دراسة مقارِنة لأحدث التطورات في خوارزمية الجدولة الدورانية

م.رشا الحجل* د.م. رأفة خازم**

(الإيداع:18 أيلول 2022،القبول:31 تشرين الأول 2022)

الملخص:

نظرا للأثر الكبير لخوارزمية الجدولة الدورانية في تحسين مدى الاستفادة من المعالج، يقوم هذا البحث بالتركيز على الإصدارات الحديثة المحسنة من هذه الخوارزمية.

يتناول هذا البحث تسع إصدارات حديثة لخوارزمية الجدولة الدورانية Round Robin من بينها إصداران حول الجدولة الدورانية في الحوسبة السحابية.

تتم المقارنة بين الخوارزميات الحديثة وكفاءتها وبين التقليدية واختبار الخوارزميات على نفس الإجرائيات (تم الاختبار والتنفيذ على اجرائيات لها أزمنة وصول مختلفة و رشقة معالج (burst time) مختلفة) وتحديد الخوارزمية الأمثل للعمل بهدف تحسين استغلالية المعالج (cpu utilization) من خلال تحسين معاملاتها الأساسية(متوسط وقت الانتظار (AWT)–عدد مرات تبديل السياق (NCS)– متوسط وقت الاستجابة (ART)– متوسط الفترة الزمنية للعمل(ATT)) .

وتحسين المعاملات الأساسية لوحدة المعالجة المركزية يتم من خلال تخفيض كل من:(average waiting time متوسط وقت الانتظار(AWT) – عدد مرات تبديل السياق (NCS)number of context switching – the متوسط الفترة الزمنية للعمل(ATT) average turnaround time)

وزيادة زمن الانتاجية Throughput time

تمت المقارنة على مرحلتين:

1–مجموعة إجراءات محددة (5 اجراءات) لها زمن وصول محدد مختلف و رشقة معالج مختلفة ومحددة.

2– مجموعة (100 اجرائية) بزمن وصول ورشقات معالج عشوائية تمت كتابة كود برمجي باستخدام لغة البرمجة python الخاصة بالخوارزميات المقترحة في البحث والتحقق منها وإظهار النتائج.

الكلمات المفتاحية: خوارزمية الجدولة الدورانية – استغلالية المعالج – رشقة المعالج.

*طالبة ماجستير في قسم هندسة الحواسيب والأتمتة –كلية الهندسة الميكانيكية والكهربائية –جامعة دمشق.

**مدرس في قسم هندسة الحواسيب والأتمتة –كلية الهندسة الميكانيكية والكهربائية –جامعة دمشق.

Comparetine study for last improvements of Round Robin algorithm

Eng.Rasha Alhajal* **Dr.Eng. Rafaa khazem****

(Received:18 September 2022,Accepted:31 October 2022)

Abstract:

Due to the significant impact of the rotational scheduling algorithm in improving the utilization of the processor, this research focuses on the modern improved versions of this algorithm. This paper deals with nine recent versions of Round Robin algorithm, including two versions about rotational scheduling in cloud computing.

A comparison is made between modern algorithms and their efficiency with traditional ones, and algorithms are tested on the same procedures (tested and implemented on procedures with different arrival times and different burst time) and the optimum algorithm was determined to work with the aim of improving utilization cpu by improving its basic parameters (AWT – NCS – ATT – ART).

Improving the main parameters of Cpu by reducing: (average waiting time (AWT) –number of context switches (NCS) – the average turnaround time (ATT)) and increase Throughput time.

The comparison took place in two stages:

1- A set of specific procedures (5 procedures) that have a different specific arrival time and a different specific burst time.

2- A set of (100 procedurals) with random latency and burst time the python codes for the proposed algorithms were written in the research, verified and showed the results.

Keywords: Round Robin algorithm – cpu utilization – burst time .

* Master's student in the Department of Computer and Automation Engineering – Faculty of Mechanical and Electrical Engineering – Damascus University.

**Professor in the Department of Computer and Automation Engineering – Faculty of Mechanical and Electrical Engineering – Damascus University.

1- المقدمة:

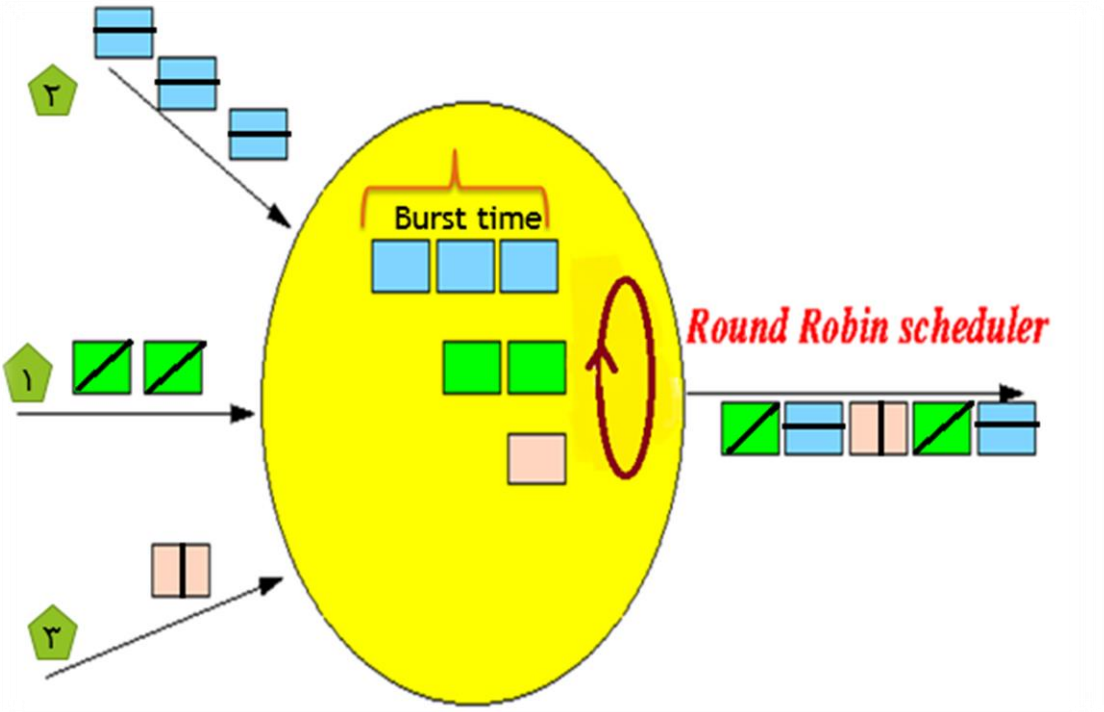
وُجدت خوارزميات الجدولة كنوع من التنظيم والاستغلال الأمثل للجهد والوقت وخاصة في علوم الحاسوب في عصر من التكنولوجيا والسرعة والاستفادة القصوى من الزمن وتم التركيز بشكل خاص على وحدة المعالجة المركزية واعتماد خوارزميات الجدولة كحل لتحسين معاملاتها ورفع أدائها. أثبتت خوارزمية الجدولة الدائرية فعاليتها في تحسين استغلالية المعالج cpu utilization من خلال تحسين معاملاتها الأساسية (AWT,NCS,ART ,ATT and Throughput time).

و مبدأ عمل خوارزمية ROUND ROBIN يقوم على:

تشريح زمن المعالج الى حصص زمنية تدعىquantum .

وعند استنفاذ الإجرائية للوقت المحدد لها يتم إيقافها وإخراجها من وحدة المعالجة المركزية وإعادتها إلى نهاية رتل الجاهزية (ready queue) و يتم تكرار العمل حتى تستنفذ الإجرائية رشقة المعالج المحددة لها.

من التطبيقات التي تستخدم فيها جدولة Round robin هي شبكات الحاسوب حيث تستخدم لجدولة حزم البيانات المرسلة في الشبكات الحاسوبية.



الشكل رقم (1):المبدأ العام لعمل خوارزمية Round Robin

ولكن تم اكتشاف عدة مشاكل تعيق عمل خوارزمية round robin وتختصر كما يلي:

1. كلما زاد الوقت المحدد (time quantum) فإننا نحصل على خوارزمية (FCFS).
2. كلما قل الوقت المحدد (time quantum) تزداد عدد مرات مقاطعة الإجرائية بعد دخولها الى وحدة المعالجة المركزية (زيادة NCS).

2- الهدف من البحث:

دراسة وتقييم عمل الخوارزميات الحديثة والمحسنة لخوارزمية الجدولة الدورانية التي ساهمت بحل المشكلتين الرئيسيتين لخوارزمية round robin وذلك من خلال الاختيار الأمثل لقيمة time quantum (TQ) ، وتحسين استغلالية المعالج cpu utilization بالإضافة الى تحقيق الأداء الأمثل ل CPU.

ويتم تحقيق الأداء الأمثل بتحسين معاملات خوارزميات الجدولة الأساسية :

1- متوسط وقت الانتظار average Waiting time (AWT) و يمثل الزمن المستغرق في بقاء الإجرائية في رتل الجهوزية (Queue Ready) لكي يخصص لها المعالج وقت التنفيذ.

2- متوسط وقت الاستجابة average Response time و يمثل المدة الزمنية منذ دخول الإجرائية إلى النظام لحين الاستجابة الأولى في التنفيذ.

3- عدد مرات تبديل السياق number of context switching(NCS)

4- متوسط وقت الإنتاجية average Throughput time تمثل عدد الإجرائيات المنفذة بشكل كامل خلال وحدة زمنية ويرتبط بمتوسط طول وظائف المجموعة.

5- متوسط الفترة الزمنية للعمل the average Turnaround time (ATT) (وقت العودة) يمثل المدة الزمنية من دخول الإجرائية للنظام والى حين اكتمال تنفيذ الإجرائية.

3- المواد وطرائق البحث:

تم استخدام طريقتين:

الأولى اعتمدت على تطبيق الإصدارات التسعة على خمس اجرائيات لها زمن وصول مختلف {0,3,4,8,10} و Burst time مختلف لكل إجرائية{18,23,10,35,14} ومن ثم حساب المعاملات (AWT – NCS – ATT – ART)

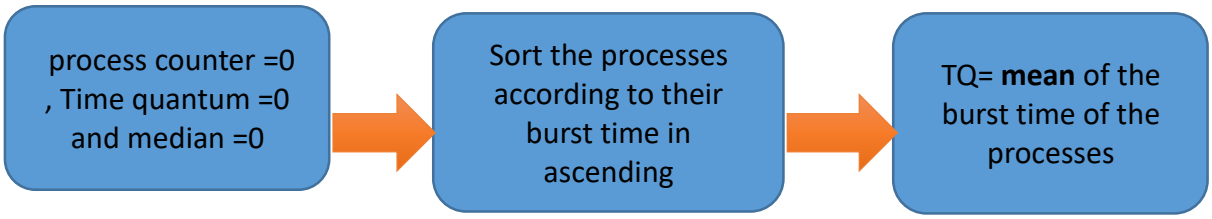
الثانية: باستخدام لغة البرمجة بايثون تمت كتابة code الخاص بكل خوارزمية ومن ثم تطبيق مجموعة من الاجرائيات وفي خرج code نحصل على المعاملين الأساسيين في تحديد الأداء الأفضل لل cpu وهما (AWT and ATT)

الدراسة المقارنة الأولى:

1- في الدراسة الأولى [8] التي تحمل العنوان A MODIFIED ROUND ROBIN CPU SCHEDULING ALGORITHM WITH DYNAMIC TIME QUANTUM يقوم الباحث باقتراح خوارزمية جديدة لجدولة وحدة المعالجة المركزية بكمية زمنية ديناميكية متغيرة، تعتمد على متوسط زمن رشقة المعالج للإجرائيات الموجودة في رتل الجاهزية (Ready Queue).

في كل دورة تنفيذ يتم تحديث قيمة رشقة المعالج بحساب متوسط زمن حتى تبقى خوارزمية واحدة فتنفذ حسب رشقة المعالج الخاصة بها، ويستمر التنفيذ حتى تنتهي كل الإجرائيات.

في الورقة البحثية تم التدقيق في نتائج المقارنة بين round robin (RR) و الخوارزمية المقترحة RRDTQ. وأثبتت هذه المقارنة أن خوارزمية RRDTQ المقترحة أكثر كفاءة من خوارزميات RR التقليدية نظرًا لأنه تكتسب متوسط وقت انتظار أقل ، ومتوسط وقت استجابة أقل ، وعددًا أقل من context switches من تلك الخاصة بـ RR . لذلك، تم توضيح الخوارزمية المقترحة لتحسين أداء أنظمة التشغيل متعددة المهام.



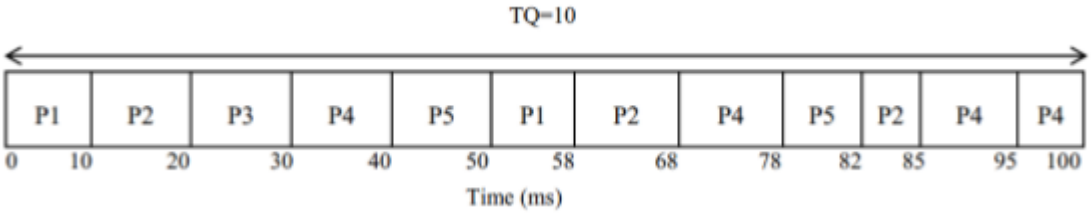
الشكل رقم (2):المخطط الصندوقي لخوارزمية RRDTQ

التنفيذ:

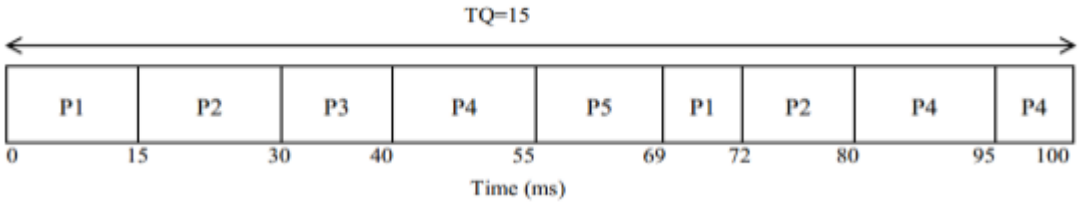
الجدول رقم (1): process with different arrival time

Process	Arrival time	Burst time
P1	0	18
P2	3	23
P3	4	10
K8P4	8	35
P5	10	14

RR ROUND ROBIN:



الشكل رقم (3): Gantt chart representation with TQ=10

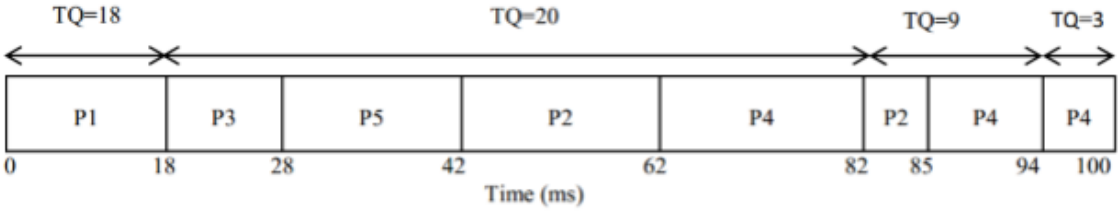


الشكل رقم(4): Gantt chart representation with TQ=15

تنفيذ الخوارزمية المقترحة في الورقة البحثية:

1- قيمة رشقة المعالج لكل جولة هي متوسط قيم رشقات المعالج للاجرائيات الموجودة في رتل الجاهزية.

2- يتم التنفيذ بدءا من الاجرائية التي تملك رشقة المعالج الاصغر في رتل الجاهزية



الشكل رقم (5):Gantt chart representation of proposed RRDTQ algorithm

النتائج بحسب الورقة البحثية:

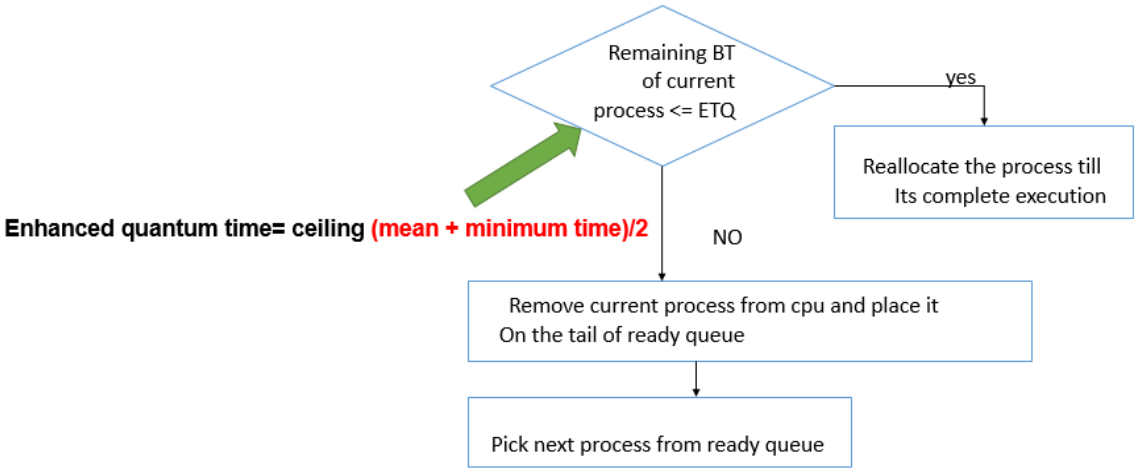
الجدول رقم (2): نتائج المقارنة بين خوارزميتي RRDTQ و Round Robin

Algorithm	Time Quantum (TQ)	Average Waiting Time (ms)	Average Turnaround Time (ms)	No. of Context Switch	Average throughput	Average response time
RR	10 , 15	46 , 49.2	66 , 67.2	10 , 7	5 , 44.2	20 , 28
Proposed RRDTQ	18 ,20 ,9 ,3	29.6	49.2	6	24.6	30

2- في الدراسة الثانية [5] A Hybrid Round Robin Scheduling Mechanism for Process Management يقدم الباحث فكرة أساسية و هي اثبات فعالية الفترة الزمنية (quntum) الديناميكية بدلا من الثابتة المستخدمة في خوارزمية الجدولة الدورانية الكلاسيكية، ويتم حساب الفترة الزمنية ديناميكيا بالاعتماد على الوسيط والزمن الأقل لرشقة المعالج لمجموعة الاجرائيات.

ولكن فعاليتها اعتمدت على المقارنة مع RR الأساسية فقط وليس مع الاصدارات الأحدث.

واذا كانت رشقة المعالج كبيرة فإن تكرار الخطوات سيكون اكثر بالإضافة الى ان نتائجها اعتمدت على قيم متقاربة لرشقة المعالج ولم تركز على التباين بين القيم.



الشكل رقم (6):المخطط الصندوقي لخوارزمية HYBIRD

المعادلة الرئيسة التي يستند عليها حساب TIME QUANTUM هي:

Enhanced quantum time= ceiling (mean + minimum time)/2

يتم التنفيذ باستخدام نفس الاجرائيات والمصطلحات التالية تعبر عن :

Mean وهو الوسيط (القيمة الوسطة من بين قيم رشقات المعالج) = 18

Minimum time وهو الزمن الأقل لرشقة المعالج = **10**

وبتنفيذ المعادلة الاساسية للخوارزمية نجد:

Enhanced quantum time= ceiling (mean + minimum time)/2=14

P1	**P1**	**P3**	**P5**	**P2**	**P2**	**P4**	**P4**	**P4**

0 14 18 28 42 56 65 79 93 100

Gantt chart representation of HYBIRD algorithm:(7) الشكل رقم

3– في الدراسة الثالثة [2] Determining Proficient Time Quantum to Improve the Performance of Round Robin Scheduling Algorithm يقدم الباحث خوارزمية تهدف الى تحسين أداء وحدة المعالجة المركزية وتسريع تنفيذ الاجرائيات وذلك عن طريق التركيز على قيمة رشقة المعالج التي تزيد وتحسن من أداء خوارزمية الجدولة الدورانية ويتم حسابها اعتمادا على الوسيط لمجموعة الاجرائيات الموجودة في رتل الجاهزية بالإضافة الى اعلى فرق بين كل اجرائيتين من مجموعة الاجرائيات.

المعادلات المستخدمة في حساب الفترة الزمنية (quantum) للخوارزمية المقترحة:

$$[i] = Pi+1 + Pi \qquad (1)$$

$$MAX_DIFF = (DI[i]) \qquad (2)$$

$$TQ = MBT + MAX_DIFF \qquad (3)$$

TQ: Time Quantum الفترة الزمنية

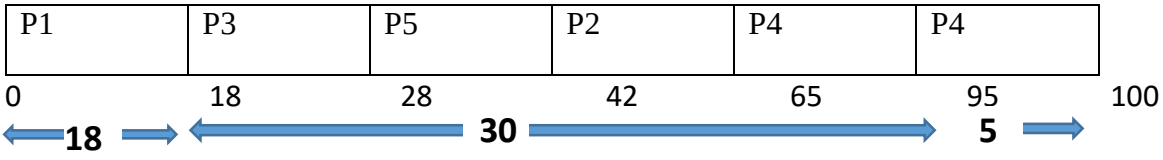
MBT هو الوسيط لمجموعة الاجرائيات الموجودة في رتل الجاهزية

MAX_DIFF هو اعلى قيمة فرق بين اجرائيتين متجاورتين في رتل الجاهزية

1- تنفذ الإجرائية التي تصل أولا حسب رشقة المعالج الخاص بها .

2- نطبق المعادلة على باقي الاجرائيات ويتم التنفيذ لمرة واحدة بعد الترتيب التصاعدي

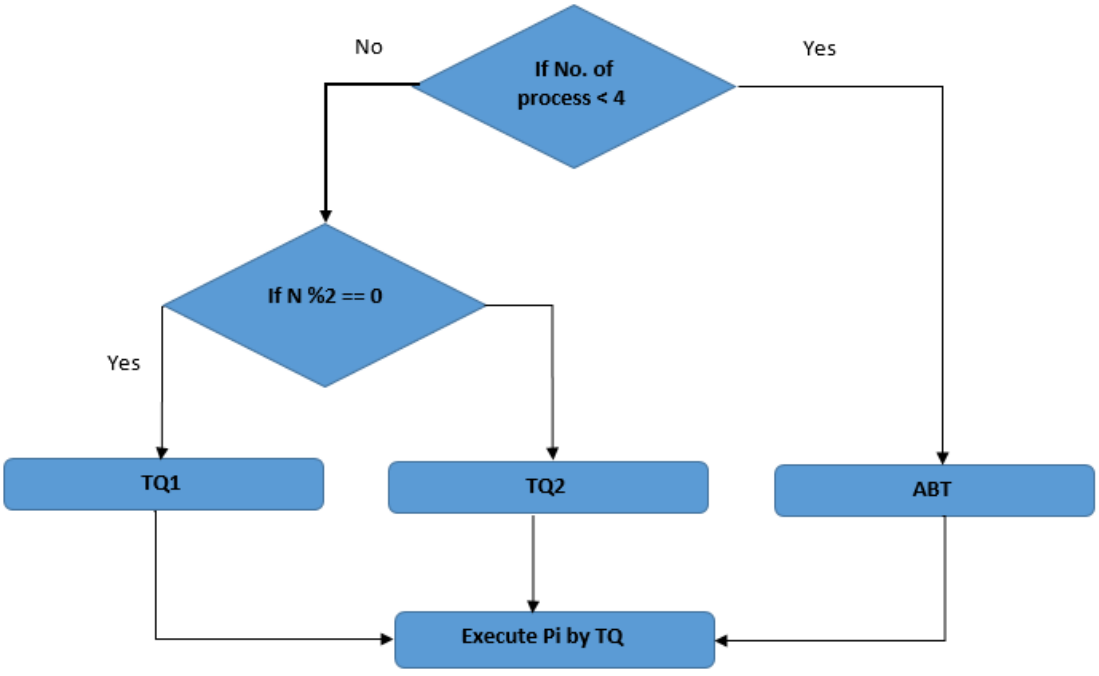
3- اذا بقي اجرائيات تنفذ حسب الباقي من رشقة المعالج الخاصة بها .



الشكل رقم(8):Gantt chart representation of DPTQRR algorithm

4- في الدراسة الرابعة [7] Improved Round Robin Scheduling Algorithm with Progressive Time Quantum ، تم اقتراح طريقة جديدة تغير رشقة المعالج بطريقة تدريجية في حالات مختلفة من رتل الجاهزية. يتم تعديل رشقة المعالج بشكل متكرر وفقًا لرشقة المعالج المتبقية (remaining burst time)للاجرائية المنفذة حاليًا.

المخطط الصندوقي للخوارزمية:



الشكل رقم(9):المخطط الصندوقي لخوارزميةIRRPTQ

حسب المعادلات :

$$TQ_1 = (P_i + P_N + P_{\frac{N}{2}} + P_{N/(2+1)})/4$$

$$TQ_2 = (P_i + P_N + P_{ceil(N/(2-1))} + P_{ceil(N/(2+1))})/4$$

$$ABT = \sum_1^n AVG(P_i)$$

في رتل الجاهزية تعبر الرموز التالية:

ABT : متوسط قيم رشقات المعالج في رتل الجاهزية

P_i : الخوارزمية الأولى

P_N: الخوارزمية الأخيرة

$P_{ceil(N/2)}$: (الوسيط) القيمة الوسطى من بين قيم رشقات المعالج للإجرائيات في رتل الجاهزية

$P_{ceil(N/(2-1))}$: الإجرائية السابقة للوسيط

$P_{ceil(N/(2+1))}$: الإجرائية اللاحقة للوسيط

يوضح الشكل (10) مخطط Gantt chart بعد تنفيذ خوارزمية IRRPTQ

P1	P3	P3	P3	P1	P5	P2	P4	
0	4	8	10	14	28	42	65	100

الشكل رقم (10): Gantt chart representation of IRRPTQ

5- في الدراسة الخامسة Improved Version of Round Robin Scheduling Algorithm Based on Analytic Mode [3] ، الهدف هو التغلب على عيوب Round Robin التقليدية من خلال اقتراح خوارزمية جدولة دزرانية محسنة باستخدام فترة زمنية متغيرة بناءً على نموذج تحليلي. يحسب النموذج التحليلي متعدد العدادات multipa- rameter المقترح تكلفة كل مهمة. يتم فرز قائمة المهام الجاهزة وفقًا لهذه التكلفة. يستخدم النهج المقترح فنرة زمنية quantum المثلى أثناء عملية التنفيذ بناءً على رشقة المعالج للعمليات الموجودة في رتل الجاهزية

$$CT(i) = \alpha Rarr(T_i) + \beta Rbut(T_i) \qquad (1)$$

حيث:

$\alpha\ \beta$ اوزان اختيارية حيث $\alpha + \beta = 1$, i هو عدد الاجرائيات.

Rarr(Ti): معدل وقت الانتظار للإجرائية

Rbut(Ti): معدل رشقة المعالج للإجرائية

$$Rarr(T_i) = \frac{Tarr(T_i)}{\max\limits_{i \le j \le N} Tarr(j)} \qquad (2)$$

حيث:

Tarr(Ti) : وقت الانتظار للإجرائية

Nعدد الإجرائيات في رتل الجاهزية

$\max\limits_{i \le j \le N} Tarr(j)$القيمة العظمى لوقت الانتظار

$$Rbut(T_i) = \frac{BT(T_i)}{\max\limits_{i \le j \le N} BT(j)} \qquad (3)$$

حيث:

BT(Ti)قيمة رشقة المعالج للاجرائية

$$CT(i) = \alpha \frac{Tarr(T_i)}{\max\limits_{i \le j \le N} Tarr(j)} + \beta \frac{BT(T_i)}{\max\limits_{i \le j \le N} BT(j)} \qquad (4)$$

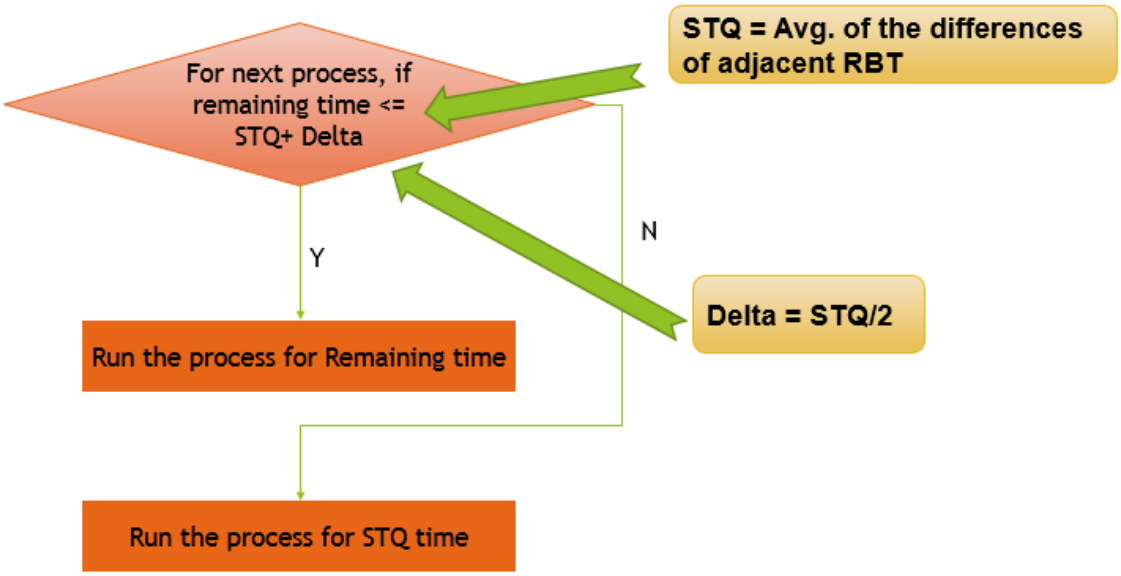
تنفيذ الخوارزمية بناء على معادلاتها الأساسية:

P1	P1	P3	P5	P2	P4
0 9		18	28	42	65 100

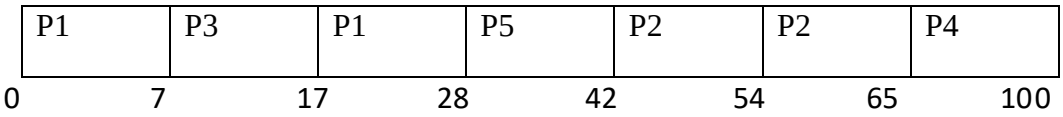
الشكل رقم(11): Gantt chart representation of IVRRAM

6- في الدراسة السادسة [6] Smart Round Robin CPU Scheduling Algorithm For Operating Systems يقترح الباحث طريقة جديدة حيث تتغير quantum (الفترة الزمنية) في كل دورة اعتمادًا على الوقت المتبقي للإجرائيات. تعطي quantum (الفترة الزمنية) المتغيرة العمليات خيارًا لأخذ quantum (الفترة الزمنية) إضافية. تعاون الاثنين ، يشكل Smart Round Robin. ينتج متوسط وقت انتظار أقل (AWT) ومتوسط وقت استجابة أقل (TAT) مقارنة بطريقة جدولة Round Robin التقليدية. توفر الخوارزمية quantum (فترة الزمنية) ديناميكية لكل دورة وتساعد على وجه التحديد المهام التي لها رشقة معالج قصيرة متبقية

تقوم هذه الدراسة بحل مشكلة الاجرائيات التي تملك رشقة معالج متبقية RBT صغيرة وتنتظر انتهاء الدورة حتى تكمل تنفيذ عملها ولكن نتائجها اعتمدت فقط على AWT,ATT لم تأخذ بعين الاعتبار وقت الوصول و المقارنة مع خوارزمية الجدولة الدورانية الكلاسيكية .



الشكل رقم(12): المخطط الصندوقي لخوارزمية SRR

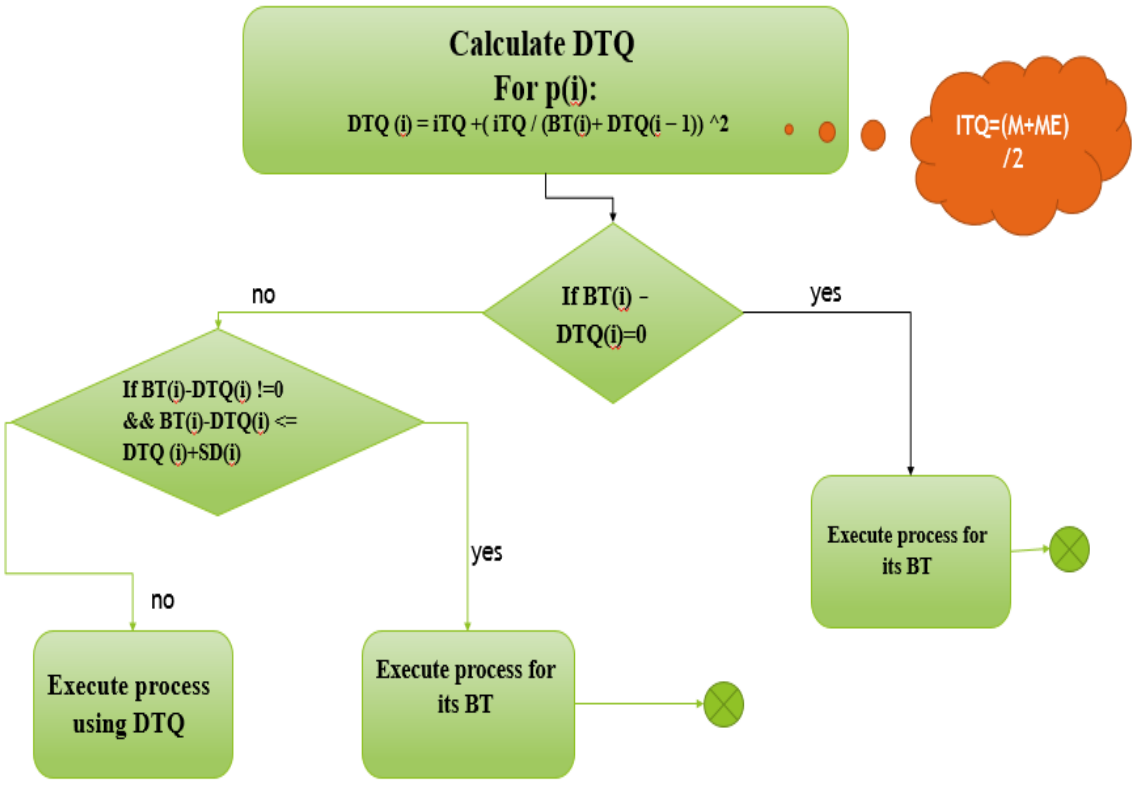


الشكل رقم(13):Gantt chart representation of SRR algorithm

7– في الدراسة السابعة [1] Advanced Dynamic Round Robin Algorithm for CPU Scheduling in Real Time Systems ، تم اقتراح خوارزمية دورانية ديناميكية متقدمة لحساب الفترة الزمنية بطريقة ديناميكية باستخدام تابع رياضي محسّن. يشتمل النموذج المصمم على حساب الفترة الزمنية لكل عملية بناءً على رشقات المعالج التي بقيت أثناء تنفيذ الاجرائيات. يمكن أن يتسبب التباين العالي في قيم رشقات المعالج في حالة مجاعة starvation للموارد للعمليات القصيرة حيث يؤدي إلى زيادة AWT و TAT و NCS ولهذا السبب تم استخدام SD من أجل الحصول على حسابات TQ أفضل.

في هذا الإصدار تم اعتماد الانحراف المعياري في حساب الفترة الزمنية و حل التشتيت العالي أو التباين بين قيم رشقات المعالج للإجرائيات.

$$SD(i) = \sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(BT(i) - M)\text{^}2}$$



الشكل رقم(14):المخطط الصندوقي لخوارزمية ADRR

P1	P3	P5	P2	P4

0 18 28 42 65 100

الشكل رقم(15): Gantt chart representation of ADRR

الجدولة الدورانية في الحوسبة السحابية:

1- في الدراسة الأولى[4] Time Efficient Round Robin Job Scheduling (NARR) in Cloud Computing يقترح الباحث فترة زمنية ديناميكية ومقارنة الأداء المختلف وذلك عن طريق حساب الوسيط بناءً على العمليات المضافة في قائمة الانتظار الجاهزة.

الفترة الزمنية هي القيمة الوسطى من بين قيم رشقات المعالج للإجرائيات في رتل الجاهزية

P1	P3	P5	P2	P4	P2	P4

0 18 28 42 60 78 83 100

الشكل رقم(16):Gantt chart representation of TERR algorithm

2– في الدراسة الثانية[9] Improved Round Robin Scheduling Algorithm for Cloud Computing ،هذا البحث لديه القدرة على إنشاء توازن بين اجرائيات الوظائف في الحوسبة السحابية في قائمة انتظار جاهزة عن طريق فرز الاجرائيات وفقًا ل رشقة المعالج الخاصة بها ثم حساب الوقت الكمي عن طريق جمع متوسط الفروق ومتوسط رشقات المعالج. من خلال تطبيق ونشر الخوارزمية المقترحة ، من الممكن تعزيز وتحسين تخصيص العملية من خلال النظام.

BTDIFF[i] متوسط الفروق بين الاجرائيات بعد ترتبها تصاعديا = 8

متوسط قيم رشقات المعالج = 22.5

TQ (الفترة الزمنية) = متوسط الفروق بين الاجرائيات + متوسط قيم رشقات المعالج =31

P1	P3	P5	P2	P4	P4
0	18	28	42	65	96 100

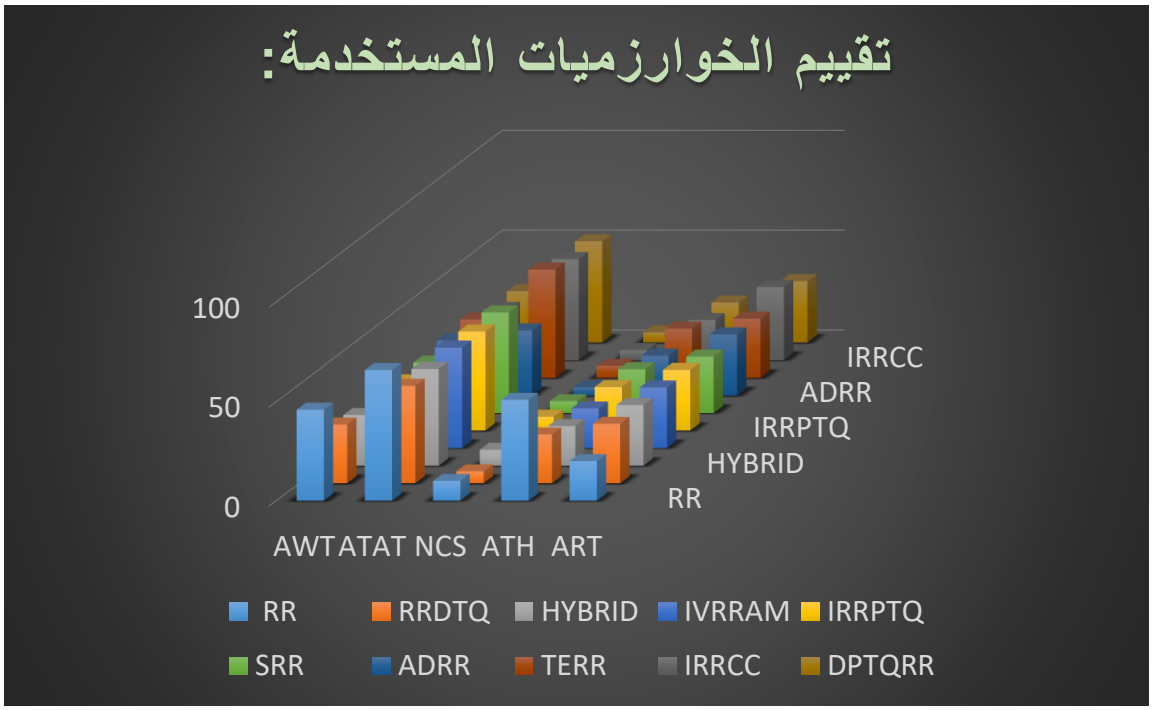
الشكل رقم (17):Gantt chart representation of IRRCC algorithm

4–النتائج النهائية للمقارنة:

يوضح الجدول (3) النتائج النهائية لتنفيذ الاجرائيات على خوارزميات الجدولة الدورانية المحسنة التي تم طرحها في البحث ومقارنتها مع خوارزمية الجدولة الدورانية الكلاسيكية بناء على معاملات الأداء الرئيسية لخوارزميات الجدولة.

الجدول رقم(3):النتائج النهائية للمقارنة بين خوارزمية RR والاصدارات المحسنة منها

Algorithm	Time Quantum (TQ)	Average Waiting Time (ms)	Average Turnaround Time (ms)	No. of Context Switch	Average throughput	Average response time
RR	10 , 15	46 , 49.2	66 , 67.2	10 , 7	51, 44.2	20 , 28
RRDTQ	18 , 20 , 9 , 3	29.6	49.2	6	24.6	30
HYBRID	14	25,6	48.8	8	20	30.6
DPTQRR	18,30,5	25.6	50.6	5	20	30.6
IRRPTQ	4,16,20,20,22,24,29	24.8	49.8	7	22	30.6
IVRRAM	9,17,19,23,29	25.6	50.6	5	20	30.6
SRR	7,10,11,14,12,11	25.4	50.4	6	22	28.4
ADRR	18,10,14,23,35	27.6	32.6	4	20	30.6
TERR	18	29.2	54.2	6	24.6	29.6
IRRCC	31	25.6	50.6	5	20	36.6



الشكل رقم (18):مخطط يعرض نتائج المقارنة بين الإصدارات المحسنة للجدولة الدورانية

دراسة المقارنة الثانية: تقييم عمل مجموعة من الخوارزميات (RR –NARR–SRR–HYBRID – RRDRQ) باستخدام PYTHON باستخدام 100 إجرائية وزمن وصول مختلف و رشقات معالج مختلفة

Round Robon -1

```
Processes     arrivetime     Burst Time     Waiting Time     Turn-Around Time

  90             90             58             88156             88214
  91             93             10             24128             24138
  92             68             5              15380             15385
  93             21             8              20195             20203
  94             52             12             27556             27568
  95             37             55             86334             86389
  96             46             48             77278             77326
  97             33             557            146646                      147203
  98             51             65             93846             93911
  99             17             98             105390                      105488
  100            20             58             88285             88343

Average waiting time = 59540.48000
Average turn around time = 59602.55000
```

الشكل رقم(19):خرج تنفيذ Python code لخوارزمية RR

RRDTQ -2

```
Processes     arrivetime     Burst Time     Waiting Time     Turn-Around Time

   90             51             58             184441                184499
   91             99             10             47398          47408
   92             1              5              24940          24945
   93             62             8              39152          39160
   94             74             12             54695          54707
   95             54             55             178660                178715
   96             82             48             161762                161810
   97             54             557            338212                338769
   98             22             65             195037                195102
   99             83             98             220060                220158
   100            66             58             184670                184728

Average waiting time = 125032.81000
Average turn around time = 125094.88000
>>>
```

الشكل رقم(20):خرج تنفيذ Python code لخوارزميةRRDTQ

NARR -3

```
Processes     arrivetime     Burst Time     Waiting Time     Turn-Around Time

   90             95             58             161031                161089
   91             76             10             43294          43304
   92             65             5              23543          23548
   93             98             8              36506          36514
   94             71             12             49460          49472
   95             15             55             156111                156166
   96             82             48             141852                141900
   97             10             557            224460                225017
   98             97             65             168895                168960
   99             15             98             185161                185259
   100            96             58             161253                161311

Average waiting time = 106323.53000
Average turn around time = 106385.60000
>>>
```

الشكل رقم(21):خرج تنفيذ Python code لخوارزمية NARR

HYBRID -4

```
Processes    arrivetime    Burst Time    Waiting Time    Turn-Around Time

  90            77            58           66465.0           66523.0
  91            87            10           19933.0           19943.0
  92            46            5            10702.0           10707.0
  93            56            8            15818.0           15826.0
  94            36            12           19970.0           19982.0
  95            93            55           64383.0           64438.0
  96            15            48           56605.0           56653.0
  97            56            557          113841.0          114398.0
  98            80            65           70035.0           70100.0
  99            81            98           80282.0           80380.0
  100           71            58           66763.0           66821.0

Average waiting time = 44561.95000
Average turn around time = 44624.02000
>>>
```

الشكل رقم(22):خرج تنفيذ Python code لخوارزمية HYBRID

SRR -5

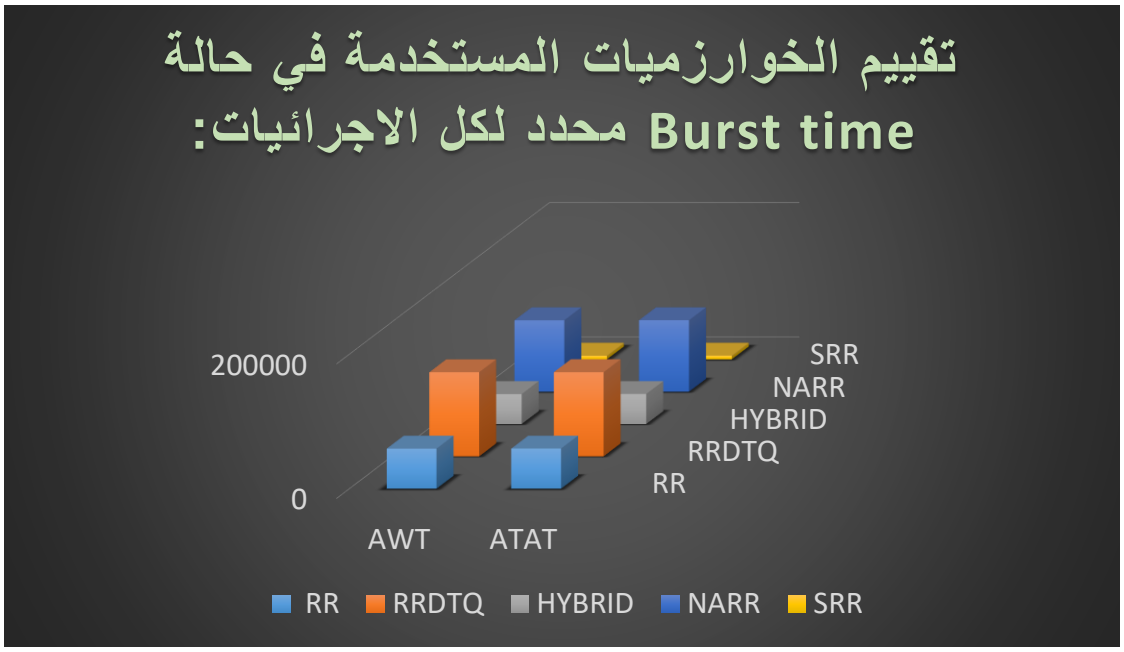
```
Processes    arrivetime    Burst Time    Waiting Time    Turn-Around Time

  90            76            64           9589              9653
  91            47            78           9700              9778
  92            29            53           9807              9860
  93            71            60           9931              9991
  94            5             26           9996              10022
  95            69            87           10091             10178
  96            60            71           10238             10309
  97            77            85           10386             10471
  98            82            64           10553             10617
  99            64            40           10681             10721
  100           54            55           10775             10830

Average waiting time = 5402.90000
Average turn around time = 5456.30000
>>>
```

الشكل رقم(23):خرج تنفيذ Python code لخوارزمية SRR

النتائج:



الشكل رقم(24): مخطط يعرض نتائج المقارنة بين الإصدارات المحسنة للجدولة الدورانية اعتماد على خرج Python

5-المناقشة:

نلاحظ من خلال دراسة وتقييم مجموعة الخوارزميات المطورة من خوارزمية الجدولة الدورانية (اعتمادا على الدراستين) يوجد تحسين في استغلالية المعالج utilization cpu من خلال تحسين معاملاتها الأساسية بشكل جزئي او كامل.

تم ذلك من خلال **تخفيض** كل من :(number of context switches(NCS) average waiting time (AWT)

(average response time (ART) – the average turnaround time (ATT)–

وذلك باستخدام الخوارزميات: HYBRID – SRR

تخفيض number of context switches(NCS) – (the average turnaround time (ATT)

عن طريق الخوارزمية: ADRR

تخفيض (average waiting time (AWT عن طريق الخوارزمية: IRRPTQ

و**زيادة** Throughput time بشكل نسبي مقارنة بتخفيض باقي المعاملات عن طريق الخوارزميات:

TERR – RRDTQ

6-الاستنتاجات: نلاحظ أن الخوارزميات الأفضل لتحقيق التطوير على الجدولة الدورانية هي:ADRR, HYBRID,SRR

7-التوصيات:

نقترح مستقبلا خوارزمية جدولة دورانية مطورة مهمتها تحسين الأداء للدرجة القصوى وتحسين المعاملات(AWT – NCS ATT – ART) انطلاقا من أفضل الخوارزميات المطورة وذلك عن طريق دمج خوارزميتين مطورتين أو أكثر أو تطوير خوارزمية واحدة وتحسينها.

8-المراجع:

1- Abd Ali,SH. M., Hasan,H. F. , Abed, Z. F. ,(2018), Advanced Dynamic Round Robin Algorithm for CPU Scheduling in Real Time Systems, Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 02-Special Issue.

2- Biswas,D. , Samsuddoha,Md. ,(2019), Determining Proficient Time Quantum to Improve the Performance of Round Robin Scheduling Algorithm, I.J. Modern Education and Computer Science, 2019, 10, 33-40 Published Online October 2019 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijmecs.2019.10.04

3- Fiad,A. , Maaza,Z.M. , Bendoukha,H. ,(2020), Improved Version of Round Robin Scheduling Algorithm Based on Analytic Model, International Journal of Networked and Distributed Computing Vol. 8(4); December (2020), pp. 195–202.

4- Jahan, A. , Mehra,R., Saini,R., (2019), Time Efficient Round Robin Job Scheduling (NARR) in Cloud Computing, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1.

5- Kumar , K. A. , PhD , Ali,K. F. , Marikal,A. ,(2020), A Hybrid Round Robin Scheduling Mechanism for Process Management, International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 36, February 2020

6- Mody,S., Mirkar,S. , (2019), Smart Round Robin CPU Scheduling Algorithm For Operating Systems, International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), DOI: 10.1109/ICEECCOT46775.2019.9114602

7- Paul,T. , Faisal,R.H. , Samsuddoha, Md., (2019), Improved Round Robin Scheduling Algorithm with Progressive Time Quantum, International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 49.

8- Sohrawordi, Md. , Ehasn Ali, U. A. Md. , Palash Uddin, Md. and Mahabub Hossain, Md. ,(2019),A MODIFIED ROUND ROBIN CPU SCHEDULING ALGORITHM WITH DYNAMIC TIME QUANTUM,INTERNASIONAL JOURNAL OF ADVANCED RESEARCH (IJAR), Article DOI:10.21474/IJAR01/8506 DOI URL: http://dx.doi.org/10.21474/IJAR01/8506

9- Lotfy, S. , (2020), Improved Round Robin Scheduling Algorithm for Cloud Computing, Computer Engineering Department ,Faculty of Engineering, Cairo University Cairo, Egypt, July 2020.