

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Operating System Principle |
| **Semester**: | 2nd semester of the academic year **2019-2020** |
| **Major**: | Software Engineering |
| **Class**: | 2019 |
| **Student Name**: | 冯春霖 |
| **Student ID:** | 222019321062074 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | Process scheduling in Linux: The CFS Scheduling | | | |
| Date | | April，2021 | Type | | □Confirmatory  □Design  √ Comprehensive |
| 1. **Objective & Requirements**    1. Understanding the concept of processor affinity    2. Learn how to set processor affinity in linux    3. Learn how to set process priority in linux    4. Understanding the CFS scheduling policy of linux    5. Review how to compile and load kernel module in linux. Learn how to pass parameters to kernel modules.    6. Learn how to access kernel scheduling information by loading kernel modules, and certify the CFS scheduling strategy. | | | | | |
| 1. **Experimental environment (**platform and software**)**   Virtualbox+Ubuntu linux | | | | | |
| 1. **Experimental content and design** (Main Content, Procedure, Codes and Results) 2. Tasks for this lab   Understand how the scheduling information is represented in PCB. Write, compile and load a kernel module to access the real runtime and virtual runtime of two processes with different priorities. Note that the two processes need to be bound to the same CPU core. To facilitate the task, you may need to learn how to pass parameters to kernel modules. Compute the ratio of real runtime and virtual runtime of the two processes, and compare the ratios with the ratio of the two processes’priorities. In this way you can certify the CFS scheduling policy.    Figure: The weights in CFS for different NICE values   1. Please provide your procedure to perform the tasks and source codes.   The modified kernal code is as follows:    First run the program to change the priority:    Load the kernel module, passing the PID as a parameter  Output the real runtime and virtual runtime of two threads    Calculated, the ratio of real running time is 3.04697, and according to the CFS algorithm, the ratio of -5 to 0 priority is 3.01891. The two values are very close and thus we can verify the use of the cfs algorithm in Linux systems | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems）   Through this experiment, I got a more comprehensive understanding of CPU scheduling on Linux systems. Firstly, I learned about processor affinity and tried to change the processor affinity of a program by command, and secondly, I learned about priority and the CFS algorithm that Linux uses for handling program priority. In completing the experiment, I also reviewed knowledge about Linux kernel modules, learned how to pass parameters like in kernel modules, and verified that the Linux system uses the CFS algorithm by loading kernel modules. Overall, the experiment was well done and achieved its intended purpose. | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |