CS 692-04: Linux Kernel Internals (Spring 2024)

Lab3 – Implementing the User-based Fair-Share Scheduling Policy

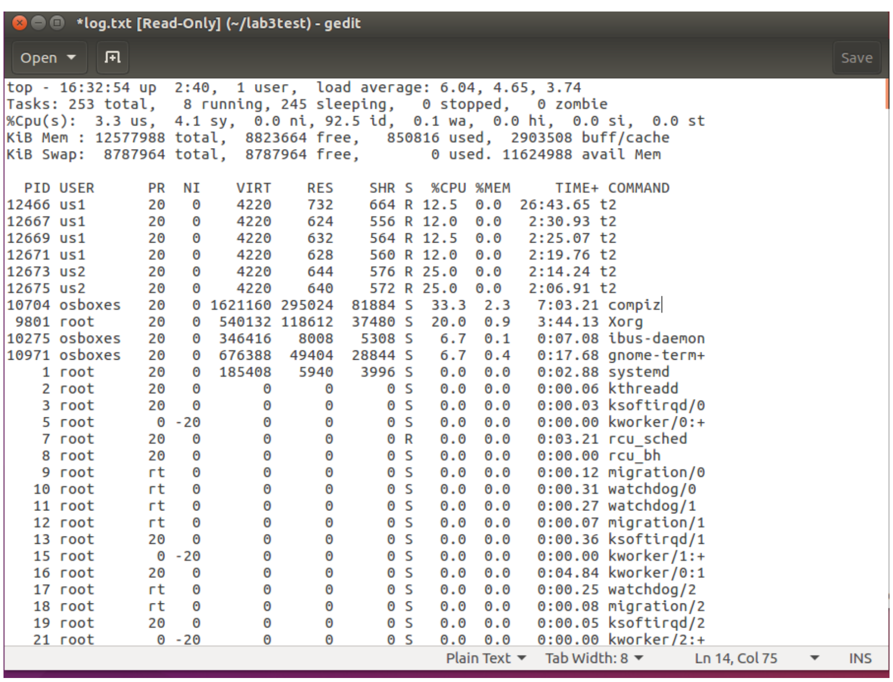
Sina Yari Karin (G01310798)

Lab 3 Procedure:

In order to implement the User-based Fair-Share Scheduling Policy, I did several things, and I will explain them in the following:

1- **I added additional fields to the data structures task_struct (include/linux/sched.h) - called "int is_fssp" which decides if the process has to be scheduled using FSSP or not. A field was also added to user_struct (include/linux/sched.h) - called atomic_t fssp_processes - which keeps track of the processes per user.**
2- **The process of FSSP is initiated when the user program invokes the system call that enables the scheduler. The systemcall checks if the user is new or not based on the count of their processes. If it is 0, the user count is incremented. The process count for that user is also incremented**
3- **I tried to evenly dirbitue the time slice evenly. So the time slice per user is calculated by dividing the required variavbel by the count of users.**
4- **The flag that enables the process to be scheduled using the time slice calculated by FSSP is set.**
5- **The process is now scheduled for its new time slice**
6- **Finally, when the user process exits, it disables FSSP. The system call decreases the process count and the user count (if there are no more processes for that user) and recalculates the TS. The scheduler is set to the default one.**

In order to test the new scheduler, I wrote a simple program that is an infinite loop. In this code, I enable the systemcall for the FSSP and schedule the code with the new scheduling policy. Also, I added two users to the system, and one of them has 4 processes, and the other one has 2 processes. In order to see the CPU usage of each user I ran the "top -n 100 -d 1 -b >log.txt" command, and the results are as following:



```
*log.txt [Read-Only] (~/lab3test) - gedit
Open   Save

top - 16:32:54 up  2:40,  1 user,  load average: 6.04, 4.65, 3.74
Tasks: 253 total,   8 running, 245 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.3 us,  4.1 sy,  0.0 ni, 92.5 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 12577988 total,  8823664 free,   850816 used,  2903508 buff/cache
KiB Swap:  8787964 total,  8787964 free,        0 used. 11624988 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
12466 us1       20   0    4220    732    664 R 12.5  0.0  26:43.65 t2
12667 us1       20   0    4220    624    556 R 12.0  0.0   2:30.93 t2
12669 us1       20   0    4220    632    564 R 12.5  0.0   2:25.07 t2
12671 us1       20   0    4220    628    560 R 12.0  0.0   2:19.76 t2
12673 us2       20   0    4220    644    576 R 25.0  0.0   2:14.24 t2
12675 us2       20   0    4220    640    572 R 25.0  0.0   2:06.91 t2
10704 osboxes   20   0 1621160 295024  81884 S  33.3  2.3   7:03.21 compiz
 9801 root      20   0  540132 118612  37480 S  20.0  0.9   3:44.13 Xorg
10275 osboxes   20   0  346416   8008   5308 S   6.7  0.1   0:07.08 ibus-daemon
10971 osboxes   20   0  676388  49404  28844 S   6.7  0.4   0:17.68 gnome-term+
    1 root      20   0  185408   5940   3996 S   0.0  0.0   0:02.88 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.06 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.03 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:+
    7 root      20   0       0      0      0 R   0.0  0.0   0:03.21 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   0:00.12 migration/0
   10 root      rt   0       0      0      0 S   0.0  0.0   0:00.31 watchdog/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.27 watchdog/1
   12 root      rt   0       0      0      0 S   0.0  0.0   0:00.07 migration/1
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.36 ksoftirqd/1
   15 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:+
   16 root      20   0       0      0      0 S   0.0  0.0   0:04.84 kworker/0:1
   17 root      rt   0       0      0      0 S   0.0  0.0   0:00.25 watchdog/2
   18 root      rt   0       0      0      0 S   0.0  0.0   0:00.08 migration/2
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.05 ksoftirqd/2
   21 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/2:+

Plain Text   Tab Width: 8   Ln 14, Col 75   INS
```

Also, I wrote a file analyzer to check the results, and the results are as follows:

Example 1:

UID	Utilization (%)
U1	12.5
	12.5
U2	12.5
	12.5
U3	12.5
	12.5
U4	12.5
	12.5

Example 2: