



IJARCCCE

International Journal of Advanced Research in Computer and Communication Engineering

Microsoft Academic, Google Scholar and Mendeley Indexed Journal

Volume 9, Issue 10, October 2020

www.ijarcce.com



International Journal of Advanced Research in Computer and Communication Engineering

A monthly peer-reviewed journal

Select Page



Call for Papers

July 2022 / August 2022

Submission: eMail paper now

Notification: within 1 day

Publication: Immediately

Downloads

Paper Format

© Copyright

Submit to ijarcce@gmail.com

[Check My Paper Status](#)

[Publication Fee](#)

Author Center

How can I publish my paper?

Why Publish in IJARCCCE

Benefits to Authors

Instructions to Authors

Frequently Asked Questions

Author Testimonials

IJARCCCE Management

Aims and Scope

Call for Papers

Editorial Board

DOI and Crossref

Publication Ethics

Policies

Subscription / Librarian

Conference Special Issue Info

Archives

Current Issues / Archives

Conference Special Issue

[Editorial Board Members](#)

Executive Advisory Board

Reviewers Board

Professor Subramaniam Ganesan

Department of Electrical and Computer Engineering, Oakland University, Rochester MI 48309, USA.

[Profile](#), [Google Scholar](#), [ResearchGate](#), [ORCID](#)

Dr. Le Quan Ha

School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK.

[ResearchGate](#)

Xiang-Fang Yu

University of Chinese Academy of Sciences, Beijing, China.

[ResearchGate](#)

Dr. Randy R. Koon Koon

Faculty of Science & Technology, University of the West Indies, Jamaica, West Indies.

[Profile](#), [ResearchGate](#)

Radi Romansky

Department of Electronics, Computer Systems and Technologies, College of Energy and Electronics at Technical University of Sofia, Sofia, Bulgaria.

[Profile](#), [ResearchGate](#), [Thomson Reuters](#)

John Joel F. Martinez

College of Engineering and Architecture –Electronics Engineering, Technological Institute of the Philippines, Quezon City, Philippines.

[ResearchGate](#)

Jeong Phil Lee

Subdivision of New & Renewable Electricity, Kyungnam College of Inform. & Tech., Busan, Korea.

[ResearchGate](#)

Yaremenko Artem

Power Supply Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kiev, Ukraine.

[Linkedin](#)

Marleen Huysman

Department of Information Systems, Logistics and Innovation, Vrije Universiteit Amsterdam, De Boelelaan, Amsterdam, The Netherlands.

[ResearchGate](#), [WordPress](#)

Ivelisse Teresa Machín Torres

Professor, Technical Department, José Martí University, Sancti Spíritus, Cuba.

Marquez

Departamento de Física de la Materia Condensada, Facultad de Ciencias, Universidad de Cadiz, Cadiz, Spain.

[ResearchGate](#)

Ljiljana Ivanković

University of Applied Sciences Velika Gorica, Velika Gorica, Croatia.

Georgios Konstantinou

Special Scientist FOSS Research Centre for Sustainable Energy, PV Technology, University of Cyprus, Nicosia, Cyprus.

[ResearchGate](#)

Đỗ Đình Thanh

Faculty of Information Technology, Ho Chi Minh University of Foreign Languages and Information Technology (HUFLIT), Vietnam.

Vinod A Prasad

Associate Professor, School of Computer Science and Engineering, Nanyang Technological University (NTU) Singapore.

[profile](#), [Googlescholar](#), [IEEE](#)

Myo Myint Maw

Lecturer, Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay, Myanmar.

[Linkedin](#), [IEEE](#)

Michael Kimwele

Dr., School of Computing & Information Technology, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya.

[Academia](#), [ResearchGate](#), [GoogleScholar](#), [Linkedin](#)

Jose A. Noh

Facultad de Matemáticas, Universidad Autónoma de Yucatán, Tizimín, México.

[Academia](#)

Indrasen Poola

Data Scientist & Artificial Intelligence – Industry Consultant, California, USA.

[ResearchGate](#), [GoogleScholar](#), [Academia](#), [figshare](#), [publons](#)

Prof. Hsiu-fei Sophie Lee

Department of Special Education, National Taitung University, Taiwan.

[Linkedin](#)

Dr. G.Sadashivappa

Dept of Telecommunication Engineering, R.V.College of Engineering, Bangalore.

[Profile](#), [Linkedin](#), [Biography](#), [Google Scholar](#), [IEEE](#)

Dr. Redahegn Sileshi

University of North Georgia, Oakwood, Georgia, USA.

Dr. Adnan Al-Rabea

Department of Information Technology, Albalqa Applied University, Salt, Jordan.

[GoogleScholar](#)

Dr. Adrian Nicolae Branga

Department of Mathematics and Informatics, Lucian Blaga University of Sibiu, Romania.

[lawrencepress](#), [Scitechnol](#), [Googlescholar](#), [SciencePG](#), [ResearchGate](#), [GoogleScholar](#)

Dr. Magdy Shayboub A. Mahmoud

Faculty of Computers and Informatics, Suez Canal University, Egypt.

[IEEE](#), [profile](#)

Kinnal Dhameliya

Electronics Engineer, Innovative electronics Corp. 750 Trumbull Drive Pittsburgh.

[Linkedin](#), [profile](#)

Dr. Ali El-Moursy

Department of Electrical & Computer Engineering, University of Sharjah, United Arab Emirates.

[Academia](#), [GoogleScholar](#), [ORCID](#), [Linkedin](#)

Dr. Aeizaal Azman Abdul Wahab

School of Electrical & Electronic Engineering, Universiti Sains Malaysia (USM), Malaysia.

[ResearchID](#), [ORCID](#), [Scopus](#), [USMExpertise](#), [Researchgate](#), [Googlescholar](#)

Mohammad Houssein Ghosn

Professor, Lebanese International University, Department of Computer Science, Beirut, Lebanon.

[Linkedin](#)

Dr. Tolga ENSARI

Computer Engineering, Istanbul University, Istanbul, Turkey.

[Linkedin](#), [ResearchGate](#), [GoogleScholar](#), [IEEE](#)

Dr. Mohd Fadzli Mohd Salleh

School of Electrical & Electronic Engineering, Universiti Sains Malaysia, Malaysia.

[ResearchGate](#), [Profile](#), [USMEXPERTISE](#), [ACM.org](#), [IEEE](#), [GoogleScholar](#)

Dr. Mohamed Abd El-Basset Matwalli

Faculty of Computers and Informatics, Zagazig University, Egypt.

Dr. Aeizaal Azman Abdul Wahab

Faculty of Computers and Informatics, Zagazig University, Egypt

Dean M Aslam, Ph.D., Professor

Professor & Member of National Academy of Inventors (NAI),

Director of BIOMEMS and Mind Laboratory, Electrical and Computer Engineering,

Michigan State University, East Lansing, MI 48824 USA.

[RESUME](#), [Profile](#), [ResearchGate](#), [Biography](#), [IEEE](#), [Googlescholar](#)

Prof. Dr.Ushaa Eswaran, B.E., M.E.,PhD.,

Professor & Dean, Department of ECE, Infant Jesus College of Engineering, Keela Vallanadu, Tamil Nadu, India.

[Linkedin](#)

S.A. Edalatpanah

Department of Applied Mathematics, Islamic Azad University of Lahijan, Iran.

[ResearchGate](#), [GoogleScholar](#), [Academia](#), [publons](#)

Timucin BARDAK, Ph.D.,

Bartın Vocational School, Bartın University, Bartın – Turkey.

[Academia](#)

Wenzhu Yang

Professor, School of Cyber Security and Computer, Hebei University, Baoding, China.

[ResearchGate](#), [Profile](#)

Dr Jayanti goyal

Head, Computer Department, Kanoria Girls PG College Jaipur, Rajasthan.

[GoogleScholar](#), [Linkedin](#), [AcademicPortal](#)

Michael Kimwele

Dr., School of Computing & Information Technology, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya.

[ResearchGate](#), [Academia](#), [Google scholar](#), [Profile](#), [Linkedin](#)

Dr. Elsanosy M. Elamin

Dept. of Electrical Engineering, Faculty of Engineering, University of Kordofan, Sudan.

[ResearchGate](#)

Dr.Dhirendra Sharma

Chief Information Security Officer (CISO), Sr. Faculty IT, UIIT Himachal Pradesh University, Shimla.

[Linkedin](#), [Profile](#)

Professor Deepika D Pai

Department of Electronics and Communication, Vemana Institute of Technology, Koramangala, Bangalore.

[Profile](#)

Georgi Tsochev

Faculty of Computer Systems and Technology, Technical University of Sofia, Sofia, Bulgaria .

[Academia](#), [ResearchGate](#), [Linkedin](#)

Dr. Jitesh Neve

Computer Science and Engineering, University of Swahili, Republic of Panama

Module Lead, Persistent System Limited, Pune.

[Linkedin](#)

A. Muse

Obafemi Awolowo University, Ile-Ife, Nigeria.

[ResearchGate](#)

Prof.(Dr.) Vijay Singh Rathore

Department of Computer Science, S.R.R.Govt.Arts & Science College, Karimnagar, Telangana.

[Profile](#), [GoogleScholar](#), [Youtube](#), [ResearchGate](#)

Janani.B

Professor, Department of CSE, Adithya Institute of Technology, Coimbatore.

Dr.S.Nagaprasad

Professor & Head, CSE, JECRC, Jaipur Engineering College & Research Centre, Jaipur.

Sile Wang

Lecturer, School of Cyber Security and Computer, Hebei University, Baoding, China.

[DBLP](#)

Shearyl U. Arenas

College of Engineering and Architecture – Electronics Engineering, Technological Institute of the Philippines, Quezon City, Philippines.

[Linkedin](#), [EJournals](#), [ResearchGate](#)

Dr. S.Rakoth Kandan, M.Tech., Ph.D.,

Professor, Department of CSE, Jayamukhi Institute of Technological Sciences, Warangal, Telangana.

[Profile](#), [Linkedin](#), [GoogleScholar](#)

Slavcho Chungurski

Associate Professor, UTMS University, Skopje, Macedonia

Information Security Management Expert, Cabinet of Deputy Prime Minister of the Republic of Macedonia.

[GoogleScholar](#), [DBLP](#), [Academia.edu](#)

Dr D.Durga Prasad, Ph.D.,

Professor, Dept of CSE, PSCMR College of Engg., & Tech, Vijayawada.

[Linkedin](#)

Dr Mohd Uruj Jaleel

Assistant Professor, College of Computing & Informatics, Saudi Electronic University, Riyadh (KSA).

[GoogleScholar](#)

Mr. R.D.Sivakumar

Assistant Professor, Department of Computer Science, Ayya Nadar Janaki Ammal College, Sivakasi.
[Profile](#), [SlideShare.net](#), [Twitter](#)

Dr. Shahzad Ashraf

Assistant Professor, Hohai University Changzhou, China

Dr. Jayeshkumar Natwarlal Modi

Assistant Professor, Computer Science, HNG University, Patan

Copyright © 2022 IJARCCCE



This work is licensed under a [Creative Commons Attribution 4.0 International License](#).



VOLUME 9, ISSUE 10, OCTOBER 2020

About Ad hoc On-Demand Multi-Path Distance Vector Routing for Mobile Ad hoc Networks

Phu Hung Le

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91001](https://doi.org/10.17148/IJARCCCE.2020.91001)

A Review Paper on games Designed & Implemented on FPGA

V.Radha Krishna, Rakesh Kumar Y

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91002](https://doi.org/10.17148/IJARCCCE.2020.91002)

Downlink Scheduling with Sharing Spectrum for Surviving LTE System: A Review

Poonam Damara, Yashika Sharma

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91003](https://doi.org/10.17148/IJARCCCE.2020.91003)

A Review on Fuzzy Based Routing Method for Improving Performance in WSN

Deepa Goel, Yashika Sharma

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91004](https://doi.org/10.17148/IJARCCCE.2020.91004)

Reactive and Bootstrapping Beem Algorithm For 3d-Clustering in Wireless Sensor Network

Manpreet Kaur, Er. Amreen Kaur

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91005](https://doi.org/10.17148/IJARCCCE.2020.91005)

Information Technology in Supply Chain Management

Shruthi K Murthy, Chethan S R, Rishi Singh, Nishitha G S, Sai Raksha S Iyengar, Janak V Kumar

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91006](https://doi.org/10.17148/IJARCCCE.2020.91006)

Open Network Automation Platform: Solution Summary

Jasneet Singh Parmar

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91007](https://doi.org/10.17148/IJARCCCE.2020.91007)

Vehicle External Airbag System

Rohit Kothawale, Amit Chandan

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91008](https://doi.org/10.17148/IJARCCCE.2020.91008)

Analysis of Random Page Replacement Algorithm in Operating System

Tri Dharma Putra, Rakhmat Purnomo, Wowon Priatna

[Abstract](#) | [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91009](https://doi.org/10.17148/IJARCCCE.2020.91009)

Bike Blind Spot Alert System

Rohit Kothawale, Amit Chandan, Prachi Kotkar, Prof. Manjusha Namewar

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91010](https://doi.org/10.17148/IJARCCCE.2020.91010)

Implementation of College Network Module using Cisco Packet Tracer Simulator

Bhavesh G. Wani, Shreyas Fegade

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91011](https://doi.org/10.17148/IJARCCCE.2020.91011)

Application of Adaptive Neuro-fuzzy Inference System (ANFIS) for Prediction of Nutrients (Nitrogen, Phosphorus, Potassium and Magnesium) level from Soil Sample in some States of Nigeria Vegetational Zone. Abdullahi Yusuf Egwoh, Emmanuel Amano Onibere, Philip Odion

Abdullahi Yusuf Egwoh, Emmanuel Amano Onibere, Philip Odion

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91012](https://doi.org/10.17148/IJARCCCE.2020.91012)

Blockchain For Securing Cyber Infrastructure

Riya Suri, Nachiket Bhuta, Sagar Gada ,Rahil Vithalani

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91013](https://doi.org/10.17148/IJARCCCE.2020.91013)

Online Arabic Handwriting Characters Recognition using Deep Learning

Khalid Mohammed Musa Yaagoup, Mohamed Elhafiz Mustafa

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91014](https://doi.org/10.17148/IJARCCCE.2020.91014)

Pneumonia Detection in Covid-19 Patients using CNN Algorithm

Vishesh S, Nishanth S, Bharath R, Amith Vishnu

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91015](https://doi.org/10.17148/IJARCCCE.2020.91015)

Development of an Electronic Incubator for Hen Eggs using a Tire

Joel Flores, Gerardo Suárez

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91016](https://doi.org/10.17148/IJARCCCE.2020.91016)

Evaluation of Multimedia Transmission for Wireless Multihop Networks using the Evalvid Tool

Ms. Renita Blossom Monteiro

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91017](https://doi.org/10.17148/IJARCCCE.2020.91017)

Two Wheelers Balancing System

Rohit Kothawale, Aniket Mahajan

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91018](https://doi.org/10.17148/IJARCCCE.2020.91018)

Traffic Signal & Street Light Automation System

Rohit Kothawale, Prince Kumar, Sahil Kumar, Aniket Mahajan

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91019](https://doi.org/10.17148/IJARCCCE.2020.91019)

A Survey on Error Detection Techniques

Nishanth S

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91020](https://doi.org/10.17148/IJARCCCE.2020.91020)

A study on Proxy Re-Encryption Schemes

Shashank Shekhar Tiwari, Hemant Kumar Baranwal, Shweta Tiwari, Ritika Yaduvanshi

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91022](https://doi.org/10.17148/IJARCCCE.2020.91022)



Obstacle Avoidance using LiDAR Sensors and Artificial Intelligence

Neelanjan Goswami

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91023](https://doi.org/10.17148/IJARCCCE.2020.91023)

Augmenting Embedded Systems for Computer Vision Applications

Neelanjan Goswami

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91024](https://doi.org/10.17148/IJARCCCE.2020.91024)

A Survey on Clustering Techniques for Wireless Sensor Networks

Alefiya Shabbir Pithapur and Ankit Tripathi

[Abstract](#) |  [PDF](#) | DOI: [10.17148/IJARCCCE.2020.91025](https://doi.org/10.17148/IJARCCCE.2020.91025)

Call for Papers

May 2022 / June 2022

Submission: eMail paper now

Notification: within 1 day

Publication: Immediately

Downloads

 Paper Format

© Copyright

Submit to ijarcce@gmail.com

 [Check My Paper Status](#)

 [Publication Fee](#)

Author Center

How can I publish my paper?

Why Publish in IJARCCCE

Benefits to Authors

Instructions to Authors

Frequently Asked Questions

Author Testimonials

IJARCCCE Management

Aims and Scope

Call for Papers

Editorial Board

DOI and Crossref

Publication Ethics

Policies

Subscription / Librarian

Conference Special Issue Info

Archives

Current Issues / Archives

Conference Special Issue

Copyright © 2022 IJARCCCE



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Analysis of Random Page Replacement Algorithm in Operating System

Tri Dharma Putra¹, Rakhmat Purnomo², Wowon Priatna³

Department of Informatics, Faculty of Computer Science,

University of Bhayangkara Jakarta Raya Jalan Perjuangan Bekasi Utara, West Java, Indonesia^{1,2,3}

Abstract: Page replacement algorithms choose pages to swap out from memory, namely when a new page needs allocation memory. Several page replacement algorithms proposed by experts. Four well known page replacement algorithms are Least Recently Used (LRU), Optimal Page Replacement (OPR), First In First Out (FIFO), Random page replacement algorithms. Each one has the objective to minimize the page fault. If the minimum page fault achieved, the performance of the process increased. The algorithm discussed is random page replacement algorithm. In this paper three case studies of random page replacement algorithm are discussed thoroughly.

Keywords: Random page replacement algorithm, page fault, page replacement algorithm, FIFO, LRU, OPR

I. INTRODUCTION

Memory management is the most important part of operating system in which all activities related to the memory are perform [1]. The replacement of page is the main concept. Pages is required to be in the main memory [2]. Computer system objective is to execute program of different sizes. The program should be in main memory when program is executed. But main memory has limited size to accommodate programs which is large. Execution of program that may not be entirely in main memory is called virtual memory [3]. Virtual memory paved the way to execute program that is larger than main memory. The process in which we provisionally moves process from primary memory to the hard disk or secondary memory, so that the memory available for other processes, which is known as swapping [4]

On operating systems which use paging for memory management, it is needed a page replacement algorithm to decide which pages to replace when a new page is entered. Good replacement can reduce the cost of page fault that result in higher performance of the system [5].

Page Fault occurs when a process references to a page that is not present in the main memory and needs to be through from secondary memory [6]. When a program currently executing, accessing a memory page that is mapped to the virtual address space, but is not loaded into physical memory. Because actual physical memory is much smaller than virtual memory, a page fault occurs. In this case, the operating system may have to replace one of the existing pages with the new page it needs. There are several page replacement algorithms for this replacement. One of them is the random page replacement algorithm [7].

II. LITERATURE SURVEYS

There are several algorithms of page replacement proposed by experts, namely :

1. First in First out (FiFO): This is the simplest page changing algorithm. In this algorithm, the operating system keeps track of all the pages in memory that are in the queue, namely page that has been in front of the queue the longest. When a page has to be replaced the page at the front of the queue is chosen to be deleted. An average of hit ratio of 50 percent is reported by FIFO algorithm. The advantages and disadvantages of FIFO algorithm: Easy to understand, very easy to be implemented, Performance not always good [8].
2. Optimal Page Replacement: In this algorithm, pages are replaced which will not be used for the next longest duration of time. Optimal page replacement is a good algorithm, but it is impossible to implement practically, because the operating system will not know the next request. The use of this algorithm is to create benchmarks so that other replacement algorithms can be analyzed against this algorithm. On this algorithm, system select victim whole reference is furthest. This algorithm has the best page fault behavior reported [9].
3. Least Recently Used: In this algorithm, the page will be replaced the least used (least recently used). An average of 60 % hit ratio is reported by LRU algorithm. This algorithm selects the victim page as that which is not been demanded for access for along time [9].

4. Random Page Replacement Algorithm: In this algorithm, the operating system will keep track of all the pages in memory which are marked as the next replaced slot. And the selection of which slots to replace is done randomly. Page replacement algorithm depends on the following parameters:

Page fault rate, indicates by how many misses occur for a set of page reference. The minimal the page fault rate, the higher the efficiency of the algorithm.

Hit rate, indicates how many hits occur on the page. The higher hit rate, the higher the efficiency of the algorithm [9].

III. ORGANIZATION STRUCTURE

This journal divides into six chapters. The first chapter is introduction, here we discuss the story behind page replacement algorithm. In the second chapter, we discuss about some literature survey about page replacement algorithms proposed by experts. The third chapter, is about organization structure. The fourth chapter is about random page replacement algorithm which is our main subject. The fifth chapter is about case studies for random page replacement algorithm. We discuss it in details in three case studies. The last chapter is conclusion.

IV. RANDOM PAGE REPLACEMENT ALGORITHM

In page replacement using a random algorithm (random) every time a page fault occurs, the replaced page is randomly selected. This technique does not use any information in determining which pages to replace, all pages in main memory have the same weight to be selected. How to select random or arbitrary pages, including the page that is currently referenced (pages that should not be replaced). This random technique is very bad, the experiment shows the random algorithm causes a very high frequency of page faults.

V. CASE STUDY

As like its name, in this random page replacement algorithm, page replacement is done randomly. The page refers to slot randomly, its slot place will be replaced by the new coming page on the queue. Here the random slot will be given an asterisk (*) sign. In this analysis we use three page frames in memory.

5.1 Case Study 1

Let us discuss case study 1, as we can see from the table below:

TABLE 1. Case Study 1

Page queue	0	1	3	4		2	0	0	4		1	3	3	3
Random Replacement	0	0	0*	4		4*	0*	^	4		4	4*	3*	^
		1	1	1*		2	2	^	2		2	2	2	^
			3	3		3	3	^	3*		1*	3	3	^
Page Fault	x	x	x	x		x	x	^	x		x	x	x	^

x = miss

^ = hit

Let us discuss the first string. First 0 gets in, page fault happens. Then 1 gets in, page fault happens. Then 3 gets into the slot, still page fault happens. The random slot to be filled in is the first page, namely 0. This slot is chosen randomly. And in the next page, namely 4, will fill in the slot of 0, which is pointed by the asterisk sign. Still in this analysis, the page fault happens. And then 1 gets chosen as the random slot.

The second string, 1 is the random slot chosen. 2 gets in into that slot, still page fault happens, since there is no hit. 4 gets chosen randomly to be filled in. Then 0 gets into the first page, replacing 4. This is also a page fault, it is a miss. Then 0 get asterisk to be the next chosen slot. Then the next page, the 0, gets a hit, since page 0 is in the first page. There is no page fault. This is a hit. Then, 4 gets into the first page, replacing 0, page fault happens. Then 3 is pointed by asterisk.

The last string. Since the third page has the asterisk, which is get the random priority, 1 will get into third page replacing 3. Then the third page, 1, get asterisk randomly. 3 is replacing 1. Then the first page, 4, gets the asterisk randomly. The next one, 3 replacing 4 and this first page gets chosen as the random slot. Page fault happens. The last one is 3. In this last page replacement, 3 gets a hit.

There are ten misses (page faults) in this algorithm and there are two hits.

5.2. Case Study 2

Let us discuss case study 2, as we can see from the table below :

TABLE 2. Case Study 2

Page queue	0	1	3	2		1	4	3	2		1	2	4	4
Random Replacement	0	0	0	0		0	0	0	0		0	^	0	^
		1	1*	2		2	2*	3	3*		1	^	1	^
			3	3*		1*	4	4*	2		2*	^	4*	^
Page Fault	x	x	x	x		x	x	x	x		x	^	x	^

x = miss

^ = hit

Take a look at the first string. 0 gets in, this is page fault. This is a miss. Then 1 gets in, this is also a miss, a page fault. Then 3 gets in into the third slot, this is also a page fault. In the meantime, 1 gets priority randomly. Then 2 replace 1. This is also a page fault. Then 3 gets asterisk.

The second string, 1 gets into the third slot, replacing 3, this is a miss, a page fault. Also 1 get asterisk. Then 4 replaces 1, this is also a page fault. 2 gets asterisk. Then 3 replaces 2, this also a miss, a page fault, with 4 gets asterisk randomly. Then 2 gets into into third slot, replacing 4, with 3 gets asterisk as the next priority.

The last string, page 1 replacing 3, because the second page has asterisk which means should be replaced. Then, the second page, 2, gets a hit in the third slot. Then the third page, 4, replacing page 2, with page 4 itself gets asterisk. The last page then gets a hit in the third slot, page 4.

There are ten misses (page faults) and two hits happen.

5.3. Case Study 3

Let us discuss case study 3, as we can see from the table below:

TABLE 3. Case Study 3

Page queue	0	2	3	2		1	0	2	2		4	1	0	2
Random Replacement	0	0	0	^		0*	^	2*	^		4	^	4*	2
		2	2*	^		1	^	1	^		1*	^	0	0
			3	^		3	^	3	^		3	^	3	3
Page Fault	x	x	x	^		x	^	x	^		x	^	x	x

x = miss

^ = hit

Take a look at the first string. 0 gets in this is a page fault. Then 2 gets into the second slot, also a page fault happens. Then 3 gets into the third slot, 2 gets asterisk randomly. Then page 2 gets into the second slot, this is a hit.

The second string, 1 replacing 2, since 2 gets asterisk, this is a page fault, and 0 get asterisk randomly. Then 0 gets a hit in the first slot. Then 2 replaces 0. Still the first slot, 2, get asterisk randomly. Then the last page, 2, gets a hit.

The last string, 4 gets into the first slot, replacing 2, and then 1 gets pointed by asterisk randomly. Then the second page, 1 gets a hit. Then the third page, 0, replacing 1, this is a page fault. And 4 get pointed by asterisk randomly. The last page, 2, replacing 4, this is also a page fault.

There are eight page faults and four hits happen.

VI. CONCLUSION

From the three case studies, the first case study we get ten misses (page faults) and two hits. For the second case study, we get ten misses (page faults) and two hits. And from the third case study, we get eight page faults and four hits. The third case study is more efficient then the two case studies earlier. So for the first and second case studies, we get the efficiency 16,7 percentile, in the meantime the third case study we have eight misses and four hits, so we get efficiency

33,3 percentile. It is concluded that this algorithm (random page replacement algorithm) is not so efficient. For future work, other algorithm like LRU can be analysed to get better efficiency.

REFERENCES

- [1] N. H. Ali A. Titinchi, "A New Method to Enhance LRU Page Replacement Algorithm Performance," *Int. J. Sci. Technol. Res.*, vol. 9, no. 2, pp. 4185–4190, 2020.
- [2] G. Rexha, E. Elmazi, and I. Tafa, "A Comparison of Three Page Replacement Algorithms: FIFO, LRU and Optimal," *Acad. J. Interdiscip. Stud.*, 2015, doi: 10.5901/ajis.2015.v4n2s2p56.
- [3] S. Shastri, A. Sharma, and V. Mansotra, "Study of Page Replacement Algorithms and their analysis with C#," 2016.
- [4] G. P. Arya, D. Prasad, and S. S. Rana, "An improved page replacement algorithm using block retrieval of pages," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i4.5.20004.
- [5] B. A. T. V. L. Kolhe, "Analysis of Various Page Replacement Algorithms in Operating System," *Int. J. Sci. Res.*, 2016.
- [6] A. Saxena, "A Study of Page Replacement Algorithms," *Int. J. Eng. Res. Gen. Sci.*, vol. 2, no. 4, pp. 385–388, 2014.
- [7] G. for Geeks, "Geeks for Geeks," *Page Replacement Algorithm*, 2020. <https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/> (accessed Oct. 08, 2020).
- [8] D. P. Juhi Kumari, Sonam Kumari, "A Comparison of Page Replacement Algorithms: A Survey," *Int. J. Sci. Eng. Res.*, vol. 7, no. 12, pp. 57–62, 2016.
- [9] R. R. Hitha Paulson, "Page Replacement Algorithms - Challenges and Trends," *Int. J. Comput. Math. Sci.*, vol. 6, no. 9, pp. 112–116, 2017.



Plagiarism Checker X Originality Report

Similarity Found: 11%

Date: Thursday, October 22, 2020

Statistics: 252 words Plagiarized / 2257 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

Analysis of Random Page Replacement Algorithm in Operating System Rakhmat Purnomo³, Tri Dharma Putra¹, Wowon Priatna² Department of Informatics, Faculty of Computer Science, University of Bhayangkara Jakarta Raya Jalan Perjuangan Bekasi Utara, West Java, Indonesia^{1,2,3} Abstract: Page replacement algorithms choose pages to swap out from memory, namely when a new page needs allocation memory. Several page replacement algorithms proposed by experts.

Four well known page replacement algorithms are **Least Recently Used (LRU)**, Optimal Page Replacement (OPR), **First In First Out (FIFO)**, Random page replacement algorithms. Each one **has the objective to minimize the** page fault. If the minimum page fault achieved, **the performance of the process** increased. The algorithm discussed is random **page replacement algorithm**. In this paper three case studies of random page replacement algorithm are discussed.

Keywords: Random page replacement algorithm, page fault, page replacement algorithm, FIFO, LRU, OPR

Introduction Memory management is the most important part of operating system in which all activities related to the memory are performed [1]. The replacement of page is the main concept. Pages are required to be in the main memory [2]. Computer system objective is to execute program of different sizes.

The program should be in main memory when program is executed. But main memory has limited size to accommodate programs which are large. Execution of program that may not be entirely in main memory is called virtual memory [3]. Virtual memory paved the way to execute program that is larger than main memory.

The process in which we provisionally move process from primary memory to the hard disk or secondary memory, so that the memory available for other processes, which is known as swapping [4]. On operating systems which use paging for memory management, it is needed a page replacement algorithm to decide which pages to replace when a new page is entered. Good replacement can reduce the cost of page fault that results in higher performance of the system [5].

Page Fault occurs when a process references to a page that is not present in the main memory and needs to be brought from secondary memory [6]. When a program currently executing accesses a memory page that is mapped to the virtual address space, but is not loaded into physical memory. Because actual physical memory is much smaller than virtual memory, a page fault occurs.

In this case, the operating system may have to replace one of the existing pages with the new page it needs. There are several page replacement algorithms for this replacement. One of them is the random replacement algorithm [7]. Some Wellknown Page Replacement Algorithms There are several other algorithms of page replacement proposed by experts, namely : First in First out (FIFO): This is the simplest page changing algorithm. In this algorithm, the operating system keeps track of all the pages in memory that are in the queue.

Page has been in front of the queue the longest. When a page has to be replaced the page at the front of the queue is chosen to be deleted. An average of hit ratio of 50 percent is reported by FIFO algorithm. The advantages and disadvantages of FIFO algorithm: Easy to understand, very easy to be implemented, Performance not always good [8].

Optimal Page Replacement: In this algorithm, pages are replaced which will not be used for the next longest duration of time. Optimal page replacement is a good algorithm, but it is impossible to implement practically, because the operating system will not

know the next request. The use of this algorithm is to create benchmarks so that other replacement algorithms can be analyzed against this algorithm.

On this algorithm, system select victim whose reference is furthest. This algorithm has the best page fault behavior reported. Least Recently Used: In this algorithm, the page will be replaced the least used (least recently used). An average of 60 % hit ratio is reported by LRU algorithm. This algorithm selects the victim page as that which is not been demanded for access for along time.

Random Page Replacement Algorithm: In this algorithm, the operating system will keep track of all the pages in memory which are marked as the next replaced slot. And the selection of which slots to replace is done randomly. Parameter of The Algorithms Page replacement algorithm depends on the following parameters: Page fault rate, indicates by how many miss occur for asset of page references.

The minimal the page fault rate, the higher the efficiency of the algorithm. Hit rate, indicates how the hit occurs on the page. The higher hit rate, the higher the efficiency of the algorithm [9]. Random Page Replacement Algorithm In page replacement using a random algorithm (random) every time a page fault occurs, the replaced page is randomly selected.

This technique does not use any information in determining which pages to replace, all pages in main memory have the same weight to be selected. How to select random or arbitrary pages, including the page that is currently referenced (pages that should not be replaced). This random technique is very bad, the experiment shows the random algorithm causes a very high frequency of page faults.

Case Study As like its name, in this random page replacement algorithm, page replacement is done randomly. The page that refers to slot randomly, its slot place will be replaced by the new coming page on the queue. In each case study, we use just three frames. Here the random slot will be given a asterisk (*) sign. 5.1 Case Study 1 Let us discuss case study 1, as we can see from the table below: TABLE 1.

Case Study 1 Page queue 0 1 3 4 2 0 0 4 1 3 1 3 Random Replacement
0 0 0* 4 4* 0* ^ 4 4 4* 1* ^ 1 1 1* 2 2 ^ 2 2 2 2 ^
3 3 3 3 ^ 3* 1* 3 3 ^ Page Fault x x x x x x x x x x x x Let us
discuss the first string. First 0 gets in, page fault happens. Then 1 gets in, page fault happens. Then 3 gets into the slot, still page fault happens.

The random slot to be filled in is the first frame, namely 0. This slot is chosen randomly.

And in the next page, namely 4, will fill in the slot of 0, which is pointed by the asterisk sign. Still in this analysis, the page fault happens. And then 1 gets chosen as the random slot. The second string, 1 is the random slot chosen.

2 gets in into that slot, still page fault happens, since there is no hit. 4 gets chosen randomly to be filled in. Then 0 gets into the first frame, replacing 4. This is also a page fault, it is a miss. Then 0 gets asterisk to be the next chosen slot. Then the next page, the 0, gets a hit, since page 0 is in the first frame. There is no page fault. This is a hit. Then, 4 gets into the first frame, replacing 0, page fault happens. Then 3 is pointed by asterisk. The last string.

Since the third frame has the asterisk, which is get the random priority, 1 will get into third frame replacing 3. Then the first frame, 4, gets the asterisk randomly. The next one, 3 replacing 4 and this first frame gets chosen as the random slot. Page fault happens. The last one is 3. In this last page replacement, 3 gets a hit. There are ten misses in this algorithm and there are two hits.

5.2. Case Study 2

Let us discuss case study 2, as we can see from the table below : TABLE 2.

Case Study 2 Page queue _0_1_3_2_ _1_4_3_2_ _1_2_4_4_ _Random Replacement
 _0_0_0_0_ _0_0_0_0_ _0_ ^ _0_ ^ _ _ _ _1_1*_2_ _2_2*_3_3*_ _1_ ^ _1_ ^ _ _ _ _3_3*_
 _1*_4_4*_2_ _2*_ ^ _4*_ ^ _Page Fault _x_x_x_x_ _x_x_x_x_ _x_x_x_ _Take
 a look at the first string. 0 gets in, this is page fault. Then 1 gets in, this is also a miss, a
 page fault. Then 3 gets in into the third slot, this is also a page fault.

In the meantime, 1 gets priority randomly. Then 2 replace 1. This is also a page fault. Then 3 gets asterisk. The second string, 1 gets into the third slot, this is a miss, a page fault. Also 1 get asterisk. Then 4 replaces 1, this is also a page fault. 2 gets asterisk. Then 3 replaces 2, this also a miss, a page fault, with 4 gets asterisk randomly.

Then 2 gets into into third slot, replacing 4, with 3 gets asterisk as the next priority. The last string, page 1 replacing page 3, because page 2 has asterisk which means should be replaced. Then, the second page, 2, gets a hit in the third slot. Then the third page, 4, replacing page 2, with page 4 itself gets asterisk. The last page also gets a hit in the third slot, page 4.

There are ten misses and two hits happen.

5.3. Case Study 3 Let us discuss case study 3, as we can see from the table below:

	TABLE 3. Case Study 3
Page queue	_0_2_3_2_1_0_2_2_4_1_0_2_
Random Replacement	_0_0_0_^_0*_^_2*_^_4_^_4*_2_2_2*_^_1_^_1_^_1*_^_0_0_3_^_3_^_3_^_3_^_3_3_
Page Fault	x_x_x_x_x_x_x_x_x_

Take a look at the first string.

0 gets in this is a page fault. Then 2 gets into the second slot, also a page fault happens. Then 3 gets into the third slot, 2 gets asterisk randomly. Then page 2 gets a hit. The second string, 1 replacing 2, this is a page fault, and 0 get asterisk randomly. Then 0 gets a hit in the first slot. Then 2 replaces 0, and the 2 itself get asterisk randomly. Then the last page, 2, gets a hit.

The last string, 4 gets into the first slot, replacing 2, and then 1 gets pointed by asterisk randomly. Then the second page, 1 gets a hit. Then the third page, 0, replacing 1, this is a page fault. And 4 get pointed by asterisk randomly. The last page, 2, replacing 4, this is also a page fault. There are eight page fault and four hits happen. Conclusion From the three case studes, the first case study we get ten misses and two hits.

For the second case study, we get ten misses and two hits. And from the third case study, we get eight page faults and four hits. The third case study is more efficient then the two case studies before, because the third case study get four hits and eight misses, in the meantime, the first study case studies only get two hits and ten misses each. The conclusion is the third case study is more efficient then first and second case study.

It is concluded that this algorithm (random page replacement algorithm) is not so efficient from other wellknown page replacement algorithms. In the first case study, we have ten misses and two hits, so we get the efficiency is 16,7 percent, in the meantime the third case study we have eight misses and four hits, so we get efficiency 33,3 percent. References

[1] N. H. Ali A. Titinchi, "A New Method to Enhance LRU Page Replacement Algorithm Performance," Int. J.

Sci. Technol. Res., vol. 9, no. 2, pp. 4185–4190, 2020. [2] G. Rexha, E. Elmazi, and I. Tafa, "A Comparison of Three Page Replacement Algorithms: FIFO, LRU and Optimal," Acad. J. Interdiscip. Stud., 2015, doi: 10.5901/ajis.2015.v4n2s2p56. [3] S. Shastri, A. Sharma, and V. Mansotra, "Study of Page Replacement Algorithms and their analysis with C#," 2016. [4] G. P. Arya, D. Prasad, and S. S.

Rana, "An improved page replacement algorithm using block retrieval of pages," Int. J. Eng. Technol., 2018, doi: 10.14419/ijet.v7i4.5.20004. [5] B. A. T. V. L. Kolhe, "Analysis of Various Page Replacement Algorithms in Operating System," Int. J. Sci. Res., 2016. [6] A. Saxena, "A Study of Page Replacement Algorithms," Int. J. Eng. Res. Gen. Sci., vol. 2, no. 4, pp. 385–388, 2014. [7] G. for Geeks, "Geeks for Geeks," Page Replacement Algorithm, 2020.

<https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/> (accessed Oct. 08, 2020). [8] D. P. Juhi Kumari, Sonam Kumari, "A Comparison of Page Replacement Algorithms: A Survey," Int. J. Sci. Eng. Res., vol. 7, no. 12, pp.

57–62, 2016. [9] R. R. Hitha Paulson, "Page Replacement Algorithms - Challenges and Trends," Int. J. Comput. Math. Sci., vol. 6, no. 9, pp. 112–116, 2017. BIOGRAPHY

<Optional> Authors have the option to publish a biography together with the paper, with the academic qualification, past and present positions, research interests, awards, etc. This increases the profile of the authors and is well received by international reader.

INTERNET SOURCES:

<1% -

<https://www.scribd.com/document/162385632/Data-Diffusion-Dynamic-Resource-Provision-And>

1% -

https://www.researchgate.net/publication/282465640_A_Comparison_of_Three_Page_Replacement_Algorithms_FIFO_LRU_and_Optimal

<1% -

<https://www.geeksforgeeks.org/program-page-replacement-algorithms-set-2-fifo/>

<1% - <https://research.ijcaonline.org/volume91/number16/pxc3895350.pdf>

<1% -

http://smithict.weebly.com/uploads/9/5/5/0/9550779/section.1.10_hardware_spec.pdf

<1% - <https://technology.blurtit.com/3545809/what-is-the-purpose-of-main-memory>

<1% -

<https://afteracademy.com/blog/what-is-virtual-memory-and-how-is-it-implemented>

1% -

https://www.researchgate.net/publication/261836721_An_Effective_Round_Robin_Algorithm_using_Min-Max_Dispersion_Measure

3% -

<https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/>

<1% - <https://study.com/academy/lesson/page-replacement-definition-algorithms.html>

1% - <https://www.youtube.com/watch?v=QbbOb8SvAPk>

<1% -

https://www.researchgate.net/publication/338026849_A_page_replacement_algorithm_based_on_a_fuzzy_approach_to_improve_cache_memory_performance

1% - <https://technobyte.org/page-replacement-algorithms-in-os/>

1% - https://en.wikipedia.org/wiki/Page_replacement_algorithm

<1% -

<https://www.ijser.org/researchpaper/A-Comparison-of-Page-Replacement-Algorithms-A-Survey.pdf>

<1% -

<https://www.slideshare.net/vtunotesbysree/solution-manual-of-operating-system-concepts-by-abraham-silberschatz-peter-baer-galvin-greg-gagne>

<1% -

https://www.researchgate.net/publication/319997969_Page_Replacement_Algorithms_-_Challenges_and_Trends

<1% -

<http://www.expertsmind.com/questions/explain-the-different-page-replacement-policies-30162918.aspx>

<1% -

<http://the-eye.eu/public/Books/robot.bolink.org/Simulation%20of%20Page%20Replacement%20Algorithms.pdf>

<1% -

<https://stackoverflow.com/questions/6398811/can-a-tlb-hit-lead-to-page-fault-in-memory>

<1% -

https://www.researchgate.net/publication/220788195_LFU-K_An_Effective_Buffer_Management_Replacement_Algorithm

<1% - <http://www.theijes.com/Vol,5,Issue,1.html>

1% -

https://www.researchgate.net/publication/327964913_An_Improved_Page_Replacement_Algorithm_Using_Block_Retrieval_of_Pages

1% - <https://www.medsci.org/ms/author>



UNIVERSITAS BHAYANGKARA JAKARTA RAYA
FAKULTAS ILMU KOMPUTER

Kampus I: Jl. Harsono RM No.67, Ragunan, Pasar Minggu, Jakarta Selatan 12550
Telepon: (021) 27808121 - 27808882
Kampus II: Jl. Raya Perjuangan, Bekasi Utara, Jawa Barat
Telepon: (021) 88955882 Fax.: (021) 88955871
Web: www.ubharajaya.ac.id/fasilkom/. Email: fasilkom@ubharajaya.ac.id

SURAT TUGAS

Nomor : ST/003/IX/2020/B-FASILKOM-UBJ


1. Dasar: Kalender Akademik Ubhara Jaya Tahun Akademik 2020/2021.
2. Dalam rangka mewujudkan Tri Dharma Perguruan Tinggi untuk Dosen di Universitas Bhayangkara Jakarta Raya maka dihimbau untuk melakukan penelitian.
3. Sehubungan dengan hal tersebut diatas, maka Dekan Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya menugaskan:

No.	NAMA	NIDN	JABATAN
1.	Tri Dharma Putra, S.T., M.Sc	302117101	Dosen Tetap Prodi Informatika
2.	Rakhmat Purnomo, S.Pd., S.Kom., M.Kom	032210820	Dosen Tetap Prodi Informatika
3.	Wowon Priatna, S.T., MTI	0429118007	Dosen Tetap Prodi Informatika

Membuat Artikel Ilmiah dengan judul '*Analysis of Random Page Replacement Algorithm in Operating System*' pada Jurnal International Journal of Advance Research in Computer and Communication Engineering. Vol.9, Issue 10, October 2020. ISSN: 2319-5940, E-ISSN: 2278-1021, October 2020.

4. Demikian penugasan ini agar dapat dilaksanakan dengan penuh rasa tanggung jawab.

Jakarta, 10 September 2020
DEKAN FAKULTAS ILMU KOMPUTER


HERLAWATI, S.Si., M.M., M.Kom.
NIP: 2001 452

Paraf:

1. Ka. Prodi Informatika 

Analysis of Random Page Replacement Algorithm in Operating System

Tri Dharma Putra¹, Rakhmat Purnomo², Wowon Priatna³

Department of Informatics, Faculty of Computer Science,

University of Bhayangkara Jakarta Raya Jalan Perjuangan Bekasi Utara, West Java, Indonesia^{1,2,3}

Abstract: Page replacement algorithms choose pages to swap out from memory, namely when a new page needs allocation memory. Several page replacement algorithms proposed by experts. Four well known page replacement algorithms are Least Recently Used (LRU), Optimal Page Replacement (OPR), First In First Out (FIFO), Random page replacement algorithms. Each one has the objective to minimize the page fault. If the minimum page fault achieved, the performance of the process increased. The algorithm discussed is random page replacement algorithm. In this paper three case studies of random page replacement algorithm are discussed thoroughly.

Keywords: Random page replacement algorithm, page fault, page replacement algorithm, FIFO, LRU, OPR

I. INTRODUCTION

Memory management is the most important part of operating system in which all activities related to the memory are perform [1]. The replacement of page is the main concept. Pages is required to be in the main memory [2]. Computer system objective is to execute program of different sizes. The program should be in main memory when program is executed. But main memory has limited size to accommodate programs which is large. Execution of program that may not be entirely in main memory is called virtual memory [3]. Virtual memory paved the way to execute program that is larger than main memory. The process in which we provisionally moves process from primary memory to the hard disk or secondary memory, so that the memory available for other processes, which is known as swapping [4]

On operating systems which use paging for memory management, it is needed a page replacement algorithm to decide which pages to replace when a new page is entered. Good replacement can reduce the cost of page fault that result in higher performance of the system [5].

Page Fault occurs when a process references to a page that is not present in the main memory and needs to be through from secondary memory [6]. When a program currently executing, accessing a memory page that is mapped to the virtual address space, but is not loaded into physical memory. Because actual physical memory is much smaller than virtual memory, a page fault occurs. In this case, the operating system may have to replace one of the existing pages with the new page it needs. There are several page replacement algorithms for this replacement. One of them is the random page replacement algorithm [7].

II. LITERATURE SURVEYS

There are several algorithms of page replacement proposed by experts, namely :

1. First in First out (FiFO): This is the simplest page changing algorithm. In this algorithm, the operating system keeps track of all the pages in memory that are in the queue, namely page that has been in front of the queue the longest. When a page has to be replaced the page at the front of the queue is chosen to be deleted. An average of hit ratio of 50 percent is reported by FIFO algorithm. The advantages and disadvantages of FIFO algorithm: Easy to understand, very easy to be implemented, Performance not always good [8].
2. Optimal Page Replacement: In this algorithm, pages are replaced which will not be used for the next longest duration of time. Optimal page replacement is a good algorithm, but it is impossible to implement practically, because the operating system will not know the next request. The use of this algorithm is to create benchmarks so that other replacement algorithms can be analyzed against this algorithm. On this algorithm, system select victim whole reference is furthest. This algorithm has the best page fault behavior reported [9].
3. Least Recently Used: In this algorithm, the page will be replaced the least used (least recently used). An average of 60 % hit ratio is reported by LRU algorithm. This algorithm selects the victim page as that which is not been demanded for access for along time [9].