Page Replacement Algorithm ReadMe and Test Cases

|  |  |
| --- | --- |
| Document Version | 0.01 |
| Document Status | Initial Draft |
| Issue Date | 2013-15-11 |
| Project | 2 |
| Term | Fall 2013 |
| Course | Operating systems : coen 283-01 |
| Author | Devi Kandasamy |

**Abstract:**

This Test Plan specifies the design of the code and the test cases for the page replacement techniques of First in First out (FIFO), Least Recently Used (LRU), Optimal and Not Frequently Used (NFU).

**Table of Contents**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Explanation of algorithm 2
2. FIFO
3. LRU
4. Optimal
5. NFU
6. Description of Code 3
7. Storage Location
8. Software Used
9. How to execute code?
10. How to parse input?
11. Page Replacement Algorithm
12. FIFO
13. LRU
14. Optimal
15. NFU
16. Output
17. Test Cases 11
18. Junit
19. FIFO
20. LRU
21. Optimal
22. NFU

# **Explanation of Algorithm**

1. **First in First Out**

The first inserted value should be removed first from the reference string

1. **Least Recently Used**

The value that is recently used stays on the page frame and the new value is replaced with the value not recently used

1. **Optimal**

The occurrence of the values in the reference string is calculated and the farthest value is replaced with new value

1. **Not Frequently Used**

The count of the occurrence of the value in string is incremented each time it occurs when it is needed to be added to the page frame. The new value is replaced with the value of least count.

# **Description of Code**

## **Storage Location**

1. Unzip the folder Project2.zip. The following files are stored under the folder,

Under Project2

* ReadMe.docx

Under Project2/SourceCode

* PageReplaceAlgo.java
* fifo.java
* lru.java
* optimal.java
* nfu.java
* fifoTest.java
* lruTest.java
* NfuTest.java
* OptimalTest.java
* AllTests.java

under folder project2/SampleInputs

* TestInput\_fifo.txt
* TestInput\_lru.txt
* TestInput\_nfu.txt
* TestInput\_optimal.txt
* Output\_fifo.txt
* Output\_lru.txt
* Output\_optimal.txt
* Output\_nfu.txt

Note: The output\_\*.txt[viewable in proper format on internet explorer and notepad++]

1. **Software Used**
2. Language Used: Java
3. Application: Eclipse and Junit
4. **How to execute code?**
5. Import project2 on eclipse in a pageReplaceAlgo project folder under pageReplaceAlgoTest package
6. Copy all the files to a **same** package
7. To compile

On linux server,

* *javac \*.java*

Compilation would compile all the \*.java files and generate the corresponding \*.class files

* *ls*

*fifo.class nfu.class pageReplaceAlgo.class TestInput\_nfu.txt*

*fifo.java nfu.java pageReplaceAlgo.java TestInput\_optimal.txt*

*lru.class optimal.class TestInput\_fifo.txt*

*lru.java optimal.java TestInput\_lru.txt*

**(or)**

On eclipse

Run -> Run(Ctrl+F11)

1. To execute

On Linux server

* *java pageReplaceAlgo (or) java pageReplaceAlgo TestInput\_fifo.txt (or) java pageReplaceAlgo TestInput\_fifo.txt > Output\_fifo.txt*
  1. Java pageReplaceAlgo

To select and parse input through **Interactive Mode**

* 1. java *pageReplaceAlgo TestInput\_fifo.txt*

Parse set of inputs from the TestInput.txt file and displays output on the console, **Batch Mode**

* *java pageReplaceAlgo TestInput\_fifo.txt > Output\_fifo.txt*

Parse set of inputs from the TestInput.txt file and output is redirected to the Output.txt, **Batch Mode**

**(or)**

On eclipse

* 1. Run -> Run(Ctrl+F11)

To select and parse input through **Interactive Mode**

* 1. Run -> Run Configurations -> Arguments -> TestInput\_fifo.txt(complete path) -> Apply -> Ok , **For Batch Mode**

1. **How to parse input?**
2. **Interactive Mode**

*>javac \*.java*

*>java pageReplaceAlgo*

Interactive Mode:

Choose the replacement algorithm that you want to simulate

1. FIFO

2. LRU

3. Optimal

4. NFU

1

Chosen option is 1

FIFO:

Enter number of pages in logical memory:

3

Enter number of frames in physical memory:

2

Enter the page-reference string: (end string with -1)

0 1 2 1 -1

0 1 2 1

pageFault: 1

0

pageFault: 2

0 1

pageFault: 3

1 2

hitRate: 1

1 2

OUTPUT:

HitRate: 1

No. of page requests: 3

Do you want to continue? Yes or No:

The text colored green are the inputs passed by the user, so the interactive mode prompts with user input to enter for page replacement selection option, number of pages in logical memory (should have the max page value), number of pages in physical memory, page-reference string(should end with -1) and an option asking on whether to continue with a “Yes” or exit with “No”

1. Batch Mode:

*>javac \*.java*

*>java pageReplaceAlgo TestInputs\_fifo.txt (or) java pageReplaceAlgo TestInputs\_fifo.txt Output\_fifo.txt*

The input from the TestInputs.txt file is parsed as input to the file. The TestInputs.txt file should follow the below format in order to properly parse the input.

**Sample input in TestInputs.txt:**

1

6

3

1 2 3 4 1 2 5 1 2 3 4 5 -1

1

8

3

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 -1

1 -----------------------------🡪 Page replacement algorithm selection option

6 -----------------------------🡪 Number of pages in logical memory **[ should always be a value greater than the largest value in the page string]**

3 -----------------------------🡪 Number of pages in physical memory

1 2 3 4 1 2 5 1 2 3 4 5 -1 -> Reference string

NOTE: Each input should be on a separate line and the reference string must be separated with a single space and end with a -1

ERROR: Inputs with any other different format are prone to errors.

1. If incorrect file name is given:

Eg: java pageReplaceAlgo TestInputs\_wrong.txt > Output.txt

Following error will be written on file

*Reading Inputs from a Batch File:*

*File Not Found Exception, Exiting..option=-2*

*End of file*

1. **Page Replacement Algorithm**
2. This is the **Main** class
3. Verifies if the program is executed through interactive or batch mode
4. If interactive mode, from step v to step vii it prompts for user input
5. If batch mode, from step v to step vii the input is read from file
6. First, an option to select the page replacement algorithm
7. If option is 1/2/3/4 (fifo, lru, optimal, nfu)
   * An object for the fifo/lru/optimal/nfu class is created
   * Prompt/Reads number of pages in logical memory
   * Prompt/Reads number of pages in physical memory
   * Prompt/Reads page-ref string
   * Validated the page string
   * Fifo/lru/optimal/nfu class pageToFrame() is invoked
     + pageToFrame – does all the process to generate the output
8. If option is invalid an error is displayed
9. If mode is interactive, prompts for “Do you want to continue?Yes or No”
10. If batch, the iteration continues until it reaches EOF
11. **First in First out**
12. The class fifo extends pageReplaceAlgo, so the variables initialized in pageReplaceAlgo can be deployed in fifo.
13. The functionality inside pageToFrame()
    1. The Queue data structure is used
    2. A while loop to check if the count is less than the length of the reference string and the page string not equals -1
       1. An if condition to check if the value is already in the frame
       2. If condition is true
          1. then hitrate and count value is incremented
       3. If condition is false,
          1. a check to check the size of frame equals to the number of frames, then
             1. remove the last frame
          2. Add next reference string onto pageframe
          3. Increment the page fault and count
14. The hitrate and the number of page requests is printed on the console
15. **Least Recently Used**
16. The class lru extends pageReplaceAlgo, so the variables initialized in pageReplaceAlgo can be deployed in lru.
17. The functionality inside pageToFrame()
    1. The Stack data structure is used
    2. A while loop to check if the count is less than the length of the reference string and the page string not equals -1
       1. An if condition to check if the value is already in the frame
       2. If condition is true
          1. Then the page is removed from the frame
          2. The page is again pushed onto the stack, to maintain the order for least recently used
          3. then hitrate and count value is incremented
       3. If condition is false,
          1. a check to check the size of frame equals to the number of frames, then
             1. remove the frame 0 on stack– as it is least recently used
          2. Push a page onto pageframe
          3. Increment the page fault and count
18. The hitrate and the number of page requests is printed on the console
19. **Optimal**
20. The class optimal extends pageReplaceAlgo, so the variables initialized in pageReplaceAlgo can be deployed in optimal.
21. The functionality inside pageToFrame()
    1. The Array list is used
    2. A while loop to check if the count is less than the length of the reference string and the page string not equals -1
       1. An if condition to check if the value is already in the frame
       2. If condition is true
          1. then hitrate and count value is incremented
       3. If condition is false,
          1. For loops, to check if the page on the page frame has the future occurrence on the page string
             1. If present then the index of the next occurrence is stored in an array
             2. Else the index is assigned to a greater value(as there is no future occurrence)
             3. The farthest future occurrence from the max array is computed
             4. The farthest page value is removed from the frame
             5. The new page string is added to the same index of the removed page(in previous step)
          2. Else, the new page is added to the frame
             1. the page fault and count value is incremented
22. The hitrate and the number of page requests is printed on the console
23. **Not Frequently Used**
24. The class nfu extends pageReplaceAlgo, so the variables initialized in pageReplaceAlgo can be deployed in nfu.
25. The functionality inside pageToFrame()
    1. The Array list and map data structure are used
    2. A while loop to check if the count is less than the length of the reference string and the page string not equals -1
       1. An if condition to check if the value is already in the frame
       2. If condition is true
          1. Then the value of the page is got from the hashmap
          2. The value is increment by 1 and put in hashmap
          3. then hitrate and count value is incremented
       3. If condition is false,
          1. a check to check the size of frame equals to the number of frames, then
             1. an Iterator is used to iterate through the hashmap and find the key of the minimum value on the hashmap
             2. The minium value key is removed from the frame

NOTE: This page replacement algorithm adds and removes value similar to **first in first out**

* + - * 1. The new page is added to the frame
        2. If the new page entry is already present in the hashmap

The page value is incremented by 1

* + - * 1. Else, the page and value are added to the hashmap
        2. Page fault and count are incremented
      1. Else, the page is added to the frame
         1. The page and value are added to the hashmap
         2. Page fault and count are incremented

1. The hitrate and the number of page requests is printed on the console

**2.10 Output:**

The Ouptpu.txt file contains the output of the file. The file redirects and saves the output that is to be displayed on the console.

* Java pageReplaceAlgo TestInputs.txt > Output.txt

This saves the Output.txt on the current directory.

# **Test Cases**

1. **Junit (On eclipse)**

Implemented a Junit test framework, were all the individual test inputs are added to the test method to notify if there is any failure on the test inputs on the implemented code.

The run code on eclipse automatically runs the Junit Test. If all the test cases are passed, then it displays “0” errors/failures and nothing on the failure trace. If there is a failure, then it displays on the failure trace.

The below java files have all the test cases of 3.2 to 3.5 sections included.

* fifoTest.java
* lruTest.java
* NfuTest.java
* OptimalTest.java

Instead, of running individual page replacement algorithm junit test cases given above, the test suite below runs the testcases of all the above page replacement algorithm junit test cases.

* AllTests.java

1. **First in First out**

* javac \*.java
* Java pageReplaceAlgo TestInput\_fifo.txt > Output\_fifo.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Logical M/y | Physical M/y | Ref-string | Output [hitrate, number of page requests] |
|  | 4 | 2 | 0 1 2 3 0 1 2 3 0 -1 | 0, 9 |
|  | 19 | 3 | 12 13 15 16 17 18 -1 | 0, 6 |
|  | 6 | 2 | 00 00 00 01 02 03 04 05 -1 | 2, 8 |
|  | 4 | 2 | 1 2 3 2 1 2 3 1 2 -1 | 1, 9 |
|  | 5 | 3 | 1 1 1 2 3 4 2 1 4 3 2 1 4 3 1 -1 | 8, 15 |
|  | 5 | 3 | 1 2 3 4 1 2 3 4 1 2 3 4 -1 | 0, 12 |
|  | 8 | 3 | 1 3 7 4 7 5 1 3 6 2 1 7 -1 | 1, 12 |
|  | 9 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | 2, 14 |
|  | 4 | 3 | 0 1 2 3 1 3 1 3 2 1 3 1 -1 | 8, 12 |
|  | 6 | 3 | 1 2 3 4 1 2 5 1 2 3 4 5 -1 | 3, 12 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 -1 | 4, 16 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 -1 | 5, 20 |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 -1 | 3, 11 |
|  | 8 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | ERROR: Invalid input  Error: As the page string value ‘8’ equals to logical memory |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 | ERROR: Invalid Page String  Error: As it does not end with -1 [when value passed from file] |
|  | 2 | 3 | 111111111111 -1 | ERROR: Wrong number format  Error:Wrong number format, So exits from the program |

1. **Least Recently Used**

* javac \*.java
* Java pageReplaceAlgo TestInput\_lru.txt > Output\_lru.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Logical M/y | Physical M/y | Ref-string | Output [hitrate, number of page requests] |
|  | 4 | 2 | 0 1 2 3 0 1 2 3 0 -1 | 0, 9 |
|  | 19 | 3 | 12 13 15 16 17 18 -1 | 0, 6 |
|  | 6 | 2 | 00 00 00 01 02 03 04 05 -1 | 2, 8 |
|  | 4 | 2 | 1 2 3 2 1 2 3 1 2 -1 | 2, 9 |
|  | 5 | 3 | 1 1 1 2 3 4 2 1 4 3 2 1 4 3 1 -1 | 5, 15 |
|  | 5 | 3 | 1 2 3 4 1 2 3 4 1 2 3 4 -1 | 0, 12 |
|  | 8 | 3 | 1 3 7 4 7 5 1 3 6 2 1 7 -1 | 1, 12 |
|  | 9 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | 2, 14 |
|  | 4 | 3 | 0 1 2 3 1 3 1 3 2 1 3 1 -1 | 8, 12 |
|  | 6 | 3 | 1 2 3 4 1 2 5 1 2 3 4 5 -1 | 2, 12 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 -1 | 5, 16 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 -1 | 8, 20 |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 -1 | 4, 11 |
|  | 8 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | ERROR: Invalid input  Error: As the page string value ‘8’ equals to logical memory |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 | ERROR: Invalid Page String  Error: As it does not end with -1 [when value passed from file] |
|  | 2 | 3 | 111111111111 -1 | ERROR: Wrong number format  Error:Wrong number format, So exits from the program |

1. **Optimal**

* javac \*.java
* Java pageReplaceAlgo TestInput\_optimal.txt > Output\_optimal.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Logical M/y | Physical M/y | Ref-string | Output [hitrate, number of page requests] |
|  | 4 | 2 | 0 1 2 3 0 1 2 3 0 -1 | 2, 9 |
|  | 19 | 3 | 12 13 15 16 17 18 -1 | 0, 6 |
|  | 6 | 2 | 00 00 00 01 02 03 04 05 -1 | 2, 8 |
|  | 4 | 2 | 1 2 3 2 1 2 3 1 2 -1 | 3, 9 |
|  | 5 | 3 | 1 1 1 2 3 4 2 1 4 3 2 1 4 3 1 -1 | 9, 15 |
|  | 5 | 3 | 1 2 3 4 1 2 3 4 1 2 3 4 -1 | 6, 12 |
|  | 8 | 3 | 1 3 7 4 7 5 1 3 6 2 1 7 -1 | 4, 12 |
|  | 9 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | 5, 14 |
|  | 4 | 3 | 0 1 2 3 1 3 1 3 2 1 3 1 -1 | 8, 12 |
|  | 6 | 3 | 1 2 3 4 1 2 5 1 2 3 4 5 -1 | 5, 12 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 -1 | 8, 16 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 -1 | 11, 20 |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 -1 | 5, 11 |
|  | 8 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | ERROR: Invalid input  Error: As the page string value ‘8’ equals to logical memory |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 | ERROR: Invalid Page String  Error: As it does not end with -1 [when value passed from file] |
|  | 2 | 3 | 111111111111 -1 | ERROR: Wrong number format  Error:Wrong number format, So exits from the program |

1. **Not Frequently Used**

* javac \*.java
* Java pageReplaceAlgo TestInput\_nfu.txt > Output\_nfu.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Logical M/y | Physical M/y | Ref-string | Output [hitrate, number of page requests] |
|  | 4 | 2 | 0 1 2 3 0 1 2 3 0 -1 | 0, 9 |
|  | 19 | 3 | 12 13 15 16 17 18 -1 | 0, 6 |
|  | 6 | 2 | 00 00 00 01 02 03 04 05 -1 | 2, 8 |
|  | 4 | 2 | 1 2 3 2 1 2 3 1 2 -1 | 3, 9 |
|  | 5 | 3 | 1 1 1 2 3 4 2 1 4 3 2 1 4 3 1 -1 | 7, 15 |
|  | 5 | 3 | 1 2 3 4 1 2 3 4 1 2 3 4 -1 | 0, 12 |
|  | 8 | 3 | 1 3 7 4 7 5 1 3 6 2 1 7 -1 | 2, 12 |
|  | 9 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | 2, 14 |
|  | 4 | 3 | 0 1 2 3 1 3 1 3 2 1 3 1 -1 | 8, 12 |
|  | 6 | 3 | 1 2 3 4 1 2 5 1 2 3 4 5 -1 | 2, 12 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 -1 | 6, 16 |
|  | 8 | 3 | 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 -1 | 7, 20 |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 -1 | 5, 11 |
|  | 8 | 3 | 4 7 3 0 1 7 3 8 5 4 5 3 4 7 -1 | ERROR: Invalid input  Error: As the page string value ‘8’ equals to logical memory |
|  | 4 | 2 | 0 1 2 1 3 1 3 2 1 3 1 | ERROR: Invalid Page String  Error: As it does not end with -1 [when value passed from file] |
|  | 2 | 3 | 111111111111 -1 | ERROR: Wrong number format  Error:Wrong number format, So exits from the program |