

Linux 内核云上实验指导

2021年4月

华为鲲鹏云服务器

使用华为云账号登陆[华为云平台](#)，统一注册的账号包含200元华为云代金券。

创建虚拟私有云(VPC)

1. 选择 产品 -> 基础服务 -> 虚拟私有云 VPC -> 创建VPC -> 访问控制台 -> 创建虚拟私有云
2. 区域选择 华东-上海一，其他默认即可
3. 访问控制 -> 安全组 中可以设置安全组。默认已放通22端口和3389端口，如有需要，可以添加规则->全部放通。

购买弹性云服务器(ECS)

1. 选择 产品 -> 基础服务 -> 弹性云服务器 ECS -> 立即购买
2. 基础配置

注：实验与华为鲲鹏云合作，ECS规格要求选择基于arm架构的鲲鹏服务器，不要选成x86服务器。

计费模式	区域	规格	镜像	系统盘
按需计费	华东-上海一	鲲鹏计算 2vCPUs 4GB	Ubuntu 18.04	至少40GB

3. 网络配置

网络	安全组	弹性公网IP
之前创建的VPC	之前创建的安全组	现在购买

4. 高级配置

设置云服务器名称 (e.g., ecs-zhangsan)、用户名 (e.g., root)、密码 (e.g., LinuxKernel2020)

云备份选择<暂不购买>

5. 确认配置

务必确认收费模式是否为<按需计费>，注意每小时金额。若误操作可能会导致账户余额不够、无法完成实验。

下载SSH工具

推荐 mobaxterm (windows)、vscode + ssh remote插件、Royal TSX (mac)

当然，mac OS的terminal自带ssh功能

登陆ECS

创建好ECS后，可以在 控制台 -> 弹性云服务器(区域务必选择上海一) 中看到弹性公网IP。

使用SSH工具，输入EIP、用户名和密码，或 `ssh usr@EIP` 即可登陆。

创建私有镜像

控制台 - 云服务器控制台 - 镜像服务 - 创建私有镜像

保存私有镜像可以方便重新购买ECS时恢复之前搭好的环境。如有需要，可以在删除ECS之前制作私有镜像。

实验零 编译内核

实验要求

将ECS的内核版本更新到5.5以上。

注：Ubuntu 18.04操作系统默认的内核版本为4.15，可通过 `uname -a` 查看。需要将其内核版本更新为5.5以上，以5.5.9为例。

下载linux kernel源码

登陆ECS，到[linux kernel官网](https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.5.x.tar.xz) 下载最新源码包。

```
$wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.5.x.tar.xz
```

注：如果ECS的下载速度太慢，也可以先下载到本地，再从本地上传到ECS(参考：scp命令、sftp工具)。

解压

```
tar xvf linux-5.5.x.tar.xz
cp -r linux-5.5.x /usr/src/
cd /usr/src/linux-5.5.x
```

更换apt源

ubuntu自带的apt源经常出现连接不上情况，最好换成国内源，这里以华为源为例。

```
cp /etc/apt/sources.list /etc/apt/sources.list.bak
rm /etc/apt/sources.list
lsb_release -c
vim /etc/apt/sources.list //注：x86对应的是ubuntu，arm对应的是ubuntu-ports
```

```
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic main restricted
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic main restricted
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates main restricted
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates main
restricted
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic universe
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic universe
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates universe
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates universe
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic multiverse
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic multiverse
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates multiverse
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-updates multiverse
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-backports main restricted
universe multiverse
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-backports main
restricted universe multiverse
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-security main restricted
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-security main
restricted
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-security universe
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-security universe
deb http://mirrors.huaweicloud.com/ubuntu-ports bionic-security multiverse
deb-src http://mirrors.huaweicloud.com/ubuntu-ports bionic-security multiverse
```

安装依赖

```
apt-get update
apt-get install gcc make libncurses5-dev openssl libssl-dev build-essential pkg-config libc6-dev bison flex libelf-dev
```

配置内核

```
cd /usr/src/linux-5.5.x
//make clean
//make mrproper
//make menuconfig                                ->图形界面
//make defconfig
->根据主机架构自动选择对应arch下的默认配置
cp /boot/config-4.15.0-70-generic .config          ->涉及虚拟硬盘、虚拟网卡等配置 默认的没有打开
vim .config    //VIRTIO_GPU    ->    #VIRTIO_GPU    ->没有GPU，所以不要编译这个模块
make olddefconfig
->按照.config来配置
```

编译和模块安装

```
//tmux new-session -s kernel    ->防止make过程中ssh断开
make -j2                        -> -jn表示用n个进程，n一般设置为core的数量
make modules_install            ->安装模块
make install                    ->会自动安装kernel、创建initramfs、更新引导
```

注：编译内核的过程可能比较久，期间如果因为网络原因断开了与服务器的连接，可能会功亏一篑。为了避免这种情况，这里推荐使用tmux工具，具体用途和命令可自行google。

重启

```
shutdown -r now
```

检查内核版本

```
uname -a
```

实验二 进程管理

实验内容

1. 为 `task_struct` 结构添加数据成员 `int ctx`，每当进程被调用一次，`ctx++`。
2. 把 `ctx` 输出到 `/proc/<PID>/ctx` 下，通过 `cat /proc/<PID>/ctx` 可以查看当前指定进程的 `ctx` 的值。

实验提示

1. 进程是如何管理的？在 linux 源码中找到 `task_struct` 的定义，在合适的位置声明 `ctx` 成员。源码文件 `include/linux/sched.h`
2. 进程是如何被创建的？在创建进程的函数中，找到合适的位置，初始化 `ctx`。源码文件 `kernel/fork.c`
3. 进程如何被调度？在调度进程的函数中，找到合适的位置，`ctx++`。源码文件 `kernel/sched/core.c`
4. 每个进程都在 `/proc` 下有自己的目录，那么这些目录是在哪里被创建的呢？如何自定义文件操作函数？参考 `pid_entry` 和 `tgid_base_stuff`。源码文件 `fs/proc/base.c`
5. 本实验要改动的代码量很少，但需要仔细阅读 linux 源码。

实现思路（仅供参考）

```
// 1. 在<include/linux/sched.h>中声明ctx
struct task_struct {
    //...
    //在合适的位置声明ctx
    //...
}

// 2. 在<kernel/fork.c>中创建进程时初始化ctx = 0

// 3. 在<kernel/sched/core.c>中调度时ctx++

// 4. 在<fs/proc/base.c>中创建proc entry

// 5. 编译内核，参考实验零

// 6. 可以使用<test.c>来检查实验效果
#include <stdio.h>
int main(){
    while(1) getchar();
    return 0;
}
```

```
# 可能用到的shell命令
ps -e | grep <program_name>
cat /proc/xxxx/ctx
```

实验效果

写一个 `test.c` 程序来循环接收输入，每接收输入一次，它得到一次调度，对应 `ctx` 加一。

作业验收与提交

验收方式：电院3号楼118向助教展示实验效果

提交渠道: Canvas

提交文件: "学号_姓名_project2.zip"

- 不要求提交源码, 但需要在报告中注明对源码的具体修改过程。
- 实验报告《学号_姓名_project2_report.pdf》, 包括但不限于实验过程、实验效果截图、实验心得。