Linux 内核编译和模块编写

1. Linux内核编译
2. 内核config文件

内核目录下的.config是实际编译用到的配置文件，可以采用make menuconfig进入可视化配置界面。

部分内核选项可以选择是否以可动态插入的模块方式编译。

1. 编译命令

make –j 64 ----编译内核，生成vmlinux

make modules\_install –安装模块（/lib/modules/\*）

make install （/boot）

1. 模块编写
2. 模块编写例子

tos\_example.c

#include <linux/init.h>

#include <linux/module.h>

#include <linux/kernel.h>

MODULE\_LICENSE(“GPL”);

MODULE\_AUTHOR(“XXX”);

MODULE\_DESCRIPTION(“A tencent os Linux module.”);

MODULE\_VERSION(“1.0”);

static int \_\_init tos\_example\_init(void) {

printk(KERN\_INFO “Hello, World!\n”);

return 0;

}

static void \_\_exit tos\_example\_exit(void) {

printk(KERN\_INFO “Goodbye, World!\n”);

}

module\_init(tos\_example\_init);

module\_exit(tos\_example\_exit);

1. Makefile / Kconfig编写

Makefile:

obj-m += tos\_example.o

Kconfig:

config TOS\_EXAMPLE

bool "TOS Driver example"

depends on XXX

default m/y

1. 任务
2. 分组
3. Github账号申请/加入OpenCloudOS社区
4. 编写一个模拟驱动

第一组： 在一个驱动中定时获取内存状况，写入一个文件中。文件大小有一定限制，达到限制则回滚。（提示：高精度定时器/内核态下如何操作文件）

第二组：在一个驱动定时获取所有cpu的占用情况，如果某个cpu上的占用大于一定阈值，输出一个warning。（提示：如何获取每个cpu的统计信息）

1. 搭建环境

Centos/fedara/tencentOS ：rpm/yum

Ubuntu：apt