

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Operating System Principle |
| **Semester**: | 2nd semester of the academic year **2020-2021** |
| **Major**: | Software Engineering |
| **Class**: | 2019SE4 |
| **Student Name**: | 吴嘉诚 |
| **Student ID:** | 222019321062111 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | C Programming, Makefile and Linux Kernel Module | | | |
| Date | | March 26, 2021 | Type | | √ Confirmatory  √ Design  □Comprehensive |
| 1. **Objective & Requirements**    1. Learn to do C programming with Linux    2. Learn how to write simple Makefile for managing C projects    3. 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, compile, and run a C program with at least two \*.c source files and one \*.h head file.   * 1. Task 2   Use Makefile and the make tool to compile your C program with at least two \*.c source files and one \*.h head file as in Task 1.   * 1. Task 3   Use kernel module to access the two values jiffies and HZ defined in the linux kernel:   * HZ: the frequency of timer interrupt * jiffies: the number of timer interrupt since system boot   Please output the value of jiffies twice, i.e. when the module is loaded and when the module is removed. Then based on the two jiffies and HZ, compute how long your kernel module stays in the kernel.   1. Please provide your procedure and source codes to perform the tasks.   **Task1:**   1. Write code:   2021-03-26 20-15-12 的屏幕截图   1. Source code:   #include <stdio.h>  #include "multiply.h"  int main(int argc, char const \*argv[]) {  printf("This is hello.c\n");  int a,b;  scanf("%d%d",&a,&b);  printf("%d \* %d = %d\n",a,b,multiply(a,b));  return 0;  }   1. One step compile and Substep compile:   2021-03-26 20-08-48 的屏幕截图  **Task2:**   1. Source code:   #include "mult.h"  int mult2(int x,int y){return x\*y;}  int mult3(int x,int y,int z){return x\*y\*z;}  #include <stdio.h>  #include "mult.h"  int main(int argc, char const \*argv[])  {  printf("This is main.c\n");  printf("3 \* 5 = %d\n",mult2(3,5));  printf("3 \* 5 \* 6 = %d\n",mult3(3,5,6) );  return 0;  }   1. Make and clean by using Makefile:   2021-03-26 21-01-45 的屏幕截图  **Task3:**   1. Check module information and insert module   2021-03-26 21-47-56 的屏幕截图   1. Check the output in the memory buffer   2021-03-26 21-51-16 的屏幕截图   1. The homework part 2. Source code:   #include <linux/module.h>  #include <linux/kernel.h>  //called when module is loaded  unsigned long jiffies\_temp;  int jiffies\_entry(void)  {  jiffies\_temp = jiffies;  printk(KERN\_INFO "The jiffies: %lu (entry)\n",jiffies\_temp);  printk(KERN\_INFO "The HZ: %d \n",HZ);  return 0;  }  //called when module is removed  void jiffies\_exit(void)  {  printk(KERN\_INFO "The delta jiffies: %lu\n",jiffies-jiffies\_temp);  printk(KERN\_INFO "The time elapsed: %ld\n",(jiffies-jiffies\_temp)/HZ);  printk(KERN\_INFO "it is exiting...\n");  }  //macros(宏) for registering module entry and exit points  module\_init(jiffies\_entry);  module\_exit(jiffies\_exit);  MODULE\_LICENSE("GPL");  MODULE\_DESCRIPTION("kernel module example : jiffies");  MODULE\_AUTHOR("zhj,");   1. Make, insmod, rmmod, dmesg: 2021-03-30 09-15-59 的屏幕截图 | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems） 2. **The three tasks are successfully finished.** 3. **Harvest** 4. **From Task1, i learned how to sub-compile an C program.** 5. **In Task2, i learned how to use MAKE, and its use.** 6. **And i learned a little of the kernel module about how to write, insert, remove a module and the printk().,dmesg command.** 7. **EXISTING PROBLEMS** 8. **Is cmake same as make? And what about QT’s qmake. I don’t know their key differences.** 9. **In Task3, i can only see my last prints-out message after my next commands of insmod,rmmod,dmesg. At last i found it might be the \n line break.** 10. **In Task3, why we don’t need to #include <linux/jiffies.h> and still can use jiffies and HZ.** | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |