Hello world kernel module

Giảng Viên: Lưu An Phú

Agenda

- Introduction
- Setup environment for development
- Create Hello world kernel module
- Home work

Introduction

- Can extend kernel code when needed
- Interact with kernel to become only one kernel process
- Code instruction has same priority with kernel

Setup environment for development

Hardware

- Board
  - Pi
    - Good support from community
    - Poor hardware document
  - Udoo
    - Good hardware document
    - Should use Udoo for development

Firmware

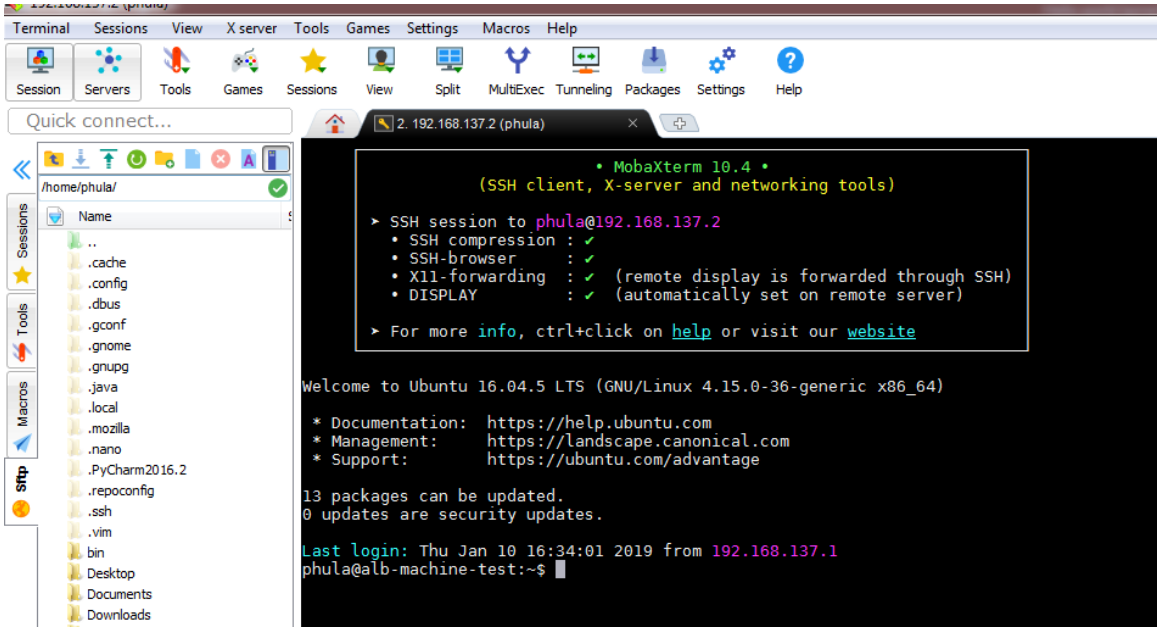
- Download firmware from main page
  - https://www.raspberrypi.org/downloads/raspbian/
- Install firmware
  - Install Window's tool to write firmware for sdcard
    - https://www.raspberrypi.org/documentation/installation/installing-images/README.md
- Plug in sdcard, power cable, mouse, keyboard and monitor for Pi

Connect PC with board

- Install MobaXterm for PC
  - https://mobaxterm.mobatek.net/
- Setup network to connect Pi with PC
  - Set static IP for Pi and PC
- Install ssh tool for Pi
- Create new session in MobaXterm to ssh with Pi

Install kernel header in Pi

- It include all source header need to develop kernel module
- It is generate when they build kernel image
- Each kernel instance has only one kernel header
- sudo apt-get install raspberrypi-kernel-headers



Final result

Create Hello world kernel module

Source code

```c
/*
 * hello_world.c - The simplest kernel module.
 */
#include <linux/module.h>	/* Needed by all modules */
#include <linux/kernel.h>	/* Needed for KERN_INFO */

int init_module(void)
{
	printk(KERN_INFO "Hello world 1.\n");

	/*
	 * A non 0 return means init_module failed; module can't be loaded.
	 */
	return 0;
}

void cleanup_module(void)
{
	printk(KERN_INFO "Goodbye world 1.\n");
}
```

Compile kernel module

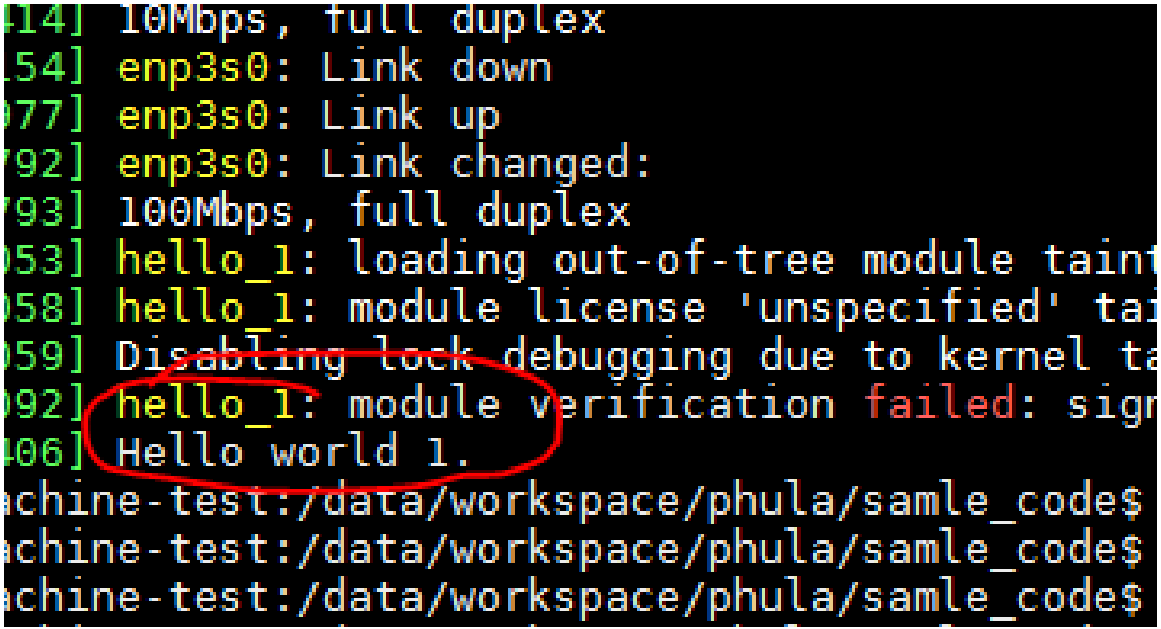
- Create makefile to compile

```
obj-m += hello_world.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

- make



Load and test result

Home work

- Kết nối đèn led với chân GPIO của board. Đọc RM để biết cách cài đặt chân GPIO của board. Viết 1 kernel module để mỗi khi load lên sẽ bật đèn led, unload sẽ tắt đèn led.
  - Tham khảo các hàm sau:
    - devm_ioremap_resource
    - __raw_writel
    - https://youtu.be/3MY8kLEZr98

Thank you