

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Operating System Principle |
| **Semester**: | 2nd semester of the academic year **2019-2020** |
| **Major**: | Software Engineering |
| **Class**: | 2019 |
| **Student Name**: | 吴嘉诚 |
| **Student ID:** | 222019321062111 |
| **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. 2. Make，编译源文件，并运行set\_priority程序，得到子父进程的pid   2021-05-09 10-21-51 的屏幕截图   1. 在另一个终端中查看到两进程的相关信息：nice值分别为0和-5，PSR值都为0，即被一同绑定在0号核上运行   2021-05-09 10-22-11 的屏幕截图   1. 插入内核并跟上两进程的pid作为参数，并显示缓冲区信息   2021-05-09 10-27-27 的屏幕截图  2021-05-09 10-26-30 的屏幕截图   1. 修改cfs.c文件，以显示virtual runtime   2021-05-09 11-33-39 的屏幕截图   1. 重新make内核模块并插入，查看信息：   2021-05-09 11-06-04 的屏幕截图  此处的ratio=10523455280/32034075572=0.328508162  nice(0)/nice(-5)=1024/3121=0.328099968  Virtual time 的值相近是没有问题的 | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems）   总结：   * 加深了对CFS的理解：所有进程的vruntime的增长速度在宏观上是相同的，而实际runtime的增长速度不同。即CFS的思想就是让每个进程的vruntime互相追赶，而每个进程的vruntime增加速度不同，权重越大的增加的越慢，这样就能获得更多的cpu执行时间。   问题：   * 运行set\_priority后发现，子进程的nice值并未变化。最后发现是因为我运行时没加sudo权限 | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |