

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Operating System Principle |
| **Semester**: | 2nd semester of the academic year **2023-2024** |
| **Major**: | Software Engineering |
| **Class**: | 2022 |
| **Student Name**: | 吴孜远 |
| **Student ID:** | 222022321062009 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | C Programming in Linux | | | |
| Date | | March, 2024 | Type | | √ Confirmatory  √ Design  □Comprehensive |
| 1. **Objective & Requirements**    1. Learn how to install the Linux system within a virtual machine    2. Grasp the command line interface of Linux    3. Learn to do C programming with Linux    4. Learn how to write simple Makefile for managing C projects    5. Learn how to write, compile, and load linux kernel modules | | | | | |
| 1. **Experimental environment (**platform and software**)**   Virtualbox + Ubuntu (or other platform+linux system combinations) | | | | | |
| 1. **Experimental content and design** (Main Content, Procedure, Codes and Results) 2. Tasks for this lab    1. Task 1   Write a C program with at least two \*.c source files and one \*.h head file. Use Makefile and the make tool to compile your C program.   * 1. Task 2   Use kernel module to access the two values jiffies and HZ defined in the Linux kernel:   * HZ: the frequency of timer interrupt (HZ is of type int) * jiffies: the number of timer interrupt since system boot (jiffies is of type unsigned long int)   Please output the value of jiffies twice, i.e. when the module is loaded and when the module is removed. Then based on the values of jiffies and HZ, compute how long your kernel module stays in the kernel. Please measure your time in seconds.   1. Please provide your procedure and source codes to perform the tasks. 2. 编写了三个.c文件，在makefile中定义了其连接关系，makefile文件内容如下：     main.c如下：    sum.c如下：    mul.c如下：    sum.h如下：    mul.h如下：    操作流程如下：  可以看到，寥寥几行命令就执行了多个编译链接操作，这就是make tool带给我的自信    2）  通过make tool来编译内核模块：    加载内核模块，查看日志    内核模块退出，    我insmod之后很快就退出了，模块在内核中的时间为16秒。  其实可以通过在每条输出中都加一个标志字符串，像老师给的example里一样（“hello“），这样就可以通过grep “…”来筛选出输出，而不是像我现在这样打印了整个日志。、  这是我的hello.c源代码：  #include <linux/module.h>  #include <linux/kernel.h>  unsigned long jiffies1;  unsigned long jiffies2;  unsigned long delta\_jiffies;  int h;  /\* This function is called when the module is loaded. \*/  int hello\_entry(void)  {  h = HZ;  jiffies1 = jiffies;  printk(KERN\_INFO "Module loaded.\n");  printk(KERN\_INFO "HZ is %d, jiffies is %lu.\n", h,jiffies1);  return 0;  }  /\* This function is called when the module is removed. \*/  void hello\_exit(void)  {  h = HZ;  jiffies2 = jiffies;  delta\_jiffies = jiffies2 - jiffies1;  printk(KERN\_INFO "HZ is %d, jiffies is %lu.\n", h,jiffies2);  printk(KERN\_INFO "Module unloaded. Delta jiffies: %lu\n", delta\_jiffies);  printk(KERN\_INFO "Time spent in the module (seconds): %lu\n", delta\_jiffies / HZ);  }  /\* Macros for registering module entry and exit points. \*/  module\_init(hello\_entry);  module\_exit(hello\_exit);  MODULE\_LICENSE("GPL");  MODULE\_DESCRIPTION("A simple kernel module example");  MODULE\_AUTHOR("wuziyuan");  Makefile文件:  obj-m += hello.o  all:  make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) modules  clean:  make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) clean | | | | | |
| 1. **Result analysis and discussion**   In this part, you are required to provide your analysis of experimental results and summing up the harvest and the existing problems; besides, you are required to provide your thinkings about the questions:   * Why printf() is not applicable in the kernel?   使用make tool，我们可以不用再每次运行项目的时候都一个一个文件编译，像那种模块化的项目还要链接。我们直接在makefile中定义好即可，用一些包装好的高级命令来操作，非常方便。  此外，学会了如何加载内核模块，查看日志，并成功运行了加载的模块，得到输出。  思考题：  Printf函数实现在用户层的libc库，内核层和用户层有隔离，printf函数在内核中无法链接使用用户层的库函数。主要问题在于，c库太大了，如果要编译链接c库到内核中，消耗时间太长，效率低，不符合内核中运行速度的要求。如果将printf的依赖项集成到内核中也会显著增加内核的大小和复杂性，从而影响其性能。 | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |