing existing connection to gitee.com:443.
HTTP request sent, awaiting response... 200 OK
length: unspecified [application/zip]
Saving to: 'openEuler-20.03-LTS-SP3.zip'

openEuler-20.03-LTS-SP3.zip          [          <=>          ] 196.22M  8.71MB/s   in 29s

2023-02-07 14:52:23 (6.68 MB/s) - 'openEuler-20.03-LTS-SP3.zip' saved [205755341]

[root@yya-hzj-ky ~]#

```

解压

unzip Euler-20.03-LTS-SP3.zip

查看 image 类型，可以看到他可以压缩格式的

make help | grep Image

```

kernel-openEuler-20.03-LTS-SP3/tools/testing/selftests/powerpc/vphn/vphn.h
[root@yya-hzj-ky ~]# cd kernel-openEuler-20.03-LTS-SP3/
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]# make help | grep Image
* Image.gz      - Compressed kernel image (arch/arm64/boot/Image.gz)
  Image        - Uncompressed kernel image (arch/arm64/boot/Image)
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#

```

编译内核前要 make 一下配置，使用的是 menuconfig，产生了一个 config 文件

```

perl-Error-1:0.17028-1.oe1.noarch
perl-XML-Parser-1:2.44-4.oe1.aarch64
webkit2gtk3-jsc-2.22.2-8.oe1.aarch64
perl-Git-2.27.0-3.oe1.aarch64
webkit2gtk3-2.22.2-8.oe1.aarch64

Complete!
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]# make openeuler_defconfig
YACC      scripts/kconfig/zconf.tab.c
LEX       scripts/kconfig/zconf.lex.c
HOSTCC    scripts/kconfig/zconf.tab.o
HOSTLD    scripts/kconfig/conf
arch/arm64/configs/openeuler_defconfig:6007:warning: override: reassigning to symbol STAGING
#
# configuration written to .config
#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#

```

然后就 build 内核，以 8 个线程来创建内核

make -j16 Image

DEPMOD 4.19.90

[illegible]

创建内核模块

```
make -j16 modules
```

[illegible]

安装到 boot 目录下

Make install

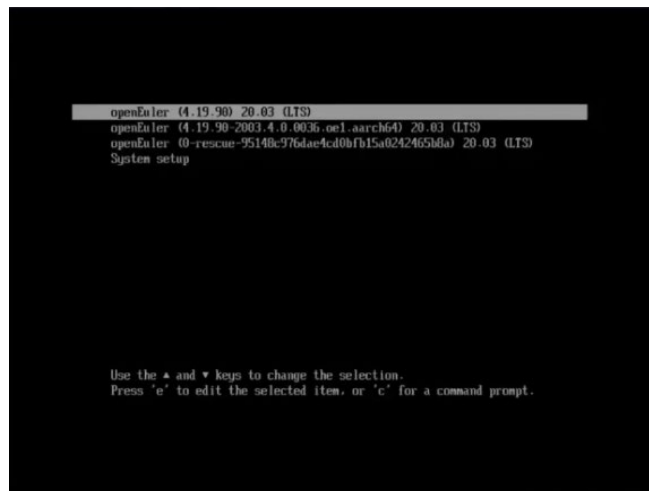

```

[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]# ls /boot
config-4.19.90-2003.4.0.0036.oe1.aarch64      initramfs-4.19.90-2110.8.0.0119.oe1.aarch64.img
config-4.19.90-2110.8.0.0119.oe1.aarch64      initramfs-4.19.90-2110.8.0.0119.oe1.aarch64kdump.img
dracut                                           loader
dtb-4.19.90-2003.4.0.0036.oe1.aarch64          symvers-4.19.90-2003.4.0.0036.oe1.aarch64.gz
dtb-4.19.90-2110.8.0.0119.oe1.aarch64          symvers-4.19.90-2110.8.0.0119.oe1.aarch64.gz
efi                                              System.map-4.19.90-2003.4.0.0036.oe1.aarch64
grub2                                           System.map-4.19.90-2110.8.0.0119.oe1.aarch64
initramfs-0-rescue-833b891167544288a72747f6690d2704.img vmlinuz-0-rescue-833b891167544288a72747f6690d2704
initramfs-4.19.90-2003.4.0.0036.oe1.aarch64.img vmlinuz-4.19.90-2003.4.0.0036.oe1.aarch64
initramfs-4.19.90-2003.4.0.0036.oe1.aarch64kdump.img vmlinuz-4.19.90-2110.8.0.0119.oe1.aarch64
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]# make install
/bin/sh ./arch/arm64/boot/install.sh 4.19.90 \
arch/arm64/boot/Image System.map "/boot"
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]# ls /boot
config-4.19.90-2003.4.0.0036.oe1.aarch64      loader
config-4.19.90-2110.8.0.0119.oe1.aarch64      symvers-4.19.90-2003.4.0.0036.oe1.aarch64.gz
dracut                                           symvers-4.19.90-2110.8.0.0119.oe1.aarch64.gz
dtb-4.19.90-2003.4.0.0036.oe1.aarch64          System.map
dtb-4.19.90-2110.8.0.0119.oe1.aarch64          System.map-4.19.90
efi                                              System.map-4.19.90-2003.4.0.0036.oe1.aarch64
grub2                                           System.map-4.19.90-2110.8.0.0119.oe1.aarch64
initramfs-0-rescue-833b891167544288a72747f6690d2704.img vmlinuz
initramfs-4.19.90-2003.4.0.0036.oe1.aarch64.img vmlinuz-0-rescue-833b891167544288a72747f6690d2704
initramfs-4.19.90-2003.4.0.0036.oe1.aarch64kdump.img vmlinuz-4.19.90
initramfs-4.19.90-2110.8.0.0119.oe1.aarch64.img vmlinuz-4.19.90-2003.4.0.0036.oe1.aarch64
initramfs-4.19.90-2110.8.0.0119.oe1.aarch64kdump.img vmlinuz-4.19.90-2110.8.0.0119.oe1.aarch64
initramfs-4.19.90.img
[root@yya-hzj-ky kernel-openEuler-20.03-LTS-SP3]#

```

可以看到有新的内核 vmlinuz-4.19.90 已经安装

Reboot 重启，以新的内核引导



可以看到，重启后内核变了

```
Last login: Tue Feb  7 15:51:52 2023 from 219.131.87.214

Welcome to 4.19.90

System information as of time: Tue Feb  7 15:54:43 CST 2023

System load:    2.13
Processes:      195
Memory used:    2.5%
Swap used:      0.0%
Usage On:       38%
IP address:     192.168.0.94
Users online:   1

[root@yya-hzj-ky ~]# ls
kernel-openEuler-20.03-LTS-SP3  openEuler-20.03-LTS-SP3.zip
[root@yya-hzj-ky ~]# uname -r
4.19.90
[root@yya-hzj-ky ~]# _
```

3.2.2 Hello, world!

将 test 文件夹上传到服务器

```
[root@yya-hzj-ky ~]# cd test
[root@yya-hzj-ky test]# cd hello-world
[root@yya-hzj-ky hello-world]# ls
cmdline.sh  hello_world.c  Makefile
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
[root@yya-hzj-ky hello-world]#
```

将编译并执行 hello-world.c

装载到内核运行， 在装载时会打印 hello, Yya, 2023

使用 rmmmod 指令卸载模块，在退出时打印了 Exit module.

```

[root@yaya-hzj-ky ~]# cd test
[root@yaya-hzj-ky test]# cd hello-world
[root@yaya-hzj-ky hello-world]# ls
cmdline.sh  hello_world.c  Makefile
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]#
[root@yaya-hzj-ky hello-world]# make
make -C /usr/lib/modules/4.19.90/build M=/root/test/hello-world modules
make[1]: Entering directory '/root/kernel-openEuler-20.03-LTS-SP3'
  CC [M]  /root/test/hello-world/hello_world.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/test/hello-world/hello_world.mod.o
  LD [M]  /root/test/hello-world/hello_world.ko
make[1]: Leaving directory '/root/kernel-openEuler-20.03-LTS-SP3'
[root@yaya-hzj-ky hello-world]# insmod hello_world.ko guy="Yya Hzj & Ky" year=2023
[root@yaya-hzj-ky hello-world]# insmod hello_world.ko guy="YyaKy" year=2023
insmod: ERROR: could not insert module hello_world.ko: File exists
[root@yaya-hzj-ky hello-world]# lsmod | grep hello
hello_world                262144  0
[root@yaya-hzj-ky hello-world]# rmmod hello_world
[root@yaya-hzj-ky hello-world]# dmesg | tail -n3
[ 1082.420678] Init module.
[ 1082.420886] Hello, Yya, 2023!
[ 1173.000035] Exit module.
[root@yaya-hzj-ky hello-world]#

```

3.2.3 char 数据类型

将 ch.c 文件上传到服务器，并用 gcc 编译执行，可以看到执行结果，如下图。

```

root@yaya-hzj-ky:~/cpp
root@yaya-hzj-ky ~]#
root@yaya-hzj-ky ~]# ./ch
-bash: ./ch: No such file or directory
root@yaya-hzj-ky ~]# ch
-bash: ch: command not found
root@yaya-hzj-ky ~]# a.out
-bash: a.out: command not found
root@yaya-hzj-ky ~]# a
-bash: a: command not found
root@yaya-hzj-ky ~]# cd
root@yaya-hzj-ky ~]# mv ch.c app
mv: overwrite 'app'?
root@yaya-hzj-ky ~]# ls
a.out  app  ch.c  kernel-openEuler-20.03-LTS-SP3  openEuler-20.03-LTS-SP3.zip  test
root@yaya-hzj-ky ~]# mkdir cpp
root@yaya-hzj-ky ~]# mv ch.c cpp
root@yaya-hzj-ky ~]# cd cpp
root@yaya-hzj-ky cpp]# ls
ch.c
root@yaya-hzj-ky cpp]# gcc ch.c -o ch
root@yaya-hzj-ky cpp]# ls
ch  ch.c
root@yaya-hzj-ky cpp]# ./ch
sizeof ch is 1, 1, 1, 1

char ch = ffh, +255, positive

signed char ch = ffh, -1, negative
unsigned char ch = ffh, +255, positive
root@yaya-hzj-ky cpp]#

```

ch.c 做的工作是创建一个值为-1 的 char 值，

分别打印在 openeuler 操作系统下，char 的长度，ch 的长度，signedchar 的长度，unsigned char 的长度。

从图中可以看出均为 1。

然后打印 ch 以本身输出，以 signedchar 格式输出，以 unsignedchar 格式输出的值

可以看到分别为 255，-1，和 255。所以在鲲鹏平台，默认 char 是无符号的数。

4 实验遇到的问题

4.1 问题一

4.1.1 问题描述

在执行以下命令时，遇到下图报错

```
yum install -y openssl-devel  
The downloaded packages were saved in cache until the next successful transaction.  
You can remove cached packages by executing 'yum clean packages'.  
Error: GPG check FAILED
```

4.1.2 解决方法

将上述命令改成下面的

```
yum install -y openssl-devel --nogpgcheck
```

5 小组分工

我们小组共同购买了一个华为云服务器，然后各自在自己电脑上完成全部实验，由于实验内容一样，本报告只粘贴了一份实验结果截图。