Page Replacement Simulation Web App

Charles Billingsley

Data Structures:

For my project, I used many different data structures to get my code up and running. I kept these structures inside of my controller, and then on any change, I would pass them into my model to update my view.

Memory References:

The memory references are kept inside of an array that is filled in once when the file loads. I then loop through each one at a time during the simulation.

Current Order:

The current order was kept in an array. Whenever a process and page number was called that was not in the array, it would be appended to the end, or most recent spot. If it was in the array already, it would be removed and appended to the end. This kept the order of use for the LRU algorithm.

PCB:

The PCB information was stored in a JSON object with the format {'P1': 1}. The key was the process name and the value was the processes page table number. This was sufficient as each process would only need one page table. It is also scalable as if I ever would like to add another category to the table, I would just make the page table number into an array of values.

Page Tables:

The page tables were my most complex data structure. In order for my web page to stay dynamic, but still keep all of the different information inside of a table, I had to nest my data deep into this data structure. It ended up being a JSON object with keys of each process name. Then the values were a list containing JSON objects for each row of the table. These JSON objects held the keys of "va" holding the value of the virtual page, and "pa" holding the physical frame. This was then looped over on my front end with KnockoutJS to display the correct results.

Free Frame List:

The free frame list was a simple array of size 16 holding each of the numbers 0-15 to display for the frames. When a frame was used I would add it to the used frame list that would update my view to color each gray.

Physical Memory:

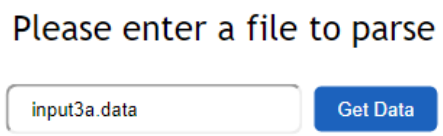
Physical memory was also an array of size 16. The index of this array represented the frame and could be used to update the string when referenced.

Operation:

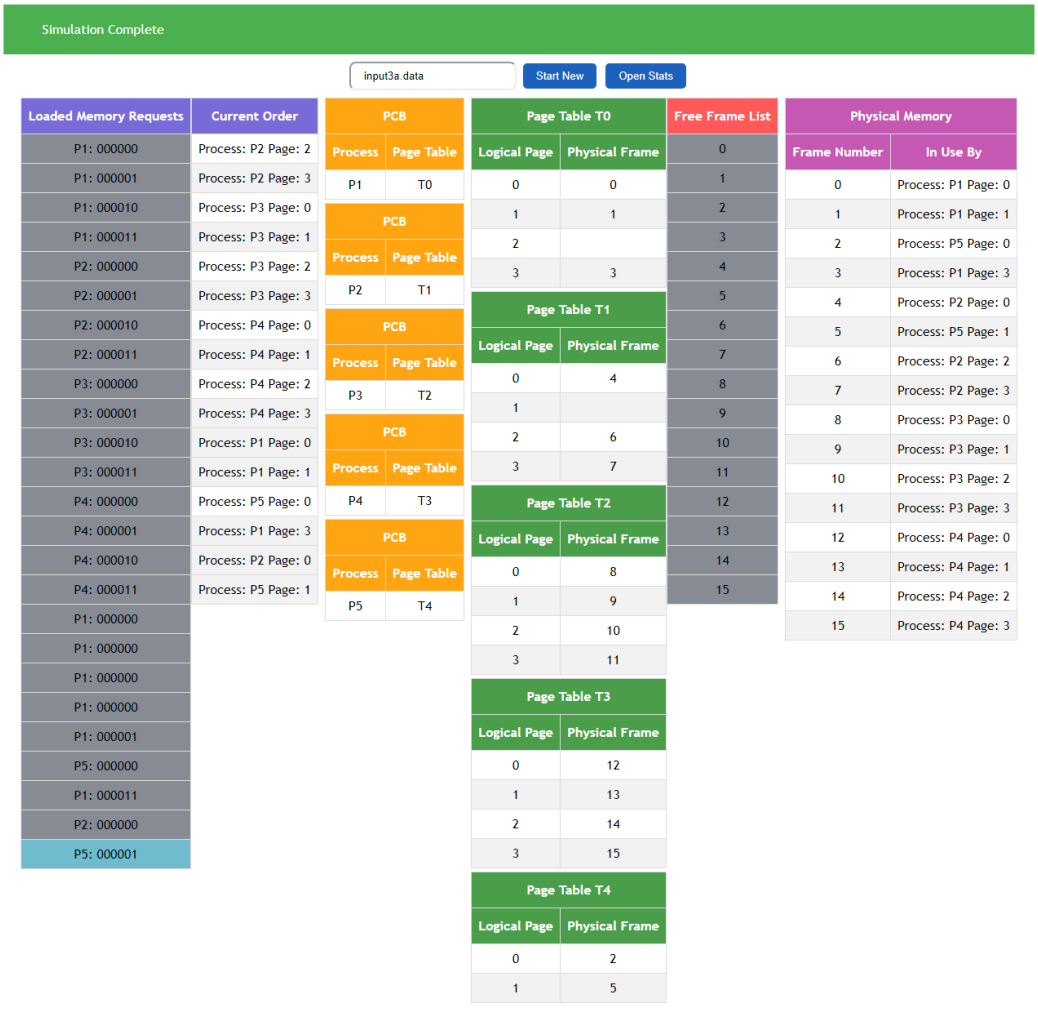
In order to run the simulation, one must first enter a file that is in the server's 'inputs' folder. An invalid file will display an error message. This step may be repeated at any time to start a new simulation by pressing 'Start New' next to the top input. Once the simulation starts, one can choose from three options to move forward. 'Step Forward' will highlight each step of the process and show the user a verbose breakdown of each piece. 'Next Fault' will move the user to the next page fault and will allow the user to continue through normally. 'Complete' will run the simulation to completion and display all tables in their final state. At any time, the user may press the 'Open Stats' button the see each processes stats. All three buttons also work on this page.

Sample Output:

Opening screen:



Completion:



Loaded Memory Requests	Current Order
P1: 000000	Process: P2 Page: 2
P1: 000001	Process: P2 Page: 3
P1: 000010	Process: P3 Page: 0
P1: 000011	Process: P3 Page: 1
P2: 000000	Process: P3 Page: 2
P2: 000001	Process: P3 Page: 3
P2: 000010	Process: P4 Page: 0
P2: 000011	Process: P4 Page: 1
P3: 000000	Process: P4 Page: 2
P3: 000001	Process: P4 Page: 3
P3: 000010	Process: P1 Page: 0
P3: 000011	Process: P1 Page: 1
P4: 000000	Process: P5 Page: 0
P4: 000001	Process: P1 Page: 3
P4: 000010	Process: P2 Page: 0
P4: 000011	Process: P5 Page: 1
P1: 000000	
P1: 000000	
P1: 000000	
P1: 000000	
P1: 000001	
P5: 000000	
P1: 000011	
P2: 000000	
P5: 000001	

PCB	
Process	Page Table
P1	T0

PCB	
Process	Page Table
P2	T1

PCB	
Process	Page Table
P3	T2

PCB	
Process	Page Table
P4	T3

PCB	
Process	Page Table
P5	T4

Page Table T0	
Logical Page	Physical Frame
0	0
1	1
2	
3	3

Page Table T1	
Logical Page	Physical Frame
0	4
1	
2	6
3	7

Page Table T2	
Logical Page	Physical Frame
0	8
1	9
2	10
3	11

Page Table T3	
Logical Page	Physical Frame
0	12
1	13
2	14
3	15

Page Table T4	
Logical Page	Physical Frame
0	2
1	5

Free Frame List
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Physical Memory	
Frame Number	In Use By
0	Process: P1 Page: 0
1	Process: P1 Page: 1
2	Process: P5 Page: 0
3	Process: P1 Page: 3
4	Process: P2 Page: 0
5	Process: P5 Page: 1
6	Process: P2 Page: 2
7	Process: P2 Page: 3
8	Process: P3 Page: 0
9	Process: P3 Page: 1
10	Process: P3 Page: 2
11	Process: P3 Page: 3
12	Process: P4 Page: 0
13	Process: P4 Page: 1
14	Process: P4 Page: 2
15	Process: P4 Page: 3

Stats:

Simulation Complete

input3a.data | Start New | Close Stats

Total Page Faults: 18

Process P1 Stats		
Total Size (Pages)	Total Number of Memory References	Total Number of Page Faults
4	10	4

Process P2 Stats		
Total Size (Pages)	Total Number of Memory References	Total Number of Page Faults
4	5	4

Process P3 Stats		
Total Size (Pages)	Total Number of Memory References	Total Number of Page Faults
4	4	4

Process P4 Stats		
Total Size (Pages)	Total Number of Memory References	Total Number of Page Faults
4	4	4

Process P5 Stats		
Total Size (Pages)	Total Number of Memory References	Total Number of Page Faults
2	2	2