

Visually simulate the scheduling process of using CFS in Linux

109598011 徐兆陽
109598073 林育德

Content

01

Slide 3

Scheduler of Linux and CFS

02

Slide 7

What Is Vruntime?

03

Slide 9

Completely Fair Scheduler Idea And Architecture

04

Slide 14

Program Flow Chart

05

Slide 15

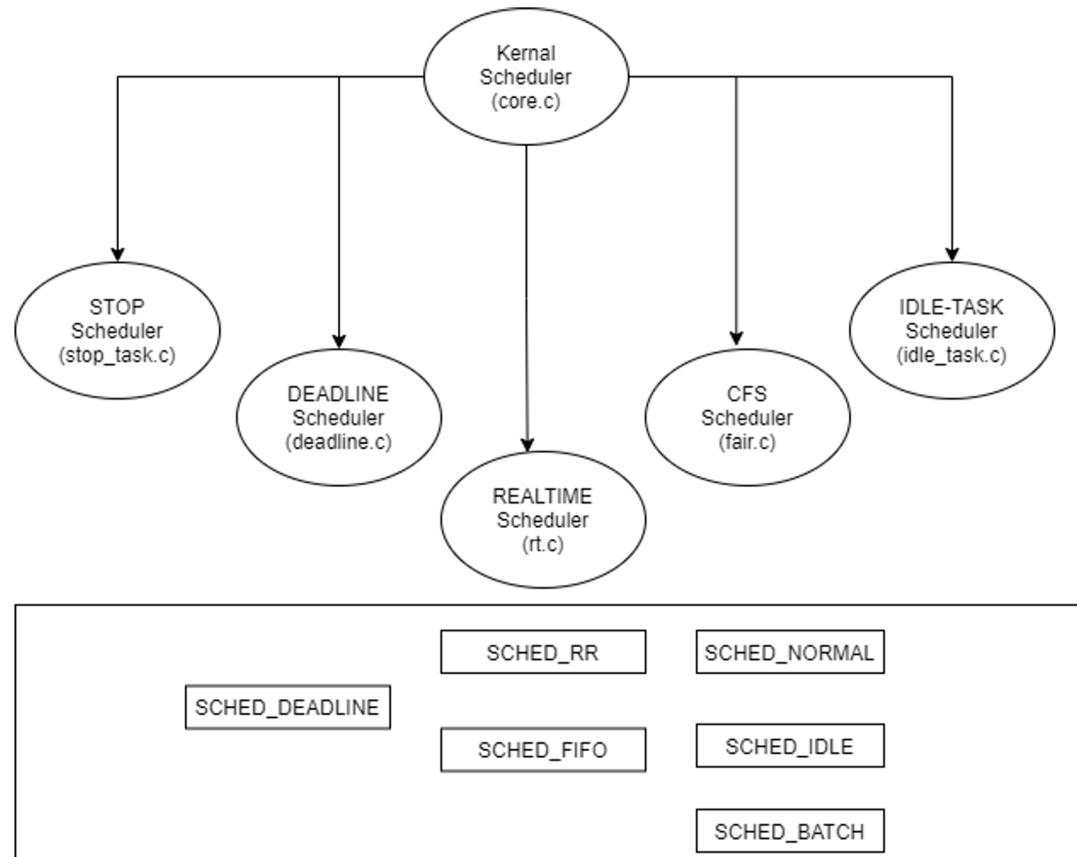
Demo and Validation

06

Slide 16

Reference

Scheduler of Linux



Fair in linux

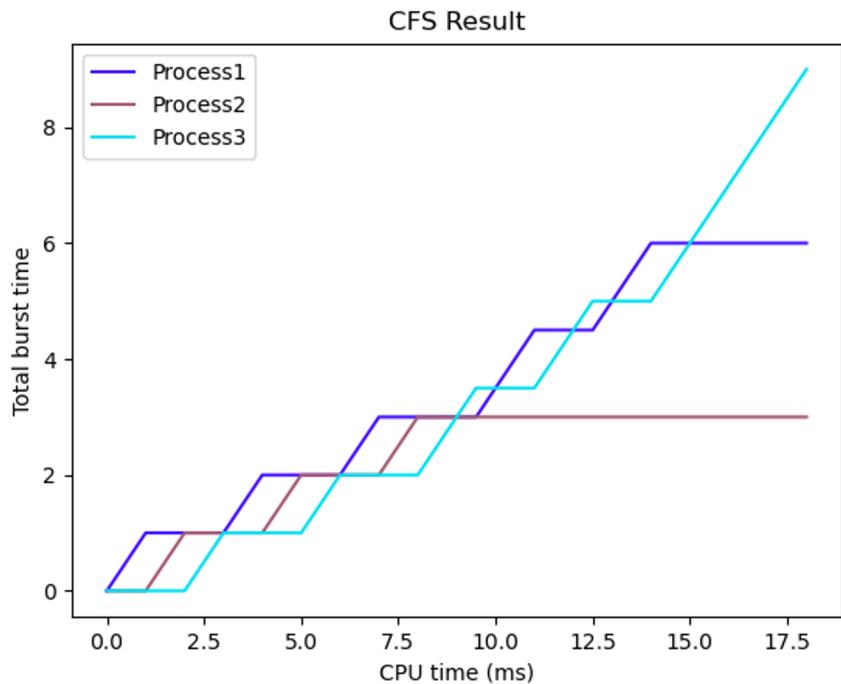
Ideal Fairness: If there are N processes in the system, each process should have got $(100/N)\%$ of the CPU time

Process	arrival time	burst time	NICE
A	0	6 ms	0
B	0	3 ms	0
C	0	9 ms	0

A	1	2	3	4.5	6			
B	1	2	3					
C	1	2	3	4.5	6	9		

Time slice = 3ms

Fair in linux



Process	arrival time	burst time	NICE
A	0	6 ms	0
B	0	3 ms	0
C	0	9 ms	0

A	1	2	3	4.5	6		
B	1	2	3				
C	1	2	3	4.5	6	9	

Time slice = 3ms

Completely Fair Scheduler-NICE to weight

```
const int sched_prio_to_weight[40] = {  
    /* -20 */     88761,      71755,      56483,      46273,      36291,  
    /* -15 */     29154,      23254,      18705,      14949,      11916,  
    /* -10 */     9548,       7620,       6100,       4904,       3906,  
    /* -5 */      3121,       2501,       1991,       1586,       1277,  
    /* 0 */       1024,       820,        655,        526,        423,  
    /* 5 */       335,        272,        215,        172,        137,  
    /* 10 */      110,         87,         70,         56,         45,  
    /* 15 */      36,          29,          23,          18,          15,  
};
```

Completely Fair Scheduler-vruntime

Runtime of the process = $\text{sysctl_sched_latency} * (\text{weight of the process}) / \text{total process}$

vruntime = Runtime of the process * weight nice=0 / (weight of the process) = runtime * weight_factor

sysctl_sched_latency : time of running total runnable tasks once
weight = $1024 * 1.25^{(-\text{nice})}$

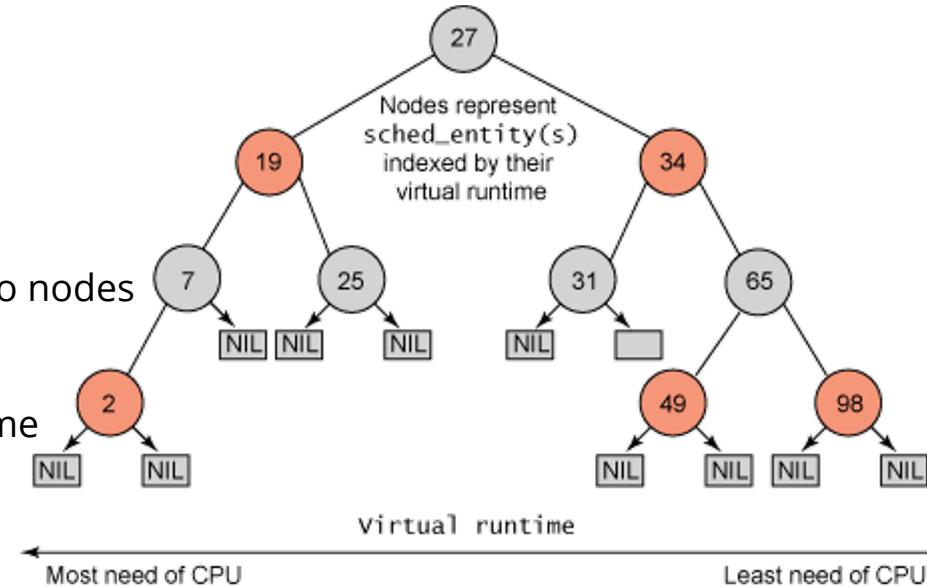
Completely Fair Scheduler

- Each node in the tree represents a runnable task

- Nodes ordered according to their vruntime

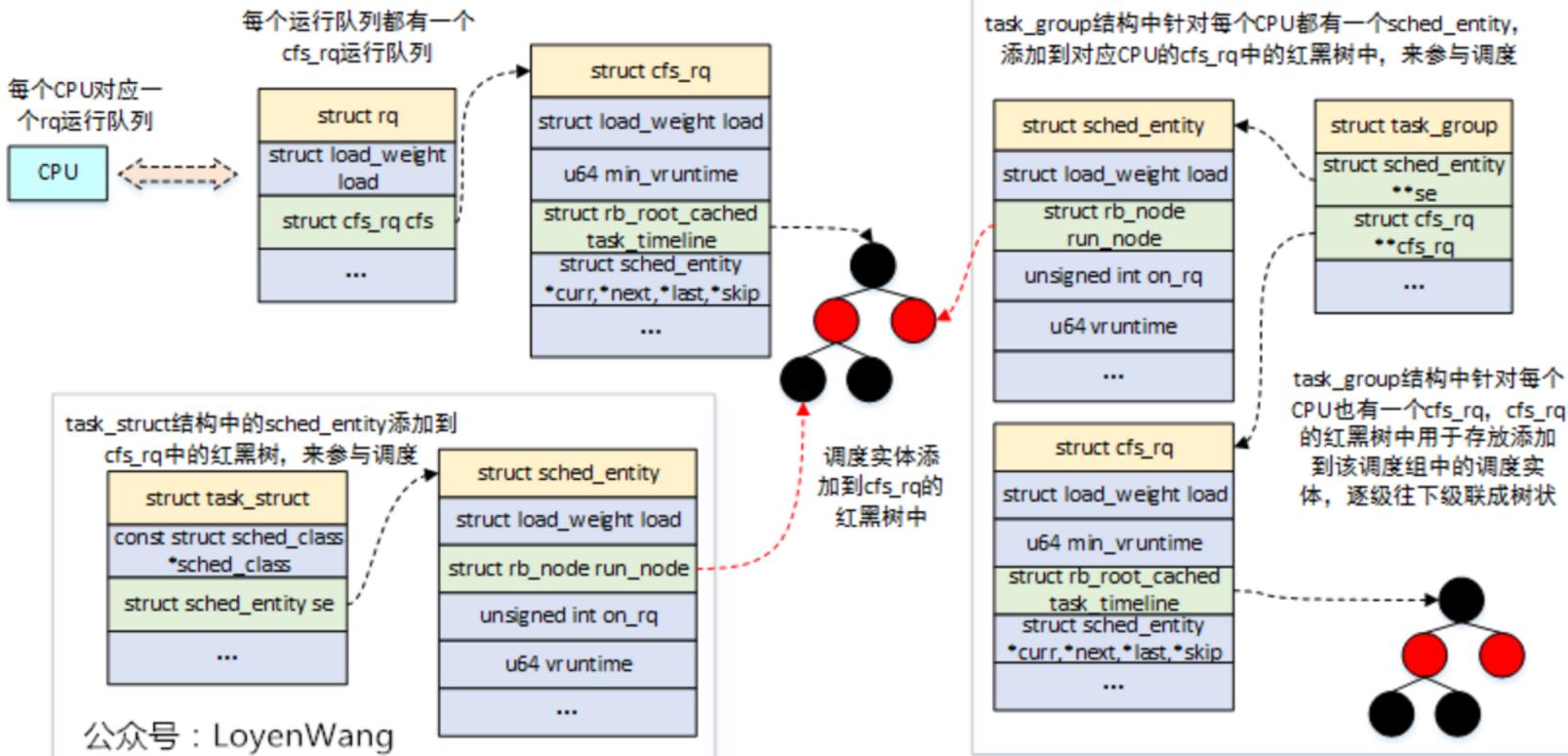
- Nodes on the left have lower vruntime compared to nodes on the right of the tree.

- The left most node is the task with the least vruntime

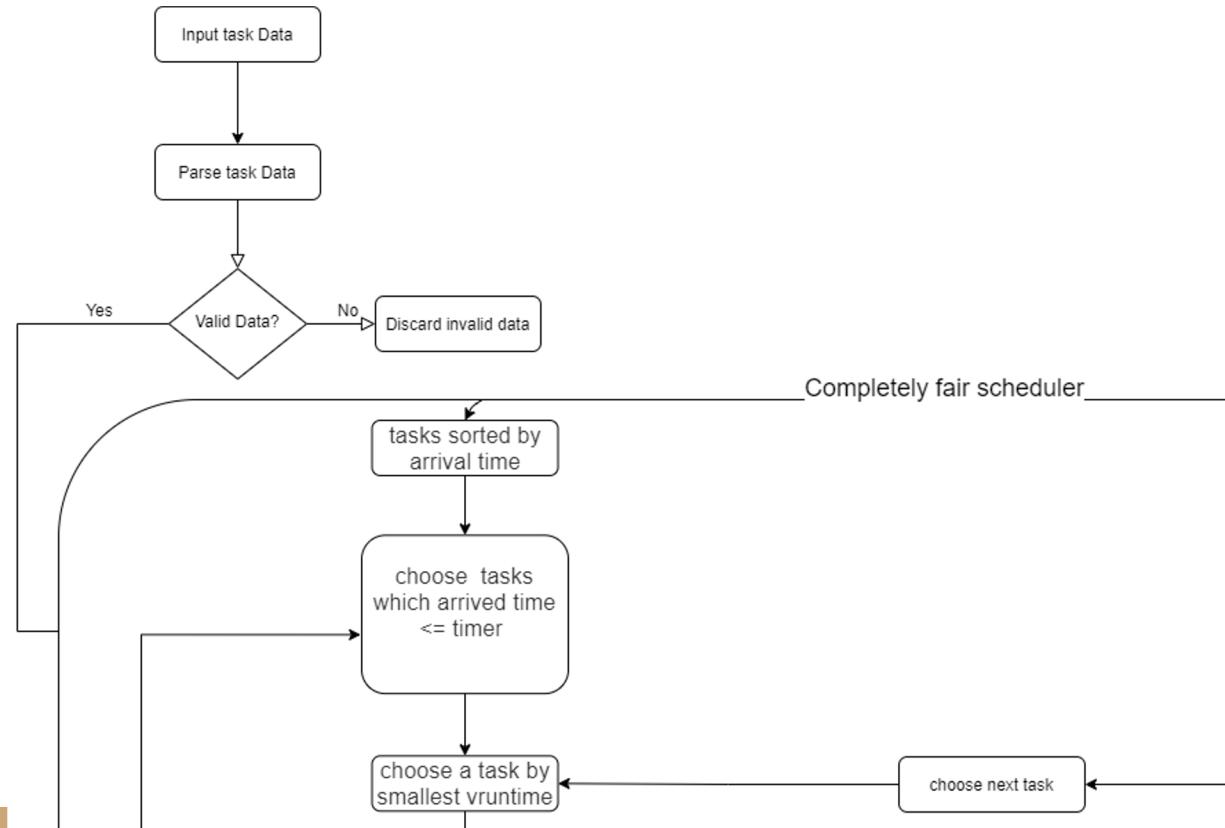


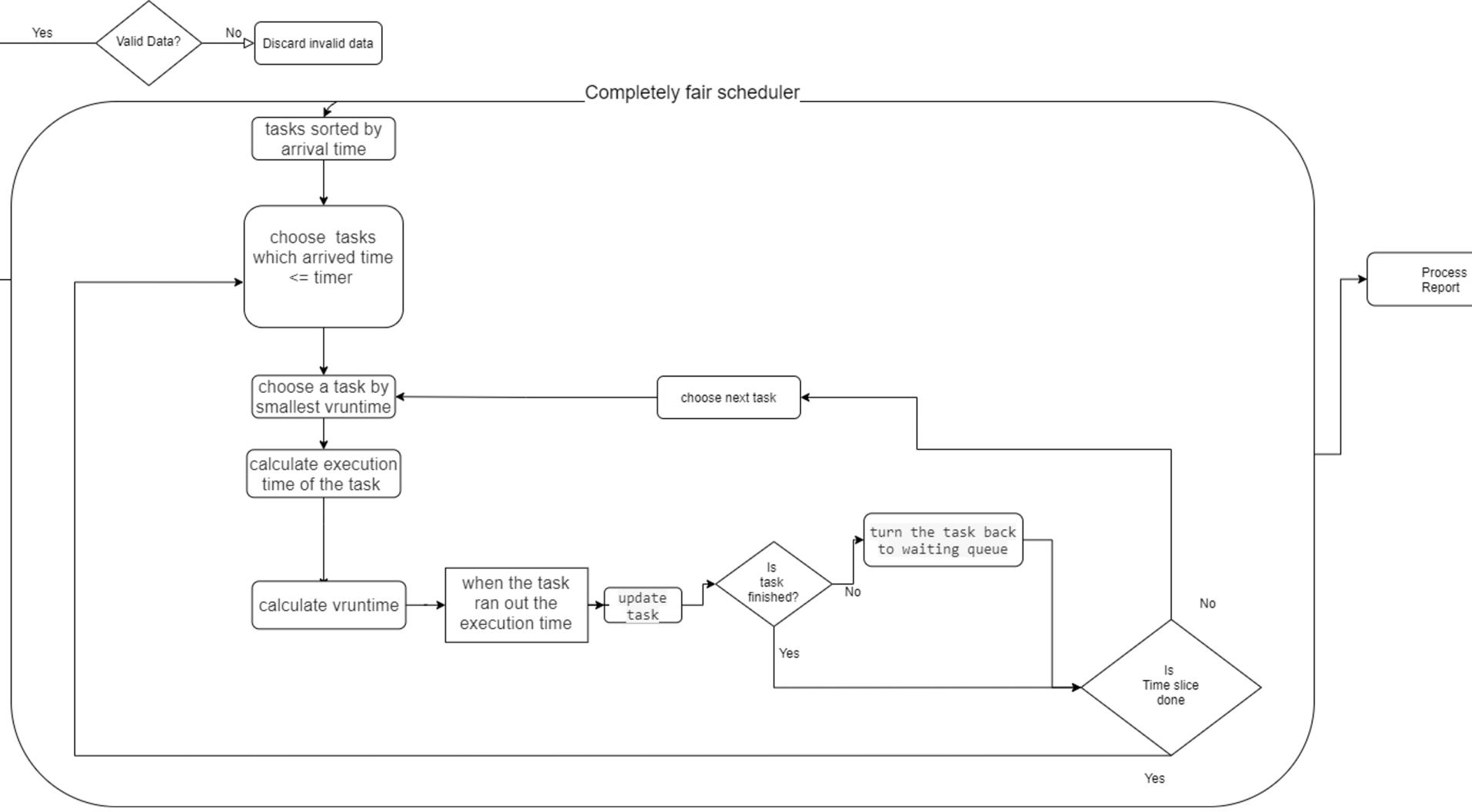
[ref]:<https://developer.ibm.com/tutorials/l-completely-fair-scheduler/>
https://www.youtube.com/watch?v=MkfufI5_hjc

Completely Fair Scheduler Architecture In Linux



Completely Fair Scheduler-flow chart

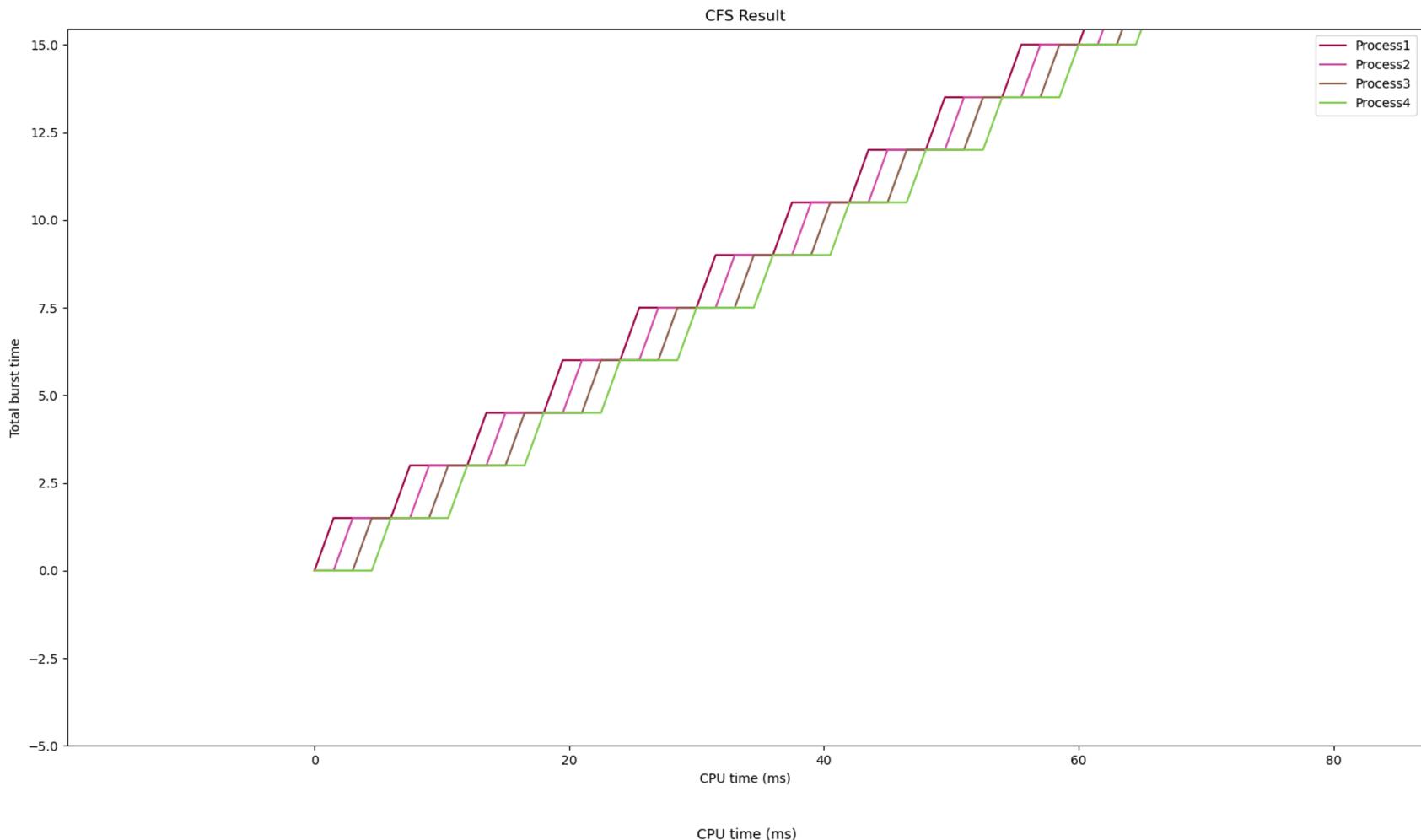




Demo And Validation

TestCase 1: Time Slice = 6ms

Process Id	Arrival Time	Burst Time(ms)	Nice Value
1	0	100	0
2	0	100	0
3	0	100	0
4	0	100	0

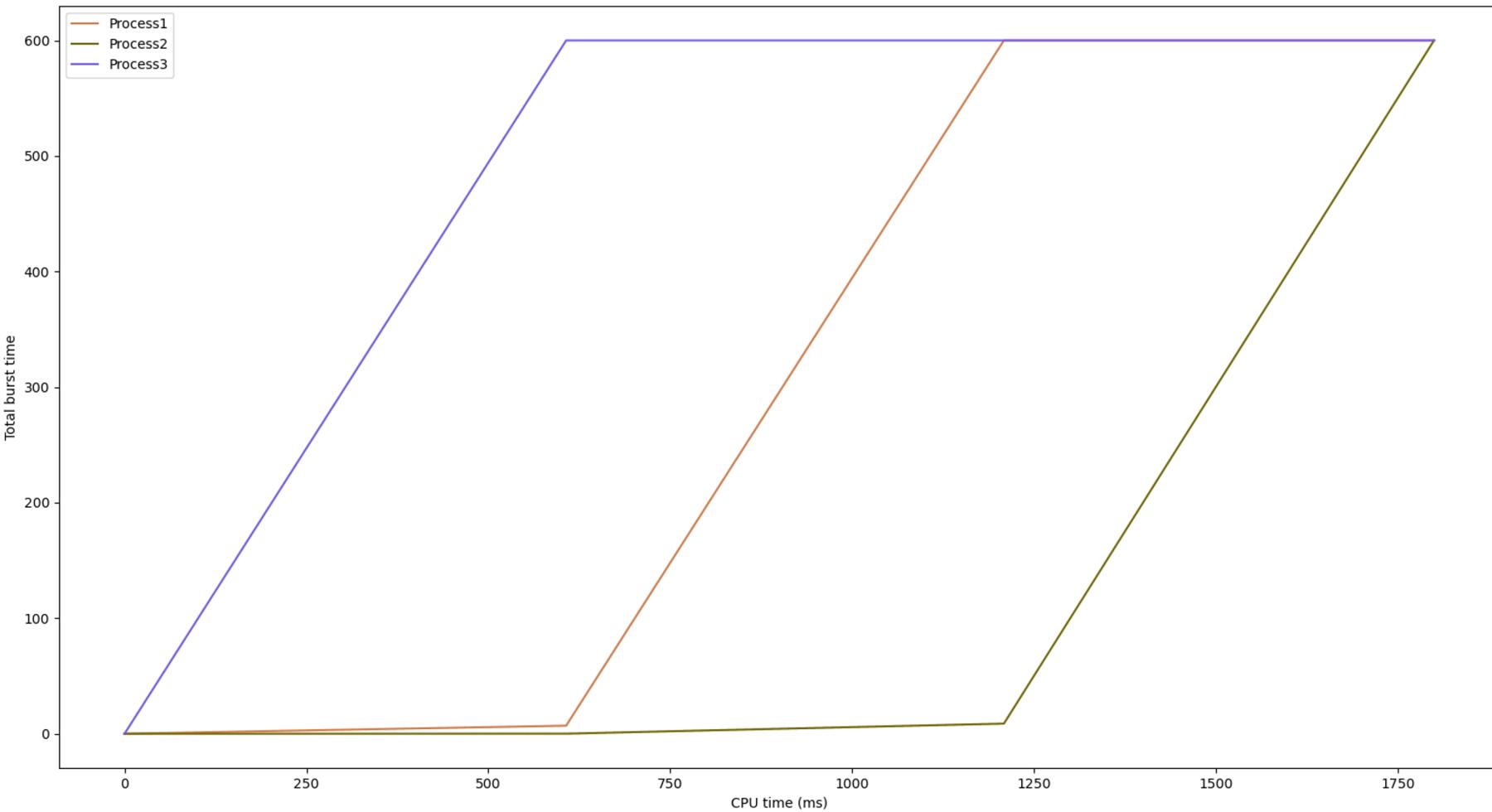


Demo And Validation

TestCase 2 Time Slice = 6ms

Process Id	Arrival Time	Burst Time(ms)	Nice Value
1	0	600	0
2	0	600	19
3	0	600	-20

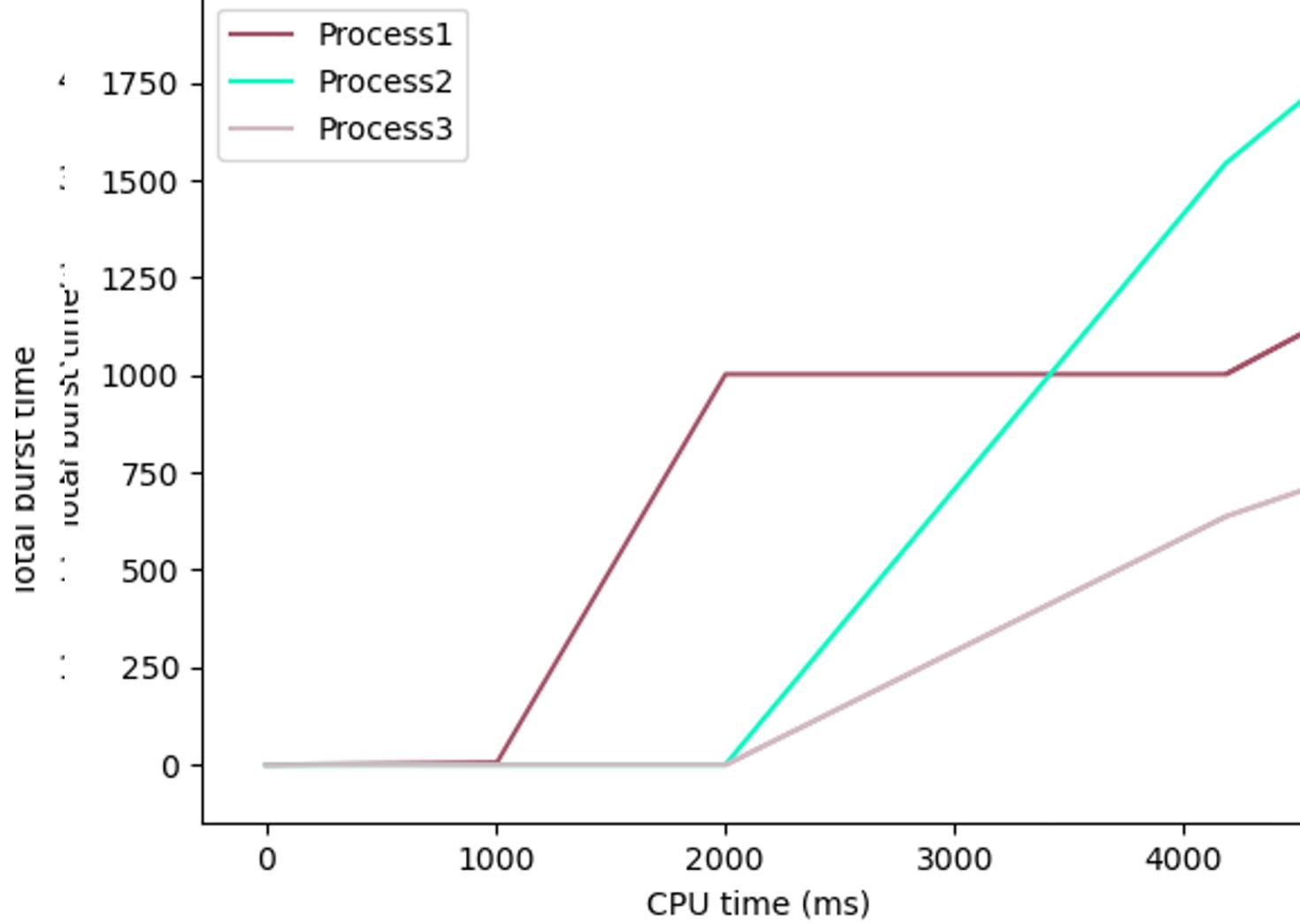
CFS Result



Demo And Validation

TestCase 3 Time Slice = 6ms

Process Id	Arrival Time(ms)	Burst Time(ms)	Nice Value
1	1000	3000	0
2	2000	4000	-2
3	2000	3000	2



Ref

[1]:Linux v4.4 core

[2]:<https://developer.ibm.com/tutorials/l-completely-fair-scheduler/>

[2]:https://www.youtube.com/watch?v=MkJfuI5_hjc

[3]:<https://elixir.free-electrons.com/linux/v4.6/source/include/linux/sched.h>

[4]:<https://stackoverflow.com/questions/33191110/reason-why-cfs-scheduler-using-red-black-tree>

[5]:<https://oakbytes.wordpress.com/2012/06/08/linux-scheduler-cfs-and-red-black-tree/>

[6]:<https://www.cnblogs.com/LoyenWang/p/12495319.html>

[7]:<https://www.slideshare.net/linaroorg/the-linux-kernel-scheduler-for-beginners-sfo17421>