

CSC 343 – Operating Systems, Fall 2020, Assignment 4, due December 3

This assignment is **DUE By 11:59 PM on Thursday December 3, 2020** via **make turnitin** on **mcgonagall** or **acad**. The standard 10% per day deduction for late assignments applies.

To get the starting code for the project please follow these steps after logging into acad:

```
cd                # This goes to your login directory.
mkdir ./OpSys    # should already be there; no error if it says so
cd ./OpSys
cp ~parson/OpSys/Paging2020.problem.zip Paging2020.problem.zip
unzip Paging2020.problem.zip
cd ./Paging2020
ssh -l YOURLOGIN mcgonagall      # -l is the lower-case letter ell
cd ./OpSys/Paging2020
```

All of your programming and testing must occur on multiprocessor **mcgonagall**. All other work must occur within your `OpSys/Paging2020` directory on **mcgonagall**.

In this assignment I am supplying a least-recently-used page replacement algorithm in file **rr_lrupage.stm**. You must fill in the comment block at the top and run **make testlru** to test it. You can then

```
cp rr_lrupage.stm rr_fifopage.stm
```

Remove all references to macro **reinsertInVictimQueue** in **rr_fifopage.stm** because FIFO page replacement does not reorder the **pcb.victimQueue** used to select a replacement page. After you remove this queue-reordering macro you can run **make testfifo** to test it. A correctly working **rr_fifopage.stm** is worth 30% of the assignment. Make sure to update comments at the top and where you make code changes to reflect FIFO page replacement.

Next, you must

```
cp rr_lrupage.stm rr_lrupage_dirty.stm
```

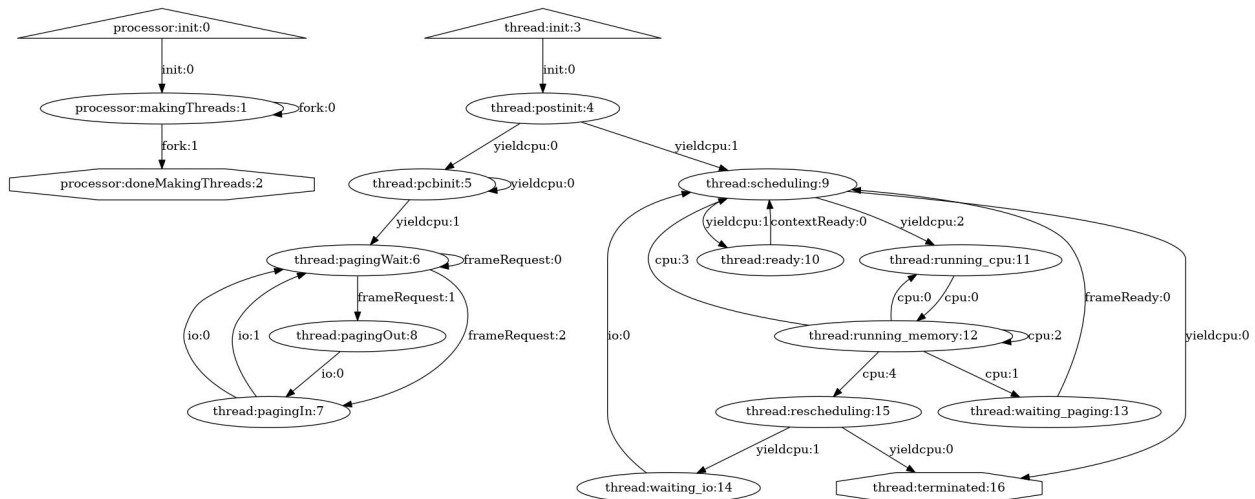
Then change the **pcb.victimQueue** constructor call to set the **ispriority** parameter to **True**. Do not change constructor calls for other **Queue** objects in this STM. Also you must change every **pcb.victimQueue.enqueue** to add the dirty bit from the **pcb.pagetable** as the second argument to **enqueue**. A clean page with a dirty bit of 0 sorts closer to the front of the **victimQueue**. You can test using **make testdirty**. A correctly working **rr_lrupage_dirty.stm** is worth 30% of the assignment. Make sure to update comments at the top and where you make code changes to reflect LRU-dirty page replacement.

After **make clean test** works correctly, **edit the README.txt file** and answer the 4 questions, each worth 10% of the project, bringing the total to 100%.

When everything runs, make sure your name is added at the top of your source files, and you have added a brief comment for every transition that you change or add. Perform **make clean test** one last time, and then **make turnitin** by the due date deadline. We will go over this assignment on November 16.

All three STMs have the same state-transition graph.

https://kuvapcsitrd01.kutztown.edu/~parson/rr_lrupage.jpg



Here is what a successful test run looks like. You can start by running **make clean test** on the handout code. Running `make graphs` plots state machine diagrams into your `public_html/` directory as before.

\$ make clean test

```
make clean test
```

```
/bin/rm -f *.o *.class .jar core *.exe *.obj *.pyc __pycache__/*.*pyc
```

```
/bin/rm -f *.out *.dif *.pyc junk parsetab.py *.vmlf
```

```
/bin/rm -f *.dot *.gif *.jpg testmachine.ck junk.* *.tmp *.log
```

```
/bin/rm -f *.crunch
```

```
/bin/rm -f /home/kutztown.edu/parson/tmp/parson_STM_*.log parson_STM_*.log __pycache__/*
```

```
/bin/rm -f rr_fifo_page_crunch.py rr_lrpage_crunch.py rr_lrpage_dirty_crunch.py
```

```
/bin/rm -f rr_fifo_page.py rr_lrpage_dirty.py rr_lrpage.py
```

```
COMPILING fifo
```

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:...\ /usr/local/bin/python3.7
```

```
/home/kutztown.edu/parson/OpSys/state2codeV17/State2CodeParser.py rr_fifo_page.stm rr_fifo_page.dot
```

```
rr_fifo_page.py CSC343Compile CSC343Compile"
```

```
INFO: Blocking function spawn is in mid-transition from thread.pcbinit -> pcbinit, so its completion event will not trigger a state change.
```

```
COMPILING COMPLETED
```

```
SIMULATING (TESTING) fifo
```

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:...
```

```
STMLOGDIR=/home/kutztown.edu/parson/tmp time /usr/local/bin/python3.7 rr_fifo_page.py 2 4 110000 12345 2"
```

```
MSG cmd line: ['rr_fifo_page.py', '2', '4', '110000', '12345', '2'], usage USAGE: python THISFILE.py NUMCONTEXTS NUMFASTIO SIMTIME SEED|None LOGLEVEL
```

```
Simulation exiting at time 110166, which meets or exceeds limit 110000.
```

```
0.26user 0.03system 0:00.41elapsed 72%CPU (0avgtext+0avgdata 12000maxresident)k
```

```
0inputs+2320outputs (0major+5829minor)pagefaults 0swaps
```

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:...\ /usr/local/bin/python3.7
```

```
crunchlog.py rr_fifo_page.log"
```

[DIFFing rr_fifo_page_crunch.py rr_fifo_page_crunch.ref](#)

OK: MEAN_waiting_paging at 8.0% tolerance, value: 28853.5
OK: MAX_waiting_paging at 8.0% tolerance, value: 59494.0

TESTING COMPLETED

COMPILING lru

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:... /usr/local/bin/python3.7  
/home/kutztown.edu/parson/OpSys/state2codeV17/State2CodeParser.py rr_lrupage.stm rr_lrupage.dot  
rr_lrupage.py CSC343Compile CSC343Compile"
```

INFO: Blocking function spawn is in mid-transition from thread.pcbinit -> pcbinit, so its completion event will not trigger a state change.

COMPILING COMPLETED

SIMULATING (TESTING) lru

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:...  
STMLOGDIR=/home/kutztown.edu/parson/tmp time /usr/local/bin/python3.7 rr_lrupage.py 2 4 110000  
12345 2"
```

MSG cmd line: ['rr_lrupage.py', '2', '4', '110000', '12345', '2'], usage USAGE: python THISFILE.py
NUMCONTEXTS NUMFASTIO SIMTIME SEED[None LOGLEVEL
Simulation exiting at time 110166, which meets or exceeds limit 110000.

0.29user 0.01system 0:00.42elapsed 73%CPU (0avgtext+0avgdata 12012maxresident)k
0inputs+2528outputs (0major+5831minor)pagefaults 0swaps

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:... /usr/local/bin/python3.7  
crunchlog.py rr_lrupage.log"
```

DIFFing rr_lrupage_crunch.py rr_lrupage_crunch.ref

OK: MEAN_waiting_paging at 8.0% tolerance, value: 28117.0
OK: MAX_waiting_paging at 8.0% tolerance, value: 58124.0

TESTING COMPLETED

COMPILING rr_lrupage_dirty

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:... /usr/local/bin/python3.7  
/home/kutztown.edu/parson/OpSys/state2codeV17/State2CodeParser.py rr_lrupage_dirty.stm  
rr_lrupage_dirty.dot rr_lrupage_dirty.py CSC343Compile CSC343Compile"
```

INFO: Blocking function spawn is in mid-transition from thread.pcbinit -> pcbinit, so its completion event will not trigger a state change.

COMPILING COMPLETED

SIMULATING (TESTING) rr_lrupage_dirty

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:...  
STMLOGDIR=/home/kutztown.edu/parson/tmp time /usr/local/bin/python3.7 rr_lrupage_dirty.py 2 4  
110000 12345 2"
```

MSG cmd line: ['rr_lrupage_dirty.py', '2', '4', '110000', '12345', '2'], usage USAGE: python THISFILE.py
NUMCONTEXTS NUMFASTIO SIMTIME SEED[None LOGLEVEL
Simulation exiting at time 110166, which meets or exceeds limit 110000.

0.29user 0.02system 0:00.42elapsed 76%CPU (0avgtext+0avgdata 12016maxresident)k
0inputs+2632outputs (0major+5837minor)pagefaults 0swaps

```
/bin/bash -c "PYTHONPATH=/home/kutztown.edu/parson/OpSys:... /usr/local/bin/python3.7  
crunchlog.py rr_lrupage_dirty.log"
```

DIFFing rr_lrupage_dirty_crunch.py rr_lrupage_dirty_crunch.ref

OK: MEAN_waiting_paging at 8.0% tolerance, value: 24062.4

OK: MAX_waiting_paging at 8.0% tolerance, value: 52395.0

TESTING COMPLETED

Each test run produces a log file (rr_fifopage.log, rr_lrupage.log, rr_lrupage_dirty.log).

Automated testing via **make clean test** is similar to assignment 3. Simulation times in ticks for critical states of the algorithm are checked for consistency with the expected times, to with a 8% allowable margin of difference. See **highlighted-in-red** lines above.

If you get an error at run time with codeTable index like this:

```
exec(__codeTable__[20],globals,locals)
```

You can run decode.py like this:

```
$ python decode.py rr_fifopage.py 20
```

```
__codeTable__[20] = compile('pcb.victimQueue = Queue(ispriority=False)','nofile','exec'),
```