

高性能计算实验报告

实验4：Linux环境多进程编程

2024秋季学期 姓名：曹馨尹 学号：2023311708

实验内容：

a. 进程cpu利用率截图

```
top - 15:15:25 up 4:38, 1 user, load average: 0.64, 0.31, 0.30
Tasks: 336 total, 3 running, 333 sleeping, 0 stopped, 0 zombie
%Cpu(s): 10.8 us, 1.7 sy, 0.0 ni, 87.5 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 3895.3 total, 143.3 free, 2762.0 used, 1419.7 buff/cache
MiB Swap: 3895.0 total, 3678.7 free, 216.3 used. 1133.3 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 2555 superso+  20   0 7168912 534280 178396 S 102.1  13.4 72:43.09 gnome-shell
 8433 superso+  20   0 27268 26112 1536 R 100.0   0.7  0:07.77 lab4fork
 8434 superso+  20   0 27268 25364 768 R 100.0   0.6  0:07.74 lab4fork
3598 superso+  20   0 32.5g 138952 117520 S 26.9   3.5  4:50.69 code
3558 superso+  20   0 1134.1g 387244 124720 S 4.1   9.7  4:08.43 code
2974 superso+  20   0 282032 110584 101140 S 3.3   2.8  0:52.58 Xwayland
3622 superso+  20   0 741440 59660 44788 S 2.1   1.5  0:16.88 gnome-terminal-
3329 superso+  20   0 218748 3100 2816 S 1.2   0.1  0:20.71 VBoxClient
7299 root       20   0 0 0 0 I 0.8   0.0  0:01.87 kworker/u32:2-events_unbound
 8436 superso+  20   0 14504 5632 3456 R 0.8   0.1  0:00.04 top
3659 superso+  20   0 1131.9g 169760 53636 S 0.4   4.3  0:22.71 code
3802 superso+  20   0 11.3g 429104 193960 S 0.4  10.8  5:48.70 firefox
 1 root      20   0 23236 13796 9188 S 0.0   0.3  0:04.96 systemd
 2 root      20   0 0 0 0 S 0.0   0.0  0:00.03 kthreadd
 3 root      20   0 0 0 0 S 0.0   0.0  0:00.00 pool_workqueue_release
 4 root      0 -20 0 0 0 I 0.0   0.0  0:00.00 kworker/R-rcu_g
 5 root      0 -20 0 0 0 I 0.0   0.0  0:00.00 kworker/R-rcu_p
```

b. pstree截图

```
supersonic@supersonic-VirtualBox:~/hpc_practice/lab4-process/ostep$ ps -ef|grep lab4fork
superso+ 8777 3643 99 15:43 pts/0 00:00:01 ./lab4fork
superso+ 8778 8777 99 15:43 pts/0 00:00:01 ./lab4fork
superso+ 8780 7754 0 15:43 pts/1 00:00:00 grep --color=auto lab4fork
```

说明：在使用 `ps -ef` 命令时找不到对应父子程序的PID，所以这里只提供了 `ps -ef|grep ..` 的截图。

c. 时间数据以下面表格的形式记录，并算出(user + sys)/real比值

父进程：

CPU时间: 8.167821秒

墙钟时间: 8.000000秒

子进程：

CPU时间: 8.198533秒

墙钟时间: 8.000000秒

name	time
real	0m8.231s
user	0m8.167s
sys	0m0.064s
(user+sys)/real	1

实验收获与问题：

- 使用 `ps tree -p` 时找不到对应父子进程的**PID**。
- 对三个**时间**的理解不足。 $user + sys = real$ 这一等式在程序比较简单时才成立。
如果涉及大量的I/O操作，那么real时间会显著增加，因为在信息写入读出过程中cpu不会工作。在代码中我把打印矩阵的那行注释掉了，所以最后时间比值几乎就等于1。
- 理解了 `argc` , `*argv[]` 的作用是从命令行向程序传入参数。
但是我还是不懂为什么在fork调用结束后在 `{ }` 外面写 `return 0;` 会报错（所以最后删掉了）。而我查到的资料表示返回值不管是不是 `void` 都是合法的。