James Peacemaker

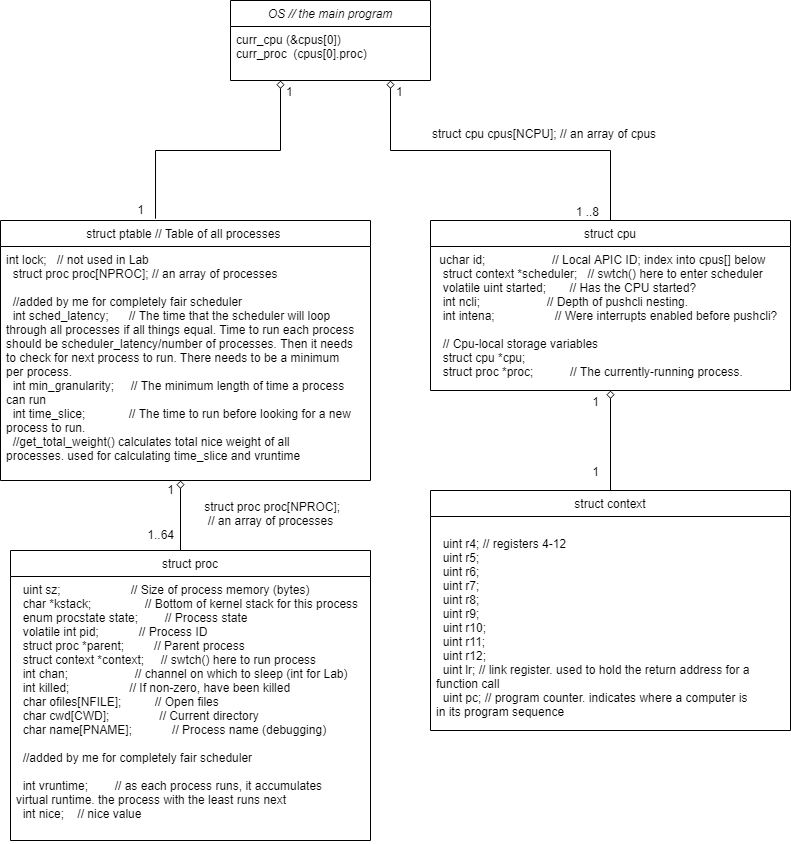
March 13, 2021

CPSC 405

Schedule Project – Answer Document

Submissions

1. Place the design of your implementation in this item. You should study the problem and create a design before trying to modify the code.
   1. Place diagrams of the data structures and how they are related here.



* 1. Place pseudo code of your algorithms and how they affect the data structures here.

// Per-CPU process scheduler.

// it changes current\_proc to the new process that needs to run. the one with the lowest vruntime. It adjust runtime based on nice value

void scheduler(void)

{

if it is time to schedule a new process (time\_slice <= 0)

{

//then schedule a new process

//stop current process from running

if (curr\_proc->state == RUNNING) then curr\_proc->state = RUNNABLE

//set process with the lowest vruntime as current by default

p\_lowest\_vruntime = curr\_proc;

//find process with the lowest vruntime

loop through all processes in ptable

//if current process is not runnable, then ignore it

if (curr\_proc->state == RUNNABLE){

if p(the current)->vruntime < p\_lowest\_vruntime(the record low)  
->vruntime then

set the current to the record low. p\_lowest\_vruntime = p

}

}

}

// Switch to chosen process and make it running if possible.

curr\_proc = p\_lowest\_vruntime;

if curr\_proc->state == RUNNABLE then curr\_proc->state = RUNNING

//set new time\_slice value

ptable.sched\_latency = 48;

ptable.min\_granularity = 6;

time\_slice = sched\_latency \* (proportionate share of total weight depending on nice value) or min\_granularity, whichever is greater

}

//else if there is still time left in the timeslice to run, reduce time\_slice by one

else

{

if (curr\_proc->state == RUNNING) //if only process available is asleep, don't change vruntime or timeslice

{

//when time\_slice still has some time left for this proc, just keep proc instead of scheduling a new one

ptable.time\_slice--

//add vruntime using nice

curr\_proc->vruntime = curr\_proc->vruntime + (total nice weight - proportionate share of nice weight);

}

}

}

* 1. Place a list of source files in the code base that you updated.

I updated defs.h, main.c, proc.h, proc.c for the completely fair scheduler and the lottery scheduler.

1. Place your test cases here. Your collection of test cases should demonstrate you have tested the entire program. You will have a collection of test cases for each of your schedulers. For each test case, first describe the objective of your test case. Then insert a copy/paste of you applying your test cases to your program. The copy/paste must demonstrate that actual output achieved the described objective.

**Completely Fair Scheduler**

Basic time slice:

It runs on a 48ms time slice by default.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

jpeacemakr@shell 2> timer 49

Running scheduler.

Time slice: 48

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Time slice: 48

Scheduler selected pid: 1

jpeacemakr@shell 3> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

jpeacemakr@shell 4> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

Test fork

Each time a process is forked, it divides the timeslice of 48 evenly. A timeslice of 6 is the minimum a process will run.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

jpeacemakr@shell 2> fork

Running scheduler.

Time slice: 24

pid: 1 forked: 2

jpeacemakr@shell 3> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 4> fork

Running scheduler.

Time slice: 16

pid: 1 forked: 3

jpeacemakr@shell 5> fork

Running scheduler.

Time slice: 12

pid: 1 forked: 4

jpeacemakr@shell 6> timer 49

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Time slice: 12

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Time slice: 12

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Time slice: 12

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Running scheduler.

Time slice: 12

Scheduler selected pid: 4

Scheduler selected pid: 4

jpeacemakr@shell 7> ps

pid: 1, parent: 0, nice: 0, vruntime: 33792, state: RUNNABLE

pid: 2, parent: 1, nice: 0, vruntime: 33792, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 33792, state: RUNNABLE

pid: 4, parent: 1, nice: 0, vruntime: 36864, state: RUNNING

jpeacemakr@shell 8> fork

Running scheduler.

Time slice: 9

pid: 4 forked: 5

jpeacemakr@shell 9> fork

Running scheduler.

Time slice: 8

pid: 1 forked: 6

jpeacemakr@shell 10> fork

Running scheduler.

Time slice: 6

pid: 1 forked: 7

jpeacemakr@shell 11> fork

Running scheduler.

Time slice: 6

pid: 1 forked: 8

jpeacemakr@shell 12> timer 48

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Time slice: 6

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Time slice: 6

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Time slice: 6

Scheduler selected pid: 6

Scheduler selected pid: 6

Scheduler selected pid: 6

Scheduler selected pid: 6

Scheduler selected pid: 6

Scheduler selected pid: 6

Running scheduler.

Time slice: 6

Scheduler selected pid: 7

Scheduler selected pid: 7

Scheduler selected pid: 7

Scheduler selected pid: 7

Scheduler selected pid: 7

Scheduler selected pid: 7

Running scheduler.

Time slice: 6

Scheduler selected pid: 8

Scheduler selected pid: 8

Scheduler selected pid: 8

Scheduler selected pid: 8

Scheduler selected pid: 8

Scheduler selected pid: 8

Running scheduler.

Time slice: 6

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Scheduler selected pid: 4

Running scheduler.

Time slice: 6

Scheduler selected pid: 5

Scheduler selected pid: 5

Scheduler selected pid: 5

Scheduler selected pid: 5

Scheduler selected pid: 5

Scheduler selected pid: 5

Running scheduler.

Time slice: 6

Scheduler selected pid: 1

jpeacemakr@shell 13> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

Test sleep

When a process is sleeping, it is not factored into the timeslice.

When it wakes up, it is given the same vruntime as the currently running process.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

Time slice: 24

pid: 1 forked: 2

jpeacemakr@shell 2> fork

Running scheduler.

Time slice: 16

pid: 1 forked: 3

jpeacemakr@shell 3> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 4> timer 5

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 5> ps

pid: 1, parent: 0, nice: 0, vruntime: 10240, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 6> sleep 1 1

Running scheduler.

Time slice: 24

Sleep Status:: 1 .

jpeacemakr@shell 7> timer 49

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Time slice: 24

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Time slice: 24

Scheduler selected pid: 3

Scheduler selected pid: 3

jpeacemakr@shell 8> ps

pid: 1, parent: 0, nice: 0, vruntime: 10240, state: SLEEPING

pid: 2, parent: 1, nice: 0, vruntime: 23552, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 24576, state: RUNNING

jpeacemakr@shell 9> wakeup 1

Running scheduler.

Time slice: 16

jpeacemakr@shell 10> ps

pid: 1, parent: 0, nice: 0, vruntime: 24576, state: RUNNABLE

pid: 2, parent: 1, nice: 0, vruntime: 23552, state: RUNNING

pid: 3, parent: 1, nice: 0, vruntime: 24576, state: RUNNABLE

jpeacemakr@shell 11> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

Test sleep all

It just shows you that the scheduler says no runnable processes when all processes are sleeping. vruntime is reset when sleeping. It is set to current process vruntime when it wakes up. They are all reset to 0 vruntime if all processes are sleeping.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

jpeacemakr@shell 2> fork

Running scheduler.

Time slice: 24

pid: 1 forked: 2

jpeacemakr@shell 3> timer 5

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 4> ps

pid: 1, parent: 0, nice: 0, vruntime: 5120, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 5> sleep 1 1

Running scheduler.

Time slice: 48

Sleep Status:: 1 .

jpeacemakr@shell 6> ps

pid: 1, parent: 0, nice: 0, vruntime: 5120, state: SLEEPING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNING

jpeacemakr@shell 7> sleep 2 2

Running scheduler.

Sleep Status:: 2 .

jpeacemakr@shell 8> ps

pid: 1, parent: 0, nice: 0, vruntime: 5120, state: SLEEPING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: SLEEPING

jpeacemakr@shell 9> timer 5

Running scheduler.

No runnable processes.

Running scheduler.

No runnable processes.

Running scheduler.

No runnable processes.

Running scheduler.

No runnable processes.

Running scheduler.

No runnable processes.

jpeacemakr@shell 10> wakeup 1

Running scheduler.

Time slice: 48

jpeacemakr@shell 11> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: SLEEPING

jpeacemakr@shell 12> timer 5

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 13> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: SLEEPING

jpeacemakr@shell 14> wakeup 2

Running scheduler.

Time slice: 24

jpeacemakr@shell 15> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 16> timer 5

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 17> ps

pid: 1, parent: 0, nice: 0, vruntime: 5120, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 18> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

Test exit

When a process is exited, it and its children get a ZOMBIE state. The scheduler is called, and they are then ignored by the scheduler.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

Time slice: 24

pid: 1 forked: 2

jpeacemakr@shell 2> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 3> timer 34

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Time slice: 24

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 4> fork

Running scheduler.

Time slice: 16

pid: 2 forked: 3

jpeacemakr@shell 5> ps

pid: 1, parent: 0, nice: 0, vruntime: 23552, state: RUNNABLE

pid: 2, parent: 1, nice: 0, vruntime: 10240, state: RUNNING

pid: 3, parent: 2, nice: 0, vruntime: 10240, state: RUNNABLE

jpeacemakr@shell 6> exit

Exit Status:: 0 .

jpeacemakr@shell 7> ps

pid: 1, parent: 0, nice: 0, vruntime: 23552, state: RUNNABLE

pid: 2, parent: 1, nice: 0, vruntime: 10240, state: ZOMBIE

pid: 3, parent: 1, nice: 0, vruntime: 10240, state: ZOMBIE

jpeacemakr@shell 8> timer 5

Running scheduler.

Time slice: 48

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 9> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

Test nice

The nice command changes how often a process runs. Lower nice value runs more. When the nice command is used, the scheduler finished the current time slice before you see the change.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

Time slice: 24

pid: 1 forked: 2

jpeacemakr@shell 2> fork

Running scheduler.

Time slice: 16

pid: 1 forked: 3

jpeacemakr@shell 3> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 4> nice 2 -10

Nice value set.

jpeacemakr@shell 5> ps

pid: 1, parent: 0, nice: 0, vruntime: 0, state: RUNNING

pid: 2, parent: 1, nice: -10, vruntime: 0, state: RUNNABLE

pid: 3, parent: 1, nice: 0, vruntime: 0, state: RUNNABLE

jpeacemakr@shell 6> timer 48

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Time slice: 39

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 7> timer 48

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Time slice: 6

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Time slice: 6

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Time slice: 39

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 8> ps

pid: 1, parent: 0, nice: 0, vruntime: 158580, state: RUNNABLE

pid: 2, parent: 1, nice: -10, vruntime: 137216, state: RUNNING

pid: 3, parent: 1, nice: 0, vruntime: 105720, state: RUNNABLE

jpeacemakr@shell 9> nice 1 20

Pid must exist. Nice value must be -20 (high priority) to 19 (low priority).

Error setting nice value.

jpeacemakr@shell 10> nice 1 -21

Pid must exist. Nice value must be -20 (high priority) to 19 (low priority).

Error setting nice value.

jpeacemakr@shell 11> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/FairScheduler$

**Lottery Scheduler**

Test fork

Time slice is set to 10 and scheduling is random among running and runnable processes.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$ ./procprog

jpeacemakr@shell 1> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

jpeacemakr@shell 2> fork

Running scheduler.

pid: 1 forked: 2

jpeacemakr@shell 3> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 4> fork

Running scheduler.

pid: 1 forked: 3

jpeacemakr@shell 5> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

pid: 3, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 6> timer 48

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 7> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

pid: 3, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 8> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$

Test sleep and wakeup

Sleeping processes are not considered when picking a lottery winner.

Scheduler is reset when sleep or wakeup command is used.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

pid: 1 forked: 2

jpeacemakr@shell 2> ps

pid: 1, parent: 0, tickets: 10, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNING

jpeacemakr@shell 3> sleep 1 1

Running scheduler.

Sleep Status:: 1 .

jpeacemakr@shell 4> ps

pid: 1, parent: 0, tickets: 10, state: SLEEPING

pid: 2, parent: 1, tickets: 10, state: RUNNING

jpeacemakr@shell 5> fork

Running scheduler.

pid: 2 forked: 3

jpeacemakr@shell 6> ps

pid: 1, parent: 0, tickets: 10, state: SLEEPING

pid: 2, parent: 1, tickets: 10, state: RUNNING

pid: 3, parent: 2, tickets: 10, state: RUNNABLE

jpeacemakr@shell 7> timer 32

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 8> wakeup 1

Running scheduler.

jpeacemakr@shell 9> timer 36

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

jpeacemakr@shell 10> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$

Test sleep all

This test puts all processes to sleep. Scheduler displays no runnable processes.

Then wakeup processes to make sure they still can work.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

pid: 1 forked: 2

jpeacemakr@shell 2> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 3> timer 4

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 4> sleep 1 1

Running scheduler.

Sleep Status:: 1 .

jpeacemakr@shell 5> ps

pid: 1, parent: 0, tickets: 10, state: SLEEPING

pid: 2, parent: 1, tickets: 10, state: RUNNING

jpeacemakr@shell 6> timer 4

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 7> sleep 2 2

Running scheduler.

No runnable processes.

Sleep Status:: 2 .

jpeacemakr@shell 8> timer 4

No runnable processes.

No runnable processes.

No runnable processes.

No runnable processes.

jpeacemakr@shell 9> wakeup 1

Running scheduler.

jpeacemakr@shell 10> timer 4

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 11> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$

Test exit

Exit turns process and its children into zombies.

Zombies are ignored by the scheduler.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

pid: 1 forked: 2

jpeacemakr@shell 2> timer 5

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

jpeacemakr@shell 3> fork

Running scheduler.

pid: 2 forked: 3

jpeacemakr@shell 4> ps

pid: 1, parent: 0, tickets: 10, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

pid: 3, parent: 2, tickets: 10, state: RUNNING

jpeacemakr@shell 5> exit 2

Running scheduler.

Exit Status:: 0 .

jpeacemakr@shell 6> ps

pid: 1, parent: 0, tickets: 10, state: RUNNING

pid: 2, parent: 1, tickets: 10, state: ZOMBIE

pid: 3, parent: 1, tickets: 10, state: ZOMBIE

jpeacemakr@shell 7> timer 15

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Running scheduler.

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

Scheduler selected pid: 1

jpeacemakr@shell 8> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$

Test tickets

Tickets for each process can be adjusted. The default is 10.

More tickets means that process has a higher chance of being scheduled.

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$ ./procprog

jpeacemakr@shell 1> fork

Running scheduler.

pid: 1 forked: 2

jpeacemakr@shell 2> fork

Running scheduler.

pid: 1 forked: 3

jpeacemakr@shell 3> ps

pid: 1, parent: 0, tickets: 10, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNING

pid: 3, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 4> tickets 1 5

Tickets value set.

jpeacemakr@shell 5> ps

pid: 1, parent: 0, tickets: 5, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNING

pid: 3, parent: 1, tickets: 10, state: RUNNABLE

jpeacemakr@shell 6> tickets 3 20

Tickets value set.

jpeacemakr@shell 7> ps

pid: 1, parent: 0, tickets: 5, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNING

pid: 3, parent: 1, tickets: 20, state: RUNNABLE

jpeacemakr@shell 8> tickets 4 4

Pid must exist. Tickets must not be less than zero.

Error setting tickets value.

jpeacemakr@shell 9> timer 35

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Running scheduler.

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Scheduler selected pid: 2

Running scheduler.

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

Scheduler selected pid: 3

jpeacemakr@shell 10> ps

pid: 1, parent: 0, tickets: 5, state: RUNNABLE

pid: 2, parent: 1, tickets: 10, state: RUNNABLE

pid: 3, parent: 1, tickets: 20, state: RUNNING

jpeacemakr@shell 11> quit

jpeacemakr@MyPrecious:/mnt/c/Users/Bettina/Desktop/cpsc405/ScheduleProject/LotteryScheduler$

1. On Canvas submit the code (.c, .h, and makefile) for your two scheduling algorithms. You must submit all code, even code that you did not modify. Your modified source files must include comments in the code that describes your modifications. If you implement two programs - one for each scheduler, you should submit two zip files - one for each solution.