Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 26 (2023), No. 5, pp. 1519–1533
DOI : 10.47974/JDMSC-1816

Operating system security with discrete mathematical structure for secure round robin scheduling method with intelligent time quantum

Anil Kumar [†]
Department of Computer Science & Engineering
Poornima Institute of Engineering & Technology
Jaipur
Rajasthan
India

Amit Kumar Gupta [*]
Deepak Panwar [§]
Sandeep Chaurasia [‡]
Department of Computer Science & Engineering
Manipal University Jaipur
Jaipur
Rajasthan
India

Dinesh Goyal [@]
Department of Computer Science & Engineering
Poornima Institute of Engineering & Technology
Jaipur
Rajasthan
India

Abstract

These days, the problems of scheduling and security are increasingly pressing as a result of the rapidly growing number of implementations that are operated in operating systems. Process scheduling and scheduling security is the most effective methodology of operating a system which can improve and degrade the performance of the computer

[†] *E-mail:* anilkumar@poornima.org
[*] *E-mail:* dramitkumargupta1983@gmail.com (Corresponding Author)
[§] *E-mail:* deepak.panwar@jaipur.manipal.edu
[‡] *E-mail:* sandeep.chaurasia@jaipur.manipal.edu
[@] *E-mail:* dinesh.goyal@poornima.org

© Taru Publications

system. The performance of scheduling algorithm in operating system can be measured by scheduling principles like average waiting time, average turnaround time, response time, throughput etc. Much research carried out over operating system security and round robin scheduling used in time sharing operating system in previous years to resolve the issues regarding to improvement in the performance of round robin (RR) scheduling and improving the security in the scheduling approach. This study discussed the security issues in operating systems, scheduling security and discrete structure of a new approach for RR scheduling. The researcher has suggested a list of policy for scheduling method security and proposed a new approach for RR scheduling and investigated the scheduling principles. The proposed methodology covers distinctive strategy for selection manner of process from ready queue than RR and used diverges time quantum. The proposed methodology has been validated experimentally by comparing with a numerous scheduling algorithm namely as Round Robin (RR), Adaptive Round Robin (ARR), Round Robin Remaining time (RRRT), improved Round Robin (IRR), an additional improvement on the improved Round Robin (AAAIRR), and an Enhanced Round Robin (ERR). Numerous simulations have been conducted to validate the effectiveness of the proposed algorithm to compelling the performance of system. At the conclusion the performance of computing system is dependent on security of operating system as well as secluding methods.

Subject Classification: *68M25,68N25.*

Keywords: *Intelligence system, Round robin, Security of operating system, Computation model, Average waiting time, Turnaround time, Context switch, Scheduling security.*

1. Introduction

Since the operating system plays such an important function in the execution of any computer system, the security (or inability thereof) of an operating system has a significant influence on the aggregate security of a software system, along with the protection of all software attempting to run within the system. An agreement of the underlying operating system will undoubtedly disclose any implementation executing the system at risk. Inadequate control and quarantine of individual software execution in an operating system may result in an intrusion or break-in from one application to another [26]. Operating system creates process to perform the task inputted by the user in the computer system [1]. From the ready where many processes are available to acquire the processor for execution are selected by the short-term scheduler on behalf of policy of scheduling methods. These scheduling algorithms like first come first serve (FCFS), shortest job first (SJF), priority and round robin (RR) [2]. The scheduling methods are evaluated their performance in terms of scheduling principles namely defined as waiting time (how much time a process has been wait to acquire the processor), turnaround time (how

much time a process takes to finish its complete execution cycle), context switching (how much time it is allocated or deallocated over the processor), response time (when the process get first response from the processor), throughput (how many process has been finished their execution over the processor) and utilization of processor (how much a processor is utilized). RR scheduling method is widely used procedure for preemptive method, which is implemented in the real time OS and time-sharing OS [3]. The procedure to deallocating a process from the processor and sent back to ready queue is commonly refers to the context switching [4]. RR allocates the processor to the processor for a constant time which is also known as time quantum (TQ) or time slice [5]. If the value of TQ is taken low, then NCS will occur in large range which includes more overheads and if the value of TQ is taken high, then the RR which implemented with FCFS will suffers from the high probability of the starvation problem. [6]. In this research the researchers have discussed the operating system (OS) security which allied with the performance of the computing system and proposed a state of art new RR approach.

2. Related work

Authors have reviewed numerous numbers of research papers to define the OS security and designing the new method for the scheduling. According to the US government's "Trusted Computer System Evaluation Criteria" the security [27]. The various include such as defined the neural network for calculation of optimal value of TQ [7]. Novel approach which has included the term instantaneous priority [8]. An approach which defines the term median and response ratio [9]. RR method implemented with K- means clustering approach Each cluster has define the value of TQ based on weight of the cluster [10]. Method for RR in which predefined threshold (TSH) is used and it select randomly value of TQ [11]. The list of processes divided in to the two halves and average of PECT of two halves. The final value of TQ is calculated based on the sum of two relative values calculated [12]. This approach used three different types of TQ depending upon the number of processes in the waiting queue. [13]. Two approaches to schedule the processes in cloud computing. First is FCFS with the SJF policy and another method implemented RR with cloud sim [14]. This paper used the concept of SJF in RR [15]. The first round of RR will have the value of TQ equal to the mean value of PECT [16]. Processes are arranged as per the process priority and the value of TQ is equal to shortest PECT in the process model [17]. Comparative study of FCFS, SJF, Priority and

RR [18]. Longest job first (LJF) is combined with combinational burst time (CBT) [19]. "An Additional Improvement in RR (AAAIRR)" [20]. Comparative analysis of RR, IRR, ERR, ARR, RRRT, AAAIRR, MMRR, IRRVQ, AMRR [21-25].

3. Operating System Security

Most modern computer systems support the multiple simultaneous successful implementations of numerous applications (concurrent execution) on a single physical computing hardware (which may contain multiple processing units). Individual solutions in these multitasking and time-sharing environments share the same system resources, such as CPU, disc, memory, and so on, under the control of the operating system. Most modern operating systems implement some abstract property of containment, such as process (or task) and TCB (Task Control Block), virtual memory space, file, port, and IPC (Inter Process Communication), etc., to protect the execution of individual application jobs from possible interference and attack by other jobs [26-28]. The capacity to communicate these reserves created a requirement for security. According to Adopters, an operating system may provide security along the following spectrum [29] **No security**: This is suitable when responsive practices are run at different times. **Isolation**: This approach assumes that each process is functioning independently of the others, with no sharing or communication. **Share everything or nothing**: The entrepreneur of an object (for example, a file or memory segment) declares whether it is public or private. **Share via access limitation**: The operating system validates each access by a particular user to a specific object. As a result, the operating system acts as an intermediary between users and objects, ensuring that only authorized access occurs. **Share via dynamic capabilities**: This extends the concept of access control to allow for the energetic creation of object sharing rights. **Limit use of an object**: This type of protection restricts not only access to an object but also its use [30-32].

4. Security of operating system scheduling approach

Securing the operating system scheduling algorithm is an important aspect of ensuring the overall security and reliability of the system. Here are some steps that can be taken to secure the operating system scheduling algorithm: **Use secure and reliable scheduling algorithms**: Operating system scheduling algorithms should be chosen based on their ability to

ensure fair allocation of system resources while also preventing malicious users from exploiting the system. The scheduler should be able to prevent denial of service attacks and ensure that the system is not vulnerable to other forms of attack. **Limit user privileges:** Users should be given only the necessary privileges to perform their tasks. This limits the amount of damage that can be caused by a malicious user who gains access to the system. **Use access controls:** Access controls can be used to limit the access of users to the system. This includes limiting access to system resources, such as the CPU, memory, and network. **Regularly update and patch the operating system:** Regular updates and patches should be installed to ensure that the operating system is up to date with the latest security patches. This reduces the likelihood of vulnerabilities being exploited. **Use intrusion detection and prevention systems:** Intrusion detection and prevention systems can be used to detect and prevent attacks on the system. These systems can monitor the system for suspicious activity and alert system administrators to potential security threats. **Monitor system logs:** System logs can be used to monitor the activity on the system. This can help detect and investigate suspicious activity and potential security threats. **Implement strong authentication and password policies:** Strong authentication and password policies can prevent unauthorized access to the system. This includes enforcing the use of strong passwords, limiting the number of failed login attempts, and requiring multi-factor authentication. By taking these steps, the operating system scheduling algorithm can be secured, and the overall security and reliability of the system can be improved.

5. Proposed Methodology for Secure RR Scheduling

It is investigated that RR performance enhancement and degradation depends upon three key factors which are first, selection of the value of TQ second, order of selection of the processes from the ready queue and third, fair allocation of processor among the processes to improve the security of scheduling algorithm. On the basis of two key factors the researchers have identified the following objective to represent the research.

1. It is needed to define the policy for the scheduler to achieve security at the level of scheduling algorithm.
2. It is needed to define the new RR methodology for calculation of optimal value of TQ for which the performance of the scheduling can be enhanced.

3. It is needed to define the new RR methodology of scheduling of the process in different manner (other than FCFS) for which the performance of the scheduling can be enhanced.
4. It is needed to develop a new optimized round robin scheduling algorithm which can minimize the average waiting time, average turnaround waiting time and number of contexts switching.

a. *Proposed Algorithm*

The proposed methods define two level of implementation in first phase the scheduler implements the security policies which have been discussed in section 4 and in second phase it proposed state of art method Novel secure Round Robin Scheduling with intelligent Time Quantum finds the optimal process based on PECT value of the processes. The pseudo code for the state of art method Secure Novel Round Robin Scheduling with intelligent Time Quantum is defined in the below.

Proposed Algorithm

1. *Input the process (P) with the process execution cycle (PECT) and arrival time (AT).*
2. *Arrange all the process in the SJF order and calculate (TET) = ΣPECT.*
3. *Set Remaining time (RT) for each process is equal to its PECT value.*
4. *Repeat steps until total execution time (TET) =0*
   a. *Calculate number of process (n) whose RT>0*
   b. *Arrange all the process in the list of processes increasing order of their RT.*
   c. *Remove the process from the process list if any process has RT=0*
   d. *Calculate the median (MD) and (MEAN) of PECT whose RT>0.*
   e. *Calculate diverges Time Quantum (DTQ)=(MD+MEAN)/2*
   f. *Select process (P_i) from the process list and allocate the processor for the value of DTQ and set*

      $RT(P_i)=PECT(P_i)-DTQ$, TET=TET-DTQ, NCS=NCS+1

      If $RT(P_i)$ <=DTQ then

         Reallocate the processor to P_i

         $RT(P_i)=0$, TET=TET-$RT(P_i)$

      Else

$RT(P_i) = RT(P_i)\text{-}DTQ$, TET=TET-$RT(P_i)$, NCS=NCS+1

Update the waiting time (WT) and turnaround time (TAT) of P_i

g. Repeat step (g) for all processes which are available in the list of process.

5. *Calculate Average Waiting Time (AWT) and Average Turnaround Time (ATAT)*
6. ***Return AWT, ATAT and NCS.***

b. *Mathematical Model for calculation of AWT, TAT and NCS*

This section has been described the mathematical model for calculation of the value of AWT, TAT and NCS for scheduling method RR [33].

Parameters Definitions:

Set of processes is defined as: $P=\{P_i \mid \text{for } i=1,2,3,\ldots\ldots\ldots\ldots N\}$ (1)

$\delta(P)$ is set of process execution time defined as: $\delta(P)=\{\delta(P) \mid \text{for } i=1,2..N\}$ (2)

€(P) is set of arrival time defined as: $€(P)=\{ €(P) \mid \text{for } i=1,2\ldots..N\}$ (3)

Process model defined as $\eta(P, \delta(P), €(P))=\{(P_i, \delta(P_i), €(P_i)) \mid \text{for } i=1,2..N\}$ (4)

(i) *Arrangement of processes:*

Sort the set of processes according to their time (which is requiring completing it) in the increasing order of their values.

Let $N, K \in \mathbb{N}$ and for all $1 \le i \le N$

let $\varphi = (P_i,\ \delta(P_i),\ €(P_i))_N^{i=1}$ be a set of tuple of process model

$\eta(P, \delta(P), €(P))$

Let $\gamma = (P_i',\ \delta(P_i'),\ €(P_i'))_N^{i=1} = sort(\varphi).$

We define $\varphi^{(K)}$ to be the set of tuple $\varphi^{(K)} = (\lambda_i)_K^{i=1}$ #(5)

of process model $\eta(P, \delta(P), €(P))$ satisfying, for all $1 \le i \le K$,

$\lambda_i = (P'_{N-K+i},\ \delta(P'_{N-K+i}),\ €(P'_{N-K+i}))$

Calculation of Average Waiting Time, Average turnaround Time and Number of Context Switching using RR:

Let ω, τ and χ are average waiting time, average turnaround time and number of context switch respectively

Let ψ is the time quantum

let k_i; $i = 1, 2, 3, \ldots, n$, number of cycles required for completion for process P_i

calculate ω, τ and χ for the value of ψ using the *RR* method

$$k_i = \begin{cases} \dfrac{\delta(P_i)}{\psi} & \text{if } \delta(P_i) \bmod \psi = 0 \\ \left\lfloor \dfrac{\delta(P_i)}{\psi} \right\rfloor + 1 & \text{otherwise} \end{cases} \qquad \#(6)$$

Remaining CPU burst time is updated after every cycle. Let $\delta(P_{i,j})$ and $\psi_{i,j}$ denote the remaining CPU burst and executed portion of CPU burst, respectively, for the i^{th} process in j^{th} cycle. Hence, following expression can be used to evaluate the remaining CPU time requirement of a process P_i in the k^{th} cycle :

$$\delta(P_{i,k}) = \delta(P_{i,k-1}) - \psi_{i,k-1} \qquad \#(7)$$

Where $\psi(i, k-1)$ can be obtained from the following :

$$\psi_{i,k-1} = \begin{cases} \delta(P_{i,k-1}) & \text{if } \delta(P_{i,k-1}) < \psi \\ \psi & \text{otherwise} \end{cases} \qquad \#(8)$$

Turnaround Time Calculation: Let is turnaround time of i[th] process.

$$\tau_i = \sum_{l=1}^{j-1} \left[\sum_{k=1}^{n} \psi_{k,l} \right] + \sum_{k=1}^{i} \psi_{k,j} - €(P_i) \qquad \#(9)$$

Waiting Time Calculation: Let is waiting time of i[th] process.

$$\omega_i = \sum_{k=1}^{i} \psi_{k,1} + \left[\sum_{k=i+1}^{n} \psi_{k,1} + \sum_{k=1}^{i-1} \psi_{k,2} \right] + \left[\sum_{k=i+1}^{n} \psi_{k,2} + \sum_{k=1}^{i-1} \psi_{k,3} \right] + \ldots\ldots\ldots + \left[\sum_{k=i+1}^{n} \psi_{k,j-1} + \sum_{k=1}^{i-1} \psi_{k,j} \right] - €(P_i) \quad (10)$$

Number of Context Switch (χ) :

$$\chi = \sum_{i=1}^{N} \left\lfloor \frac{\delta(P_i)}{\psi} \right\rfloor + 1 \qquad \#(11)$$

6. Experiments and Validation

This section described the experiments which are performed by researchers to validate his proposed method. The six-case study has been shown. Six process model define in the Table 1. The Table 2 shows the efficiency parameter comparison of proposed method and AIRR, ARR, ERR, IRR, RR and RRRT. Figure 1 shows graphical representation of the efficiency parameter comparison of proposed method and AIRR, ARR, ERR, IRR, RR and RRRT.

Table 1

Processes with CPET

Case Study-1			Case Study-2			Case Study-3		
P1	0	24	P1	0	45	P1	0	18
P2	0	40	P2	0	36	P2	0	38
P3	0	5	P3	0	30	P3	0	6
P4	0	12	P4	0	18	P4	0	28
P5	0	34	P5	0	10	P5	0	21
Case Study-4			Case Study-5			Case Study-6		
P1	0	15	P1	0	70	P1	0	20
P2	0	19	P2	0	53	P2	0	55
P3	0	23	P3	0	26	P3	0	38
P4	0	47	P4	0	20	P4	0	24
P5	0	56	P5	0	15	P5	0	40

6. Conclusion

In this research the researcher has been proposed two level of implementation, in the first phase define the security policies (Use secure and reliable scheduling algorithms, Limit user privileges, Use access controls, Regularly update and patch the operating system, Use intrusion detection and prevention systems, Monitor system logs, Implement strong authentication and password policies) for the scheduler and in second phase state of art method Secure Novel Round Robin Scheduling with intelligent Time Quantum. The proposed method basically finds the optimal job from the pool of available jobs based on burst time and scheduled the process for a time quantum that is calculated dynamically which is namely called Diverges time quantum. This algorithm shall

Table 2
Output using Proposed Method and AIRR, ARR, ERR, IRR, RR and RRRT

Case Study-1					Case Study-2					Case Study-3				
Method	**AWT**	**TAT**	**NCS**	**TQ**	**Method**	**AWT**	**TAT**	**NCS**	**TQ**	**Method**	**AWT**	**TAT**	**NCS**	**TQ**
Proposed Method	27.6	50.6	5	23.5	Proposed Method	38	65.8	5	28.9	Proposed Method	29.6	51.8	5	21.6
AAAIRR	36.2	59.2	8	12	AAAIRR	51.8	79.6	7	15	AAAIRR	32	54.2	6	14
ARR	32.4	55.4	7	24	ARR	44	71.8	7	30	ARR	33.8	56	7	21
ERR	43.4	66.4	8	12	ERR	68.8	96.6	7	15	ERR	41.8	64	6	14
IRR	46.8	69.8	9	12	IRR	73.8	101.6	9	15	IRR	45.2	67.4	7	14
RR	49.2	72.2	11	12	RR	87.2	115	11	15	RR	59	81.2	10	14
RRRT	36.4	59.4	10	11	RRRT	45.8	73.6	9	13	RRRT	31.8	54	8	11
Case Study-4					**Case Study-5**					**Case Study-6**				
Algorithm	**AWT**	**TAT**	**NCS**	**TQ**	**Algorithm**	**AWT**	**TAT**	**NCS**	**TQ**	**Algorithm**	**AWT**	**TAT**	**NCS**	**TQ**
Proposed Method	42	74	6	27.5	Proposed Method	45	81.8	6	31.4	Proposed Method	53.6	89	5	36.7
AAAIRR	46	78	7	20	AAAIRR	68.4	105.2	7	25	AAAIRR	66.8	102	6	22
ARR	51.2	83.2	9	23	ARR	55.4	92.2	9	26	ARR	61.2	96.6	7	38
ERR	46	78	7	20	ERR	87.8	124.6	7	25	ERR	69.6	105	6	22
IRR	46	78	7	20	IRR	87.8	124.6	7	25	IRR	69.6	105	6	22
RR	58	90	10	20	RR	109.4	146.2	10	25	RR	96	131	10	22
RRRT	45.2	77.2	8	16	RRRT	48.6	85.4	8	18	RRRT	63.8	99.2	9	17

Figure 1

Comparison Graph between Proposed Method and AIRR, ARR, ERR, IRR, RR and RRRT.

overcome the issues prevalent in older processor scheduling algorithm and enhanced the processor scheduling performance principles too. The researcher has also compared the proposed scheduling method with the RR, ARR, RRRT, IRR, AAAIRR, ERR. The researcher has conducted the experiments on the proposed scheduling method and the RR, ARR, IRR, RRRT, AAAIRR, ERR and concluded that the proposed scheduling method provide the better results than these in terms of minimizing decreasing

Average waiting time, Average Turnaround Time, Number of Context Switches. The researcher also inferred that the proposed method is also solving the problem security issues towards the scheduling algorithm and improvement in RR by selection of time quantum by using varying time quantum method.

References

[1] Chandiramani, K.; Verma, R.; Sivagami, M. A Modified Priority Preemptive Algorithm for CPU Scheduling. Procedia Comput. Sci., 165, 363–369 (2019).

[2] Silberschatz, A.; Galvin, P.B.; Gagne, G. Operating System Concepts, 10th ed.; Wiley Publ.: Hoboken, NJ, USA (2018).

[3] Sunil, J.G.; Anisha Gnana, V.T.; Karthija, V. Fundamentals of Operating Systems Concepts; Lambert Academic Publications: Saarbrucken, Germany (2018).

[4] Shafi, U.; Shah, M.; Wahid, A.; Abbasi, K.; Javaid, Q.; Asghar, M.; Haider, M. A Novel Amended Dynamic Round Robin Scheduling Algorithm for Timeshared Systems. *Int. Arab. J. Inf. Technol.*, 17, 90–98 (2019).

[5] Tajwar, M.M.; Pathan, N.; Hussaini, L.; Abubakar, A. CPU Scheduling with a Round Robin Algorithm Based on an Effective Time Slice. *J. Inf. Process. Syst.*, 13, 941–950 (2017).

[6] Mostafa, S.M.; Rida, S.Z.; Hamad, S.H. Finding Time Quantum of Round Robin CPU Scheduling Algorithm in General Computing Systems Using Integer Programming. *Int. J. New Comput. Archit. Appl.*, 5, 64–71 (2010).

[7] Zouaoui, S., Boussaid, L., & Mtibaa, A. Improved time quantum length estimation for round robin scheduling algorithm using neural network. *Indonesian Journal of Electrical Engineering and Informatics* (IJEEI), 7(2), 190–202 (2019). https://doi.org/10.11591/ijeei.v7i2.464.

[8] Pandey, A., Singh, P., Gebreegziabher, N. H., & Kemal, A. Chronically Evaluated Highest Instantaneous Priority Next: A Novel Algorithm for Processor Scheduling. *Journal of Computer and Communications*, 04(04), 146–159 (2016). https://doi.org/10.4236/jcc.2016.44013.

[9] Nie, B., Du, J., Xu, G., Liu, H., Yu, R., & Wen, Q. (n.d.). A New Operating System Scheduling Algorithm. 5.

[10] Mostafa, S. M., & Amano, H. Dynamic Round Robin CPU Scheduling Algorithm Based on K-Means Clustering Technique. Applied Sciences, 10(15), 5134 (2020). https://doi.org/10.3390/app10155134.

[11] Mostafa, S., & Amano, H. An Adjustable Round Robin Scheduling Algorithm in Interactive Systems. Information Engineering Express, 5(1), 11–18 (2019). https://doi.org/10.52731/iee.v5.i1.353.

[12] Gull, H., Iqbal, S. Z., Saeed, S., Alqatani, M. A., & Bamarouf, Y. Design and Evaluation of CPU Scheduling Algorithms Based on Relative Time Quantum: Variations of Round Robin Algorithm. *Journal of Computational and Theoretical Nanoscience*, 15(8), 2483–2488 (2018). https://doi.org/10.1166/jctn.2018.7524.

[13] Gibet Tani, H., & El Amrani, C. Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data. Journal of Data Mining & Digital Humanities, Special Issue on Scientific..., 3104 (2018). https://doi.org/10.46298/jdmdh.3104.

[14] Gibet Tani, H., & El Amrani, C. Cloud Computing CPU Allocation and Scheduling Algorithms using CloudSim Simulator. *International Journal of Electrical and Computer Engineering* (IJECE), 6(4), 1866 (2016). https://doi.org/10.11591/ijece.v6i4.10144.

[15] Elmougy, S., Sarhan, S., & Joundy, M. A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud Computing*, 6(1), 12 (2017). https://doi.org/10.1186/s13677-017-0085-0

[16] Lipika Datta,"Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice", *IJEME*, vol.5, no.2, pp.10-19 (2015). DOI: 10.5815/ijeme.2015.02.02.

[17] Chandiramani, K., Verma, R., & Sivagami, M. A Modified Priority Preemptive Algorithm for CPU Scheduling. Procedia Computer Science, 165(2019), 363–369 (2019). https://doi.org/10.1016/j.procs.2020.01.037.

[18] Ali, S. M., Alshahrani, R. F., Hadadi, A. H., Alghamdi, T. A., Almuhsin, F. H., & El-Sharawy, E. E. A Review on the CPU Scheduling Algorithms: Comparative Study. *International Journal of Computer Science and Network Security*, 21(1), 19–26 (2021). https://doi.org/10.22937/IJCSNS.2021.21.1.4

[19] Abdullahi, I., & B Junaidu, S. Empirical Framework to Mitigate Problems in Longer Job First Scheduling Algorithm LJF+CBT.

International Journal of Computer Applications, 75(14), 9–14 (2013). https://doi.org/10.5120/13178-0763.

[20] Abdulrahim, A., E. Abdullahi, S., & B. Sahalu, J. A New Improved Round Robin (NIRR) CPU Scheduling Algorithm. *International Journal of Computer Applications*, 90(4), 27–33 (2014). https://doi.org/10.5120/15563-4277.

[21] ElDahshan, K., Elkader, A. A., & Ghazy, N. Round Robin based Scheduling Algorithms, A Comparative Study. 17(2), 14 (2017).

[22] Kumar Mishra, M., & Rashid, F. An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum. *International Journal of Computer Science, Engineering and Applications*, 4(4), 1–8 (2014). https://doi.org/10.5121/ijcsea.2014.4401.

[23] Khatri, J. An Enhanced Round Robin CPU Scheduling Algorithm. *IOSR Journal of Computer Engineering*, 18(04), 20–24 (2016). https://doi.org/10.9790/0661-1804022024.

[24] Hiranwal, S., & Roy, D. K. C. Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice. 2(2), 5 (2011).

[25] Sharma, Kakhani, A., Gaurav. Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm. *International Journal of Computer Science and Mobile Computing* (2015).

[26] Cui-Qing Yang, Operating System Security and Secure Operating Systems Version 1.4b, , GIAC practical repository, Option 1 for GSEC January 2003, SANS Institute 2003, Operating System Security and Secure Operating Systems (giac.org) (2003).

[27] DOD 5200.28-STD, "DOD Trusted Computer System Evaluation Criteria" (Orange Book), 26 December 1985, http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf.

[28] Chris Dalton and Tse Huong Choo, "An Operating System Approach to Securing E-Services", Communications of the ACM, V. 44, No. 2, p. 58 (2001).

[30] Ipsita Vashista, Harsha Suri, Disha Papneja, Operating Systems Security – A Review, IJIRT | Volume 1 Issue 5 (2014). ISSN : 2349-6002, IJIRT100350_PAPER.pdf.

[31] Amit Kumar Gupta, Pushpa Gothwal, Dinesh Goyal & Carlos M. Travieso-Gonzalez. IoT-Galvanized pandemic special E-Toilet for generation of sanitized environment, *Journal of Discrete Mathematical Sciences and Cryptography* (2022), DOI: 10.1080/09720529.2022.2068607.

[32] Amit Kumar Gupta, Vijander Singh, Priya Mathur & Carlos M. Travieso-Gonzalez. Prediction of COVID-19 pandemic measuring criteria using support vector machine, prophet and linear regression models in Indian scenario, *Journal of Interdisciplinary Mathematics* (2020), DOI:https://doi.org/10.1080/09720502.2020.1833458.

[33] Jitendra Singh Yadav, Amit Kumar Gupta & Arjit Saxena. A review on gender identification using machine learning based on fingerprints, *Journal of Information and Optimization Sciences*, 40:5, 1121-1129 (2019), DOI: https://doi.org/10.1080/02522667.2019.1638002.

[34] Neetu Goel, Dr. Abhinav Saxena, Dr. R. B. Garg, An Optimal Mathematical Approach for CPU Scheduling – OMDRRS vs. RR, *International Journal of Computational and Applied Mathematics*. ISSN 1819-4966 Volume 12, Number 1, pp. 65-80 (2017).