Fair Share Scheduling in Linux

Group: 4

Institute of Engineering and Technology, Ahmedabad University

April 30, 2015

Anmol Anubhai - 121004
Pooja Shah - 121035
Rahul Patel - 121040
Shashwat Sanghavi - 121049

Overview

What are Schedulers?

- The main goals of a scheduler is to increase the system utilization while not compromising on the wait time and response time of jobs.
- Performs key role for operating system efficiency

Linux Schedulers

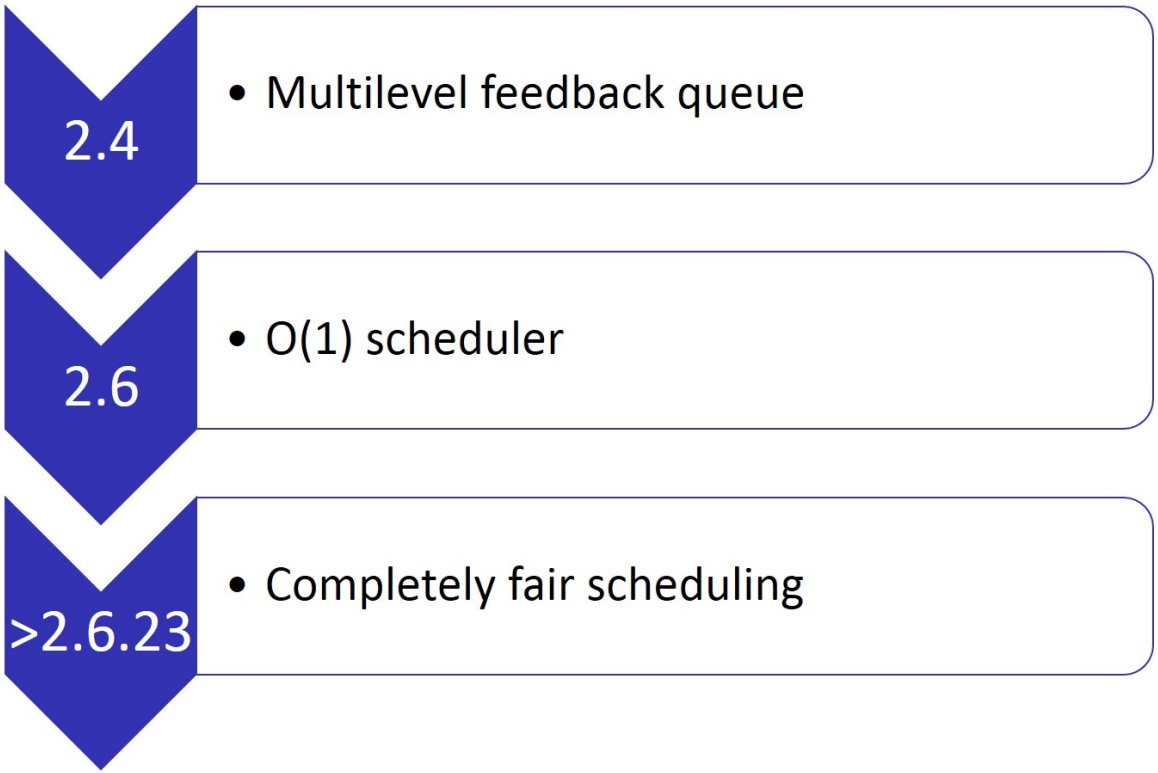

Figure : Linux kernels and their schedulers

Multilevel Feedback Queue

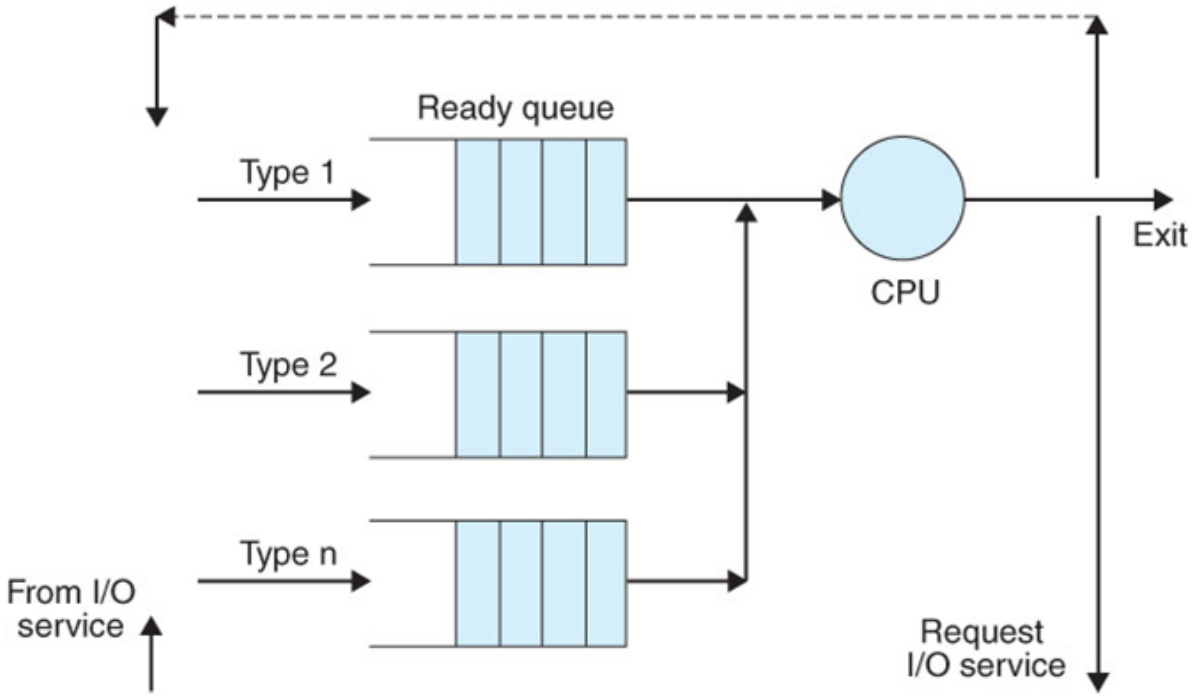

Figure : Multilevel Priority Topics

O(1) Scheduler

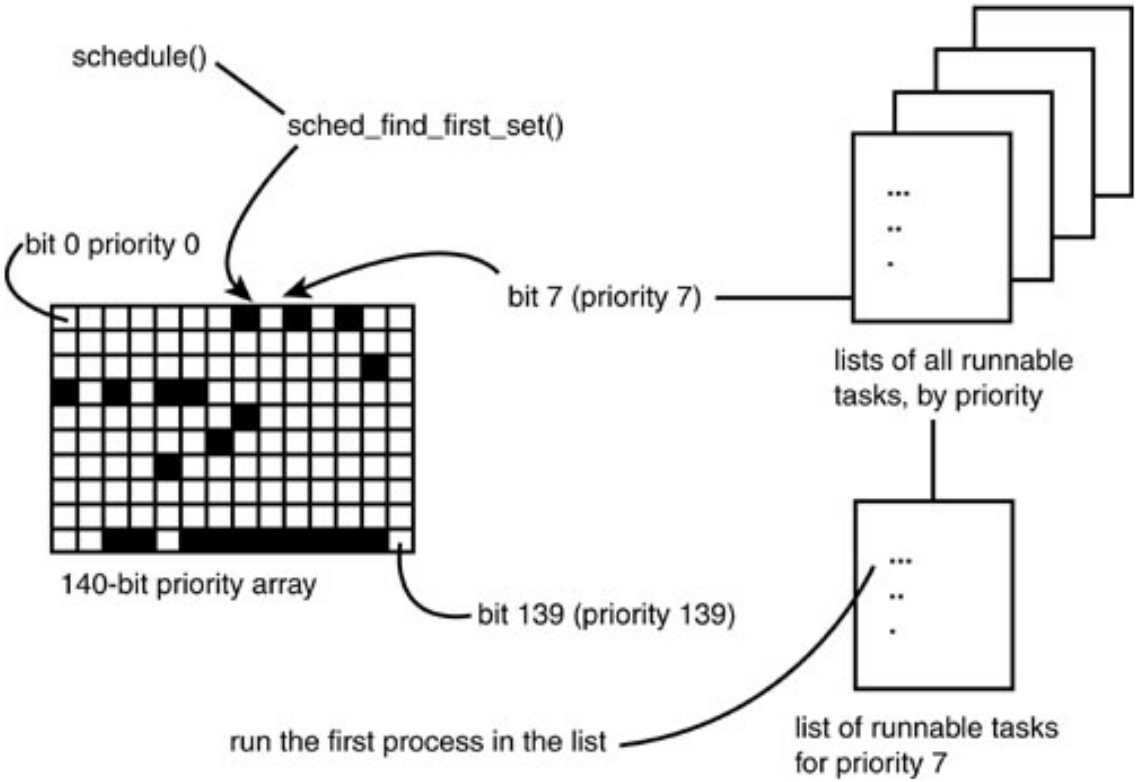

Figure : O(1) Scheduler

Completely Fair Share Scheduling

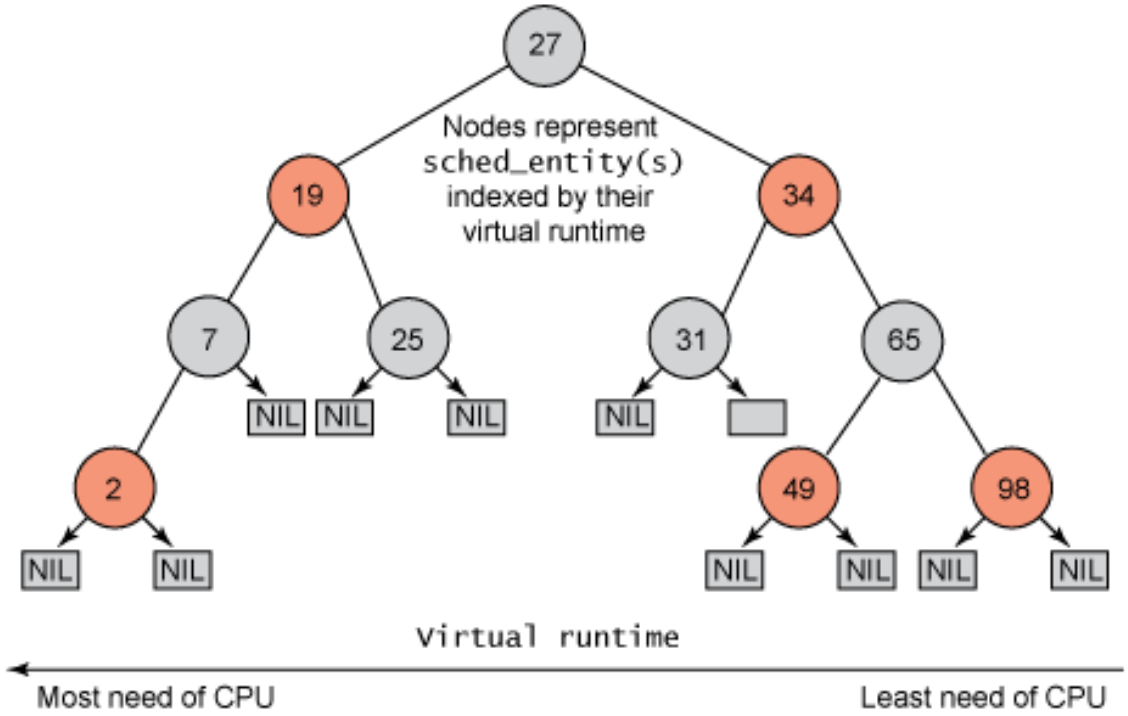

Figure : CFS RB Tree

Fair share scheduling

- Every user gets an equal share of the CPU.
- Every processes of one user gets equal amount of CPU resources.

User	Processes	CPU time per user	CPU time per process
A	3	33%	11%
B	2	33%	16.5%
C	4	33%	8.25%

Figure : FSS example

Background

- Choice of Linux kernel 2.6.38
- File of interest: sched.c
- Some useful functions
  - schedule()
  - sched_fork()
  - finish_task()
  - task_cred()
- Useful structure: task_struct
- Multiple priority runqueue

Block diagram-New process is forked

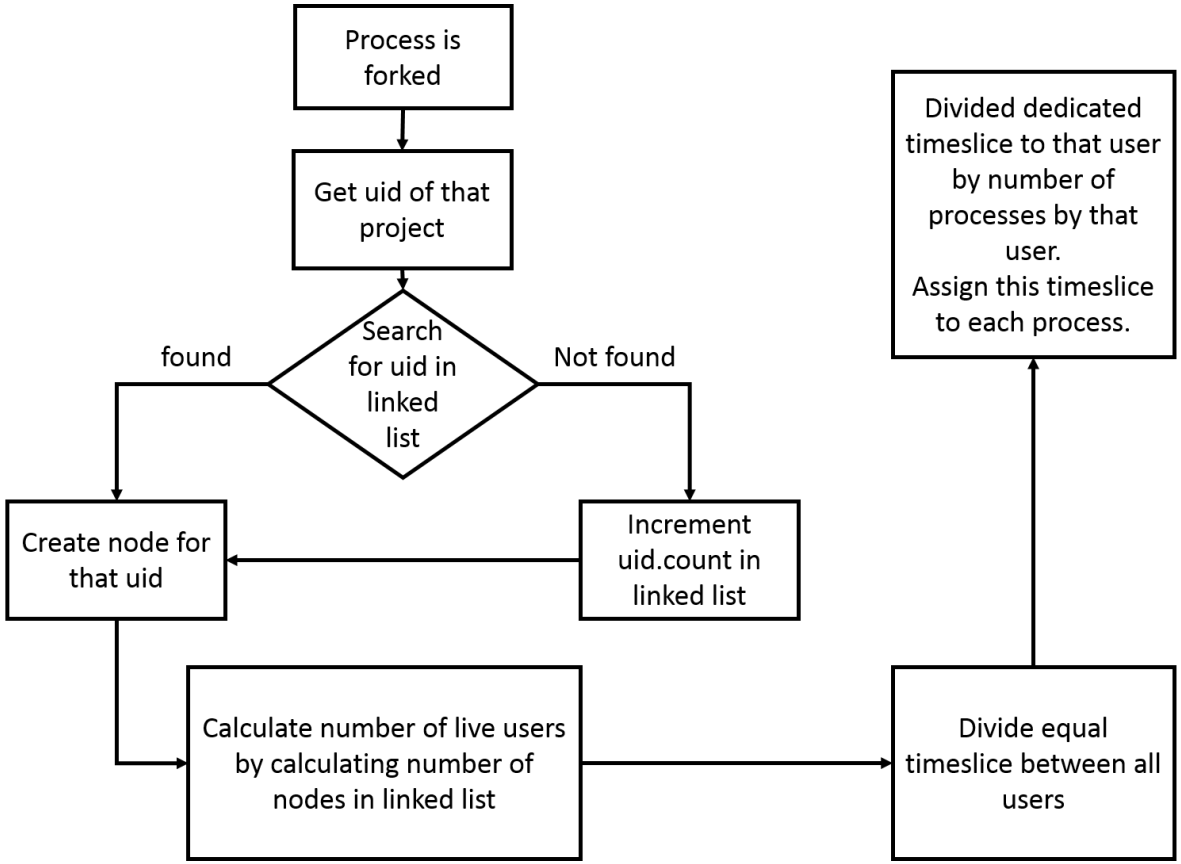

Figure : Algorithm which should be followed when new process is forked

Block diagram-Process is terminated

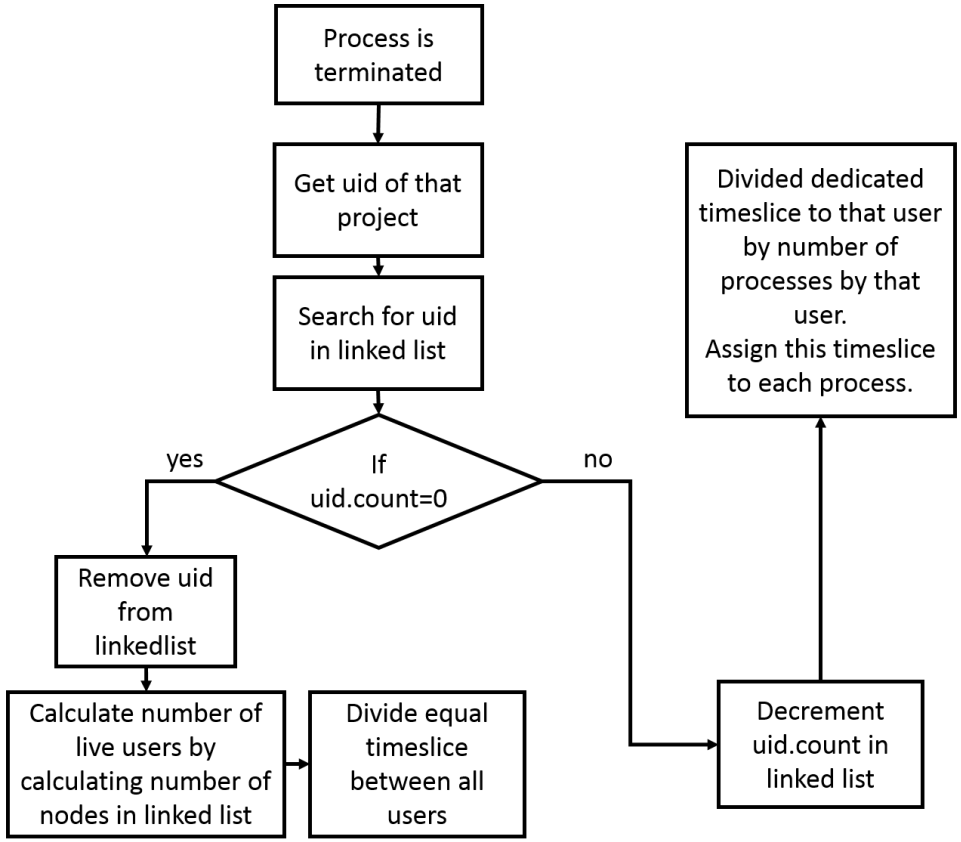

Figure : Algorithm which should be followed when process is terminated

Results

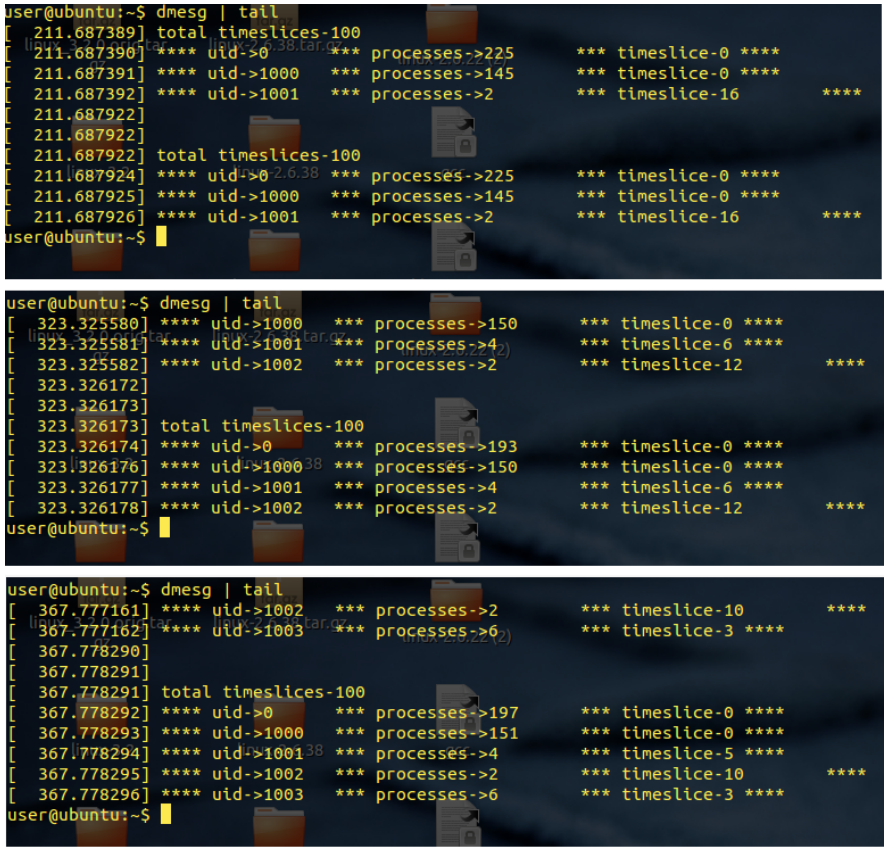

Figure : Results showing the effect of fork exit and user login over the timeslice division

Conclusion

- Our algorithm gives desired output for FSS.
- But, it is O(n) algorithm where n is the number of users.
- Hence, cannot be integrated in linux kernel since it strictly follows O(1) scheduler policy.

Thank You.

REFERENCES


Scheduling in Linux http://www.cs.montana.edu/~chandrima.sarkar/AdvancedOS/CSCI560_Proj_main/


Process Scheduling in Linux
http://www.ittc.ku.edu/~kulkarni/teaching/EECS678/projects/scheduling/materials/scheduling.pdf


Implementing a new real-time scheduling policy for Linux: Part 1
http://www.embedded.com/design/operating-systems/4204929/Real-Time-Linux-Scheduling-Part-1


Background Process Scheduling http://web.cs.wpi.edu/ claypool/courses/3013-A05/projects/proj1/


Linux Kernel Map http://www.makelinux.net/books/lkd2/ch04lev1sec2


Modify the Linux Scheduler to limit the CPU usage of a process family
http://www.csd.uoc.gr/~hy345/assignments/2013/cs345_front4.pdf


LinSched: The Linux Scheduler Simulator http://www.cs.unc.edu/~jmc/linsched/


Understanding the Linux 2.6.8.1 Process Scheduler

http://cs.boisestate.edu/~amit/teaching/597/scheduling.pdf

REFERENCES


CFS Scheduler https://www.kernel.org/doc/Documentation/scheduler/sched-design-CFS.txt


Fair-Share Scheduling http://dasl.mem.drexel.edu/~ducNguyen/courses/cs370-operating-systems/cs370-p4-fair-share-scheduling/


Linux Kernel 2.6.22.19 Scheduler

http://dasl.mem.drexel.edu/~ducNguyen/2013/08/15/linux-kernel-2-6-22-19-scheduler/